Multi-object Tracking Based on a Novel Feature Image with Multi-modal Information

Yi An, *Member, IEEE*, Jialin Wu, Yunhao Cui, and Huosheng Hu, *Life Senior Member, IEEE*

Abstract—**Multi-object tracking technology plays a crucial role in many applications, such as autonomous vehicles and security monitoring. This paper proposes a multi-object tracking framework based on the multi-modal information of 3D point clouds and color images. At each sampling instant, the 3D point cloud and image acquired by a LiDAR and a camera are fused into a color point cloud, where objects are detected by the Point-GNN method. And, a novel height-intensity-density (HID) image is constructed from the bird's eye view. The HID image truly reflects the shapes and materials of objects and effectively avoids the influence of object occlusion, which is helpful to object tracking. In two sequential HID images, a new rotation kernel correlation filter is proposed to predict the objects. Furthermore, an object retention module and an object re-recognition module are developed to overcome the object matching failure in the in-between frames. The proposed method takes full advantage of the multi-modal data and effectively achieves the information complementation to improve the accuracy of multi-object tracking. The experiments with the KITTI dataset show that the proposed method has the best performance among the existing traditional multi-object tracking methods.**

Index Terms—**Image processing, kernel correlation filter, multi-object tracking, point cloud, tracking by detection.**

I. Introduction

OBJECT tracking is to estimate the trajectory of an object as it moves in a scene [1]. In the driver assistance systems of intelligent vehicles, object tracking is a key technology of environment perception [2-3]. Object tracking lays a foundation for obstacle avoidance, path planning, and adaptive cruise [4]. Besides, object tracking is also widely used in military defense and security monitoring [5-6].

According to the number of tracked objects, object tracking is divided into single-object tracking and multi-object tracking (MOT). Since most application scenes contain multiple objects of interest, MOT has become a major research area recently. MOT generally follows the tracking-by-detection paradigm that the objects of interest are detected first, and then the same object in subsequent frames is associated into a trajectory.

In general, MOT methods are divided into deep learning methods and traditional methods. The deep learning MOT methods tend to have higher accuracy. However, they require a large number of training samples. We need to label the relationships between the objects in multiple frames. The labeling process is expensive and time-consuming.

The traditional MOT methods directly obtain correlation information from data and don't need to label the relationships between the objects in multiple frames. This greatly reduces the costs of the application and improves the universality. Most of the existing traditional MOT methods are based on images, but images are easily affected by illumination and occlusion. Rapid improvements in 3D rangefinder technology allow us to digitize the shape and surface characteristics of objects accurately and conveniently. 3D point clouds acquired by LiDARs provide the geometric information of the sampling points on the surfaces of physical objects. And, the 3D point clouds are unaffected by illumination. They are the effective supplements to images and contribute to achieve accurate positioning and tracking. Thus, this paper uses the multi-modal information of 3D point clouds and images to improve the accuracy of MOT.

This paper proposes a MOT framework based on the multi-modal information of 3D point clouds and color images. At each sampling instant, the 3D point cloud and image acquired by a LiDAR and a camera are fused into a color point cloud. Then, we obtain the point cloud within the visual field of the camera, which is used to construct a novel height-intensity-density (HID) image. At the same time, the objects are detected from the 3D color point cloud by the Point-GNN method [7]. In two sequential HID images, a new rotation kernel correlation filter (RKCF) is proposed to predict the objects. Then, the predicted objects are associated with the detected objects by the Kuhn-Munkres algorithm. Furthermore, an object retention module and an object re-recognition module are designed to overcome the object matching failure in the in-between frames. The proposed method in Fig. 1 makes full use of multi-modal

Manuscript received 11 March 2022; revised 9 July 2022, 25 December 2022 and 5 February 2023; accepted 18 March 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62173055 and Grant 61673083, in part by the Science and Technology Major Project of Shanxi Province under Grant 20191101014, and in part by the Key Research and Development Project of Xinjiang Uygur Autonomous Region under Grant 2022C1001-1. *(Corresponding author: Yi An.)*

Yi An is with the School of Control Science and Engineering, Dalian University of Technology, Dalian 116023, China, and also with the School of Electrical Engineering, Xinjiang University, Urumqi 830046, China (e-mail: anyi@dlut.edu.cn).

