3D Multi-Object Tracking with Adaptive Cubature Kalman Filter for Autonomous Driving

Ge Guo, *Senior Member, IEEE*, and Shijie Zhao

***Abstract*—A crucial and challenging issue in autonomous driving is dynamic road environment detection and 3D multi object tracking. In this article, we propose a novel framework for online 3D multi-object tracking to eliminate the influence of inherent uncertainty and unknown biases in point cloud. A constant turn rate and velocity (CTRV) motion model is employed to estimate the future motion state, which are smoothed by a cubature Kalman filter (CKF) algorithm. A new affinity model is introduced to evaluate the similarity between trajectories and candidate detections for accurate and reliable data association which can be formulated as a bipartite matching problem. An adaptive cubature Kalman filter (ACKF) is given to remove the influence of unknown bias and to robustly update the tracked state. Accuracy and speed of the proposed tracking method are evaluated on the KITTI 3D multi-object tracking dataset, showing superior performance than the baselines.**

***Index Terms*—3D multi-object tracking, point cloud, data association, Kalman filter (KF)**

I. Introduction

REAL-TIME accurate multi-object tracking is an important technology for intelligent transportation systems (ITS) applications and vehicular technology, such as traffic monitoring system and autonomous driving. In order to avoid potential traffic accident and guarantee safe driving, it's important to capture the dynamic road environment information by monitoring the motion states of the surrounding objects. It also plays an important role in dangerous scenarios detection such as a vehicle overtaking another or a pedestrian crossing the road. Existing 3D multi-object tracking (MOT) approaches are developed based on camera sensors [1]–[3]. However, such solutions are susceptible to variations in extreme high lighting or severe weather conditions. The operation range and performance of camera sensors depend on the intensity of light similar to the human eye. Hence, as a result, some implementations are unable to detect vehicles under strong light sources, which was reportedly a major cause of Tesla Autopilot-related deaths in Florida in May 2016 [4].

One of the most widely used sensing techniques in multi-object tracking applications is LiDAR (i.e., Light Detection and Ranging), due to its rich information on environment perception via 3D point cloud [5], [6]. This article is concerned with LiDAR-based 3D multi-object tracking for autonomous vehicles (AVs) applications. LiDAR data is heavily dependent on surface conditions, which is very challenging in practical road environments, making the raw point cloud data subject to noise and uncertainties. This may degrade the capability of the algorithm for continuous reliable tracking.

Impressive advances have been achieved in Kalman filter (KF)-based 3D MOT [7]–[11], and most of the recent approaches follow the tracking-by-detection framework. A well known result is the one in [7], which is based on a KF algorithm comprising a predictor-corrector. The optimality of a KF requires completely known of both motion and statistical model of the tracking system. However, in real-time applications, because the real-time measurement contains clutters or noise, the system models could contain parameters deviating from the nominal values by model drift or unknown random time-varying biases.

Meanwhile, it is difficult to effectively formulate the affinity between the predicted and candidate detected state used for accurate data association. Recently, intersection-over-union (IoU)-based 3D multi-object tracking approaches [8] employ IoU from 3D bounding boxes as the affinity model for data association. Caused by the motion state prediction errors, this method will decrease precision when prediction and detection do not overlap. Another approach [8] is distance-based affinity which takes into account the uncertainty about the prediction provided by the Kalman Filter. Although point cloud deep features can be extracted for 2D MOT [12], comprehensive affinity formulation for LiDAR-based 3D MOT is still under-explored.

These limitations constitute a major challenge for 3D MOT, which can lead to instability when dealing with multi-object in a dense and clutter environment. Although the tracking-by-detection framework is useful for multi-object tracking, it is still an open problem to deal with uncertainty and unknown random bias for online tracking. Critical steps to overcome the problems are using more advanced models for 3D point cloud and introducing a more accurate and robust data association strategy to reduce the impact of noise and confusing data.

In this article, a novel online 3D multi-object tracking framework (see Fig 1) is proposed to attain better performance in term of accuracy and reliability. We employ a constant turn rate and velocity (CTRV) motion model to predicted the motion states, which is smoothed using a cubature Kalman filter (CKF). An affinity model is proposed to associate the predicted and candidate detected states. We then introduce an adaptation mechanism into the filter to address issues of unknown bias or model drift, and to update the tracked states. The main contributions of this article are summarized

This work was supported by the National Natural Science Foundation of China under Grant 62173079 and Grant U1808205.

Ge Guo is with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China (e-mail: geguo@yeah.net).

Shijie Zhao is with the College of Information Science and Engineering, Northeastern University, Shenyang 110819, China (e-mail: 1971837@stu.neu.edu.cn).

