IEEE Open Journal of
Intelligent Transportation
Systems

Received 10 February 2023; revised 28 April 2023; accepted 21 May 2023. Date of publication 8 June 2023; date of current version 23 June 2023.

Digital Object Identifier 10.1109/OJITS.2023.3283768

3-D Vehicle Detection Enhancement Using Tracking Feedback in Sparse Point Clouds Environments

YEQIANG QIAN [1] (Member, IEEE), XIAOLIANG WANG [2,3], HANYANG ZHUANG [4] (Member, IEEE), CHUNXIANG WANG [2,3], AND MING YANG [2,3] (Member, IEEE)

[1]Global Institute of Future Technology, Shanghai Jiao Tong University, Shanghai 200240, China

[2]Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China

[3]Key Laboratory of System Control and Information Processing, Ministry of Education, Shanghai Jiao Tong University, Shanghai 200240, China

[4]University of Michigan–Shanghai Jiao Tong University Joint Institute, Shanghai Jiao Tong University, Shanghai 200240, China

CORRESPONDING AUTHOR: M. YANG (e-mail: mingyang@sjtu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62103261, Grant 62203294, and Grant 62173228; and in part by the 2022 Open Project Program of Guangxi Key Laboratory of Automobile Components and Vehicle Technology under Grant 2022GKLACVTKF03.

ABSTRACT In recent years, vehicle detection in intelligent transportation systems using 3D LIDAR point clouds based on deep neural networks has made substantial progress. However, when the point clouds are very sparse, the detection model cannot generate proposals efficiently, resulting in false negative results. Considering that the object tracking technology accurately predicts vehicles based on historical measurements and motion models, and these prediction results can become proposals for object detection. Therefore, this paper proposes a novel object detection paradigm based on tracking feedback to address the false negative problem based on sparse point clouds. According to the distribution of the state vector from the Kalman prediction, multiple proposals are sampled and fed back to the second stage of two-stage detection models. After regression and non-maximum suppression, the false negative results can be effectively reduced. This method enhances the vehicle detection capability of classical neural networks. Comparing the recall metric of multiple detection models at different distances in the public KITTI and nuSences datasets, the proposed method can promote up to 5.31% compared to the previous method, which reflects the effectiveness and versatility of the proposed method.

INDEX TERMS Vehicle detection, tracking feedback, sparse point clouds.

I. INTRODUCTION

DRIVEN by data, computation, and algorithms, deep learning-based vehicle detection methods have been greatly improved in terms of accuracy and robustness [1]. Since the 3D LIDAR point cloud can directly obtain the depth information of the target, the deep learning method based on the 3D LIDAR point cloud has also been developed rapidly [2], [3]. However, such method has encountered the problem of point cloud sparsity, and sparsity increases with distance. Fig. 1 shows the change in the number of points for the same vehicle at different distances. When the distance of the vehicle is 12.21 m, the point clouds are dense, and the vehicle outline is roughly visible. However, when the distance is 62.46 m, there are only 6 points, and the point cloud is very sparse, which makes vehicle detection based on sparse point clouds more challenging. Sparse point clouds are a common phenomenon in industrial applications [4].

Specifically, when the object point cloud is very sparse, the neural networks are unable to generate candidate proposals, or the generated proposals are ignored due to low confidence. It is difficult to extract semantic features of the point clouds, resulting in false negative results. For the two-stage detection model, the number and quality of candidate proposals impose the upper bound of detection recall. Thus, improving the number and quality of candidate proposals is the critical issue.

Since the actual point cloud is continuous data, we naturally try to solve this problem with object tracking

The review of this article was arranged by Associate Editor Guoyuan Wu.

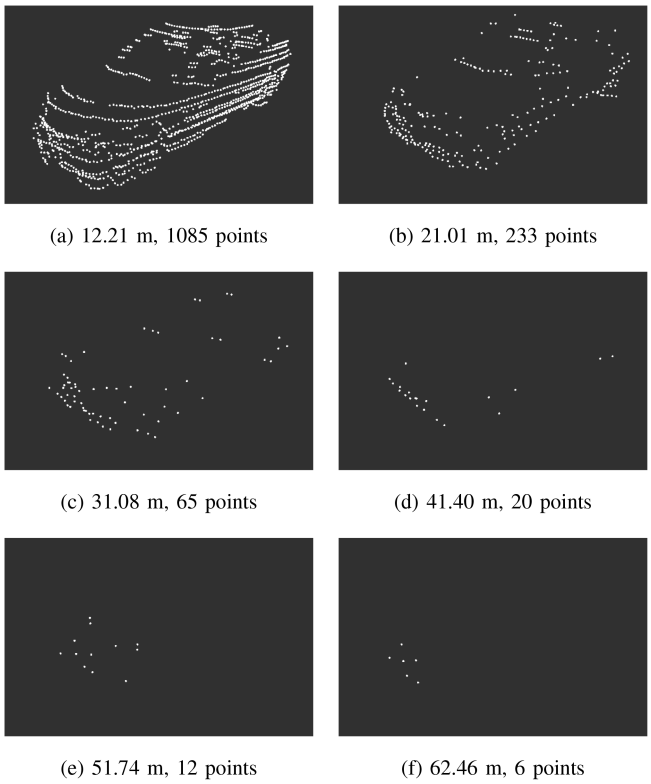


FIGURE 1. Changes in the number of points at different distances from the same vehicle, the data is from the KITTI dataset [5].

technology. Object tracking uses continuous multi-frame detection results to filter inaccurate measurements and estimate high-order states (velocity, acceleration, etc.). For vehicles in a structured road environment, their motion is regular [6], and object tracking can predict the objects' states accurately by using the historical motion state and motion model. These prediction results can be used as candidate proposals for object detection to improve the number of proposals.

