2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

3D Multi-Object Tracking using Random Finite Set-based Multiple Measurement Models Filtering (RFS-M^3) for Autonomous Vehicles

Su Pang, Daniel Morris and Hayder Radha

***Abstract*— Multiple object tracking (MOT) is a critical module for enabling autonomous vehicles to achieve safe planing and navigation in cluttered environments. In tracking-by-detection systems, there are inevitably many false positives and misses among learning-based input detections. The challenge for MOT is to combine these detections into tracks, and filter them based on their uncertainties, states, and temporal consistency to achieve accurate and persistent tracks. In this paper, we propose to solve the 3D MOT problem for autonomous driving applications using a random finite set-based (RFS) Multiple Measurement Models filter (RFS-M^3). In partiuclar, we propose multiple measurement models for a Poisson multi-Bernoulli mixture (PMBM) filter in support of different application scenarios. Our RFS-M^3 filter can naturally model these uncertainties accurately and elegantly. We combine the learning-based detections with our RFS-M^3 tracker through incorporating the detection confidence score into the PMBM prediction and update step. The superior experimental results of our RFS-M^3 tracker on *Waymo*, *Argoverse* and *nuSceness* datasets illustrate that our RFS-M^3 tracker outperforms state-of-the-art deep learning-based and traditional filter-based approaches. To the best of our knowledge, this represents a first successful attempt for employing an RFS-based approach in conjunction with 3D learning-based amodal detections for 3D MOT applications with comprehensive validation using challenging datasets made available by industry leaders.**

I. Introduction

Autonomous vehicles need robust and accurate 3D perception to achieve safe maneuvering within a cluttered environment. There are three main problems in 3D perception: 3D object detection, multiple object tracking (MOT) and object trajectory forecasting. In a modular perception system, MOT is a critical module that connects detection and forecasting.

For tracking-by-detection approaches, the impact of the quality of input detections that are provided by the underlying detector is of a paramount importance. However, due to the complexity of cluttered environments and limitations of learning-based detectors, there are many false positives, misses and inaccurate detections among input detections. Therefore, the main challenges for MOT in autonomous driving applications are threefold: (1) uncertainty in the number of objects; (2) uncertainty regarding when and where the objects may appear and disappear; (3) uncertainty in objects' states.

The family of Random Finite Set (RFS) [1], [2], [3] based approaches are theoretically sound Bayesian frameworks that naturally model the aforementioned uncertainties accurately and elegantly. While traditional filtering based methods, such as Kalman filtering [4], [5], [6], perform well in state estimation, they are not designed to model the unknown number of objects, or the so-called *birth and death* phenomena of objects. RFS-based MOT algorithms address these problems from a Bayesian perspective and have been shown to be very effective in radar-based applications [7], [8]. Among RFS-based approaches, Poisson multi-Bernoulli mixture (PMBM) filtering has shown superior performance and favourable computational cost [9]. However, the 3D input detection format (3D bounding boxes) for modern autonomous driving systems is significantly different from raw radar signals (points and Doppler velocity). Applying RFS for 2D/3D amodal detections (bounding boxes) from learning-based detections has not been well explored. Existing works in this area either under-perform state-of-the-art trackers or they have been tested using a small dataset that do not reflect broad and truly challenging scenarios [10], [11], [12].

There are multiple evaluation metrics for 3D MOT corresponding to different application scenarios. The main dif-

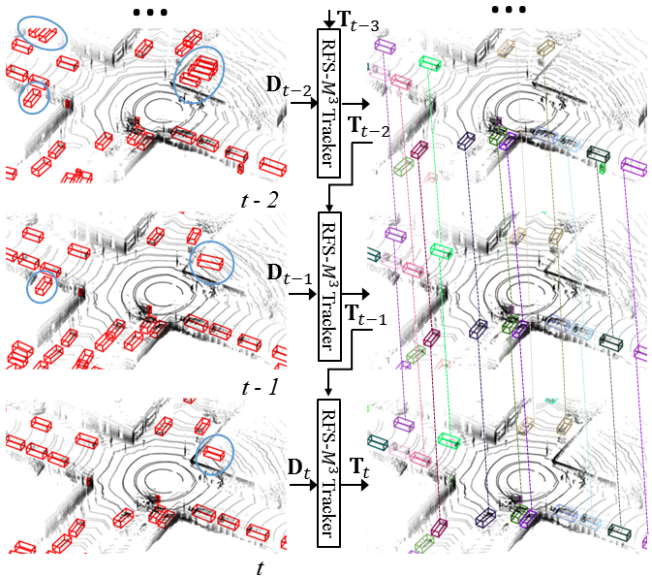

Fig. 1: Overview of the proposed RFS-M^3 tracker pipeline. For each frame, many 3D detections are generated by a neural-network-based 3D detector, as the red bounding boxes on the left column. Our RFS-M^3 tracker successfully tracks targets and filters out false positives (boxes shown within the blue ellipses). For figures in the right column, different bounding box colors correspond to different unique tracked IDs. Some tracks with the same IDs are connected with dashed lines to help visualization. Best viewed in color.