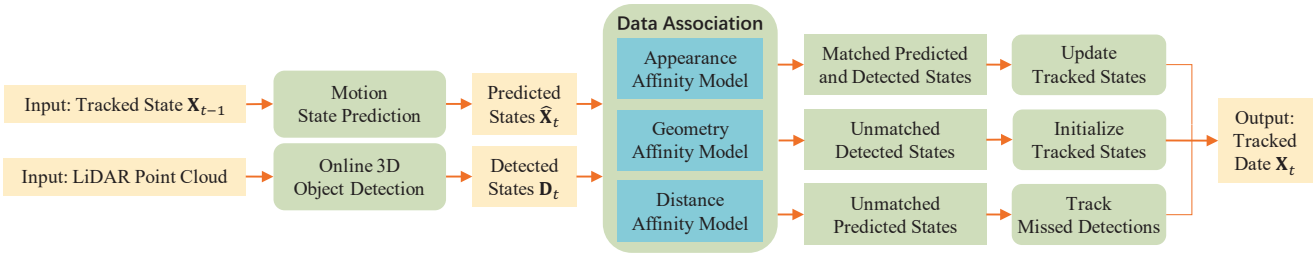

Fig. 1. The framework of the proposed 3D MOT method in points cloud. At time step t, input 3D point cloud data and tracked states $\mathbf{X}_{t-1}$ at time step $t-1$. First, obtain detected state $\mathbf{D}_t$ by using 3D object detection algorithm. Second, we estimate motion states $\hat{\mathbf{X}}_t$ of each object. Third, we conduct data association between predicted states $\hat{\mathbf{X}}_t$ and candidate detected states $\mathbf{D}_t$ to obtain the near-optimal matched pairing. Finally, update tracked states $\mathbf{X}_t$ with adaptive cubature Kalman filter algorithm and initialize new trajectories with the unmatched detected states. In addition, we keep the unmatched predicted states, in order to track temporarily missed detections.

as follows.

1) A novel online 3D multi-object tracking method is presented for autonomous driving, in which an CTRV motion model is involved to predict the motion states.
2) An enhanced cubature Kalman filter (CKF) with an adaptation mechanism is derived to update the tracked states, which is capable of handling inherent uncertainty and unknown random bias for online tracking.
3) An affinity model taking full advantages of the appearance, geometry, and distance correlation of the objects in point cloud is designed for better data association.

The remaining parts of this article are organized as follows. Section II gives the formulation of the 3D multi-object tracking problem in point cloud. Our proposed tracking method is presented in Section III. Evaluation results and ablation experiments are reported in Section IV, following the conclusion and future work in Section V

II. Problem Formulation

Considering 3D multi-object tracking problem under nonlinear cluttered driving environment, the 3D point cloud data from LiDAR mounted onboard the ego-vehicle can be used for tracking of surrounding objects, i.e., vehicles and pedestrians. Each object such as vehicles and pedestrians is set to move in a 3D plane assuming contact with the ground, and is represented by a state vector $X = [x, y, z, \theta, \mathrm{v}, \dot{\theta}, \dot{z}]^T$ composed of position, orientation angle and velocity, namely. 3D multi-object tracking system modeling includes two main parts: a motion model to predict the motion of each object; and a measurement model for the observation of the object's current motion states. Consider the motion model and measurement model in the following time step (assuming in the sampling time Δt is the scanning interval of LiDAR sensor) in the Cartesian coordinate system, the general form of the state-space model is given as below:

$$\begin{cases} X_t = f(X_{t-1}) + W_{t-1} \\ D_t = h(X_t) + V_t = HX_t + V_t \end{cases} \tag{1}$$

where $f(\cdot)$ and $h(\cdot)$ denote the functions of the state transition and measurement processes, respectively. X_{t-1} is the tracked state at time step $t-1$. We estimate current motion states at time step t using tracked state X_{t-1} to perform an accurate and robust data association. The dimension of the predicted states $\hat{X}_t$ is the same as the tracked states. $D_t = [x_t, y_t, z_t, w_t, h_t, l_t, \theta_t]^T$ is the candidate detected state at time step t produced by the detection algorithm at time step t, and encapsulates information of a 3D bounding box including global coordinates, orientation angle and size. W_t and V_t are system noise and measurement noises vectors with covariance matrix Q_t and R_t, respectively. Meanwhile, those noises are uncorrelated zero-mean white Gaussian noise sequences, and satisfy

$$E\left\{W_k {W_l}^T\right\} = Q_k \delta(k,l) \tag{2}$$

$$E\left\{V_k {V_l}^T\right\} = R_k \delta(k,l) \tag{3}$$

where $\delta(\cdot)$ is the Kronecker delta function

$$\delta(k,l) = \begin{cases} 1 & \text{if } k = l \\ 0 & \text{if } k \neq l \end{cases} \tag{4}$$

Assume that the initial states X_0 is Gaussian random variables with

$$E\{X_0\} = \hat{X}_0 \tag{5}$$

$$E\left\{\left[X_0 - \hat{X}_0\right]\left[X_0 - \hat{X}_0\right]^T\right\} = P_0 > 0 \tag{6}$$

Suppose there are m tracks of objects and n candidate detections. The set of tracked states and detection states are, respectively, defined as:

$$\mathbf{X}_{t-1} = \left\{X_{t-1}^1, X_{t-1}^2, \ldots, X_{t-1}^m\right\} \tag{7}$$

$$\mathbf{D}_t = \left\{D_t^1, D_t^2, \ldots, D_t^n\right\} \tag{8}$$

The 3D multi-object tracking problem addressed in this article can be formulated as the processing of a sequence of 3D point cloud $\mathbf{I}_t$ into a sequence of *posteriori* estimation $\mathbf{X}_t$ of the set of tracked states,

$$I_0, I_1, \ldots, I_t \quad \Rightarrow \quad X_0, X_1, \ldots, X_t \tag{9}$$

In our work, we follow the tracking-by-detection framework which is shown in Fig. 2, by match the candidate detected state set $\mathbf{D}_t$ with the predicted state $\hat{\mathbf{X}}_t$ which is estimated from the tracked states set $\mathbf{X}_{t-1}$ at the previous time step, to update the tracked states and obtain the new tracked states $\mathbf{X}_t$.