Although object state prediction integrates historical measurements and motion models, prediction results are often incorrect due to the inaccuracy of measurements and the noise of the motion model. When the vehicle is gradually far away from the sensor, the inaccuracy of measurements increases, which leads to a larger error of the prediction results. Therefore, if prediction results are directly used as the candidate proposals, the network may not return the correct results. To solve this problem, this paper samples multiple prediction proposals as candidates according to the probability distribution of the predicted state to further improve the number and quality of the proposals.

Therefore, to address the challenge of false negative results caused by sparse point clouds, this paper proposes an object detection method based on tracking feedback, which modifies the previous tracking-by-detection paradigm and uses the sampled prediction proposals generated by object tracking as candidates to improve the detection recall. The main contributions of this paper are as follows:

1) A novel object detection paradigm based on tracking feedback to address the false negative problem is proposed.
2) The method adapts to various two-stage object detection networks, and enhances the long-distance vehicle detection capability of the classical neural network.
3) A sampling strategy to further improve the number and quality of candidates based on the probability distribution of predicted states is introduced.

The remainder of this paper is organized as follows: Section II presents the related work of object detection and tracking. In Section III, the proposed novel object detection paradigm is presented in detail. Experimental results are presented in Section IV. Finally, we conclude the paper in Section V.

II. RELATED WORKS

Since we aim to address the false negative issue based on tracking feedback, the following related works are carried out in both 3D object detection and 3D object tracking.

A. 3D OBJECT DETECTION

Object detection methods using 3D LIDAR point clouds can be divided into two categories: rule-based methods and deep learning-based methods. The rule-based method decomposes object detection into multiple steps: ground segmentation, object clustering, and bounding box generation. The deep learning-based methods use deep neural networks to extract features from data and generate end-to-end object bounding boxes.

Given the 3D point cloud data, the rule-based method first uses the ground segmentation algorithm to segment it into ground point clouds and obstacle point clouds [7]. The ground point cloud refers to the points hit on the ground and the obstacle point cloud refers to the points hit on obstacles, such as vehicles, pedestrians, and trees. Since ground segmentation is a process of binary segmentation, obstacle point clouds are discrete points without instance-level information. Therefore, it is necessary to cluster the obstacle point cloud and divide it into independent object clusters [8]. For the point cloud clusters obtained by object clustering, their sizes and shapes vary, which is inconvenient for the use of subsequent tracking and prediction modules. Therefore, these point cloud clusters need to be uniformly characterized as bounding boxes [9].

Due to high accuracy and robustness, deep learning methods have become the mainstream method of object detection in the last few years, and can be further divided according to the representation of the point cloud as 2D projection representation-based methods, 3D voxel representation-based methods, raw point cloud representation-based methods, and multiple representation fusion-based methods [10].

Since deep learning-based 2D object detection in images has made great progress, many researchers use a plane or spherical projection to transform the 3D point cloud to 2D to ensure that standard 2D object detection methods can

be used. Among them, the method based on plane projection generally projects the point cloud onto a birds-eye view, where each pixel contains information, such as height, intensity, density, etc. The only difference in the image detection method is that the orientation of the object needs to be estimated [11]. The spherical projection-based method generates a 2D image with the height of the number of laser beams and the width of rotation resolution, which preserves the neighborhood relationship between points [12]. Although the methods based on 2D projection representation can use mature 2D detection algorithms, they inevitably cause the loss of accuracy or introduce scale variation in the image.

The 3D voxel representation method divides the 3D physical space into neatly arranged voxels to ensure that the irregular point cloud can be characterized by a 3D matrix, and the 2D convolution operation can also be extended to 3D [13]. In contrast from the method based on 2D projection representation, it retains the structural information of the point cloud, and thus, the detection accuracy is higher. However, due to the sparseness of the point cloud, there are many empty voxels, which seriously affect the efficiency of the detection algorithm. Therefore, some scholars have proposed sparse convolution, which can greatly reduce the computational cost by constructing a hash table [14]. Although the method based on 3D voxel representation makes full use of 3D information of the point cloud, it inevitably brings information loss to the process of voxelization.

To reduce the information loss caused by projection or voxelization, a method based on the raw point cloud representation is proposed [15]. The pioneering work of this method is led by PointNet, which uses a multilayer perceptron to extract features for each point, and then aggregates global features through a pooling layer to ensure the independence of point order [16]. PointNet++ extends it to encode more complex features through a hierarchical structure, which improves the feature expression and receptive field [17].

To make full use of the advantages of different representations, an increasing number of methods consider the fusion of multiple different representations. Reference [18] combined the raw point cloud representation and the 3D voxel representation. The 3D voxel branch uses 3D sparse convolution to encode features and generate high-quality candidate proposals. The raw point cloud branch aggregates semantic features in voxels at different scales and predicts the weight of the points.

B. 3D OBJECT TRACKING

For object tracking, tracking-by-detection has emerged as the preferred paradigm, which first uses the object detection algorithm to locate the position of all objects in the scene. Then, data association and state estimation are performed across multiple frames to generate the object trajectory. Thus, the following mainly expands on two aspects: data association and state estimation.