Su Pang, Daniel Morris and Hayder Radha are with the Department of Electrical and Computer Engineering, College of Engineering, Michigan State University, 220 Trowbridge Road, East Lansing, Michigan, 48824, United States. Email: pangsu@msu.edu, dmorris@msu.edu, radha@egr.msu.edu

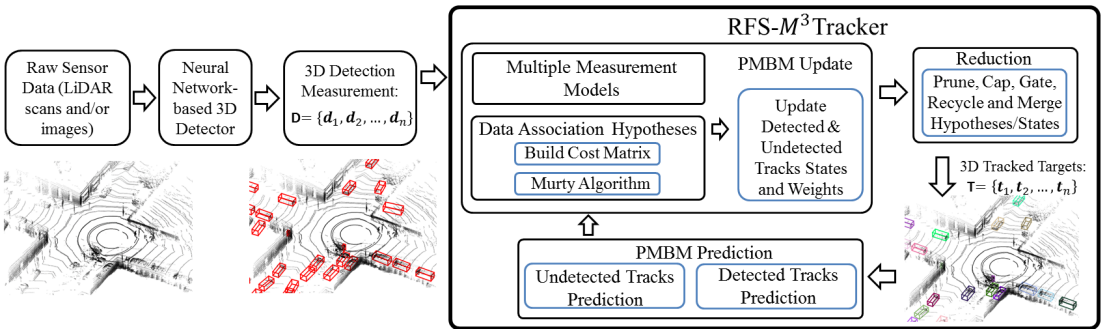

Fig. 2: RFS-M^3 system architecture. The system works in a recursive way. Started at tracks from previous timestep, PMBM prediction is done first to predict each track's state and weight, then according to the detection measurements for the current timestep, we form different global association hypotheses from possible combinations of single target hypotheses (track-measurement pairs). Then the update of tracks is done based on the data association. Finally the update outputs are sent to reduction module to remove unlikely and redundant results and output filtered tracks for the current timestep.

ference resides in the definition of *True Positive metrics* (TP metrics). Waymo [13] uses the strictest TP metric: 3D intersection over union (IoU). This requires trackers to not only perform well in data association but also perform well in estimating tracks' 3D locations, dimensions and orientations. The TP metrics for Argoverse [14] and nuScenes [15] are calculated only using a center distance threshold at 2 meters during matching. These different metrics and application scenarios require different system designs for optimizing onboard computing resources.

We propose an RFS-based Multiple Measurement Models filter (RFS-M^3) to solve the 3D MOT problem with amodal detections for autonomous driving applications. In particular, we propose multiple measurement models ranging from 3D bounding box model to point measurement model for a PMBM filter in support of different TP metrics and optimize the usage of computing resources. Furthermore, our framework supports *amodal* detections, which implies the ability to track objects that are only partially visible due to partial occlusions by other objects. This partial occlusion phenomena represents a challenging and realistic condition that should be addressed by any viable tracking solution. The contributions of our paper are as follows:

- To the best of our knowledge, this represents a first successful attempt for employing an RFS-based approach that incorporates amodal 3D detections from a neural network for 3D MOT.
- Multiple measurement models including 3D bounding box model and point model are incorporated in our RFS-M^3 in support of different MOT application scenarios.
- Our RFS-M^3 is an online real-time tracker with multiple global hypotheses maintained, and can run at an average rate of 20 Hz on a standard desktop.
- We validate the performance of our RFS-M^3 tracker using three extensive open datasets provided by three industry leaders – Waymo [13], Argoverse [16] and nuSceness [15]. It is worth noting that among entries that use the organizer provided detections, our RFS-M^3 ranks **No.2** in all-class MOTA[1] on the Waymo dataset.

The rest of the paper is organized as follows. We first review related works in section II. Then, we introduce the standard problem formulation for 3D MOT under autonomous driving applications as well as our proposed system overview in section III. In section IV, we illustrate the details of our RFS-M^3 tracker. We report and analyze our experimental results on Waymo, Argoverse and nuScenes datasets in section V. In section VI, we conclude the paper.

II. Related Work

In this section we focus on MOT methods for autonomous driving applications.

A. MOT using Traditional Filtering

Kalman filter [4] and its variants are the most popular approaches in this category. Weng *et al* [5] uses a combination of 3D Kalman filter and Hungarian algorithm for state estimation and data association. This method can achieve reasonable performance with very low computational cost and became the baseline methods for many 3D tracking competitions [15], [16], [13]. Chiu *et al* [6] modified [5] by using stochastic information from the Kalman filter in the data association step by measuring the Mahalanobis distance between predicted object states and detections. But the simple birth and death management of targets and single distance-based association make these methods struggle in cluttered environments.

