Interaction-Aware Labeled Multi-Bernoulli Filter

Nida Ishtiaq, Amirali Khodadadian Gostar®, Alireza Bab-Hadiashar®, Senior Member, IEEE, and Reza Hoseinnezhad®

Abstract—Tracking multiple objects through time is an important part of an intelligent transportation system. Random finite set (RFS)-based filters are one of the emerging techniques for tracking multiple objects. In multi-object tracking (MOT), a common assumption is that each object is moving independent of its surroundings. However, in many real-world applications, objects interact with one another and the environment. Such interactions are rarely considered within the tracking process. In this paper, we present a novel approach to incorporate target interactions within the prediction step of a RFS-based multi-target filter, i.e. labeled multi-Bernoulli (LMB) filter. The proposed method is capable of explicitly incorporating the effect of target interactions within the filtering process, with little to no changes made for specific applications. We have developed the proposed filter for two practical applications, i.e. tracking a coordinated swarm and vehicles. The method has been tested for a complex vehicle tracking dataset and compared with the PHD filter and LMB filter through OSPA metric, and with the LMB filter through OSPA(2) metric and cardinality error. The results demonstrate that the proposed interaction-aware method depicts considerable performance enhancement over the other methods in terms of the selected metrics.

Index Terms—Multi-object tracking, random finite sets, labeled multi-Bernoulli, interaction-aware tracking.

I. INTRODUCTION

One of the core requirements of an intelligent transportation system (ITS) is its ability to track surrounding objects. Although an intelligent vehicle should be able to track the surrounding stationary and non-stationary objects, but the tracking of surrounding vehicles is of primary concern. Information regarding trajectories of surrounding vehicles (tracking information) is the key enabling tool to achieve a comprehensive understanding of the environment. This information can be vital for ‘smart’ decision-making in both local (vehicle) and central (e.g. control centre) levels. Such smart decisions are the backbone of efficiency and safety of driver assist [1], collision avoidance [2] and trajectory planning [3] in the local level, and traffic flow management in the control level [4]. The number of nearby vehicles, their positions and speeds vary as a vehicle moves along a road. Therefore, tracking of a randomly varying number of targets (due to target birth and death or entry and exit) is a challenging problem with its difficulty being compounded if there are possible miss-detections and false alarms (clutter) in sensor measurements [5]. In early 21st century, a new class of stochastic multi-object filters was invented by Mahler [6], [7], [8]. They were called random finite set (RFS) filters and were designed by treating the multi-object entity as a random set of single-object entities (targets) with random cardinality (number of elements in the set). Various approximations, followed by efficient implementations of RFS filters, were then proposed [9], [10], [11], [12], [13] and applied in different domains [14], [15], [16]. The latest generation of RFS filters, called labeled RFS filters append the label of each target into its single-target state and propagate target labels with their states to directly create target trajectories [17], [18], [19], [20], [21]. The labeled multi-Bernoulli (LMB) filter [19] is of particular interest in this paper and will be explored further in Section III.

A major advantage of using stochastic multi-object filters for multi-target tracking, especially with RFS filters, is that they allow us to directly incorporate the environment-related information about targets (regardless of measurements) such as target birth- and death-related information, into the prediction step of the filtering process in a mathematically principled manner. This is more evident in the multi-Bernoulli filter and its labeled version, the LMB filter: The birth process is modeled by a set of possibly existing targets, each distributed around one possible area of target entry, and target death is modeled by a state-dependent probability of survival that is small in possible areas of target exit. Incorporation of extra information, if modeled accurately, is expected to result in improvements in the overall cardinality and state estimation and tracking performance of the filter. Nowadays, machine learning-based methods are becoming increasingly common for many applications, including detection-based tracking. Tomakov et. al. [22] considers the concept of object permanence to develop detection methods for occluded objects. However, learning-based methods have not been deemed very feasible for model-based measurements as these methods require an immense amount of data for training and require sequential detections, which is not the focus of this paper. A recent work [23] highlights this issue and develops a deep learning-based method (which is a class of machine learning) and has comparable performance to the state-of-the-art Bayesian methods like δ-generalized labeled multi-Bernoulli (δ-GLMB) filter [21] for simplistic scenarios. Hence for complex model-based tracking, such as the applications discussed in this paper, Bayesian methods are the preferred choice owing
to the many drawbacks of deep learning methods highlighted in [24].

Interactions between targets and how they may affect target motion are something that have been rarely taken into account in formulating stochastic multi-object filters as solutions for multi-target tracking problems. To facilitate the simplicity of the developed method, it is often assumed that each object moves independently, with no limitations posed by the other targets or the environment. This leads to tracking solutions which may perform poorly in a realistic scenario, specifically in highly dense and dynamic environments. One major example is transportation systems, where slight errors in location estimation can lead to hazardous consequences [25], [26], [27], [28], [29], [30]. Therefore, fundamental solutions capable of identifying and utilizing such interactions are highly desirable.

This paper presents a new approach for incorporating a precise model of target interactions directly into the prediction step of the LMB filter. The LMB filter is a class of Bayesian filters that incorporate the estimation of the number and states of an unknown number of moving targets and it has demonstrated superior performance in challenging multi-target tracking applications. We have selected the LMB filter because it is suitable for real-time applications and does not impose the additional computational complexity as opposed to the alternative filters such as the GLMB filter. The core contribution of this work lies in the general applicability of the proposed method. Without using external information, the filter can identify certain types of interactions among targets, resulting in more accurate and realistic estimates. We have developed a novel prediction formulation where the interactions influence the predicted target states instead of using external interaction-aware motion modeling, without compromising the target independence assumption commonly followed by Bayesian filters. We also show how various intuitive deterministic interaction models can be turned into interaction-aware LMB filters for accurate tracking of numerous targets in very challenging applications. Our experiment involves tracking of a large number of vehicles from aerial photos in a complex multi-road junction area. Significant improvements in tracking performance, in terms of optimal sub-pattern assignment (OSPA) metric [31], OSPA[2] and cardinality error are demonstrated.

This paper represents a substantial advancement of our previous work as outlined in [32], where the main concept was first introduced. Here, we provide a more in-depth examination, including mathematical and experimental details, for a variety of practical applications. Additionally, the simulation results presented in this paper depict a realistic and challenging scenario, as opposed to the original synthetic data simulations presented in [32]. The key contributions of this paper are summarized as follows:

- A mathematical proof for LMB prediction when considering target interactions.
- Implementation details for interaction-aware tracking in the RFS framework, specifically in context of the LMB filter.
- Reformulation of existing deterministic interaction models into stochastic forms to be used within our proposed interaction-aware LMB filtering scheme.
- Evaluation of performance on a practical challenging vehicle tracking scenario that involves tracking of numerous targets with a significant number of ongoing interactions.

The rest of this paper is organised as follows. Section II presents an overview of prominent related work. Section III introduces a background to labeled RFS filters and the notation used in formulating the relevant techniques. Our proposed solution is illustrated in Section IV. One possible approach for Sequential Monte Carlo (SMC) implementation of the proposed interaction-aware LMB filter is presented in Section V which also includes examples of how the solution can be implemented in vehicle tracking applications. In Section VI, experimental results are provided and discussed. The paper is concluded in Section VII.