Data association aims to find the correspondence between multiple objects across multiple frames and mainly includes two key issues: similarity measurement and association problem solving. In multi-object tracking, the most commonly used similarity metric is the distance between objects, such as the Euclidean distance and the Mahalanobis distance. The distance metric is simple and efficient, but it is not robust to handle complex scenes [19]. To tackle this challenge, many researchers have proposed other similarity measurement metrics, including direction distance, bounding box size distance, point num distance, histogram distance and a combination of multiple metrics [20]. However, these handcrafted metrics have limited representability. To increase the discrimination between different objects, recent works explore the use of deep structured support vector machines, convolutional neural networks and recurrent neural networks to measure similarity [21].

To solve the association problem, the nearest neighbor method successfully takes the pair with the largest similarity as the association and then deletes this pair from the object list. These two steps iterate until the similarity score is less than a given threshold [19]. To achieve global optimization, the Hungarian algorithm regards the data association problem as a linear assignment problem and achieves maximum matching by finding an augmentation path [22]. Compared with the nearest neighbor algorithm to find a one-to-one correspondence between multiple objects, an object in Probabilistic Data Association can correspond to multiple agents with different probabilities [23]. For decoupling, Joint Probabilistic Data Association first calculates the joint probability between objects, then calculates the marginal probability, and finally uses the Probabilistic Data Association [24]. Compared with the previous method, which only processes current observations, multiple hypothesis tracking considers the connection of observations between multiple frames [25]. In the last few years, deep learning methods have also been applied to solve association problems, such as recurrent neural networks and convolutional neural networks [26].

State estimation refers to filtering the multi-frame detection results of the same object and estimating high-order states, such as velocity and acceleration. The Kalman filter is the most popular method in state estimation, which uses the linear system state equation to optimally estimate the system state [27]. The essence of the Kalman filter is to iteratively fuse two normal distributions and obtain a new normal distribution so it can only deal with linear problems. In real applications, the motion model of the object is generally nonlinear, and thus, some researchers have proposed the extended Kalman filter. The extended Kalman filter uses the first-order Taylor expansion to approximate the nonlinear function and calculates the Jacobian matrix [28]. However, when the motion model is more complicated, it may be difficult to calculate the Jacobian matrix analytically, and when the motion model is not continuous, the Jacobian matrix cannot be calculated. To solve these problems, the unscented

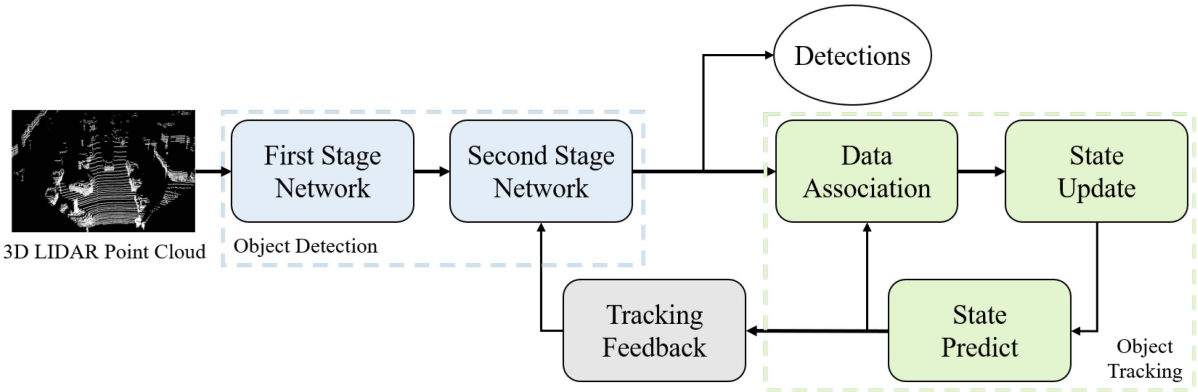


FIGURE 2. The framework of our vehicle detection method based on tracking feedback.

Kalman filter is proposed to approximate the probability distribution [29]. Specifically, the unscented Kalman filter selects a set of minimum sampling points so that its mean and variance are the same as the probability distribution. Therefore, the nonlinear function transforms these sampling points to calculate the new state. To break the normal distribution assumption, the particle filter expresses unknown distributions by sampling a large number of particles into the space [30]. The sampled distribution may be random at first, but it will finally converge to a specific distribution as the system runs.

III. PROPOSED METHOD

A. FRAMEWORK OVERVIEW

In this work, we modify the conventional tracking-by-detection paradigm by feeding back the sampled predictions as detection candidates to tackle the false negative challenge over a long distance. The overall framework is shown in Fig. 2, which consists of three modules: object detection, object tracking and tracking feedback.

The work in this paper is aimed at a two-stage object detection network, and the network structure has been changed. In the traditional two-stage network, the first-stage network is feature extraction and candidate box generation, which is input to the second-stage network, and the second-stage network is candidate box pooling and position optimization. In our work, the candidate boxes input to the second-stage network are not only generated in the first stage, but also from the tracking feedback module. Our tracker will sample multiple prediction boxes as candidates according to the distribution of the predicted state of the target state, which greatly increases the number of candidate boxes entering the second-stage network, improves the recall rate of the vehicle, and thus improves the vehicle detection performance.

For object tracking, data association is applied to find the correspondence between objects in adjacent frames, and state estimation filters the historical measurements of the same object. Among them, state estimation mainly includes two parts: state prediction and state updating, where the result of state prediction is used for tracking feedback. For tracking feedback, we sample multiple candidate proposals according to the probability distribution of the state vector obtained by state prediction, and then feed these proposals back to the second stage of the detection model for further optimization.