Jialin Wu is with the School of Control Science and Engineering, Dalian University of Technology, Dalian 116023, China (e-mail: wjl980718@mail.dlut.edu.cn).

Yunhao Cui is with the School of Mechanical Engineering, Dalian University of Technology, Dalian 116023, China (e-mail: cyhlovegl@mail.dlut.edu.cn).

Huosheng Hu is with the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: hhu@essex.ac.uk).

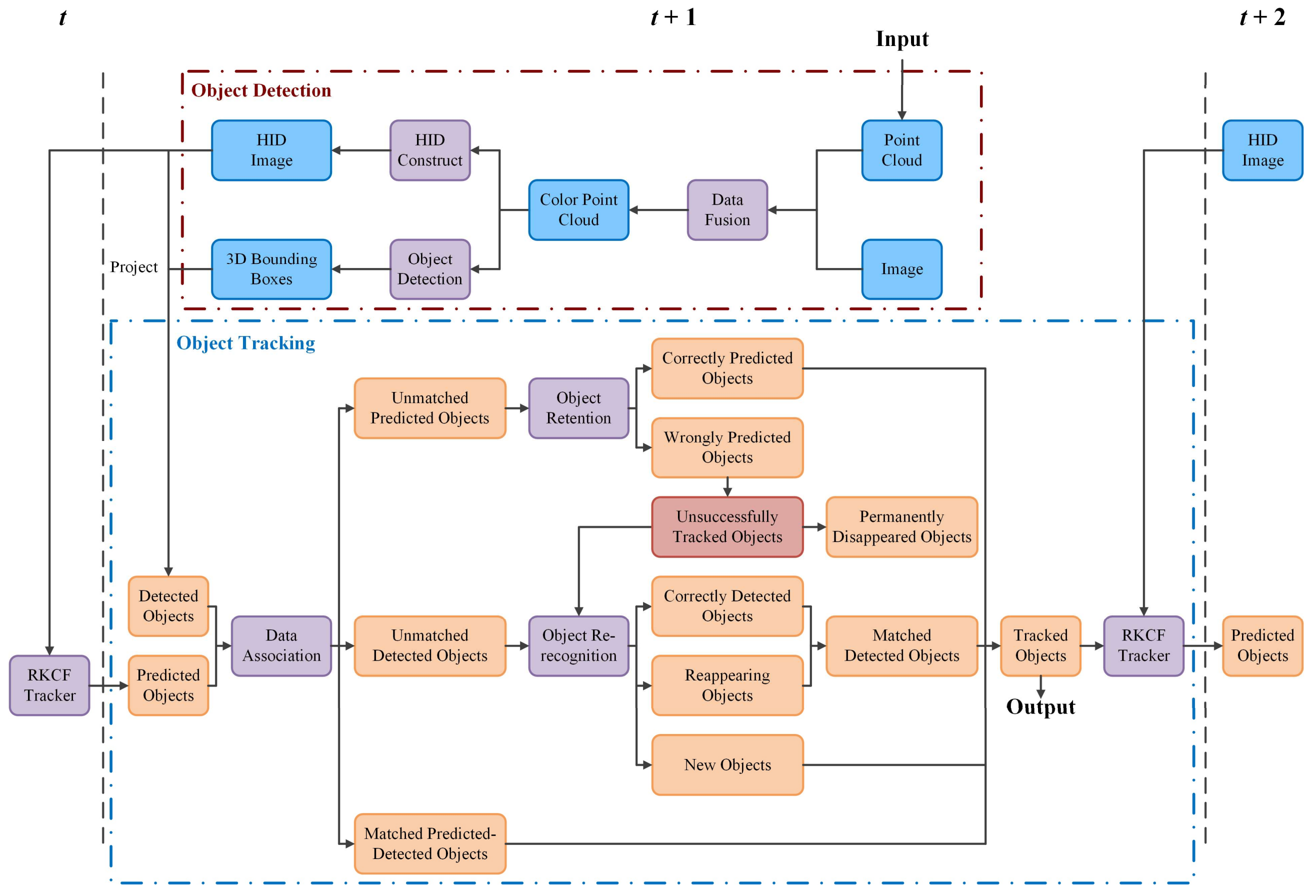


Fig. 1. Framework of our multi-object tracking.