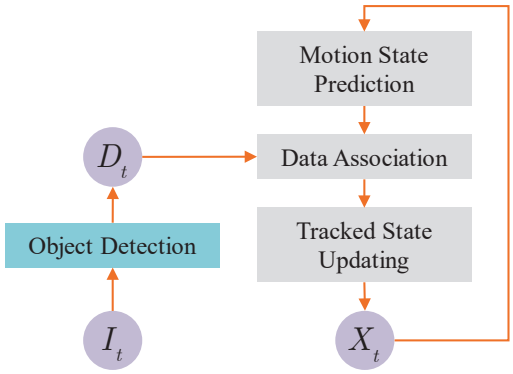

Fig. 2. The illustration of 3D multi-object tracking problem formulation, which can be formulated as optimally matching the candidate detections with the predicted states which can be estimated from tracked states at the previous time step.

III. The Object Tracking Method

In this section, we present a new online 3D multi-object tracking method in point cloud for AVs developments. The proposed method follows the tracking-by-detection framework and can be divided into the following four steps: (1) detect objects in point cloud by employing a deep-learning based 3D object detection algorithm, (2) estimate the motion states of tracked objects based on a constant turn rate and velocity (CTRV) motion model and smooth the predicted state by cubature Kalman filter (CKF) algorithm, (3) match the predicted states with the candidate detected states by solving a bipartite matching problem with our proposed affinity model, (4) introduce an adaptation mechanism into the CKF algorithm to address issues such as model drift, and update the tracked states.

A. Motion State Prediction

The missed detections greatly hinder 3D multi-object tracking performance, especially obtain reliably detection of distant or occluded objects. To address these issues, most of 3D MOT methods [8], [9], [14] predict motion state based on the constant velocity (CV) model. But these linear motion models cannot capture the nonlinear motion state in the complex driving environment. Deep learning approach which can automatically identify patterns and extract features via unsupervised/supervised learning was recently presented using Long Short-Term Memory (LSTM) to predict the motion states [2]. However, deep-learning based methods are much slower than the motion-model-based method. To achieve a comprehensive performance in both accuracy and reliability, we employ a constant turn rate and velocity (CTRV) motion model to estimate the future motion states.

The CTRV motion model is used to simulate an object for making a cornering movement with fixed turning rate and constant angular velocity. For each tracked state X_{t-1}, we predict the tracked state $\hat{X}_t$ at time step t using the CTRV model. Thus, the system motion estimation model as following:

$$\begin{aligned} X_t &= f\left(X_{t-1}\right) \\ &= X_{t-1} + \begin{bmatrix} \frac{v}{\dot{\theta}}(\sin(\theta+\dot{\theta}\Delta t)-\sin(\theta)) \\ \frac{v}{\dot{\theta}}(-\cos(\theta+\dot{\theta}\Delta t)+\cos(\theta)) \\ \dot{z}\Delta t \\ \dot{\theta}\Delta t \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{aligned} \tag{10}$$

where, Δt is the sampling time. In the case of zero turning rate (i.e. $\dot{\theta}=0$) ,

$$\begin{aligned} X_t &= f\left(X_{t-1}\right) \\ &= X_{t-1} + \begin{bmatrix} v\cos\theta & v\sin\theta & \dot{z}\Delta t & \dot{\theta}\Delta t & 0 & 0 & 0 \end{bmatrix}^T \end{aligned} \tag{11}$$

We employ the cubature Kalman filter [15] to optimize the predicted states for better accuracy. The CKF algorithm is a novel nonlinear filtering algorithm which can solve the problem that KF algorithm has is less effective estimation when the state variables are high-dimensional. Based on the CKF algorithm, the motion state prediction can be divided into the following two steps.

1) *Calculate cubature points:* According to the third-degree spherical-radial rule [15], evaluate the cubature points. And evaluate the propagated cubature point set according to the CRTV motion model mentioned above.

$$P_{t-1} = S_{t-1}S_{t-1}^T \tag{12}$$

$$\chi_{k,t-1} = X_{t-1} + S_{t-1}\xi_k, k = 1, 2, \cdots, 2n \tag{13}$$

$$\xi_k = \begin{cases} \sqrt{n}[1], k = 1, 2, \ldots, n \\ -\sqrt{n}[1], k = n+1, n+2, \ldots, 2n \end{cases} \tag{14}$$

$$\hat{\chi}_{k,t} = f\left(\chi_{k,t-1}\right) \tag{15}$$

2) *Time update:* Compute the predicted motion states at time step t and the prediction error covariance matrix.

$$\hat{X}_t = \frac{1}{2n}\sum_{k=1}^{2n}\hat{\chi}_{k,t} \tag{16}$$

$$\hat{P}_t = \frac{1}{2n}\sum_{k=1}^{2n}\hat{\chi}_{k,t}\hat{\chi}_{k,t}^T - \hat{X}_t\hat{X}_t^T + Q_{t-1} \tag{17}$$

where ξ_k denotes the cubature point with corresponding weight, n is the dimension of the tracked state vector, $\hat{X}_t$ represents a *priori* estimate of the state vector, $\hat{P}_t$ is a *priori* estimate error covariance matrix, and P_{t-1} is a *posteriori* estimate error covariance matrix.