B. OBJECT DETECTION

As mentioned above, deep learning-based 3D object detection has become the most popular method, and can be further divided into one-stage methods and two-stage methods. The one-stage method uses the powerful feature learning capabilities of deep neural networks to infer the category and position of the object in one network, but the accuracy cannot be guaranteed. The two-stage method first generates candidate proposals, and then verifies and optimizes them in terms of score and position. Thus, the detection accuracy of the two-stage method is often higher than that of the one-stage method, and it is usually used for long-distance vehicle detection [10].

Specifically, for the two-stage object detection method, the first-stage network extracts semantic features of point clouds by different representations. Then, it generates candidate proposals using anchor-free or anchor-based methods. The second-stage network pools semantic features and spatial features together, refining the bounding box in terms of position, size and score. Note that most two-stage object detection networks can be applied into our methods.

C. OBJECT DATA ASSOCIATION

In actual road scenes, there are generally multiple targets. To continuously track the same target, it is necessary to determine the corresponding relationship between the current frame and the previous frame to perform target data association. As the core problem in multitarget tracking, many classic methods have been proposed in the last few decades, such as the nearest neighbor, global nearest neighbor, probabilistic data association, joint probabilistic data association, and multi-hypothesis tracking.

A data association matrix is a method to describe the problem of target association in two frames through a

two-dimensional matrix. It stores the similarity between targets. We use the current frame measurement to construct a detection list with a dimension of m, and the target construction dimension of the previous frame prediction is n for the prediction list. Then, the data association matrix is as follows:

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{bmatrix} \tag{1}$$

Among them, C is the data association matrix, and element c_{ij} represents the mth detection target and the similarity of the nth predicted target. For similarity calculation, we choose the three-dimensional intersection of union (IoU) of two target bounding boxes for this paper. Among them, B_g and B_p are the bounding boxes of the two targets as follows:

$$c_{ij} = \frac{|B_g \cap B_p|}{|B_g \cup B_p|} \tag{2}$$

Among them, the numerator is the intersection of two bounding boxes and the denominator is the union of two bounding boxes. According to the equation, it can be seen that it not only describes the similarity of the two target positions but also includes the similarity of the target size and orientation.

For each frame, we suppose there are m detected objects and n predicted objects, and c_{ij} is the similarity measured by the 3D intersection of Union between two objects. The data association problem can be regarded as a linear allocation problem, and its mathematical form is as follows:

$$\begin{aligned} &\max \quad E = \sum_{i}^{m}\sum_{j}^{n} w_{ij}c_{ij} \\ s.t. \quad &\textstyle\sum_{j=1}^{n} w_{ij} = 1, \qquad i = 1\ldots, m; \\ &\textstyle\sum_{i=1}^{m} w_{ij} = 1, \qquad j = 1\ldots, n; \\ &w_{ij} \in \{0, 1\}, \qquad i = 1\ldots, m, \quad j = 1\ldots, n; \end{aligned} \tag{3}$$

where w_{ij} is the element of the permutation matrix, $w_{ij} = 1$ means that the i_{th} detected object and the j_{th} predicted object are the same, and $w_{ij} = 0$ means that it is different.

D. OBJECT STATE ESTIMATION

After target data association, the detection results of the same target in different frames are obtained. The primary purpose of target state estimation is to reasonably estimate the motion state position, velocity, and acceleration, as well as to carry out the state in the future forecast. In the actual tracking process, there is noise in the sensor data and errors in the target detection results, which have an unstable effect on the tracking results. Target state estimation can filter these errors and noises to make the tracking results more stable. When the target is missed, the state of the target can also be predicted. This prediction is used as a candidate region in the next section. Common target state estimation methods include the Kalman filter, the histogram filter, the particle filter, etc. Among them, the Kalman filter is simple and efficient, and it is the most commonly used method in target tracking. This article is also based on the Kalman filter for state estimation.

For 3D object detection, the object is represented as $(x, y, z, l, w, h, \theta)$, where (x, y, z) is the object's position and (l, w, h) are the length, width and height of the object, respectively. θ is the object's orientation about the z-axis. In the urban road environment, objects generally move along the ground plane, and thus, the movement along the z-axis can be ignored. At the same time, the size of the object changes little during tracking. Thus, we model the object's state as follows:

$$x = \begin{bmatrix} x & y & \theta & v & \dot{\theta} \end{bmatrix}^T \tag{4}$$

where v and $\dot{\theta}$ are the linear and angular speeds of the object, respectively.

For object state prediction, the constant turn rate and velocity (CTRV) model is applied, which assumes that the linear and angular velocities of the object are constant. Thus, it is consistent with the movement in a real traffic scene.

According to the definition of the CTRV model, the differential expression of the state vector is as follows:

$$\dot{x} = \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \\ \dot{\theta} \\ 0 \\ 0 \end{bmatrix} \tag{5}$$

Given this motion model and the object's previous state x_k, the state at the current frame $x_{k+1|k}$ can be predicted using the extended Kalman filter as follows:

$$x_{k+1|k} = g(x_k) + \nu \tag{6}$$

$$P_{k+1|k} = JP_kJ^T + Q \tag{7}$$

where $g(x)$ is the state transition function, P is the state covariance matrix and its initial value is generally specified manually. Q is the process covariance matrix, or the noise vector $\nu \sim \mathcal{N}(0, Q)$. For more details, refer to [31].