data and effectively achieves information complementation to improve the tracking accuracy. The contributions of this paper are as follows:

1) A novel weighted height-intensity-density (HID) image is constructed for MOT. Due to the weighted merge of the height, intensity, and density information from point cloud data, the HID image contains more information that truly reflects the shapes and materials of objects.

2) A rotation kernel correlation filter (RKCF) is developed to predict the objects in the HID images. The rotation mechanism makes the base sample more accurate and avoids information loss in feature extraction, which effectively reduces the accumulated error and ensures the accuracy of prediction.

3) An object retention module and an object re-recognition module are developed based on the color image to overcome the object matching failure in the in-between frames. The multi-modal information of 3D point clouds and color images complement each other in our proposed multi-object tracking framework.

These technical features make our multi-object tracking method accurate and reliable. Our method is based on the multi-modal information of the point cloud and the color image. The weighted HID image is constructed from the point cloud. The object retention module and the object re-recognition module are developed based on the color image. In a word, this paper proposes a multi-object tracking framework based on the multi-modal information of 3D point clouds and color images.

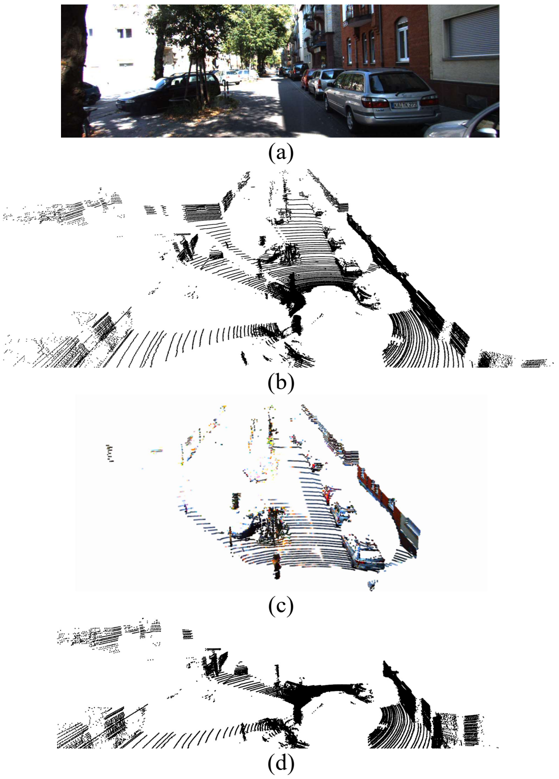
The rest of the paper is organized as follows. Section II reviews the previous work related to this research. The HID image construction is described in Section III. In Section IV, the MOT framework is detailed. Experimental results are presented in Section V to show the performance of our method. Section IV concludes this paper.

II. Related Work

A. Deep learning MOT Methods

Zhang et al. [8] designed mmMOT, which uses Vgg-net and PointNet++ to extract image and LiDAR point cloud features respectively, and establishes a multi-modal fusion module. Luiten et al. [9] proposed MOTSFusion, which combines the depth map, color image, optical flow, and camera pose to achieve object tracking by using optical flow estimation. Erkan et al. [10] proposed FANTrack that uses CNN for data association and builds Siamese networks to model the similarities between tracks and detections.