B. MOT using Neural Networks

Compared to traditional filtering-based approaches, recently developed neural network-based methods can capture the descriptive features and temporal motion features from raw sensor data for MOT. Frossard and Urtasun apply a convolutional neural network (CNN)-based Match Network to compute a matching cost score to formulate the data association problem as a linear program [17]. Similarly, FANTrack [18] uses a CNN to learn a similarity function

[1]Multi-object tracking accuracy, MOTA in short, is the primary metric for ranking in most MOT benchmarks.

that combines cues from both image and spatial features of objects. Weng *et al* proposes GNN3DMOT to use a graph neural network to improve the discriminative feature learning for MOT [19]. Some other methods [20], [21], [22], [23] combine MOT with detection or forecasting to reduce the system complexity. These learning-based approaches use complicated networks and require a great deal of training. They have great potential but the tracking performance is so far similar or slightly worse compared to many filtering-based methods according to popular open 3D tracking benchmark leaderboards [24], [14], [25].

C. MOT using RFS

A recent family of MOT algorithms are based on RFS [1], [2], [3], including probability hypothesis density (PHD) filter [26], cardinalized PHD filter [27], generalized labeled multi-Bernoulli (GLMB) [28] and PMBM [29], [30]. PHD filter and CPHD filter are two examples of moment approximations of the multi-object density. GLMB and PMBM are examples of using multi-object conjugate priors. Among these RFS-based filtering methods, PMBM filtering has shown superior performance and favourable computational cost [9] when compared to other RFS-based approaches. RFS-based MOT algorithms have been shown to be very effective for point target and extended shape target measurement models MOT applications [7], [8], [12]. However, the 3D input detection format (classified 3D/2D bounding boxes) for modern autonomous driving systems is significantly different from point/extended target models. Applying RFS for 2D/3D amodal detections (bounding boxes) from learning-based detections has not been well explored. Existing works in this area either under-perform state-of-the-art trackers or they have been tested using a small dataset that do not reflect broad and truly challenging scenarios [10], [11], [12].

III. Problem Formulation and System Overview

Our problem formulation and system pipeline are summarized in Fig 2. This includes an RFS-M^3 tracker designed to solve the 3D MOT problem given a set of noisy 3D detections, as well as estimate the number of objects, and maintain identity and state of each object. The details of the RFS-M^3 tracker is further illustrated in section IV

Our work is an online tracking-by-detection system with 3D detections as input for each step. The standard format for n 3D detections in one frame can be defined as follows:

$$\begin{aligned}\mathbf{D} &= \{\mathbf{d_1}, \mathbf{d_2}, ...\mathbf{d_n}\}, \\ \mathbf{d_i} &= [x_i, y_i, z_i, h_i, w_i, l_i, \theta_i, s_i, cls_i]\end{aligned} \tag{1}$$

where $\mathbf{D}$ is the set of all n detections in one frame, for i_{th} detection $\mathbf{d_i}$, x_i, y_i, z_i is the the center location in 3D space, h_i, w_i, l_i denote the height, width and length of the 3D bounding box, θ_i is the orientation around z axis, s_i and cls_i are the confidence score and class name (vehicle, pedestrian and so on) respectively. Note that our RFS-M^3 tracker module can work with any 3D detectors that output standard 3D detections. The 3D detectors can be either image-based [31], [32], LiDAR-based [33], [34] or multi-sensor fusion-based [35], [36]. But the impact of the quality of input detections is of a paramount importance.

The ouput of our RFS-M^3 tracker for each frame is a set of m tracked 3D targets $\mathbf{T}$ defined as follows:

$$\begin{aligned}\mathbf{T} &= \{\mathbf{t_1}, \mathbf{t_2}, ...\mathbf{t_m}\}, \\ \mathbf{t_j} &= [x_j, y_j, z_j, h_j, w_j, l_j, \theta_j, weight_j, cls_j, id_j]\end{aligned} \tag{2}$$

where $weight_j$ is the weight of the target, it models the probability of the corresponding data association hypothesis. id_j denotes the global unique tracked ID of the target.

IV. RFS-M^3 Tracker with Multiple Measurement Models

The high-level system of the proposed RFS-M^3 MOT tracker architecture is shown in Fig 2. This illustrates the four primary components of the tracker: (1) PMBM Predictions; (2) Data Association; (3) PMBM Update; and (4) Reduction.

A. Detected and Undetected Tracks

Under the PMBM model [29], [30], the set of tracks $\mathbf{x_t}$ at timestamp t is the union of detected tracks $\mathbf{x}_t^d$ and undetected tracks $\mathbf{x}_t^u$. Detected tracks $\mathbf{x}_t^d$ are tracked objects that have been detected at least once. Undetected tracks $\mathbf{x}_t^u$ are potential objects that have not been detected. Note that we are not explicitly tracking the undetected tracks, which is impossible under a tracking-by-detection framework. Instead, we have a representation of their possible existences.