Given the correspondence between the detected objects and the predicted objects, the state at the current frame can be updated as follows:

$$K_{k+1} = P_{k+1|k}H^T\left(HP_{k+1|k}H^T + R\right)^{-1} \tag{8}$$

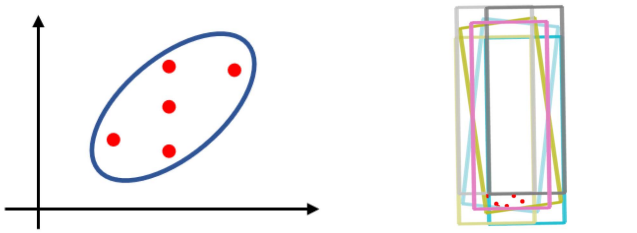
$$x_{k+1} = x_{k+1|k} + K_{k+1}\left(z_{k+1} - Hx_{k+1|k}\right) \tag{9}$$

$$P_{k+1} = (I - K_{k+1}H)P_{k+1|k} \tag{10}$$

where $z = \begin{bmatrix} x & y & \theta \end{bmatrix}^T$ is the measurement vector, $H_{3\times5} = \begin{bmatrix} I & 0 \end{bmatrix}$ is the measurement matrix, R is the measurement noise covariance matrix, K is the Kalman gain, and I is the identity matrix.

In this way, the overall process of object tracking is as follows:

1) We predict the object state from the previous frame using equations (6) and (7).



(a) Sampled points (b) Sampled bounding boxes

FIGURE 3. An illustration of state sampling.

2) We associate detected objects and predicted objects by solving equation (3) using the Hungarian algorithm.
3) We update the object state at the current frame using equations (8) to (10).
4) We iterate repeatedly 1) to 3).

E. TRACKING FEEDBACK

Since the Kalman filter is an optimal estimation algorithm, it assumes that the state vector obeys a normal distribution and selects the mean of the state vector as output. Although the probability of the true state falling at the mean position for the normal distribution is the highest, more states can be obtained by sampling the probability distribution to ensure that the probability of including the true state and the accuracy of the bounding box regression is further improved.

For state sampling of a normal distribution, we refer to the selection of σ points in unscented transformation [32]. One of the sampling states is located at the mean of the normal distribution, and the other sampling states are symmetrically distributed at the main axis of the covariance matrix (each dimension has two). For an n-dimensional normal distribution with mean μ and variance Σ, $2n+1$ sampling states can be obtained as follows:

$$\begin{aligned} x^0 &= \mu \\ x^i &= \mu + \left(\sqrt{(n+\lambda)\Sigma}\right)_i, \quad i = 1, 2, \ldots, n \\ x^i &= \mu - \left(\sqrt{(n+\lambda)\Sigma}\right)_{i-n}, \quad i = n+1, n+2, \ldots, 2n \end{aligned} \tag{11}$$

where $\lambda = \alpha^2(n+\kappa) - n$ is the regulating parameter, which determines the distance between the sampling state and the mean value. The hyperparameter α determines the spread of the sigma points and is usually set to a small positive value. In our experiment, α is set to 1e-3. The hyperparameter κ is a secondary scaling parameter that is set to 0 in our experiment. $(\sqrt{(n+\lambda)\Sigma})_i$ represents the i_{th} row of the square root of the matrix. To calculate the square root of the matrix, the Cholesky decomposition is applied, which decomposes the symmetric positive definite matrix into the product of the lower triangular matrix and its transpose. Fig. 3(a) shows an illustration of state sampling for a 2D normal distribution, where the blue ellipse is the covariance ellipse and the red points are sampling results.

For state vector $x = \left[x\ y\ \theta\ v\ \dot{\theta}\right]^T$, v and $\dot{\theta}$ are second-order and have no relationship with the object's bounding box. Thus, only the position and orientation of the object are applied for sampling. For each object, the mean and the variance of the predicted state are obtained using equations (6) and (7). By taking the first three dimensions to form a 3D normal distribution, the state is sampled according to equation (11). As a result, seven sampling states are obtained. To acquire the complete bounding box of objects, the sampled states are expanded using the size of the mean state, and the final sampled bounding boxes are shown in Fig. 3(b). The red points are the object point cloud, the pink bounding box is the sampled state at the mean location, the bounding boxes of other colors are symmetrically distributed near the mean location, and all bounding boxes have the same size and height.

After state sampling, multiple bounding boxes are obtained and fed into the second-stage network together with the feature extracted by the first-stage network. In this way, after further optimization and non-maximum suppression of the second-stage network, the final detection results are obtained with fewer false negative results.

IV. EXPERIMENT

To validate the effectiveness of the proposed method, we conduct a comprehensive experiment on the KITTI benchmark dataset and nuSences benchmark dataset. First, we give a brief introduction to the two datasets. Second, the detection accuracy and the recall before and after tracking feedback are evaluated. Finally, the experimental results are visualized and further analyzed.

A. DATASET

Among the datasets commonly used in autonomous driving systems, the most famous are the KITTI dataset [5] and the nuSences dataset [33]. The KITTI dataset includes several tasks, such as object detection, object tracking, and road segmentation. The data in the object detection benchmark are randomly scrambled, but in the object tracking benchmark, they are stored continuously. Since we employ tracking results to solve the false negative problem, temporal information is needed, and we validate the proposed method in the object tracking benchmark.