B. Association affinity model

Giving the tracked states set, we should associate each track with a candidate detection, in order to spawn or end a trajectory. The affinity model determines how well a trajectory and a detection are matched. To evaluate the similarity between trajectories and candidate detections and achieve high accuracy and reliability, we propose a new affinity model between the

predicted and candidate detected states as a joint probability function of appearance, geometry and distance correlation between their corresponding objects in point cloud. Our proposed affinity model is defined as

$$\Lambda(\hat{X}_t, D_t) = \Lambda^A(\hat{X}_t, D_t)\Lambda^D(\hat{X}_t, D_t)\Lambda^G(\hat{X}_t, D_t) \tag{18}$$

Each affinity score is computed as follows:

$$\Lambda^A(\hat{X}_t, D_t) = \exp(-1/\left\| f_{\hat{X}_t} - f_{D_t} \right\|_2^2) \tag{19}$$

$$\Lambda^G(\hat{X}_t, D_t) = \exp(-1/\sum_{k\in\{h,w,l\}} \frac{|k_{\hat{X}_t} - \hat{k}_{D_t}|}{k_{\hat{X}_t} + \hat{k}_{D_t}}) \tag{20}$$

$$\Lambda^D(\hat{X}_t, D_t) = \mathcal{N}(\sqrt{\eta_t^T(H\hat{P}_t H + R_t)\eta_t}) \tag{21}$$

where $\eta_t = D_t - H\hat{X}_t$ is the innovation, $\mathcal{N}(\cdot)$ denotes the normalization function.

For the appearance affinity $\Lambda^A(\hat{X}_t, D_t)$, we reactivate the appearance features extracted from the object detection algorithm forward propagation for computational efciency, and compute their affinity with L2 distance. The geometry affinity $\Lambda^G(\hat{X}_t, D_t)$ is calculated with their height, width and length. $\Lambda^D(\hat{X}_t, D_t)$ is the distance correlation affinity defined by the Mahalanobis distance [9].

Given the affinity model between all predicted and detected states, we can formulate the data association as a bipartite matching problem to obtain the near-optimal pairing. Suppose there are m tracked states of objects and n candidate detection are available. detections are available. The data association is to assign each trajectory to each candidate detection for minimize the total cost which is defined by the affinity model.

Denote the bipartite weighted graph (see Fig. 3) as $G = (V, U, E)$, where $|V| = m$ and $|U| = n$ are the sets of nodes. Weight matrix E is the set of edges whose weights store in a cost matrix:

$$C = \begin{bmatrix} \Lambda(\hat{X}_t^1, D_t^1) & \cdots & \Lambda(\hat{X}_t^1, D_t^n) \\ \vdots & \ddots & \vdots \\ \Lambda(\hat{X}_t^m, D_t^1) & \cdots & \Lambda(\hat{X}_t^m, D_t^n) \end{bmatrix} \tag{22}$$

where $c_{ij} = \Lambda(\hat{X}_t^i, D_t^j)$ is the cost of i^{th} tracked state assigned to j^{th} candidate detection. G is a bipartite graph, so V and U are two nonoverlapping sets. The assignment problems can be formulated as finding a one-to-one matching in graph G and the total edge weight optimized to minimum. Let M_{ij} defined as the matching matches i^{th} node from U to j^{th} node from V. Mathematically the min-weight matching problem is:

$$\min_{M_{i,j}} \quad \sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}M_{ij}, \tag{23}$$

$$\text{s.t.} \quad \sum_{i=1}^{m} M_{ij} = 1, \tag{24}$$

$$\sum_{j=1}^{n} M_{ij} = 1. \tag{25}$$

where $M_{ij} = 1$ if i^{th} track is assigned to j^{th} detection, otherwise, $M_{ij} = 0$.

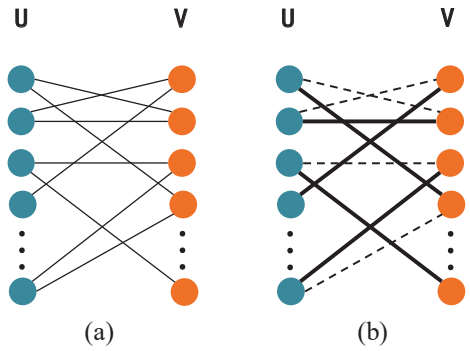

Fig. 3. An illustration of the bipartite graph and its assignment.

We can apply a greedy algorithm to solve the problem which can be defined as a minimum-weight perfect matching problem. After data association, we can obtain the matched pairing $(\hat{X}_t, D_t)$, which denotes the near-optimal match of the predicted states and the candidate detected states. Unmatched detected states are also obtained which we use to initiate new tracked stated. We can also obtain unmatched predicted states, which caused by objects naturally disappear or miss detected by the detection algorithm. Therefore, to distinguish the two cases, we set a threshold N. We will delete the trajectory if a predicted state couldn't be associated with any detected state over N frames which treated as disappearing. Otherwise, the predicted states treated as missed detection and iteratively perform data association at the next time step.