B. Object States with Different Measurement Models

There are two versions of object states corresponding to two measurement models: a 3D bounding box version and a point version. For 3D bounding boxes, the object state, $\mathbf{x^{3D}}$, and measurement, $\mathbf{z^{3D}}$, are defined as:

$$\begin{aligned}\mathbf{x^{3D}} &= [x, y, z, h, w, l, \theta, v_x, v_y, v_z, v_\theta] \\ \mathbf{z^{3D}} &= [x, y, z, h, w, l, \theta]\end{aligned} \tag{3}$$

where v_x, v_y, v_z represent the velocity of objects in 3D space, v_θ denotes the yaw angle velocity. x, y, z, h, w, l, θ are measurable states, while v_x, v_y, v_z, v_θ are unmeasureable states. Similarly, for point objects, the state, $\mathbf{x^{pt}}$, and measurement, $\mathbf{z^{pt}}$, are defined as:

$$\begin{aligned}\mathbf{x^{pt}} &= [x, y, v_x, v_y] \\ \mathbf{z^{pt}} &= [x, y]\,.\end{aligned} \tag{4}$$

As discussed in the Experiment section, each version has its own unique attributes for handling specific tracking TP metrics. Note that besides the 3D bounding box version, the outputs from the point model version are also tracked 3D bounding boxes for 3D targets; however, the missing 3D information in the 2D point-target state is added from the associated 3D detection measurement.

C. Data Association Hypotheses

For each timestamp, there are multiple hypotheses for data association. In our measurement-driven framework, each measurement is one of three options: a newly detected target, a previously detected target, or a false positive detection. We form different global association hypotheses from possible

combinations of the single target hypothesis (STH). In our framework, one measurement can only be associated to one object in one global association hypotheses.

One of the advantages of our RFS-M^3 tracker is that we maintain multiple global hypotheses instead of only one. For the k_{th} global hypothesis in one frame, assuming there are n tracked targets (detected tracks) given m detection measurements, the m by $(n+m)$ cost matrix is built as follows:

$$L^k = \begin{bmatrix} -l_{1,1,k} & -l_{1,2,k} & \cdots & -l_{1,n,k} & -l_{1,0} & \infty & \cdots & \infty \\ -l_{2,1,k} & -l_{2,2,k} & \cdots & -l_{2,n,k} & \infty & -l_{2,0} & \cdots & \infty \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -l_{m,1,k} & -l_{m,2,k} & \cdots & -l_{m,n,k} & \infty & \infty & \cdots & -l_{m,0} \end{bmatrix} \quad (5)$$

The left $m \times n$ block contains the negative logarithm pair costs for associating each of m detection measurements to each of n previously tracked targets (detected tracks). For element $-l_{i,j,k}$ in the i_{th} row and j_{th} column, it represents the negative logarithm cost of associating the j_{th} tracked target to the i_{th} detection measurement and is calculated as $-(log(w^d_{i,j}) - log(w^d_{0,j}))$. Here $w^d_{i,j}$ is the target hypothesis-measurement pair weight, and $w^d_{0,j}$ is the missed detection weight. The calculation details of these weights are given in Section IV-F. The right $m \times m$ square block contains negative logarithm costs for generating new targets from detection measurements.

Murty's algorithm [37], an extension of the Hungarian algorithm [38] is used to generate K best global hypotheses instead of only one. The weights for each global hypothesis is based on the product of its detected tracks' weights.

D. PMBM Density

Under the PMBM model, Poisson RFS, also named as Poisson point process (PPP), is used to represent undetected tracks, and a multi-Bernoulli mixture (MBM) RFS is used to represent detected tracks [29], [30]. The PMBM density can be expressed as:

$$\mathcal{PMBM}_t(\mathbf{x}) = \sum_{\mathbf{x}^u \uplus \mathbf{x}^d = \mathbf{x}} \mathcal{P}_t(\mathbf{x}^u) \mathcal{MBM}_t(\mathbf{x}^d) \quad (6)$$

where $\mathbf{x}$ represents all the objects in the surveillance area, and $\mathbf{x}$ is the disjoint union set of undetected tracks $\mathbf{x}^u$ and detected tracks $\mathbf{x}^d$. Symbols $\mathcal{P}(\cdot)$ and $\mathcal{MBM}(\cdot)$ are the Poisson point process density and multi-Bernoulli mixture density, respectively. We assume that the PPP mixture intensity and Bernoulli state probability density functions are Gaussian. Therefore, the parameters for N^u undetected tracks PPP densities are $\left\{w^u_i, \boldsymbol{\mu}^u_i, \boldsymbol{\Sigma}^u_i\right\}_{i=1}^{N^u}$, where w^u_i is the weight for i_{th} undetected track, $\boldsymbol{\mu}^u_i$ and $\boldsymbol{\Sigma}^u_i$ are the state mean and covariance respectively. The parameters for N^d detected tracks MBM densities are: $\left\{w^d_j, r^d_j, \boldsymbol{\mu}^d_j, \boldsymbol{\Sigma}^d_j\right\}_{j=1}^{N^d}$, where w^d_j and r^d_j are the weight of j_{th} detected track and the probability of existence, and $\boldsymbol{\mu}^d_j$ and $\boldsymbol{\Sigma}^d_j$ are the Gaussian state variables. The PMBM density parameters are the combination of PPP density parameters and MBM density parameters. The PMBM prediction and update are the processes of predicting and updating the PMBM parameters.