II. RELATED WORK

In this section we present some existing strategies for incorporating interactions between targets in the context of multi-object tracking. One existing methodology for catering target interactions is forming target groups according to the locations, motion parameters, or other application-specific parameters [33], [34]. The targets in a group are considered to be behaving in a similar fashion. However, most methods in this category face implementation issues due to the splitting and merging of groups. The formation of target groups also introduces direct dependence between targets, often referred to as data association. The identification and formulation of target groups is a tedious task, making these methods both mathematically and computationally expensive. While some methods utilize data association for tracking interacting targets [35], [36], it is often undesirable for Bayesian methods, where data association between large number of targets can lead to mathematical intractability. In the target tracking literature, particularly in visual tracking and vehicle tracking fields, there have been several attempts to model the interactions between targets. Those models are dominantly deterministic. Examples include the interactive motion-based vehicle tracking [37], [38], car-following and lane-changing models [39], [40], [41], [42], social and group behaviour models for visual tracking [43], [44], [45].

Within stochastic filters, there have been a number of works that their design would consider target interactions and their effect on tracking performance. The most prominent examples are the interactive Kalman filter [46] and the unscented Kalman filter [47], [48], and the multiple model filters [49], [50]. Kalman filters are naturally designed as single-object filters and when used for multi-target tracking, they are designed as a stack of filters based on availability of prior knowledge about the number of targets. In the multiple model filters, several possible dynamic models are considered for each target movement, and switching from each model to the next is governed by a state machine and constant or state-dependent transition probabilities. In addition to these
methods, as far as we know, except for the generalized labeled multi-Bernoulli (GLMB) filter [21], in other stochastic multi-object filtering solutions, random variations of target states are assumed to be independent from each other, i.e. zero interaction is assumed between targets. Even in GLMB filter, target interactions are not directly modeled and incorporated into the filter. Information about such interactions are of the same type of death and birth information, in the sense that they are not measurement-related and could be incorporated into the prediction step of a Bayesian multi-object filter.

Another approach that could consider interactions between targets, is direct tracking from image observations (also called track-before-detect) [51], [52]. In such works, the multi-target likelihood is a function of image observation (rather than detection extracted from the image observations) and interactions could be indirectly incorporated into the formulation of the likelihood function. Another approach is to use multiple sensors and fuse the information in a robust [53], [54], [55], [56] or distributed fashion [57], [58], [59], [60] that despite treating target movement as independent in the prediction step of the stochastic filter, the comprehensive target-related information used in the update step compensate for lack of information on target interactions and still deliver accurate tracking. Alternatively, one could use controllable sensors that would be actuated/scheduled/selected to obtain information-rich measurements towards compensation for target interaction-related information [61], [62], [63], [64], [65], [66].

### III. BACKGROUND AND NOTATION

The notations and definitions used in this paper are summarised in Table I. The rest of this section provides a brief review of the general multi-object filter and its prediction and update steps, and the LMB filter.

#### A. The Bayesian Multi-Object Filter

Let us denote the labeled multi-object state at time $k$ by $X_k \subset \mathbb{X}$ and the multi-object observation by $Z_k \subset \mathbb{Z}$. Both $X_k$ and $Z_k$ are modeled as random finite sets. The multi-object random set distribution is recursively predicted and updated by the filter. We also denote the labeled multi-object prior density (at time $k - 1$) by $\pi_{k-1}(\cdot|Z_{1:k-1})$, where $Z_{1:k-1}$ is the collection of finite measurements up to time $k - 1$.

The prediction step of the filter is governed by Chapman-Kolmogorov equation given in (1),

$$
\pi_{k|k-1}(X_k|Z_{1:k-1}) = \int \pi_{k|k-1}(X_k|X) \pi_{k-1}(X|Z_{1:k-1}) \delta X,
$$

where the integral is the labeled set integral defined in [17].

In the update step, Bayes’ rule returns the following multi-object posterior:

$$
\pi_k(X_k|Z_{1:k}) = \frac{g_k(Z_k|X_k) \pi_{k|k-1}(X_k|Z_{1:k-1})}{\int g_k(Z_k|X) \pi_{k|k-1}(X|Z_{1:k-1}) \delta X},
$$

where $Z_k$ is generally comprised of some measurements each associated with an object (with some objects possibly missed), and some false alarms or clutter. Both the number of object-related measurements and the number of false alarms randomly vary with time. Hence, $Z_k$ is an RFS with its stochastic variations characterized by a multi-object likelihood function $g_k(Z_k|X_k)$.

#### B. The Labeled Multi-Bernoulli RFS

An LMB RFS is the union of a number of possibly existing single-object sets that are assumed statistically independent, and is denoted by (3) as:

$$
X = \bigcup_{\ell \in \mathbb{L}} X^{(\ell)},
$$

where each $X^{(\ell)}$ is a labeled RFS representing one possibly existing target. The statistical variations of each $X^{(\ell)}$ is characterised by (1) its probability of existence denoted by $r^{(\ell)}$, and (2) its single-object density $p^{(\ell)}(x)$ conditioned on its existence.

As Reuter et. al [19] have shown, the LMB distribution is completely described by its component parameters, i.e. $\pi = \{r^{(\ell)}, p^{(\ell)}(x)\}_{\ell \in \mathbb{L}}$. Indeed, given the component parameters, the LMB RFS density is given by

$$
\pi(X) = \Delta(X) w(L) [p]^X,
$$

where $\Delta(X)$ is included to ensure uniqueness of object labels, and is defined to be one if $|X| = |\mathbb{L}(X)|$ and zero otherwise. The remaining symbols in (4) are described by (5) and (6):

$$
[p]^X = \prod_{(s, \ell) \in X} p^{(s)}(x),
$$

$$
w(L) = \prod_{\ell \in \mathbb{L} - L} \left(1 - r^{(\ell)}\right) \left[\prod_{\ell \in L} 1_{\mathbb{L}}(\ell) r^{(\ell)}\right],
$$

where $w(L)$ is the probability of joint existence of all objects with labels $\ell \in L$ and non-existence of all other labels [19].

---

**TABLE I**

| Symbol | Definition |
|--------|------------|
| lower-case letters (e.g. $x, a$) | single-object state |
| upper-case letters (e.g. $X, X'$) | multi-object state |
| blackboard letters (e.g. $X, Z, \mathbb{L}$) | spaces for variables or labels |
| $\pi, \pi'$ | a multi-object density |
| $|X|$ | The cardinality (number of elements) of $X$ |
| $(f, g)$ | labeled entities |
| $[p]^X$ | inner product of two functions |
| $G_X[k]$ | Probability Generating Functional (PGF) of RFS variable $X$ |

**NOTATION AND DEFINITIONS**
An important characteristic of the LMB RFS is its PGFL. Considering that PGFL of the union of independent RFSs equals the product of their individual PGFLs [6], for the LMB RFS given in (3) we have:

\[ G_X[h] = \prod_{\ell \in \mathbb{L}} G_{X^{(\ell)}}[h] \]

(7)

where the PGFL of each single-Bernoulli component is given by:

\[ G_{X^{(\ell)}}[h] = \int [h]^X \pi^{(\ell)}(X) \delta X = [h]^p \pi^{(\ell)}(\emptyset) + \int [h]^x \pi^{(\ell)}([x]) \, dx. \]

(8)

Noting that

\[ [h]^0 = 1, \quad \pi^{(\ell)}(\emptyset) = 1 - r^{(\ell)}, \quad \pi^{(\ell)}([x]) = r^{(\ell)} p^{(\ell)}(x), \]

we have:

\[ G_{X^{(\ell)}}[h] = \left(1 - r^{(\ell)} \right) + \left(r^{(\ell)} h(\cdot), r^{(\ell)} p^{(\ell)}(\cdot) \right). \]

(9)

Therefore, the PGFL of an LMB RFS density \( \{G^{(\ell)}, p^{(\ell)}(\cdot)\}_{\ell \in \mathbb{L}} \) is given by (10) as follows:

\[ G_X[h] = \prod_{\ell \in \mathbb{L}} \left[ \left(1 - r^{(\ell)} \right) + \left(r^{(\ell)} h(\cdot), r^{(\ell)} p^{(\ell)}(\cdot) \right) \right]. \]

(10)

C. The Multi-Object System Model

The target birth and death, the application constraints and the measurement information are all encapsulated in the multi-object filter through a multi-object system model. This model is comprised of two parts: the multi-object dynamic model and the multi-object measurement model. The former is employed within the prediction step of the filter, and the latter in the update step.