The object tracking benchmark in the KITTI dataset consists of 21 training sequences and 29 test sequences, where the test sequences can only be evaluated on the private KITTI dataset evaluation server with object tracking metrics (MOTA, MOTP, etc.). The detection models employed in this paper are trained in object detection benchmarks. Therefore, we test our method on the training sequences with 8008 aggregated frames. The nuSences dataset contains 1000 driving sequences for both object detection and tracking.

TABLE 1. Comparison of AP before and after tracking feedback in the KITTI dataset.

Model	Before(AP)	After(AP)	Promote(AP)	Before(recall)	After(recall)	Promote(recall)
PointRCNN	80.24	84.65	4.41	82.47	88.27	5.80
Part-A^2Net	85.06	86.78	1.72	87.98	90.17	2.19
PV-RCNN	88.69	89.80	1.11	91.98	93.40	1.42

TABLE 2. Comparison of the AP before and after tracking feedback in the nuSences dataset.

Model	Before(AP)	After(AP)	Promote(AP)
PointRCNN	70.6	75.9	5.3
Part-A^2Net	76.5	78.9	2.4
PV-RCNN	81.5	83.7	2.2
PointPillars	68.4	73.5	5.1
CenterPoint	85.2	85.9	0.7
Pointformer	82.3	84.6	2.3

B. PERFORMANCE EVALUATION

Our method adapts to various two-stage object detection networks without extensive modification. To validate this, we employ three classic two-stage detection models: PointRCNN [15], Part-A^2Net [34], PV-RCNN [18], PointPillars [35], CenterPoint [36] and Pointformer [37].

We first evaluate the most widely used metric AP of three detection models before and after tracking feedback, and the results are shown in Tables 1 and 2. Among the three detection models, the performance of PV-RCNN is better than Part-A^2Net and further better than PointRCNN. After tracking feedback, the AP of all three detection models has been improved. Notably, the PointRCNN detection model has been promoted by 4.41% in the KITTI dataset and 5.3% in the nuSences dataset, and the gap with Part-A^2Net has been dramatically reduced. It can be observed that all model progress has been improved to varying degrees. We also notice that our method improves significantly for models with lower accuracy.

Since this paper focuses on tackling the false negative challenge and the recall metric reflects the missing rate of objects, the recall of the three detection models is evaluated and shown in Tables 1 and 2. After tracking feedback, the recall of the three detection models has been improved, especially for the PointRCNN detection model. Its recall has been improved by up to 5.80% in the KITTI dataset, which is why its AP has increased considerably.

C. ABLATION STUDY ACCORDING TO THE DISTANCE

To evaluate the effectiveness of the proposed method for long-distance objects, the recall metrics of three detection models are evaluated at different distances and shown in Table 3. The experiment is conducted in the KITTI dataset. It can be seen that for three detection models, the improvement of detection recall increases with distance after tracking feedback. For the PointRCNN detection model, when the object distance is 40-50 m, the detection recall is increased up to 6.40%. For the Part-A^2Net detection model, the improvement in recall reaches 2.85% when the distance is 40-50 m. Although the improvement is reduced when the distance is 60-70 m, it also reaches 2.21%. The PV-RCNN detection model has the best detection performance. The improvement is relatively small compared to the other two methods, reaching a maximum of 1.73% when the object distance is 50-60 m.

In summary, the proposed tracking feedback method can effectively reduce false negative results, especially for long-distance objects with sparse point clouds. Due to the reduction of a false negative results, the average precision of object detection has also been improved.

D. ABLATION STUDY OF THE STATE VECTOR SAMPLING

We compare the scheme that only samples the mean of the state variable, because the probability of the true state falling at the mean is highest. The result is shown in Table 4. The table shows that state vector sampling has a good effect, because state vector sampling provides more states, corresponding to more candidate bounding boxes, and the probability of containing the real state will be further improved.

E. RUNTIME

In addition to evaluating the accuracy of the proposed method, we test the runtime on all frames of the training sequences. The algorithm is implemented in Python, running on a desktop computer with Intel Core i7-4790 CPU and GeForce GTX TITANX GPU. The running time (ms) of the three detection models before and after tracking feedback is shown in Table 5. It can be seen that the increase in runtime is only approximately 10-20 ms, which has little effect on the raw detection models.

F. QUALITATIVE RESULTS

For an intuitive analysis, we visualize the quantitative results before and after the tracking feedback of two detection models, PointRCNN and PV-RCNN. Fig. 4 shows the result of the PointRCNN detection model. The object with ID 7 is detected successfully after tracking feedback approximately 45 m away from the ego-car. It is worth noting that there is another object approximately 60 m away from the ego-car behind object 7, but since the object distance is too far, the object cannot be detected and tracked for multiple frames, and therefore, cannot be fed back.

The qualitative results of the PV-RCNN detection model before and after tracking feedback are shown in Fig. 5. The

TABLE 3. Comparison of the recall before and after tracking feedback for three detection models at different distances.