E. PMBM Prediction

A crucial aspect of the PMBM filter is its *conjugacy* property, which was proved in [30]. The notion of conjugacy is critical for robust and accurate Bayesian-based MOT. In summary, the conjugacy of the PMBM filter inplies that if the prior is in a PMBM form, then the distribution after the Bayesian prediction and update steps will be of the same distribution form. Therefore, the prediction stage of a PMBM filter can be written as:

$$\mathcal{PMBM}_{t+1|t}(\mathbf{x}_{t+1}) = \int p(\mathbf{x}_{t+1}|\mathbf{x}_t) \mathcal{PMBM}_{t|t}(\mathbf{x}_t) \delta \mathbf{x}_t \quad (7)$$

where $p(\mathbf{x}_{t+1}|\mathbf{x}_t)$ represents the transition density. A constant velocity model is used as the motion model in this work for simplicity. The undetected and detected track can be predicted independently. P_s is defined as the probability of survival, namely the probability that an object survives from one time step to the next. Standard Bayesian prediction is applied for both detected and undetected track states. The weight of each undetected track w^u_i is scaled by P_s in the prediction step. For detected tracks, which are modeled as multi-Bernoulli mixture (MBM) RFSs, each MB process can also be predicted independently of the others. The probability of existence for each MB-modeled object is decreased by a factor P_s in order to account for the higher uncertainty of existence within the Prediction stage.

Unlike the standard PMBM filter that uses a constant for P_s, we incorporate confidence scores from previously associated measurements in P_s for detected tracks. This is because detections with higher confidence scores indicate a higher chance of existence and survival across frames.

F. PMBM Update

By adding information from the measurement model $p(\mathbf{z}_t|\mathbf{x}_t)$, the PMBM density can be updated with:

$$\mathcal{PMBM}_{t+1|t+1}(\mathbf{x}_{t+1}) = \frac{p(\mathbf{z}_{t+1}|\mathbf{x}_{t+1}) \mathcal{PMBM}_{t+1|t}(\mathbf{x}_{t+1})}{\int p(\mathbf{x}_{t+1}|\mathbf{x}'_{t+1}) \mathcal{PMBM}_{t+1|t}(\mathbf{x}'_{t+1}) \delta \mathbf{x}'_t} \quad (8)$$

The update steps for different hypotheses are different, including detected/undetected tracks, being associated with a detection measurement or not. The association metrics vary depending on the measurement models. For the 3D bounding box measurement model, the association metric uses the Mahalanobis distance and 3D IoU between predicted 3D measurements and real 3D measurements. The point measurement model only uses the Mahalanobis distance between the center points.

1) Update of Undetected Tracks without Associated Measurement: The undetected tracks that do not have any measurement associated with them remain undetected. The Bayesian update will thus not change the states or variances of the Poisson distributions since no new information is added. The weight of each undetected track is thus decreased by a factor of $(1 - P_d)$ to account for the decreased probability of existing [29], [30]. P_d is the probability of detection, namely the by which it ought to be detected.

2) Update of Potential New Tracks Detected for the First Time: Our RFS-M^3 tracker is a measurement-driven framework: an object must be connected to a measurement in order to be classified as a new track (detected for the first time). All undetected tracks and corresponding gated measurements are considered to generate the new tracks. Note that the detections from a neural network have confidence scores attached to them. This confidence score is an invaluable indicator of the object probability of existence. So unlike a standard PMBM filter, we incorporate the detection confidence score into the update step of objects detected for the first time. For detections with confidence scores larger than a threshold (dependent on different learning-based 3D detector), we generate a potential new target by adding a new Bernoulli process, and plug the negative logarithm weight in the right $m \times m$ blocks diagonal in cost matrix L discussed in Section IV-C. For detections with lower confidence score, since we are not certain about their existences and require more evidences from the future, an undetected track with PPP density is generated for each of them.

3) Update of Detected Tracks without Associated Measurement: If there is no measurement associated with the detected tracks, which was detected from a previous frame, we maintain the object predicted state unchanged. Furthermore, the probability of existence and weight updated with:

$$\begin{aligned} r^d_{t+1|t+1} =& \frac{r^d_{t+1|t}(1-P_d)}{1-r^d_{t+1|t}+r^d_{t+1|t}(1-P_d)} \\ w^d_{t+1|t+1} =& w^d_{t+1|t}(1-r^d_{t+1|t}+r^d_{t+1|t}(1-P_d)) \end{aligned} \tag{9}$$

This weight is the missed detection weight $w^d_{0,j}$ mentioned in Section IV-C. Here P_d is the confidence score of the detection that is associated with this detected track in the previous frame. Unlike standard Kalman filter-based trackers, the survival time of detected tracks without measurment varies based on the tracking status from the previous time period.