The multi-object dynamic model incorporates all the non-measurement type information that exist about the targets, including their possible state transitions (e.g. movements) from each time step to the next, possible regions of their entry to the scene (the birth process) and possible regions of their exit (target death). A model for interactions between targets is best to be incorporated into this part of the multi-object system model. Hence, we provide a brief overview of the model here (which will be expanded further as the proposed solution is presented in the paper).

Target death is usually modeled by a state-dependent (normally a location-dependent) probability of survival denoted by \( p_S(x) \). The target dynamics itself is usually modeled by a state transition density. Given a target state \( x_{k-1} \) at time \( k-1 \), the probability density of the next state of the target (at time \( k \)) is denoted by \( f_{k|k-1}(x_k|x_{k-1}) \). For a survey of most common target dynamic models, readers are referred to [67].

Given a target state \( x_{k-1} \in X_{k-1} \) at time \( k-1 \), its behaviour at time \( k \) is modeled via a Markov shift, resulting in a Bernoulli RFS \( S_{k|k-1}(x_{k-1}) \) with probability of existence of \( r = p_S(x_{k-1}) \) and density of \( p(\cdot) = f_{k|k-1}(\cdot|x_{k-1}, \ell) \). Hence, given a labeled set of targets \( X_{k-1} \) at time \( k-1 \), its transition to time \( k \) is modeled by the LMB in (11):

\[ T_{k|k-1}(X_{k-1}) = \bigcup_{x \in X_{k-1}} S_{k|k-1}(x). \]

(11)

The appearance of new objects (the birth process) at time \( k \) is modeled by an LMB RFS of spontaneous births, denoted by:

\[ \Gamma_k = \left\{ \left( r^{(\ell)}(\cdot), p^{(\ell)}(\cdot) \right) \right\}_{\ell \in \mathbb{L}_k} \]

(12)

where \( \mathbb{L}_k \) denotes the space of labels of targets that may be born at time \( k \). Consequently, the labeled RFS of multi-object state \( x_k \) at time \( k \) is itself an LMB given by the union of (11) and (12) as:

\[ X_k = T_{k|k-1} \cup \Gamma_k. \]

(13)

The multi-object state transition density \( f_{k|k-1}(X_k|X_{k-1}) \) is the density of an LMB with the parameters:

\[ \left\{ \left( p_S(x), f_{k|k-1}(\cdot|x) \right) \right\}_{x \in X_{k-1}} \cup \left\{ \left( r^{(\ell)}(\cdot), p^{(\ell)}(\cdot) \right) \right\}_{\ell \in \mathbb{L}_k}. \]

(14)

In an LMB filter, assume that the multi-object prior is an LMB density with parameters

\[ \pi_{k-1} = \left\{ \left( t^{(\ell)}_{k-1}, p^{(\ell)}_{k-1}(\cdot) \right) \right\}_{\ell \in \mathbb{L}_{k-1}} \]

(15)

where \( \mathbb{L}_{k-1} \) denotes the space of target labels at time \( k-1 \). Note that this space is gradually extended with time as new targets are born, according to:

\[ \mathbb{L}_k = \mathbb{L}_{k-1} \cup \mathbb{L}_k. \]

(16)

Substituting from equations (14) and (15) in the Chapman-Kolmogorov equation (1), leads to the following parametrisation of the predicted multi-object density [19]:

\[ \pi_{k|k-1} = \left\{ \left( t^{(\ell)}_{k|k-1}, p^{(\ell)}_{k|k-1}(\cdot) \right) \right\}_{\ell \in \mathbb{L}_{k-1}} \]

(17)

where the components \( t^{(\ell)}_{k|k-1} \) and \( p^{(\ell)}_{k|k-1}(\cdot) \) are given by (18) and (19) as:

\[ t^{(\ell)}_{k|k-1} = \left( r^{(\ell)}_S(\cdot), p^{(\ell)}_{k-1}(\cdot) \right) \]

(18)

\[ p^{(\ell)}_{k|k-1}(\cdot) \]
prediction involving the multi-object transition with SMC implementation would need sampling of RFSs which would include sampling the cardinality density and for each cardinality sample, again sampling the multi-object state. This approach is computationally very expensive and in the presence of numerous objects, it may be intractable. The core idea presented in this paper is that we can model any prior information into the state transition density, with including the parameters of the overall multi-object LMB prior as additional parameters. Implementing such a model with LMB densities is straightforward because at time $k$, each possibly existing object with label $\ell$ is separately associated with parameters $r_k^{(\ell)}$ and $p_k^{(\ell)}(\cdot)$. Note that the density parameter is usually replaced with weights, means and covariance matrices (if a Gaussian-mixture implementation is applied) or with particles and their weights (if a sequential Monte-Carlo implementation is applied).

We envisage that at any time $k$, interactions between each target $\ell$ and other targets can be modeled by making its transition density dependent on the true states of other targets at time $k - 1$. However, in the absence of prior knowledge of true states, we propose to use the available information, i.e., using the estimated multi-object density at the previous time $k - 1$. Note that the density is parameterized and its parameters will be used as the parameters of single-object transition density.

Let us denote the Bernoulli parameters of each target label $\ell$ at time $k$ by (20):

\[ \pi_k^{(\ell)} \triangleq \left( r_k^{(\ell)}, p_k^{(\ell)}(\cdot) \right) \]

and denote the set of all such parameters except for label $\ell$ by (21)

\[ \psi_k^{(\ell)} \triangleq \bigcup_{\ell' \in L_{k-1} \setminus \{\ell\}} \pi_{k-1}^{(\ell')} \].

An interaction-aware single-object transition density is denoted by $f_{k|k-1}(x|x_{k-1}, \ell; \psi_k^{(\ell)})$. The most straightforward example of how the parameters $\psi_k^{(\ell)}$ given by (21) can be incorporated into the state transition density is by inferring a multi-object estimate at time $k - 1$ and use it as the parameters in the transition density. There are two common approaches for this inference. The first is to extract only those Bernoulli components whose probabilities of existence are greater than a given threshold (e.g., 0.50) and the second is to calculate the cardinality estimate, $|\hat{X}_{k-1}|$, as the sum of all probabilities of existence, then consider only those Bernoulli components whose probabilities of existence are among the $|\hat{X}_{k-1}|$-th largest. In both cases, if a Bernoulli component with label $\ell$ is chosen, its state estimate is given by (22):

\[ \hat{x}_{k-1}^{(\ell)} = \int \pi_{k-1}^{(\ell)}(x) \, dx \]

and the interaction-aware single-object state transition density is denoted by $f_{k|k-1}(x|x_{k-1}, \ell; \hat{x}_{k-1}^{(\ell)})$.