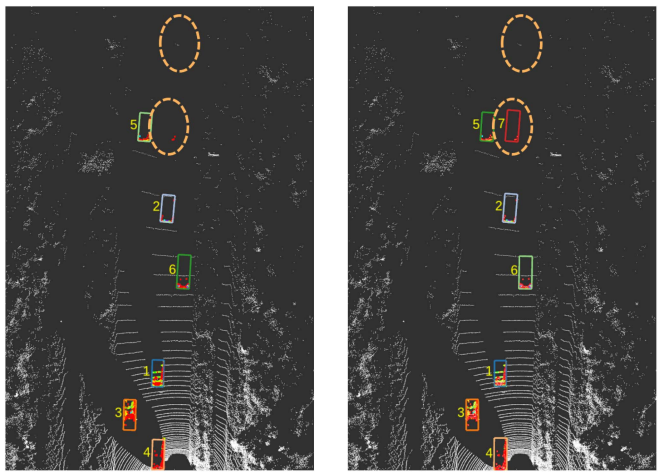
Distance(m)	PointRCNN			Part-A^2Net			PV-RCNN		
	Before	After	Promote	Before	After	Promote	Before	After	Promote
0-10	98.55	99.16	0.61	98.45	98.35	-0.10	98.26	98.36	0.10
10-20	97.72	99.03	1.31	98.32	98.65	0.33	98.41	98.25	-0.16
20-30	91.68	94.95	3.27	95.60	96.55	0.95	96.23	97.19	0.96
30-40	84.67	89.70	5.03	93.51	95.14	1.63	96.41	97.08	0.67
40-50	80.58	86.98	6.40	85.21	88.06	2.85	92.49	93.88	1.39
50-60	53.50	59.73	6.23	63.96	66.66	2.70	78.66	80.39	1.73
60-70	26.70	31.08	4.38	44.66	46.87	2.21	62.70	64.29	1.59

TABLE 4. Ablation study of the state vector sampling based on PointRCNN.

Strategy	AP	Recall
Original	80.24	82.47
With state vector sampling	84.65	88.27
Without state vector sampling	82.77	85.92

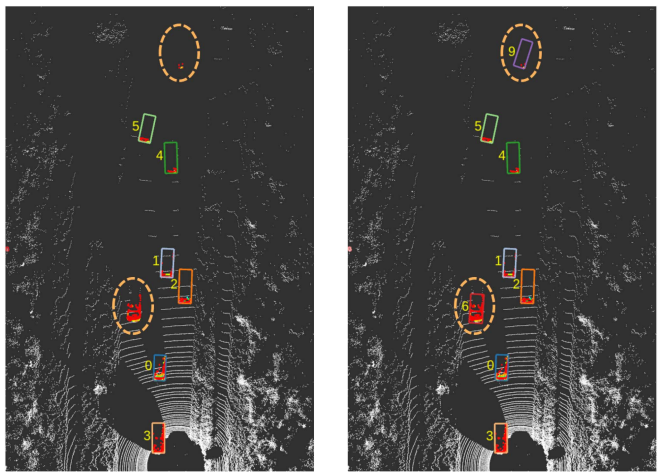
TABLE 5. Comparison of runtime before and after tracking feedback for three detection models.

Model	Before(ms)	After(ms)	Diff(ms)
PointRCNN	272.12	285.77	13.65
Part-A^2Net	276.39	290.04	13.65
PV-RCNN	226.13	248.97	22.84



(a) before feedback (b) after feedback

FIGURE 4. Comparison of tracking feedback results for the PointRCNN detection model.



(a) before feedback (b) after feedback

FIGURE 5. Comparison of tracking feedback results for the PV-RCNN detection model.

object with ID 6 is approximately 25 m away from the ego-car and not detected due to the design or training defect of the model itself. However, the object is detected successfully after tracking feedback. That is the reason why the maximum promotion of the PV-RCNN detection model is approximately 20 m. At the same time, for the long-distance object with ID 9, it cannot be detected because there are only 6 points, but our tracking feedback method can also successfully detect it.

V. CONCLUSION AND PROSPECT

In this paper, we propose a vehicle detection method based on tracking feedback, which can effectively address the challenge of false negative results due to the sparse point clouds. For object detection, the two-stage detection model with higher accuracy is employed. For object tracking, the Hungarian algorithm is applied for data association, and the extended Kalman filter is applied for state estimation. The predicted state is sampled and fed into the second-stage network for better refinement using tracking feedback, and more candidate proposals with higher quality are generated. Experiments show that the detection recall and the precision of the three detection models have been improved, especially for long-distance vehicles.

Our method builds the tracker based on the CTRV model, so it is more accurate when the speed changes of surrounding vehicles are small. This method will fail if the speed or direction of the vehicle changes drastically. This is the limitation of this method. The method in this paper may work better on the highway. In the future, for different scenes,

integrating different motion model trackers to make the method perform better in more scenes will be an important work in the future.

REFERENCES

[1] X. Zhang, B. A. Story, and D. Rajan, "Night time vehicle detection and tracking by fusing vehicle parts from multiple cameras," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8136–8156, Jul. 2021.

[2] Y. Cui et al., "Deep learning for image and point cloud fusion in autonomous driving: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, pp. 722–739, Feb. 2020.

[3] W. Liu, W. Sun, and Y. Liu, "DLOAM: Real-time and robust LiDar SLAM system based on CNN in dynamic urban environments," *IEEE Open J. Intell. Transp. Syst.*, early access, Sep. 1, 2021, doi: 10.1109/OJITS.2021.3109423.

[4] B. Schlager, T. Goelles, M. Behmer, S. Muckenhuber, J. Payer, and D. Watzenig, "Automotive LiDar and vibration: Resonance, inertial measurement unit, and effects on the point cloud," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 426–434, 2022.

[5] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The Kitti dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[6] S. Mukherjee, A. M. Wallace, and S. Wang, "Predicting vehicle Behavior using automotive radar and recurrent neural networks," *IEEE Open J. Intell. Transp. Syst.*, vol. 2, pp. 254–268, 2021.

[7] B. Douillard et al., "Hybrid elevation maps: 3D surface models for segmentation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2010, pp. 1532–1538.