4) Update of Detected Tracks with Associated Measurement: For a detected track with associated measurements, the predicted state is updated by weighting in the information contained in the measurement, and a standard Kalman filter [4] is used to update the state vector. Also the updated probability of existence is set to 1 because one can not associate a measurement to an object that does not exist. The updated weight can be calculated as:

$$w^d_{t+1|t+1} = w^d_{t+1|t} r^d_{t+1|t} P_d \mathcal{N}(\boldsymbol{z_t}; \boldsymbol{H}\boldsymbol{x}^d_{t+1|t+1}, \hat{\boldsymbol{S}}) \tag{10}$$

where $w^d_{t+1|t}$ and $r^d_{t+1|t}$ are the predicted weight and probability of existence from the prediction stage. $\hat{\boldsymbol{S}}$ and $\mathcal{N}(\boldsymbol{z_t}; \boldsymbol{H}\boldsymbol{x}^d_{t+1|t+1}, \hat{\boldsymbol{S}})$ are the innovation covariance and measurement likelihood, and $\boldsymbol{H}$ is the observation matrix. Here we set probability of detection P_d equal to the associated detection confidence score. This weight is the target hypothesis-measurement pair weight $w^d_{i,j}$ mentioned in the Section IV-C.

G. Reduction

The general assignment problem in MOT is NP-hard [41], and hence reducing the number of hypotheses is necessary for managing computational complexity and maintaining real-time performance. Five reduction techniques are used in this work: pruning, capping, gating, recycling and merging. *Pruning* is used to remove objects and global hypotheses with low weights. *Capping* is used to set an upper bound for the number of global hypotheses and detected tracks. *Gating* limits the search distance for data association using the Mahalanobis distance instead of Euclidean distance. *Recycling* moves detected tracks with lower weights to undetected track set rather than discarding them. There may be non-unique global hypotheses, and *merging* combines these identical global hypotheses into one.

V. Experiment

A. Settings

Dataset. We evaluate our method on three popular open dataset provided by three industry leaders: Waymo [13], Argoverse [16] and nuScenes [15]. The basic information for each dataset is shown in Table I.

Since our method is not a learning based method, we don't use the training set; but the validation set is used for parameter tuning. The field of view of these datasets are all 360 degrees. Note that for these open datasets, ground truth labels are only available for training and validation sets. For fairness of evaluation of testing samples, one needs to submit the tracking results to the relevant server.

Detections. We use 3D object detections provided by the dataset organizer.

Evaluation Metric. Standard evaluation metrics [42] for MOT are used, including MOTA, MOTP (multi-object tracking precision), FP (False Positives), IDS (ID switches) and so on. The details of the metrics for each benchmark can be found in [13], [16], [25].

TABLE I: Basic information of Waymo, Argoverse and nuScenes Datasets

Dataset	#Scenes/Segments	#LiDAR Frames	#Annotated Frames	#Evaluation Class
Waymo	1152	∼230.4K	∼230.04K	3
Argoverse	113	∼22.4K	∼22.4K	2
nuScenes	1000	∼400K	∼40.04K	7

TABLE II: Quantitative comparison of LEVEL_2 difficulty 3D MOT evaluation results on Waymo *test set*.

Method	Class	MOTA ↑ (Primary) (%)	MOTP ↓	FP(%) ↓	Misses (%) ↓
Waymo Baseline [13]	All	25.92	0.263	13.98	64.55
AB3DMOT-style KF [16]	All	29.14	0.270	17.14	53.47
Probabilistic KF [6]	All	36.57	0.270	8.32	54.02
RFS-M^3-Point	All	38.51	0.270	7.74	52.86
RFS-M^3-3D	All	41.73	0.270	8.98	49.01

TABLE III: Quantitative comparison of 3D MOT evaluation results on Argoverse *test set*.

Method	Class	MOTA (%) ↑	IDF1 ↑	MOTP ↓	MOTP-I* ↓	#False Possitive ↓	#Misses ↓	#IDS ↓
Argoverse Baseline [16]	All	57.11	0.685	0.355	0.190	20626	49374	624
RFS-M^3-Point	All	60.12	0.705	0.370	0.195	14202	48443	1145
RFS-M^3-3D	All	58.55	0.705	0.345	0.180	11536	52939	1520

*MOTP-I: amodal shape estimation error, computed by the 1-IoU of 3D bounding box projections on xy plane after aligning orientation and centroid

TABLE IV: Quantitative comparison on nuScenes *test set*.

Method	Class	AMOTA* ↑ (%)	AMOTP* ↓	MOTA (%) ↑	MOTP ↓
nuScenes Baseline [15]	All	15.1	1.501	15.40	0.402
Probabilistic KF [6]	All	55.0	0.798	45.9	0.353
RFS-M^3-Point	All	61.9	0.752	52.4	0.387
RFS-M^3-3D	All	61.4	0.716	51.1	0.363

*AMOTA & AMOTP: average MOTA and MOTP across different thresholds [5], [15].

TABLE V: Comparison of RFS-M^3 with different input detections on Waymo *val set*, all metrics in LEVEL_2 difficulty.