Remark 1: Note that if $\ell \not\in L(\hat{X}_{k-1})$, then we simply have $\hat{x}_{k-1}^{(\ell)} = \hat{x}_{k-1}$.

In Section V, we will present a number of examples to demonstrate how common and intuitive interactions between vehicle targets can be modelled into a previous estimate-dependent transition density $f_{k|k-1}(x|x_{k-1}, \ell; \hat{x}_{k-1}^{(\ell)})$.

Remark 2: With incorporating the interaction-aware single-object density, whether in the general form of $f_{k|k-1}(x|x_{k-1}, \ell; \psi_k^{(\ell)})$ or the more specific form of $f_{k|k-1}(x|x_{k-1}, \ell; \hat{x}_{k-1}^{(\ell)})$, the Bernoulli components of the multi-object LMB at time $k$ can still be assumed conditionally independent, given the prior. This is similar to the conditional independence that is commonly assumed for measurements, given the object state.

The conditional independence of target states discussed in Remark 2 can be elaborated with the help of a practical example shown in Fig. 1, where two vehicles (\(\ell\)) (orange car) and (\(\ell'\)) (red car) can be seen moving along a road, maintaining a safe distance between them, which is shown as 35m. The velocities are represented by lines with arrows ahead of each vehicle (magnitude is depicted by length of the line). Possible locations at next time step are shown by faded vehicles. Owing to some external factors (road closure in this case), the front vehicle (\(\ell'\)) may be required to drastically change its speed, which would force the following vehicle $\ell$ to slow down as well. Due to the independence assumption in traditional MOT, the predicted particles would not reflect the change in speed required by vehicle $\ell$, as depicted by black circles in Fig. 1 (a), causing the distance between the two vehicles to reduce significantly (by up to 10 m as shown in this example). In case of applications such as automated driving, this can be fatally hazardous. However, when the tracking filter is capable of identifying interactions, it is capable to predict the possible next location of the following vehicle to incorporate the adjustment in speed, as shown in Fig. 1 (b) by red circles (predicted particles).

Proposition 1: In an LMB filter, assume that the multi-object prior is an LMB density with parameters given in (15) and the single-object state transition density is interaction-aware and parametrised as $f_{k|k-1}(x|x_{k-1}, \ell; \psi_k^{(\ell)})$. Then, applying the Chapman-Kolmogorov equation (1), leads to an interaction-aware predicted LMB in which the predicted probabilities of existence are given by (18) and the predicted single-object
densities are given by:
\[ p_{k|k-1}^{(i)}(x) = \left\{ \begin{array}{ll}
p_S(x) f_{k|k-1}(x|\ell; \psi_{k-1}^{(i)}) , & \text{if} \quad p_{k-1}^{(i)}(\cdot) \\
p_S(x) p_{k-1}^{(i)}(\cdot) & \end{array} \right. \]  
(23)

Proof: Based on the observation made in Remark 2 and the PGFI for LMB RFS given in (10), the PGFI of the possibly surviving targets, conditional on prior multi-object state \( X_{k-1} \), is given by
\[
G_{T_{k|k-1}}[h|X_{k-1}] = \prod_{\mathbf{x} \in X_{k-1}} \left[ (1 - p_S(x)) + \{ h(\cdot), \right.
\[
\left. p_S(x) f_{k|k-1}(x|\ell; \psi_{k-1}^{(i)}) \} \right].
\]
(24)

The PGFI of the predicted multi-object RFS is then given by
\[
G_{X_{k|k-1}}[h] = \int [h]^Y \pi_{k|k-1}(X) \delta X
\]
(25)

Let us consider to add the birth LMB process at the end of the prediction step. Thus, discarding the birth terms, the term inside the bracket in (25) equals \( G_{T_{k|k-1}}[h|Y] \). Thus we have:
\[
G_{X_{k|k-1}}[h] = \int G_{T_{k|k-1}}[h|X] \pi_{k-1}(X) \delta X
\]
(26)

where
\[
h'(x) = (1 - p_S(x)) + p_S(x) \{ h(\cdot), f_{k|k-1}(x|\ell; \psi_{k-1}^{(i)}) \}.
\]

Thus, \( G_{X_{k|k-1}}[h] = G_{X_{k-1}}[h'] \). Noting that \( X_{k-1} \) is an LMB with parameters \( \left\{ p_{k-1}^{(i)}(\cdot) \right\}_{i \in L(k-1)} \), substituting these parameters in equation (10) results in (27):
\[
G_{X_{k|k-1}}[h] = \prod_{\ell \in L(k-1)} \left[ (1 - r_{k-1}^{(i)}) + \{ h'(), r_{k-1}^{(i)} p_{k-1}^{(i)}(\cdot) \} \right]
\]
(27)

In (27), the inner product can be further expanded by substituting \( h'() \) as follows:
\[
\langle h'(), r_{k-1}^{(i)} p_{k-1}^{(i)}(\cdot) \rangle = \int \left[ (1 - p_S(x)) + p_S(x) \{ h(\cdot), f_{k|k-1}(x|\ell; \psi_{k-1}^{(i)}) \} \right] \pi_{k-1}(x) dx.
\]
(28)

Substituting the inner product term in (27), followed by some algebraic manipulation (that we omit for the sake of brevity) will lead to
\[
G_{X_{k|k-1}}[h] = \prod_{\ell \in L(k-1)} \left[ (1 - \rho^{(i)} + \{ h(\cdot), \rho^{(i)} q^{(i)}(\cdot) \} \right]
\]
(29)

where
\[
\rho^{(i)} = \left\{ p_{k}^{(i)}(\cdot), p_{k-1}^{(i)}(\cdot) \right\}
\]

and
\[
q^{(i)}(X) = \frac{\left\{ p_S(x) f_{k|k-1}(x|\ell; \psi_{k-1}^{(i)}) \right\}}{\left\{ p_S(x), p_{k-1}^{(i)}(\cdot) \right\}}.
\]

Equation (29) matches the mathematical form of the PGFI of an LMB density given by (10). Therefore, the predicted multi-object RFS, \( X_{k|k-1} \) turns out to be LMB with parameters
\[
J_{k-1}^{(i)} = \rho^{(i)} = \left\{ p_{k}^{(i)}(\cdot), p_{k-1}^{(i)}(\cdot) \right\}
\]

\[
p_{k-1}^{(i)}(x) = q^{(i)}(x) = \frac{\left\{ p_S(x) f_{k|k-1}(x|\ell; \psi_{k-1}^{(i)}) \right\}}{\left\{ p_S(x), p_{k-1}^{(i)}(\cdot) \right\}}.
\]

V. SEQUENTIAL MONTE CARLO IMPLEMENTATION

An overall flowchart of the method is presented in Fig. 2, where the LMB filter steps are shown in clear boxes and the steps added in the proposed method are presented in green color. In an SMC implementation (also called particle implementation) of the LMB filter, the density of each Bernoulli component is approximated by particles and their weights as follows [19],
\[
p_{k-1}^{(i)}(x) = \sum_{j=1}^{J_{k-1}^{(i)}} \omega_{k-1,j}^{(i)} \delta (x - x_{k-1,j}^{(i)})
\]
(32)

where \( \sum_{j=1}^{J_{k-1}^{(i)}} \omega_{k-1,j}^{(i)} = 1 \).