[8] M. Montemerlo et al., "Junior: The stanford entry in the urban challenge," *J. Field Robot.*, vol. 25, no. 9, pp. 569–597, 2008.

[9] D. Kim, K. Jo, M. Lee, and M. Sunwoo, "L-shape model switching-based precise motion tracking of moving vehicles using laser scanners," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 598–612, Feb. 2018.

[10] Y. Wu, Y. Wang, S. Zhang, and H. Ogai, "Deep 3D object detection networks using LiDAR data: A review," *IEEE Sensors J.*, vol. 21, no. 2, pp. 1152–1171, Jan. 2021.

[11] M. A. Javed, M. Tahir, and K. Ali, "Attitude in motion: Constraints aided accurate vehicle orientation tracking in harsh environment," *IEEE Trans. Ind. Informat.*, vol. 19, no. 5, pp. 6450–6459, May 2023.

[12] G. P. Meyer, A. Laddha, E. Kee, C. Vallespi-Gonzalez, and C. K. Wellington, "LaserNet: An efficient probabilistic 3D object detector for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 12677–12686.

[13] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 4490–4499.

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

[15] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 770–779.

[16] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 652–660.

[17] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.

[18] S. Shi et al., "PV-RCNN: Point-Voxel feature set abstraction for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 10529–10538.

[19] H.-K. Chiu, A. Prioletti, J. Li, and J. Bohg, "Probabilistic 3D multi-object tracking for autonomous driving," 2020, *arXiv:2001.05673*.

[20] "3D obstacle perception." Apollo. 2023. [Online]. Available: https://github.com/ApolloAuto/apollo

[21] S. Giancola, J. Zarzar, and B. Ghanem, "Leveraging shape completion for 3D siamese tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 1359–1368.

[22] 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)*, 2020, pp. 10359–10366.

[23] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *IEEE Control Syst. Mag.*, vol. 29, no. 6, pp. 82–100, Dec. 2009.

[24] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Multi-target tracking using joint probabilistic data association," in *Proc. IEEE Conf. Decis. Control Including Symp. Adapt. Processes*, 1980, pp. 807–812.

[25] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. 24, no. 6, pp. 843–854, Dec. 1979.

[26] E. Baser, V. Balasubramanian, P. Bhattacharyya, and K. Czarnecki, "FANTrack: 3D multi-object tracking with feature association network," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2019, pp. 1426–1433.

[27] X. Li, K. Wang, W. Wang, and Y. Li, "A multiple object tracking method using Kalman filter," in *Proc. IEEE Int. Conf. Inf. Autom. (ICIA)*, 2010, pp. 1862–1866.

[28] S. Yang and M. Baum, "Extended Kalman filter for extended object tracking," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 4386–4390.

[29] S. Thrun, "Probabilistic robotics," *Commun. ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[30] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Robust tracking-by-detection using a detector confidence particle filter," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2009, pp. 1515–1522.

[31] T. Omeragić and J. Velagić, "Tracking of moving objects based on extended Kalman filter," in *Proc. Int. Symp. ELMAR*, 2020, pp. 137–140.

[32] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proc. Adapt. Syst. Signal Process., Commun., Control Symp.*, 2000, pp. 153–158.

[33] H. Caesar et al., "nuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 11621–11631.

[34] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2647–2664, Aug. 2021.

[35] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12697–12705.

[36] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3D object detection and tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11784–11793.

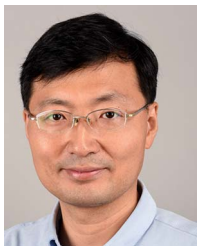
[37] X. Pan, Z. Xia, S. Song, L. E. Li, and G. Huang, "3D object detection with pointformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7463–7472.

YEQIANG QIAN (Member, IEEE) received the Ph.D. degree in control science and engineering from Shanghai Jiao Tong University, Shanghai, China, in 2020, where he is currently a Postdoctoral Fellow with the University of Michigan–Shanghai Jiao Tong University Joint Institute. His main research interests include computer vision, pattern recognition, machine learning, and their applications in intelligent transportation systems.

XIAOLIANG WANG received the M.S. degree in control science and engineering from Shanghai Jiao Tong University, Shanghai, China, in 2020. His current research interests include ground segmentation, object detection, and classification using LIDAR sensors. At the same time, he has a large interest in multisensor fusion, including camera, LIDAR, and RADAR.

HANYANG ZHUANG (Member, IEEE) received the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2018, where he is currently an Assistant Research Professor implementing research works related to intelligent vehicles. He has worked in industry on ADAS system development and validation. His research focus is on improving the safety and efficiency of intelligent vehicles through V2X technology and cooperative intelligent transportation systems.

CHUNXIANG WANG received the Ph.D. degree in mechanical engineering from the Harbin Institute of Technology, Harbin, China, in 1999. She is currently an Associate Professor with the Department of Automation, Shanghai Jiao Tong University, Shanghai, China. Her research interests include robotic technology and electromechanical integration.



MING YANG (Member, IEEE) received the master's and Ph.D. degrees from Tsinghua University, Beijing, China, in 1999 and 2003, respectively. He is currently a Full Tenure Professor with Shanghai Jiao Tong University and the Deputy Director of the Innovation Center of Intelligent Connected Vehicles. He has been working in the field of intelligent vehicles for more than 20 years. He participated in several related research projects, such as the THMR-V project (first intelligent vehicle in China), European CyberCars and CyberMove projects, CyberC3 project, CyberCars-2 project, ITER transfer cask project, and AGV.