Method (Tracker+Detector)	Class	Detection APH* ↑	MOTA↑	MOTP ↓
RFS-M^3-3D+PPBA[39]	Car	49.4%	40.4%	0.182
RFS-M^3-3D+CenterPoint[40]	Car	64.2%	50.9%	0.171

*APH: average precision weighted by heading [13]

B. Experimental Results

The results for Waymo, Argoverse and nuScenes dataset are shown in Table II, Table III and Table IV respectively. RFS-M^3-Point represents our RFS-M^3 with point measurement model and RFS-M^3-3D denotes RFS-M^3 with 3D bounding box measurement model. As shown in these three Tables, our method outperforms other state-of-the-art tracker significantly. At the time of submission (Oct 2020), among all entries that use organizer-provided detections, our RFS-M^3 tracker ranked **No.2** on the Waymo 3D tracking leaderboard; For Argoverse leaderboard, we ranked **No.1** in Vehicle MOTA, **No.2** in all-class MOTA and **No.3** in averaged ranking (primary metric). For tracking-by-detection, the quality of input detections is of a paramount importance. Although some methods used better, self-generated detections than provided by the organizer, nevertheless our performance is very competitive. In Table V, we show results of RFS-M^3 with different input detections. The APH for CenterPoint[40] is 14.8% higher than PPBA (organizer provided detections)[39], and this results in a 10.5% improvement in MOTA. We believe our RFS-M^3 would perform better in these test benchmarks with better input detections than organizer precomputed ones.

Our RFS-M^3-3D performs significantly better than RFS-M^3-Point on the Waymo dataset (Table II), while has similar performance on the Argoverse and nuScenes datasets (Table III,IV). This is because Waymo uses the strictest *True Positive metrics* – 3D IoU, while Argoverse and nuScenes use ground plane center distance. The temporal 3D bounding box information contained in RFS-M^3-3D provides limited assistance in reducing center distance, but is crucial for estimating 3D bounding boxes in higher accuracy. This is also supported by the result that compared to RFS-M^3-Point, RFS-M^3-3D has smaller MOTP-I on Argoverse dataset.

TABLE VI: Ablation study of RFS-M^3-3D on Waymo *val set*, all metrics in LEVEL_2 difficulty.

Method	Class	MOTA ↑ (Primary) (%)	MOTP ↓	FP(%) ↓	Misses (%) ↓
Single Global Hypo	Vehicle	38.82	0.183	9.63	51.29
Constant P_d	Vehicle	27.83	0.183	7.13	63.74
Gating with Euclidean dis	Vehicle	38.57	0.183	9.51	51.24
RFS-M^3	Vehicle	40.40	0.182	9.90	49.60

C. Ablation Study

We evaluate the contribution of each important component in our RFS-M^3. The results are shown in Table VI. One advantage of our RFS-M^3 tracker is that we maintain multiple global hypotheses instead of only one to achieve improved data association across frames. This is because the ambiguities within input detections could lead to wrong data associations having higher weights than correct associations. Keeping the correct associations even with lower weights could help correct associations in future frames. P_d is the probability of detection. Incorporating the detection confidence score into P_d, rather than using a constant as standard PMBM. better models the uncertainties within the prediction and update. Gating is applied to limit the search distance for data association, and compared to Euclidean distance, Mahalanobis distance performs better because it takes the estimated state uncertainties into consideration.

VI. Conclusion

In this paper, we proposed an RFS-M^3 tracker to solve the 3D amodal MOT problem with multiple measurement models in support of different autonomous driving scenarios. Our framework can naturally model the uncertainties in the MOT problem. This represents a first successful attempt for employing an RFS-based approach in conjunction with 3D neural network-based detectors and with comprehensive testing using large-scale datasets. The experimental results on Waymo, Argoverse and nuScenes datasets demonstrate that our approach outperforms previous state-of-the-art methods by a large margin. Finally, we hope that our results motivate further research on RFS-based trackers for autonomous system applications.

ACKNOWLEDGMENT

This work has been supported in part by the Semiconductor Research Corporation (SRC) and by Amazon Robotics under the Amazon Research Award (ARA) program.

REFERENCES

[1] B.-N. Vo, S. Singh, and A. Doucet, "Sequential monte carlo methods for multitarget filtering with random finite sets," *IEEE Transactions on Aerospace and electronic systems*, vol. 41, no. 4, pp. 1224–1245, 2005.

[2] R. P. Mahler, *Statistical multisource-multitarget information fusion*. Artech House Norwood, MA, 2007, vol. 685.

[3] B.-T. Vo, B.-N. Vo, and A. Cantoni, "Bayesian filtering with random finite set observations," *IEEE Transactions on signal processing*, vol. 56, no. 4, pp. 1313–1326, 2008.

[4] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

[5] X. Weng, J. Wang, D. Held, and K. Kitani, "3d multi-object tracking: A baseline and new evaluation metrics," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[6] H.-k. Chiu, A. Prioletti, J. Li, and J. Bohg, "Probabilistic 3d multi-object tracking for autonomous driving," *arXiv preprint arXiv:2001.05673*, 2020.

[7] B.-T. Vo and B.-N. Vo, "A random finite set conjugate prior and application to multi-target tracking," in *2011 Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing*. IEEE, 2011, pp. 431–436.

[8] F. Papi, B.-T. Vo, M. Bocquel, and B.-N. Vo, "Multi-target track-before-detect using labeled random finite set," in *2013 International Conference on Control, Automation and Information Sciences (ICCAIS)*. IEEE, 2013, pp. 116–121.

[9] Y. Xia, K. Granstrcom, L. Svensson, and Á. F. García-Fernández, "Performance evaluation of multi-bernoulli conjugate priors for multi-target filtering," in *2017 20th International Conference on Information Fusion (Fusion)*. IEEE, 2017, pp. 1–8.