To prevent particle death, the particles are commonly resampled at the end of previous iteration which leads to a new set of particles with equal weights. In this case we have:
\[
\forall j \in [1 : J_{k-1}^{(i)}], \quad \omega_{k-1,j}^{(i)} = 1/J_{k-1}^{(i)}.
\]

From equation (18) the predicted probability of existence is simply given by:
\[
r_{k|k-1}^{(i)} = \eta_{k|k-1}^{(i)} r_{k-1}^{(i)}
\]
(33)
In such applications, the probability of survival $p_S$ is constant, and from equation (37) the weights of the newly sampled particles remain unchanged.

To implement equation (31), we consider two cases: (i) the interaction-aware single-object density $f_{k|k-1}(x_k|x_{k-1}, \ell, \psi_k^{(i)})$ can be directly sampled, (ii) the interaction-aware single-object density is modeled as the product of two components as given by (38), one that is based on a motion model with no interaction included, and one that is focused on modeling target interaction, i.e.

$$
\tilde{f}_{k|k-1}(x_k|x_{k-1}, \ell, \psi_k^{(i)}) \propto \tilde{f}_{k|k-1}(x_k|x_{k-1}, \ell) \times g_{k|k-1}(x_k|x_{k-1}, \psi_k^{(i)}).
$$

In the former case, similar to the original LMB filter, each particle is propagated to a new one according to (39)

$$
\forall j \in [1 : J_{k-1}^{(i)}], \quad x_{k,j}^{(i)} \sim f_{k|k-1}\left(\left|x_{k-1,j}^{(i)}\right|, \ell, \psi_k^{(i)}\right)
$$

and their weights remain unchanged for a constant $p_S$, otherwise simply scaled by $p_S(\cdot)$ values at each particle – see (37). In the latter case, not only the particles are propagated according to

$$
\forall j \in [1 : J_{k-1}^{(i)}], \quad x_{k,j}^{(i)} \sim f_{k|k-1}\left(\left|x_{k-1,j}^{(i)}\right|, \ell\right)
$$

but also their weights are scaled (even with constant $p_S$) according to:

$$
\omega_{k|k-1,j}^{(i)} \propto \omega_{k-1,j}^{(i)} \cdot p_S(x_{k,j}^{(i)} \mid x_{k-1,j}^{(i)}, \psi_k^{(i)}).
$$

### A. Tracking of Coordinated Swarm Targets

Consider an application where targets are known to be likely to move in a coordinated swarm, i.e. while each target maneuvers, it is likely to maintain its distance from the closest target. Also, targets may die anywhere (hence, the survival probability $p_S$ is constant). Each target state transitions from time $k-1$ to time $k$ according to the linear model given by (42)

$$
x_k = Fx_{k-1} + e_k
$$

where $e_k$ is a sample of system noise distributed according to a Gaussian with zero mean and covariance matrix $\Sigma$. Thus, the single-target motion is modelled by the state transition density

$$
\tilde{f}(x_k|x_{k-1}) = \mathcal{N}(x_k; Fx_{k-1}, \Sigma).
$$

To implement target’s maneuvers based on the above model, we first propagate each particle $j$ according to (44):

$$
x_{k,j}^{(i)} = Fx_{k-1,j}^{(i)} + e_k.
$$

Due to the constant probability of survival, the particle weights remain the same.

To model target interactions, we note that the $g_{k|k-1}(x_k|x_{k-1}, \psi_k^{(i)})$ merely affects the particle weights as an importance function – see equation (41). Consider $\hat{X}_{k-1}^{(i)}$ to be the multi-target estimate inferred from LMB parameters at time $k - 1$, i.e. from $\psi_k^{(i)}$. Note that $\hat{X}_{k-1}^{(i)}$ includes all target state estimates except for the one with label $\ell$. A noise-free next state estimate of each single-target
state estimate \( \hat{x}_{k-1} \in \hat{X}_{k-1} \) can be calculated from (42) as follows:

\[
\hat{x}_{k|k-1} = F \hat{x}_{k-1}.
\]  

(45)

Note that labels remain unchanged when calculating (45). We denote the ensemble of such predicted estimates by \( \hat{X}_{k|k-1} \).

For each Bernoulli component in \( \pi_{k-1} \) with label \( \ell \), we first compute a state estimate according to the following:

\[
x^{(\ell)}_{k-1} = \sum_{j=1}^{N^{(\ell)}_{k-1}} a^{(\ell)}_{k-1,j} x^{(\ell)}_{k-1,j}.
\]

Then we find the closest distance from the above estimate to any element of the estimated target states in \( \hat{X}^{(\ell)}_{k-1} \):

\[
d^{(\ell)}_{k-1} = \min_{x \in \hat{X}^{(\ell)}_{k-1}} \text{dist}(x^{(\ell)}_{k-1}, x) \tag{46}
\]

where \( \text{dist}(\hat{x}_{k-1}, x) \) is a distance measure. For example, if the Cartesian coordinates of the location of \( \hat{x}^{(\ell)}_{k-1} \) and \( x \) are \([p_{xk-1}^{(\ell)}, p_{yk-1}^{(\ell)}]^T\) and \([p_{x}, p_{y}]^T\), respectively, then the Euclidean distance between the two is given by (47):

\[
\text{dist}(\hat{x}^{(\ell)}_{k-1}, x) = \left[ (p_{x} - p_{x}^{(\ell)})^2 + (p_{y} - p_{y}^{(\ell)})^2 \right]^{\frac{1}{2}}. \tag{47}
\]

We also denote the label of the closest target estimate by \( l_k^{(\ell)} \), and the state estimate itself by \( \hat{x}^{(l_k^{(\ell)})}_{k} \). The predicted state estimate for this target is a member of \( \hat{X}_{k|k-1} \) given by

\[
\hat{x}^{l_k^{(\ell)}}_{k|k-1} = F \hat{x}^{(l_k^{(\ell)})}_k.
\]

Since target interaction is stated as maintaining the distance from closest target, for each particle \( x_{k|j} \), the closer its distance from \( \hat{x}^{l_k^{(\ell)}}_{k|k-1} \) is to the distance \( d^{(\ell)}_{k-1} \), the more its weight should be increased. Therefore, we have (48):

\[
g_{k|k-1}(x_{k|j}, \psi^{(\ell)}_{k}) = \mathcal{N}(e_{k|j}^{(\ell)}, 0, \sigma_d^2) \tag{48}
\]

where

\[
e_{k|j}^{(\ell)} = d^{(\ell)}_{k-1} - \text{dist}(x_{k|j}, \hat{x}^{(l_k^{(\ell)})}_{k|k-1})
\]

and \( \sigma_d \) is a user-defined parameter to decide how much change in distance to closest target from time \( k-1 \) to \( k \) is acceptable.

Figure 3 shows an example of how particles and their weights would be distributed. With no interactions incorporated into the filter, the particles will be distributed around the next predicted state of the target, all with the same weight, i.e. same shade of red in Fig. 3(a). With the interaction-aware filter, however, the weights of the particles change. As the scenario in Fig. 3 shows, the movements of the closest target to target \( (\ell) \) are in such a way that at time \( k \), the target \( (\ell) \) is more likely to stay at the same location to maintain its distance from the closest target, rather than move according to its motion model. Hence, in Fig. 3(b), although the particles are distributed according to the motion model, those closer to the current location of the target \( (\ell) \) are assigned larger weights (darker shade of red colour in the figure).

**B. Front-Vehicle Follower Model**

Consider another application where numerous vehicles are to be tracked in a busy multi-lane road. The most prominent interaction between vehicles is that every target maintains its distance from the closest target in its front. Incorporating this type of interaction is more complicated than interactions between targets in a coordinated swarm.