Mono3DT [11] and QD-3DT [12] use 3D box depth-ordering matching for robust instance association. Zhou et al. [13] proposed CenterTrack to localize objects and predict the associations with the previous frame. Tokmakov et al. [14] presented PermaTrack based on CenterTrack with a recurrent memory module. Wu et al. [15] proposed PC-TCNN, which introduces a tracklet proposal design on point clouds. Wang et al. [16] proposed DiMOT, which extracts both spatial feature and temporal feature to learn the correlation relationship. LGM [17] and TrackMPNN [18] achieve object tracking without


(a)

(b)

(c)

(d)

Fig. 2. Multi-model information: (a) the color image, (b) the 3D point cloud, (c) the 3D color point cloud, and (d) the 3D colorless point cloud.

appearance information based on graph neural networks.

The accuracy of these supervised deep learning MOT methods depends on the number of training samples. Supervised deep learning methods usually require a large amount of time and manual work to label the relationship between objects in different frames. Unsupervised deep learning MOT methods can avoid these problems to some extent.

Gonzalez et al. [19] defined an affinity measure based on position, appearance, and optical flow affinity. It uses an unsupervised neural network to select key points to be tracked by the optical flow. Favyen et al. [20] proposed a method which constructs two different inputs for the same sequence of video, calculates trajectories by applying an unsupervised RNN model on each input independently, and trains the model to produce consistent trajectories across both inputs. Shyamgopal et al. [21] proposed SimpleReID, which is categorized as learning by generating labels. It first generates tracking labels using a traditional MOT method and trains a ReID network to predict the generated labels.

Although these unsupervised deep learning methods avoid the expense of extensive annotation, they are not sufficiently accurate. One reason is that they only use single-modal information. In addition, the labels of some unsupervised MOT methods are generated from the predictions of traditional tracking methods. If the predictions are incorrect, the trained model will also be inaccurate, resulting in poor accuracy of the tracking result.

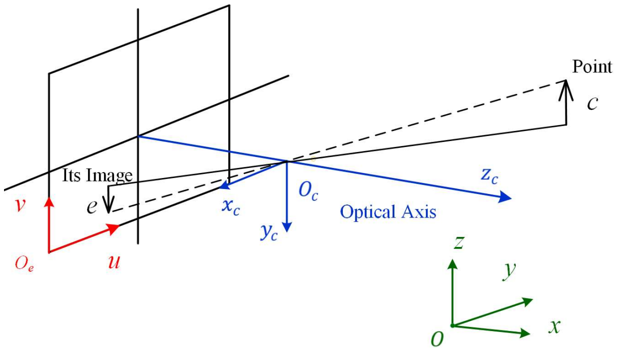


Fig. 3. Pinhole camera model.

B. Traditional MOT Methods

Hostettler et al. [22] proposed a method for vehicle tracking on roadways, which uses measurements of magnetometers and accelerometers fused by particle filtering. This method only uses motion state information for tracking. An et al. [23] proposed a contour-based method, which uses the contours of the objects to obtain a robust association between detection and tracking. This method only uses LiDAR information. Weng et al. [24] proposed AB3DMOT, which uses a 3D Kalman filter and Hungarian algorithm for state estimation and data association. Reich et al. [25] presented an extended Kalman filter for 3D state estimation. Kim et al. [26] proposed EagerMOT, which integrates object observations from LiDAR and the color image by two-stage association. The three methods only use 2D or 3D detections obtained from the color image or the point cloud to perform 3D tracking, not using any appearance cue. Kanmasekera et al. [27] presented MASS, which defines a dissimilarity measure based on the motion, appearance, structure, and size of an object.

Most of the above traditional MOT methods only use single-modal information, resulting in lower accuracy. Our method takes full advantage of point cloud information and image information. We not only construct a novel weighted height-intensity-density (HID) image by the point cloud, but also develop an object re-recognition module and an object retention module by using the color image.