C. Tracked State Updating

To obtain the updated tracked states based on the matched pairing of predicted states and the corresponding detected states after data association, most of the existing tracking methods directly use Kalman filter (KF). However, In the actual tracking scenario, the unknown deviation caused by the inaccurate system motion model and the system noise statistical characteristics will seriously affect the performance of the filter algorithm estimation, and even lead the algorithm to diverge, resulting in tracking failure. In this case, the lter algorithms can not produce the reliable state estimation only when the nonlinear model errors are considered and resolved. To address this issue, we introduce an adaptation mechanism into the cubature Kalman filter.

The innovation of the cubature Kalman filter can be influenced by unknown errors such as modeling inaccuracies, initial states or the unknown bias. Meanwhile, because the unaccounted errors involved in the calculation of the innovation, the innovation covariance shows the impact of unknown errors. The innovation covariance of the CKF is

$$C_t = E\left[\eta_t \eta_t^T\right] = H\hat{P}_t H^T + R_t \tag{26}$$

The innovation covariance matrix is equal to C_t when the exact system motion model is obtained. However, for complex traffic scenarios, it is not possible to obtain the exact motion equation of a nonlinear stochastic system in most cases. Meanwhile, in the scenario where the parameters are

uncertain or the noise is large, the prediction model does not match the real model under unknown strong noise, resulting in both obtaining a biased motion state prediction result and increasing the error covariance matrix. In this case, we can estimate the increased innovation covariance matrix as [16]:

$$C_t' = \begin{cases} \eta_{t=0}\eta_{t=0}^T, & t=0 \\ \frac{\rho \cdot C_{t-1}' + \eta_t \eta_t^T}{1+\rho}, & t \geq 1 \end{cases} \tag{27}$$

where $\rho \in [0,1]$ is a weighting factor and determines the weight between the *priori* state and the current state.

To address the effects of modeling inaccuracies, we assume that the exact motion model of the system and the measurement equation are unknown. Therefore, the ltering process becomes adaptive by correcting the innovation sequence [17]. The relationship between corrected innovation covariance C_t' and C_t can be expressed as:

$$C_t' = \alpha_t C_t \tag{28}$$

$$\alpha_t = \max\left\{1, \frac{tr\left(C_t'\right)}{tr\left(C_t\right)}\right\} \tag{29}$$

where $tr(\cdot)$ denotes the trace operation of the covariance matrix.

In case of biases or uncertainties in motion model, the prediction error covariance will increase. We introduce an adaptive factor $\lambda_t \geq 1$. which is used to modify the error covariance, so that the impact of uncertainties and biases can be reduced. The error covariance can be represented as

$$\hat{P}_t' = \lambda_t \hat{P}_t \tag{30}$$

$$\lambda_t = \max\left\{1, \frac{tr\left(C_t' - R_t\right)}{tr\left(\hat{P}_{D,t} - R_t\right)}\right\} \tag{31}$$

According the adaptation mechanism described above, we perform the adaptive cubature Kalman filter (ACKF) algorithm to obtain the updated tracked states.

Calculated cubature points from the modified prediction error covariance matrix and the prediction state.

$$\hat{P}_t' = \lambda_t \hat{P}_t = S_t S_t^T \tag{32}$$

$$\chi_{k,t} = \hat{X}_t + S_t \xi_k \tag{33}$$

Compute the innovation covariance of ACKF algorithm.

$$\begin{aligned} C_t' &= \tau_k C_t \\ &= \tau_k \left[H \hat{P}_t H^T + R_t\right] \\ &= H \hat{P}_t' H^T + R_t \end{aligned} \tag{34}$$

According to the measurement model, the cubature points transformation can be expressed as

$$D_{k,t} = h\left(\chi_{k,t}\right) \tag{35}$$

Compute the *priori* detection state

$$\hat{D}_t = \frac{1}{2n}\sum_{k=1}^{2n} D_{k,t} \tag{36}$$

The cross variance and covariance can be given by

$$\hat{P}_{D,t} = \frac{1}{2n}\sum_{k=1}^{2n} D_{k,t} D_{k,t}^T - \hat{D}_t \hat{D}_t^T + R_t \tag{37}$$

$$\hat{P}_{XD,t} = \frac{1}{2n}\sum_{k=1}^{2n} \chi_{k,t} D_{k,t}^T - \hat{X}_t \hat{D}_t^T \tag{38}$$

Finally, the tracked state and covariance updating equations is expressed as

$$P_t = \hat{P}_t' - K_t \hat{P}_{D,t} K_t^T \tag{39}$$

$$X_t = \hat{X}_t + K_t\left(D_t - \hat{D}_t\right) \tag{40}$$

where the Kalman gain matrix $K_t = \hat{P}_{XD,t}\hat{P}_{D,t}^{-1}$.

IV. THE EXPERIMENTS

In this section, we use a real-world dataset for autonomous driving applications to evaluate our proposed method, showing that our method outperforms several well-known baselines. In addition, we perform ablation studies for better understanding the impact of each component on the general performance of our method.

A. Dataset and Settings

Our method is tested on the real-world computer vision dataset KITTI [13] benchmark suite which collected driving sequences using a multi-modal sensor suite including LiDAR in various environment. Cars and Pedestrians are noticed in the KITTI tracking dataset. The point cloud in the dataset were produced by the recording platform which equipped with a Velodyne HDL-64E LiDAR scanner. In our experiments, we only make use of point cloud and IMU/GPS data. The 2D images only employed for better visualization. The proposed tracking method is implemented using python and run on a server computer with Intel(R) i5-10400 CPU and NVIDIA TITAN Xp GPU.