Consider the car labeled \( \Xi \) in the scenario shown in Fig. 4. Here, incorporating the interaction is not merely maintaining the shortest distance between car \( \Xi \) and all other cars but only cars that are travelling (i) in the same direction as \( \Xi \) and (ii) in front of it. This way, assuming that at time \( k-1 \) all the cars shown in the figure are detected and their labels and state estimates appear within \( \hat{X}_{k-1} \), care should be taken in the design of the \( g_{k|k-1}(\cdot | x_{k-1}, \psi^{(\ell)}_{k}) \) function to ensure that for car \( \Xi \), the closest car (for its distance to be maintained) is not chosen as \( \Xi \) or \( \Xi \), but \( \Xi \).

In the following, we explain how such a \( g_{k|k-1}(\cdot | x_{k-1}, \psi^{(\ell)}_{k}) \) function can be mathematically devised. The main difference here is the consideration of target velocity direction in identifying the interacting targets. As with the coordinated swarm target case, we first propagate particles according to the linear model in (44). For the purpose of this application, let us assume that the linear motion model \( F \) is described by the
nearly constant velocity (NCV) motion model. After particle propagation, for each unique target label, we identify if there is a possible interaction with a close front vehicle/target. If no such interaction is found, the predicted particle weights are calculated using (37). However, if a possible interaction is identified, the particle weights are calculated according to (41).

In order to determine if a target/vehicle with label \( \ell \) is following another target/vehicle closely, we utilize the estimates from time \( k-1 \), \( \hat{X}_{k-1} \). According to the complexity of the road and the general speed of vehicles, we set a threshold on distance between vehicles \( d_{th} \). If the distance is less than threshold for two vehicles, the vehicle in pursuit is expected to be impacted by the motion of its front vehicle. Therefore, we first calculate the distances between the estimated location of a target with label \( \ell \), \( \hat{x}^{(\ell)}_{k-1} \), and all other target estimates from time \( k-1 \), \( \hat{X}_{k-1}^{(\ell)} \), as described in (47). The set of targets for which the distance from target \( \ell \) falls below \( d_{th} \) are considered to be “near” target \( \ell \), with their labels collected in the set \( L^{(\ell)}_{\text{near}} \):

\[
L^{(\ell)}_{\text{near}} \triangleq \left\{ \ell \in \mathbb{L} - \{ \ell \}; \; \text{dist}(\hat{x}^{(\ell)}_{k-1}, x) \leq d_{th} \right\}. \tag{49}
\]

Remark 3: We note that the detection of interactions is commonly based on criteria that are dependent on vehicles’ states and practically, the estimates returned by the filter will be used. Hence, estimation errors can lead to incorrect detection of interactions or missing existing interactions. To avoid such cases as much as possible, we need to ensure that interaction criteria is defined in such a way that it is robust to state estimation errors. For instance, in our experiments where the distance between two vehicles is interrogated to discover interaction, the threshold is much larger than the average error of distance calculation that may occur due to state estimation errors of the filter.

Let us denote the state estimates of near targets in \( L^{(\ell)}_{\text{near}} \) by \( \hat{X}_{\text{near}}^{(\ell)} \) as a subset of \( \hat{X}_{k-1}^{(\ell)} \). Vehicles in the same lane are bound to be moving in the same direction. The use of NCV motion model indicates that each target state is comprised of the target location and velocity in \( x \) and \( y \) coordinates,

\[
x = [p_x \; \dot{p}_x \; p_y \; \dot{p}_y]^{\top}. \tag{50}
\]

The velocity components \( \hat{v}_x = [\dot{p}_x \; \dot{p}_y]^{\top} \) of the estimates can be used to find the angle between velocity vectors of the target \( \ell \) and all the near vehicles with labels in \( L^{(\ell)}_{\text{near}} \). If this angle is smaller than a threshold \( \alpha_{th} \), the targets can be assumed to have similar direction of motion. This is shown in Fig. 5 where the vehicles with labels \( \ell \) and \( \ell' \) are moving in the same direction hence the angle between their velocity vectors should be very small (even after consideration that noise makes the velocity direction seem slightly varying), as opposed to the angle between velocities of vehicles \( \ell \) and \( \ell' \). According to the angle between velocities of vehicles \( \ell \) and \( \ell' \), we may update the set of near targets to \( \ell \), \( L^{(\ell)}_{\text{near}} \) as follows:

\[
L^{(\ell)}_{\text{near}} \leftarrow \left\{ \ell' \in L^{(\ell)}_{\text{near}}; \; \angle \left( \hat{v}_{x,k-1}^{(\ell)}, \hat{v}_{x,k-1}^{(\ell')} \right) \leq \alpha_{th} \right\}
\]

Fig. 5. An example demonstrating how we can determine whether “near” vehicles are moving in the same or opposite direction by investigating the angle between velocity vectors.

where

\[
\angle \left( \hat{v}_{x,k-1}^{(\ell)}, \hat{v}_{x,k-1}^{(\ell')} \right) \triangleq \cos^{-1} \frac{\left( \hat{v}_{x,k-1}^{(\ell)} \right)^{\top} \left( \hat{v}_{x,k-1}^{(\ell')} \right)}{\left| \left| \hat{v}_{x,k-1}^{(\ell)} \right| \right| \left| \left| \hat{v}_{x,k-1}^{(\ell')} \right| \right|}.
\]

The above condition excludes all vehicles travelling in a direction different from vehicle \( \ell \) from possibly interacting targets. However, it may still include a vehicle travelling closely behind vehicle \( \ell \). Therefore, we check the position vector of each vehicle in the updated set of “near” labels, relative to vehicle \( \ell \):

\[
\hat{p}^{(\ell \rightarrow \ell')} \triangleq \left[ \left( \hat{p}_{x,k-1}^{(\ell)} - \hat{p}_{x,k-1}^{(\ell')} \right) \left( \hat{p}_{y,k-1}^{(\ell)} - \hat{p}_{y,k-1}^{(\ell')} \right) \right]^{\top} \tag{51}
\]

and investigate whether it is in around the same direction as the vehicle velocity vector \( \hat{v}_{x,k-1}^{(\ell)} \) or not. This is simply done by comparing the x and y components of the position of vehicles \( \ell \) and \( \ell' \), according to the x and y components of \( \hat{v}_{x,k-1}^{(\ell)} \), respectively. Furthermore, we ascertain if a “near” vehicle is actually moving in front of the vehicle \( \ell \) and not behind it. Figure 6 shows an example to demonstrate how we run this check. Consider the two “near” vehicles \( \ell'_1 \) and \( \ell'_2 \). For \( \ell'_1 \) which is behind the target vehicle \( \ell \), the angle between its relative position vector \( \hat{p}^{(\ell \rightarrow \ell'_1)} \) and the target vehicle velocity vector \( \hat{v}_{x,k-1}^{(\ell)} \) is very large (close to 180°), while this angle is small for the vehicle \( \ell'_2 \) that is in front of \( \ell \). Thus, we can shrink the set of “near” vehicle labels further as follows:

\[
L^{(\ell)}_{\text{near}} \leftarrow \left\{ \ell' \in L^{(\ell)}_{\text{near}}; \; \angle \left( \hat{v}_{x,k-1}^{(\ell)}, \hat{p}^{(\ell \rightarrow \ell')} \right) \leq \beta_{th} \right\}
\]

where \( \beta_{th} \) is a threshold.