III. The HID Image Construction

A. Multi-modal Data Fusion

Our tracking method is mainly used for autonomous vehicles and mobile robots. These intelligent systems use different kinds of sensors to collect the environmental data and perceive the environment. Generally, at each sampling instant, a 3D point cloud (a set of discrete points) and a color image are collected by a laser rangefinder (LRF) and a camera respectively, as shown in Fig. 2(a) and Fig. 2(b). To take full advantage of these multi-modal data in multi-object tracking, the 3D point cloud and the color image are fused to a 3D color point cloud by the geometric mapping relationship, i.e. perspective projection, between the LRF and the camera as follows

$$s\vec{\boldsymbol{e}} = A[R,\boldsymbol{t}]\vec{\boldsymbol{p}} \tag{1}$$

where s is an arbitrary scale factor, A is the intrinsic parameter matrix, and $[R\ \boldsymbol{t}]$ is the extrinsic parameter matrix between the

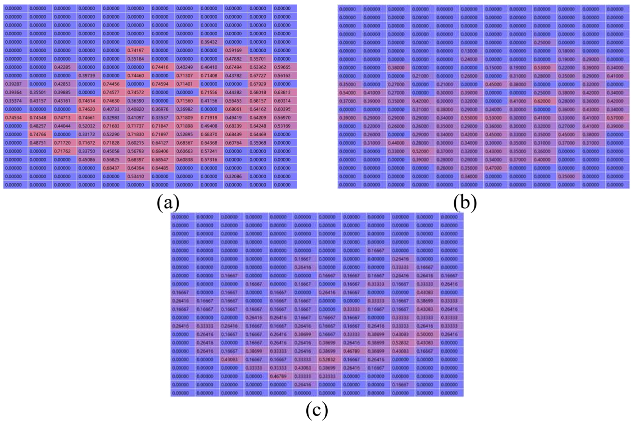


Fig. 4. The value distribution of a HID image patch: (a) the height channel, (b) the intensity channel, and (c) the density channel.

LRF and the camera. $\boldsymbol{p} = [x, y, z]^T$ denotes a laser point in the 3D point cloud in the laser coordinate system and $\boldsymbol{e} = [u, v]^T$ is its image projection in the pixel coordinate system. $\vec{\boldsymbol{p}} = [x, y, z, 1]^T$ and $\vec{\boldsymbol{e}} = [u, v, 1]^T$ are their homogeneous coordinates. The laser point is projected to the imaging plane and is colored by its corresponding pixel in the color image. Therefore, a color point cloud is obtained, as shown in Fig. 2 (c). The geometric mapping relationship, i.e. the intrinsic parameter A and the extrinsic parameter $[R\ \boldsymbol{t}]$, is obtained by the calibration of the LRF and camera according to the pinhole camera model [28], as shown in Fig. 3.

In general, the visual field of the LRF is 360°, while the visual field of the camera is about 60°. Because of the difference of visual fields between two sensors, some laser points, which are within the visual field of the camera, can be projected to the imaging plane and colored by their image projection. The other laser points, which are out of the visual field of the camera, can't be projected to the imaging plane and have no image projection. Therefore, we can divide the 3D point cloud $P = \{\boldsymbol{p}_i = (x_i, y_i, z_i) | 1 \le i \le n\}$ into two parts: 3D color point cloud $P_c = \{\boldsymbol{p}_{ci} = (x_{ci}, y_{ci}, z_{ci}) | 1 \le i \le n_c\}$ and 3D colorless point cloud $P_e = \{\boldsymbol{p}_{ei} = (x_{ei}, y_{ei}, z_{ei}) | 1 \le i \le n_e\}$, as shown in Fig. 2(c) and Fig. 2(d). The 3D color point cloud P_c can also be denoted by $P_c = \{\boldsymbol{p}_{ci} = (x_{ci}, y_{ci}, z_{ci}, R_{ci}, G_{ci}, B_{ci}) | 1 \le i \le n_c\}$, where $R_{ci} = R(u_{ci}, v_{ci})$, $G_{ci} = G(u_{ci}, v_{ci})$, and $B_{ci} = B(u_{ci}, v_{ci})$ denote the three-primary color of the image projection $\boldsymbol{e}_{ci} = (u_{ci}, v_{ci})$ of $\boldsymbol{p}_{ci}$. In addition, the LRF can also measure the reflection intensity t_{ci} of each laser point $\boldsymbol{p}_{ci}$. Therefore, we obtain a 7D laser point $\boldsymbol{p}_{ci} = (x_{ci}, y_{ci}, z_{ci}, R_{ci}, G_{ci}, B_{ci}, t_{ci})$.