[10] B. Kalyan, K. Lee, S. Wijesoma, D. Moratuwage, and N. M. Patrikalakis, "A random finite set based detection and tracking using 3d lidar in dynamic environments," in *2010 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 2010, pp. 2288–2292.

[11] K. W. Lee, B. Kalyan, S. Wijesoma, M. Adams, F. S. Hover, and N. M. Patrikalakis, "Tracking random finite objects using 3d-lidar in marine environments," in *Proceedings of the 2010 ACM Symposium on Applied Computing*, 2010, pp. 1282–1287.

[12] K. Granström, S. Renter, M. Fatemi, and L. Svensson, "Pedestrian tracking using velodyne data—stochastic optimization for extended object tracking," in *2017 ieee intelligent vehicles symposium (iv)*. IEEE, 2017, pp. 39–46.

[13] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2446–2454.

[14] *Argoverse 3D Tracking Benchmark Leaderboard*. [Online]. Available: https://eval.ai/web/challenges/challenge-page/453/leaderboard/1278

[15] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 621–11 631.

[16] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8748–8757.

[17] D. Frossard and R. Urtasun, "End-to-end learning of multi-sensor 3d tracking by detection," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 635–642.

[18] E. Baser, V. Balasubramanian, P. Bhattacharyya, and K. Czarnecki, "Fantrack: 3d multi-object tracking with feature association network," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 1426–1433.

[19] X. Weng, Y. Wang, Y. Man, and K. M. Kitani, "Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6499–6508.

[20] H.-N. Hu, Q.-Z. Cai, D. Wang, J. Lin, M. Sun, P. Krahenbuhl, T. Darrell, and F. Yu, "Joint monocular 3d vehicle detection and tracking," in *Proceedings of the IEEE international conference on computer vision*, 2019, pp. 5390–5399.

[21] X. Weng, Y. Yuan, and K. Kitani, "Joint 3d tracking and forecasting with graph neural network and diversity sampling," *arXiv preprint arXiv:2003.07847*, 2020.

[22] M. Liang, B. Yang, W. Zeng, Y. Chen, R. Hu, S. Casas, and R. Urtasun, "Pnpnet: End-to-end perception and prediction with tracking in the loop," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 553–11 562.

[23] W. Luo, B. Yang, and R. Urtasun, "Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 3569–3577.

[24] *Waymo 3D Tracking Benchmark Leaderboard*. [Online]. Available: https://waymo.com/open/challenges/3d-tracking/

[25] *NuScene 3D Tracking Benchmark Leaderboard*. [Online]. Available: http://engineering.purdue.edu/ mark/puthesis

[26] R. P. Mahler, "Multitarget bayes filtering via first-order multitarget moments," *IEEE Transactions on Aerospace and Electronic systems*, vol. 39, no. 4, pp. 1152–1178, 2003.

[27] R. Mahler, "Phd filters of higher order in target number," *IEEE Transactions on Aerospace and Electronic systems*, vol. 43, no. 4, pp. 1523–1543, 2007.

[28] B.-T. Vo and B.-N. Vo, "Labeled random finite sets and multi-object conjugate priors," *IEEE Transactions on Signal Processing*, vol. 61, no. 13, pp. 3460–3475, 2013.

[29] J. L. Williams, "Marginal multi-bernoulli filters: Rfs derivation of mht, jipda, and association-based member," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 3, pp. 1664–1687, 2015.

[30] Á. F. García-Fernández, J. L. Williams, K. Granström, and L. Svensson, "Poisson multi-bernoulli mixture filter: direct derivation and implementation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 4, pp. 1883–1901, 2018.

[31] G. Brazil and X. Liu, "M3d-rpn: Monocular 3d region proposal network for object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9287–9296.

[32] X. Ma, Z. Wang, H. Li, P. Zhang, W. Ouyang, and X. Fan, "Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6851–6860.

[33] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 529–10 538.

[34] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[35] S. Pang, D. Morris, and H. Radha, "Clocs: Camera-lidar object candidates fusion for 3d object detection," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[36] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun, "Multi-task multi-sensor fusion for 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7345–7353.

[37] K. G. Murthy, "An algorithm for ranking all the assignments in order of increasing costs," *Operations research*, vol. 16, no. 3, pp. 682–687, 1968.

[38] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[39] S. Cheng, Z. Leng, E. D. Cubuk, B. Zoph, C. Bai, J. Ngiam, Y. Song, B. Caine, V. Vasudevan, C. Li, *et al.*, "Improving 3d object detection through progressive population based augmentation," *arXiv preprint arXiv:2004.00831*, 2020.

[40] T. Yin, X. Zhou, and P. Krähenbühl, "Center-based 3d object detection and tracking," *arXiv:2006.11275*, 2020.

[41] P. Emami, P. M. Pardalos, L. Elefteriadou, and S. Ranka, "Machine learning methods for solving assignment problems in multi-target tracking," *arXiv preprint arXiv:1802.06897*, 2018.

[42] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008.