B. Evaluation Metrics

The performance of our method is evaluated using the CLEAR MOT performance measures [18] with addition of Mostly Tracked (MT), Mostly Lost (ML), ID Switches (IDS) and fragmentation (FRAG) which calculates the times of a trajectory changes from tracked to untracked [19]. The metrics above are also employed by the KITTI benchmark. Note that, the 3D MOT algorithm can be summarized in two main metrics, namely, Multi-Object Tracking Accuracy (MOTA) and Multi-Object Tracking Precision (MOTP), which can show the overall performance. MOTA is used to measure the overall tracking accuracy and defined as:

$$MOTA = 1 - \frac{\sum_t m_t + fp_t + mme_t}{\sum_t g_t} \tag{41}$$

where m_t, and fp_t represent the number of misses, of false positives, and of mismatches, respectively, at time step t.

TABLE I
RESULTS OF QUANTITATIVE PERFORMANCE ON THE VALIDATION DATASET OF KITTI 3D MULTI-OBJECT TRACKING DATASET. THE SYMBOL ↑ INDICATES BETTER PERFORMANCE FOR LARGER VALUE. THE BEST PERFORMANCE ARE MARKED WITH BOLD FONT

Method	Modality	MOTA ↑	MOTP ↑	MT ↑	ML ↓	IDS ↓	FPAG ↓
CIWT [3]	Camera	74.38%	82.85%	49.59%	9.8%	26	131
DSM [21]	Camera/LiDAR	70.66%	83.08%	72.17%	**4.85%**	54	239
FANTrack [22]	Camera/LiDAR	76.52%	84.81%	73.14%	9.25%	1	54
AB3DMOT [7]	LiDAR	86.24%	78.43%	-	-	**0**	**15**
Ours	LiDAR	**88.73%**	**86.81%**	**85.62%**	5.01%	8	68

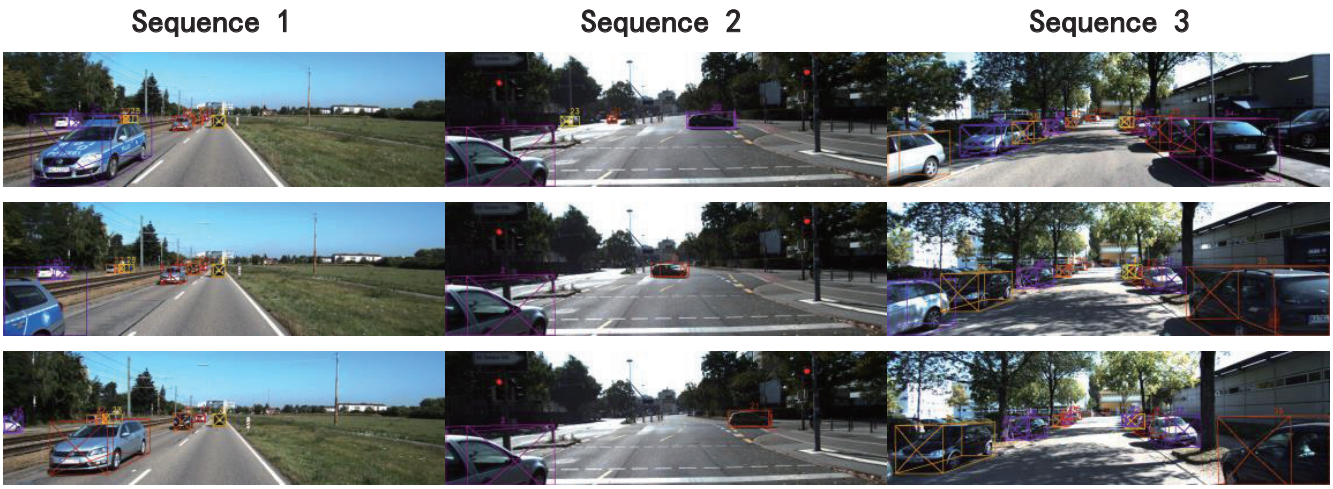

Fig. 4. Examples of 3D MOT results on the KITTI tracking dataset show the tracked vehicles on the 2D image plane projected from the 3D point cloud tracking results. We place different object ID above, and correspond to 3D bounding box which is represents different vehicles. Sequence 1 shows the effectiveness tracking of vehicles at varying depths. In sequence 2, turning vehicles with different poses are well tracked in an intersection. While sequence 3 shows an accurate tracking of the occluding vehicle and a small part of the vehicle is visible but successfully tracked.

While MOTP is concerned with the ability of localization precision and is independent of the ability of recognizing configurations. The MOTP is defined as:

$$MOTP = \frac{\sum_{i,t} d_t^i}{\sum_t c_t} \tag{42}$$

where c_t is the number of all matched pairs, and d_t^i is the metric distance in i^{th} frame.

C. Quantitative Results Analysis

We project the tracking results onto the 2D image plane for comparison with other methods. Table I shows the performance of our method compared to 3D-based tracking methods on KITTI tracking dataset. As can be seen, the MOTA of our tracking method achieves 88.73% which reaches 2.45% improvement compared with the AB3DMOT [8] which use the same detection algorithm named PointRCNN [23]. In addition, high performance of our method is shown in MOTP and ML which achieves 86.81% and 2.16%, respectively, which can be credited with adaptation mechanism employed to address issues such as motion model drift.