The last step is to find the closest vehicle that is still included as a “near” vehicle in \( L^{(\ell)}_{\text{near}} \) (i.e. the vehicle that is driven right in front of \( \ell \) and is not too far from it, hence
interacting with it:

$$t_k^{(l)} = \arg \min_{\ell' \in L_{local}^{(k)}} \text{dist}(\hat{x}_{k-1}^{(l)}, \hat{x}_{k-1}^{(\ell')}).$$ (52)

Once the label of the interacting target, $$t_k^{(l)}$$, is found, similar to Section V-A, first its distance is calculated according to $$d_{k-1}^{(l)} = \text{dist}(\hat{x}_{k-1}^{(l)}, \hat{x}_{k-1}^{(t_k^{(l)})})$$, then its estimate is propagated to time $$k$$ via $$\hat{x}_{k|k-1}^{(l)} = F \hat{x}_{k-1}^{(t_k^{(l)})}$$ then for each particle $$x_{k,j}^{(l)}$$, the error term $$e_{k,j}^{(l)} = d_{k-1}^{(l)} - \text{dist}(x_{k,j}^{(l)}, \hat{x}_{k|k-1}^{(l)})$$ is calculated and used within the density term in equation (48), which is finally used to increase or decrease the weights of those particles in effect of interaction.

VI. EXPERIMENTAL RESULTS

The proposed method has been tested on a complex road intersection scenario, where interactions between targets are identified using the steps outlined in Section V-B. An aerial video of the Swindon M4 motorway junction (J-16) in the U.K has been selected, which is shot by a high functioning drone camera (DJI inspire 2). A sample image from the dataset is shown in Fig. 7. The dataset contains a large number of vehicles at any time, usually more than 60 vehicles per frame. The intersection has different challenges like heavy, fast traffic as well as traffic lights which control the motion of vehicles on the road. The size of the target vehicles is very small in the images, making it a challenging tracking task. The vehicles undergo various maneuvers like varying speeds, turns and lane changes. The proposed method is capable of identifying all the mentioned maneuvers, rather than ignoring some maneuvers for simplicity. Furthermore, we identify target interactions based on distance between two vehicles, without using any dataset-specific information. A total of 245 consecutive frames have been selected from the video for our implementation. The camera is assumed to be stationary for all images. The ground truth for 200 vehicles has been manually annotated for the dataset as part of this research, using the MATLAB’s ground truth labeler application, and MATLAB is used for producing all simulation results. The measurements have been obtained by applying a noisy detector to the ground truth. The proposed method uses LMB filter for tracking where there is no interaction, and the interactive weight update from (41) has been implemented for vehicles with identified interactions. In addition to the performance comparison discussed in this section, a results video is provided as supplementary material to this paper. The video shows each frame of the dataset at a frame rate of 2 frames per second (fps). For each frame, ground truth has been represented with green dots on the centre point of the target vehicle and estimates are represented with red boxes around the vehicles, with target labels written in red to identify any possible label switching.

As the estimated target velocities are used for identifying any possible interactions, the interaction model is implemented after 5 time frames so that the estimated velocities have been well adjusted according to the measurement and motion models. The NCV motion model is used in which the state of each target is described according equation (50). In this model, the single-target state transition density is given by:

$$f (x_k|x_{k-1}) = N'(x_k; Fx_{k-1}, Q)$$ (53)

where $$F$$ is the state transition matrix and $$Q$$ is the process noise covariance matrix:

$$F = \text{diag} (A, A) \quad Q = \text{diag} (B, B)$$ (54)

and $$A$$ and $$B$$ are matrices given by:

$$A = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad B = \sigma_{\text{motion}}^2 \begin{bmatrix} T^3/3 \\ T^2/2 \\ T \end{bmatrix}$$ (55)

in which $$T$$ is the time step, which is set equal to 1 s. The $$\sigma_{\text{motion}}^2$$ is 7 pixels/s². The probability of target detection $$P_d$$ is 0.995 and probability of survival $$P_s$$ is 0.99. The observation noise is a zero mean Gaussian with variance 3 pixels/s². The target birth is generated at all entry points on the road in the scenario, as well as some central points at the intersection. The birth model used is an LMB RFS according to $$(r_B^{(l)}, p_B^{(l)})_{l \in \mathbb{R}}$$ where $$r_B^{(l)} = 0.2$$ for all birth components.

An example of the target estimation achieved by the proposed method is shown in Fig. 8. Here, the frame 70 of the dataset is shown, with the ground truth location for each target/vehicle is shown using green square and the estimated location is shown by a red square. The red squares are intentionally kept bigger to make the image easier to read. The velocity of each vehicle is shown by yellow vectors as estimated by the proposed method. It can be seen that the proposed method is not only estimating the locations of vehicles efficiently, but also realistically estimating the
velocity magnitude and direction for each vehicle, even in the presence of noise. The slight variation in estimated velocity direction is due to the presence of noise.

The performance of the proposed interaction-aware tracking method has been compared with one of the fundamental RFS-based filters, i.e., the PHD filter [68] and the LMB filter [19] using the sequential Monte-Carlo implementation, with the same simulation parameters used for all methods. We use $N = 200$ particles and tracks with probabilities of existence lower than $10^{-4}$ have been pruned. Owing to a large number of vehicles present in each frame and the size of each target vehicle being very small, interaction identification proved to be very challenging. Therefore, in the results presented, we have not implemented the step presented in Fig. 6 for interaction identification, as it resulted in some adjacent lane vehicles identified as interacting with each other. Performance evaluation has been conducted using a number of metrics, first of which is the optimal sub-pattern assignment (OSPA). The OSPA metric measures the error between ground truth and estimated tracks. It has three components, the overall OSPA error, the localisation error and cardinality error. The OSPA metric has two parameters, i.e. the cut-off for error $c$, which has been set to 100 and the order $p$, which has been set to 2 for the purpose of performance evaluation. It can be seen from Fig. 9 that the proposed method outperforms the PHD filter significantly and the LMB filter for most time steps. While the PHD filter is able to maintain the cardinality estimate better than the other methods, it should be noted that this is due to the presence of false alarm estimates obtained in the PHD filter tracking, as shown for a section of scenario in Fig. 10. Due to these false estimates, the filter is able to track the cardinality estimate but the localisation error is much greater than the other two methods. Owing to this as well as the absence of labels, we have not included PHD filter in the subsequently presented performance evaluation. Since most of the error values are seen for the overall OSPA error and the OSPA cardinality, we have calculated an average error difference for these for the LMB filter and the proposed method. The overall average of OSPA error difference is calculated to be 1.666 and OSPA-cardinality error as 2.108.

Another commonly used metric for evaluation of tracking performance is the OSPA$(^2)$, which calculates cumulative error over a specific window length. In addition to the cut-off and order parameters, which have been kept same as OSPA metric for consistency, the OSPA$(^2)$ window length has been set to 5. It can be seen from the OSPA$(^2)$ plots in Fig. 11 that the proposed method outperforms the LMB filter. The overall average error difference for LMB and proposed method for OSPA$(^2)$ is 0.195 and for OSPA$(^2)$ cardinality is $-0.115$.