We present challenging tracking sequences in Fig.4, where for better visualization we project our tracking results onto the 2D image plan. For example, the first sequence shows the effectiveness of tracking vehicles at the varying depths. In the second sequence, turning vehicles with different poses are well tracked at an intersection. And the last sequence shows an accurate tracking of the occluding vehicle and a small part of the vehicle is visible but successfully tracked.

TABLE II
RESULTS OF USING DIFFERENT 3D OBJECT DETECTION ALGORITHMS ON THE KITTI VALIDATION DATASET

Detection	MOTA ↑	MOTP ↑	IDS ↓	FPAG ↓
VoxelNet [24]	82.34%	82.44%	53	132
SECOND [25]	84.69%	85.53%	36	98
PointRCNN [20]	**88.73%**	**86.81%**	**8**	**86**

TABLE III
RESULTS OF USING THE SAME 3D OBJECT DETECTION ALGORITHM WITH DIFFERENT TRACKING METHODS ON THE KITTI VALIDATION DATASET

Detection	Tracking	MOTA ↑	MOTP ↑	IDS ↓	FPAG ↓
PointRCNN	AB3DMOT [7]	86.24%	78.43%	**0**	**15**
	Ours	**88.73%**	**86.81%**	8	68

D. Ablation Study

We employ three ablation studies on the KITTI 3D multi-object tracking dataset to demonstrate the effectiveness of our proposed method.

1) **Influence of the Object Detection algorithm**

TABLE IV
RESULTS OF USING DIFFERENT PREDICTION METHODS ON THE KITTI VALIDATION DATASET

Prediction	MOTA ↑	MOTP ↑	IDS ↓	FPS ↑
LSTM	**88.84**%	10	70	51
Ours	88.73%	**8**	**68**	**214**

TABLE V
RESULTS OF USING DIFFERENT AFFINITY MODEL ON THE KITTI VALIDATION DATASET, WHERE IOU., APP., GEO., DIS., DENOTE IOU-BASED, APPEARANCE, GEOMETRY AND DISTANCE CORRELATION AFFINITY MODEL, RESPECTIVELY

Cost				MOTA ↑	IDS ↓	FRAG ↓
IoU	App.	Geo.	Dis.			
✓	-	-	-	82.96%	54	122
-	-	✓	✓	85.68%	18	82
-	✓	-	✓	84.33%	90	145
-	✓	✓	-	87.35%	11	77
-	✓	✓	✓	**88.73**%	**8**	**68**

The performance of the 3D object detection algorithm has great impact on the tracking-by-detection method. We conducted an ablation experiment on three detection methods to explore the effects of our tracking method with different detection algorithms. PointRCNN achieves better performance in the point cloud compared with VoxelNet [24] and SECOND [25]. Using the detections and the deep features produced by the different algorithm, the performance is as shown in Table II. Obviously, the better detection results make the better tracking performance. In addition, as shown in Table III, our method achieves MOTA and MOTP improvement comparing with the AB3DMOT which uses the same detection algorithm.

2) **Performance of the Motion Prediction**
We compared our proposed prediction method with deep-learning based method to reveal the performance. Frames Per Second (FPS) is used to measure the running speed of the prediction algorithm. Keeping the other components unchanged such as the object detection algorithm, as shown in Table IV, CKF is only inferior in MOTA and superior in speed than LSTM.

3) **Effectiveness of the Affinity Model**
We separately removed each component and kept others unchanged, to evaluate the effectiveness of every component in the affinity model. Besides, we compared the affinity model with the IoU-based methods [2],[8]. As shown in Table V, each component has a positive impact on our tracking method. Eventually, with all components, our method achieves the best performance improvement on MOTA.

V. CONCLUSIONS AND FUTURE WORK

Traffic monitoring systems such as online 3D multi-object tracking are the prerequisite for intelligent transportation systems (ITS) applications and autonomous vehicles (AVs). The LiDAR (i.e., Light Detection and Ranging) that provide high accuracy measurements, is becoming one of the most important sensors. In this article, we proposed a novel 3D MOT method with an adaptive cubature Kalman filter and a new affinity model for data association. We perform a comparative verification to evaluate the accuracy and reliability compared with the state-of-the-art methods for 3D-MOT in point cloud. The results of experiments on the KITTI tracking dataset show the superiority of our method in both accuracy and reliability.

However, the tracking method we proposed has some limitations. The objects could be miss detected by the detection algorithm and lead to tracking failures when the point cloud data is extraordinarily sparse. To address this issue, we could utilize additional modality such as image data to enhance the tracking performance. However, that will cause lower computational efficiency. Besides, our tracking method considers the motion prediction noises as Gaussian noises which may lead to tracking failures in non-Gaussian conditions. To enhance the tracking performance and further monitor the dynamic road environment information for AVs development, future work will focus on taking into account non-Gaussian noises and handling different motion dynamics with developing behavioral.

REFERENCES

[1] C. Wojek, S. Walk, S. Roth, K. Schindler, and B. Schiele, "Monocular visual scene understanding: Understanding multi-object traffic scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, pp. 882-897, 2013.

[2] H.-N. Hu et al., "Joint monocular 3D vehicle detection and tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5390-5399.