In order to further depict the efficacy of the proposed method, we have plotted a comparison of the cardinality error for all time frames, where cardinality error has been calculated as:

$$\text{Card. Error} = \text{Ground Truth Card.} - \text{Estimated Card.}$$

The cardinality errors for LMB filter and the proposed method are depicted in Fig. 12. The portions of the plot highlighted with red ellipses show examples of the case when the cardinality error for LMB filter is higher than the proposed method, while the blue color ellipses show samples of cases where our method has a higher cardinality error. It can be clearly seen that our proposed interaction-aware LMB filter returns a more accurate cardinality than the traditional LMB filter in a large portion of the time frames. The cardinality
error highlights the fact that while the LMB filter sometimes loses a track, our method is far less prone to this issue. The interaction-aware LMB filter may result in over-estimation for a few frames, as shown by negative cardinality error. However, it should be noted that in these instances the LMB filter has a greater cardinality error (missing targets) than the interaction-aware LMB filter for the set of measurements used. We have investigated the over-estimation achieved by the proposed filter by plotting the estimates of targets on the dataset frames for both the proposed filter and LMB filter. One example of this can be seen at frame 116 of the dataset, as shown in Fig. 13. The green boxes represent the ground truth for the frame, red squares depict the target estimates, and the cyan ellipse depicts the cardinality error present in the estimation for the frame. The cardinality error for this frame represents the incorrect estimation of target with label (1, 9), which exited the scene at the previous frame, i.e. frame 115. This type of cardinality error is common in MOT applications, especially when there is a large number of targets. However, it should be noted that this target is removed from the estimates from the very next frame, which leads to zero cardinality error for subsequent frames. Furthermore, this cardinality error only occurs for 4 frames out of a total of 245 frames, which clearly shows that it is not a significant issue compared to missed targets, which are seen many frames for the LMB filter. Overall, it can be seen from the metrics presented that the interaction-aware LMB filter has generally improved performance even in the selected scenario, which is quite complex due to the large and varying number of targets, the number of maneuvers, and the varying speeds of targets in different areas of the scene. The difference in tracking is highlighted most in terms of cardinality error.

VII. CONCLUSION

A novel approach was presented to incorporate target interactions into the prediction step of a RFS-based multi-target filter. The filter of choice in this paper is the LMB filter, but the proposed approach can be directly formulated into other similar filters such as the δ-GLMB filter. The main idea was to incorporate interactions in the form of extra parameters involved in the single-target state transition density, with the parameters being time-varying, chosen as parts of the multi-target density information from the previous time step.

In two practical examples, we elaborated further on how target interactions can be incorporated into the SMC implementation of the LMB filter, resulting in particle weight changes in the prediction step of the filter. We presented a challenging tracking scenario in which a large number of vehicles are moving and interacting in a complex multi-way intersection. The results demonstrated how the incorporation of interaction into the filter improves the tracking results in terms of both the OSPA and OSPA(2) error metrics.

In general, this paper clearly demonstrates how the incorporation of information into the Bayesian filtering process can improve the estimation of states and labels of multiple targets in ITS applications. In essence, RFS filters were originally invented with this purpose in mind. They provided a mathematically solid way to incorporate all target- and scene-related information into the prediction step and all measurement-related information into the update step of the filter. The motion model, the probability of survival, and the birth model are examples of the target- and scene-related information. The likelihood function, the probability of detection and the clutter model are examples of measurement-related information. In our work, we added the information that are available regarding interactions between targets (emphasizing that they are target-related information) into the prediction step of the most advanced RFS filter. This could be further enhanced by incorporating more information such as the road information (scene-related) into the prediction step as well. Indeed, we are currently working on this as an area of further research.

REFERENCES

[1] H. Lee, J. Yoon, Y. Jeong, and K. Yi, “Moving object detection and tracking based on interaction of static obstacle map and geometric model-free approach for urban autonomous driving,” IEEE Trans. Intell. Transp. Syst., vol. 22, no. 6, pp. 3275–3284, Jun. 2021.
[2] H. Lee, H. Lee, D. Shin, and K. Yi, “Moving objects tracking based on geometric model-free approach with particle filter using automotive LiDAR,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 10, pp. 17863–17872, Oct. 2022.
[3] Y. Liu et al., “Dynamic lane-changing trajectory planning for autonomous vehicles based on discrete global trajectory,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 7, pp. 8513–8527, Jul. 2022.
[4] X. Chen, J. Yin, K. Tang, Y. Tian, and J. Sun, “Vehicle trajectory reconstruction at signalized intersections under connected and automated vehicle environment,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 10, pp. 17986–18000, Oct. 2022.
C. Fantacci, B.-N. Vo, B.-T. Vo, G. Battistelli, and L. Chisci, “Robust fusion for multisensor multioject tracking,” IEEE Signal Process. Lett., vol. 25, no. 5, pp. 640–644, May 2018.

S. Li, W. Yi, R. Hoseinnezhad, G. Battistelli, B. Wang, and L. Kong, “Robust distributed fusion with labeled random finite sets,” IEEE Trans. Signal Process., vol. 66, no. 2, pp. 278–293, Jan. 2018.

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.

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.

B. Wang, W. Yi, R. Hoseinnezhad, S. Li, L. Kong, and X. Yang, “Distributed fusion with multi-Bernoulli filter based on generalized covariance intersection,” IEEE Trans. Signal Process., vol. 65, no. 1, pp. 242–255, Jan. 2017.

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.

G. Li, G. Battistelli, W. Yi, and L. Kong, “Distributed multi-sensor multi-view fusion based on generalized covariance intersection,” Signal Process., vol. 166, Jan. 2020, Art. no. 107246. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165168419302920

X. Wang, A. K. Gostar, T. Rathnayake, B. Xu, A. Bab-Hadiashar, and R. Hoseinnezhad, “Centralized multiple-view sensor fusion using labeled multi-Bernoulli filters,” Signal Process., vol. 150, pp. 75–84, Sep. 2018.

A. K. Gostar et al., “Cooperative sensor fusion in centralized sensor networks using Cauchy–Schwarz divergence,” Signal Process., vol. 167, Feb. 2020, Art. no. 107278.

A. K. Gostar, R. Hoseinnezhad, and A. Bab-Hadiashar, “Multi-Bernoulli sensor control via minimization of expected estimation errors,” IEEE Trans. Aerosp. Electron. Syst., vol. 51, no. 3, pp. 1762–1773, Jul. 2015.

A. K. Gostar, R. Hoseinnezhad, and A. Bab-Hadiashar, “Multi-Bernoulli sensor-selection for multi-target tracking with unknown clutter and detection profiles,” Signal Process., vol. 119, pp. 28–42, Feb. 2016.

A. K. Gostar, R. Hoseinnezhad, A. Bab-Hadiashar, and W. Liu, “Sensor-management for multitarget filters via minimization of posterior dispersion,” IEEE Trans. Aerosp. Electron. Syst., vol. 53, no. 6, pp. 2877–2884, Dec. 2017.

A. K. Gostar, R. Hoseinnezhad, T. Rathnayake, X. Wang, and A. Bab-Hadiashar, “Constrained sensor control for labeled multi-Bernoulli filter using Cauchy–Schwarz divergence,” IEEE Signal Process. Lett., vol. 24, no. 9, pp. 1313–1317, Sep. 2017.

X. Wang, R. Hoseinnezhad, A. K. Gostar, T. Rathnayake, B. Xu, and A. Bab-Hadiashar, “Multi-sensor control for multi-object Bayes filters,” Signal Process., vol. 142, pp. 260–270, Jan. 2018.

X. R. Li and V. P. Jilkov, “Survey of maneuvering target tracking. Part I. Dynamic models,” IEEE Trans. Aerosp. Electron. Syst., vol. 39, no. 4, pp. 1333–1364, Oct. 2003.

B.-N. Vo, S. Singh, and A. Doucet, “Sequential Monte Carlo implementation of the PHD filter for multi-target tracking,” in Proc. 6th Int. Conf. Inf. Fusion, 2003, pp. 792–799.