[3] A. Osep, W. Mehner, M. Mathias, and B. Leibe, "Combined image-and world-space tracking in trafc scenes," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2017, pp. 1988-1995.

[4] "Highway accident report: Collision between a car operating with automated vehicle control systems and a tractor-semitrailer truck near williston, Florida," Nat. Transp. Saf. Board, Washington, DC, USA, Accident Rep. NTSB/HAR-17/02, PB2017-102600, 2017.

[5] L. Huang and M. Barth, "Tightly-coupled LIDAR and computer vision integration for vehicle detection," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2009, pp. 604-609.

[6] Z. Luo, M. Attari, S. Habibi, and M. V. Mohrenschildt, "Online multiple maneuvering vehicle tracking system based on multi-model smooth variable structure filter," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 2, pp. 603-616, 2020.

[7] R. E. Kalman, "A new approach to linear ltering and prediction problems," *Trans. ASME-J. Basic Eng.*, vol. 82, no. 1, pp. 35-45, 1960.

[8] X. Weng, J. Wang, D. Held and K. Kitani, "3D Multi-Object Tracking: A Baseline and New Evaluation Metrics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 10359-10366.

[9] H. -K. Chiu, J. Li, R. Ambruş and J. Bohg, "Probabilistic 3D Multi-Modal, Multi-Object Tracking for Autonomous Driving," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2021, pp. 14227-14233.

[10] H. Wu, W. Han, C. Wen, X. Li and C. Wang, "3D Multi-Object Tracking in Point Clouds Based on Prediction Confidence-Guided Data Association," *IEEE Trans. Intell. Transp. Syst.*, doi: 10.1109/TITS.2021.3055616.

[11] K. Burnett, S. Samavi, S. L. Waslander, T. D. Barfoot, and A. P. Schoellig, "AUToTrack: A lightweight object detection and tracking system for the SAE AutoDrive challenge," 2019, *arXiv:1905.08758.* [Online]. Available: http://arxiv.org/abs/1905.08758

[12] W. Zhang, H. Zhou, S. Sun, Z. Wang, J. Shi, and C. C. Loy, "Robust multi-modality multi-object tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2365-2374.

[13] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The kitti vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354-3361.

[14] S. Scheidegger, J. Benjaminsson, E. Rosenberg, A. Krishnan, and K. Granstrom, "Mono-camera 3D multi-object tracking using deep learning detections and PMBM ltering," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 433-440.
[15] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *IEEE Trans. Automat. Contr.*, vol. 54, no. 6, pp. 1254-1269, 2009.
[16] Y. Liang, D. X. An, D. H. Zhou, and Q. Pan, "A finite-horizon adaptive Kalman filter for linear systems with unknown disturbances," *Signal Processing*, vol. 84, no. 11 SPEC. ISS., pp. 2175-2194, 2004.
[17] M. A. Gandhi and L. Mili, "Robust Kalman lter based on a generalized maximum-likelihood-type estimator," *IEEE Trans. Signal Process.*, vol. 58, no. 5, pp. 2509-2520, 2010.
[18] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The clear MOT metrics," *EURASIP J. Image Video Process.*, vol. 2008, pp. 1-10, Dec. 2008.
[19] Y. Li, C. Huang, and R. Nevatia, "Learning to associate: Hybridboosted multi-target tracker for crowded scene," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 2953-2960.
[20] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 770-779.
[21] D. Frossard and R. Urtasun, "End-to-end learning of multi-sensor 3D tracking by detection," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2018, pp. 635-642.
[22] E. Baser, V. Balasubramanian, P. Bhattacharyya, and K. Czarnecki, "FANTrack: 3D multi-object tracking with feature association network," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 1426-1433.
[23] D. Frossard and R. Urtasun, "End-to-end learning of multi-sensor 3D tracking by detection," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2018, pp. 635-642.
[24] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4490-4499.
[25] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, Oct. 2018.

Ge Guo (Senior Member, IEEE) received the B.S. degree and the PhD degree from Northeastern University, Shenyang, China, in1994 and 1998, respectively.

From May 2000 to April 2005, he was with Lanzhou University of Technology, China, as a Professor and the director of the Institute of Intelligent Control and Robots. He then joined Dalian Maritime University, China, as a Professor with the Department of Automation. Since 2018, he has been a Professor with Northeastern University and the Dean of the School of Control Engineering at Qinhuangdao Campus. He has published over 150 international journal papers within his areas of interest, which include intelligent transportation systems, cyber-physical systems, etc.

Dr. Guo is an Associate Editor of the *IEEE Transactions on Intelligent Transportation Systems,* the *IEEE Transactions on Intelligent Vehicles*, the *Information Sciences*, the *IEEE Intelligent Transportation Systems Magazine*, the *ACTA Automatica Sinica*, the *China Journal of Highway and Transport* and the *Journal of Control and Decision*. He was an honoree of the New Century Excellent Talents in University, Ministry of Education, China, in 2004, and a nominee for Gansu Top Ten Excellent Youths by the Gansu Provincial Government in 2005. He won the CAA Young Scientist Award in 2017 and the First Prize of Natural Science Award of Hebei Province in 2020.

Shijie Zhao received the B.S. degree from Northeastern University, Shenyang, China, in 2019. He is currently pursuing the Master's degree in Northeastern University. His research interests include 3D point cloud processing and 3D vision.