B. HID Image Construction of 3D Point Clouds

We construct a 2D grid on the xoy plane and place the 3D point cloud P_c without RGB information into the 2D grid. The size of the grid is set to 0.1m. The points in a cell are $C(u, v) = \{\boldsymbol{p}_{ci} = (x_{ci}, y_{ci}, z_{ci}) | \boldsymbol{p}_{ci} \in P_c, x_{min} + (u-1)d \le x_{ci} \le x_{min} + ud, y_{min} + (v-1)d \le y_{ci} \le y_{min} + vd\}$ where u and v are the column and row numbers of the cell, x_{min} and y_{min} are the minima of x and y-coordinates, and d is the size of the cell. $1 \le u \le \widetilde{m}$ and $1 \le v \le \widetilde{n}$, where $\widetilde{m} = \lceil (x_{max} - x_{min})/d \rceil$ and $\widetilde{n} = \lceil (y_{max} - y_{min})/d \rceil$, x_{max} and y_{max} are the maxima of x and y-coordinates, and $\lceil \cdot \rceil$ denotes ceil$(\cdot)$.

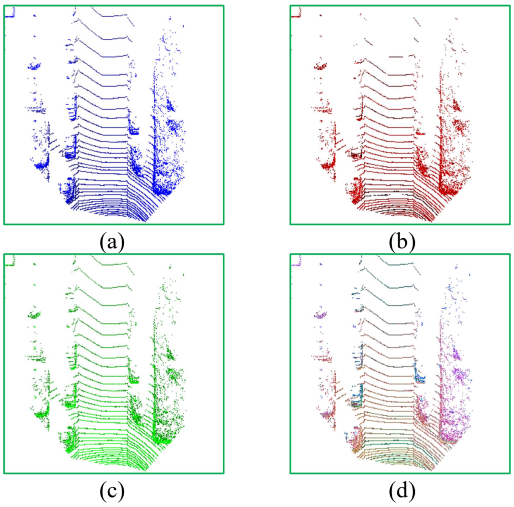


Fig. 5. HID image construction: (a) the height channel, (b) the intensity channel, (c) the density channel, and (d) the HID image.

For each cell, we compute three features: height, intensity, and density as follows

$$h(u, v) = \frac{\max\{z_{ci} | \boldsymbol{p}_{ci} \in C(u, v)\} - z_{min}}{z_{max} - z_{min}} \quad (2)$$

$$i(u, v) = t_{ci} \text{ corresponding to } \max\{z_{ci} | \boldsymbol{p}_{ci} \in C(u, v)\} \quad (3)$$

$$d(u, v) = \min\left\{1.0, \frac{\log(N_{uv} + 1)}{\log 64}\right\} \quad (4)$$

where z_{min} and z_{max} represents the maximum and minimum height of point clouds respectively, t_{ci} is the reflection intensity of $\boldsymbol{p}_{ci}$ and N_{uv} is the number of the points in $C(u, v)$. The values of $h(u, v)$, $i(u, v)$ and $d(u, v)$ are all between 0-1.

Fig. 4 visualizes the value distribution of a HID image patch in the height channel, the intensity channel, and the density channel separately. The brighter the color in the distribution map, the larger the value. It can be seen obviously that the feature of the height channel is stronger than others. If three channels are directly merged into the non-weighted HID image, the effective information of the density channel and the intensity channel will be covered by the height channel. The weight is set as:

$$r_h : r_i : r_d = \frac{1}{SUM_h} : \frac{1}{SUM_i} : \frac{1}{SUM_d} \quad (5)$$

we compute the sum of the values of each point in the three channels separately as follows:

$$SUM_h = \sum_u \sum_v h(u, v) \quad (6)$$

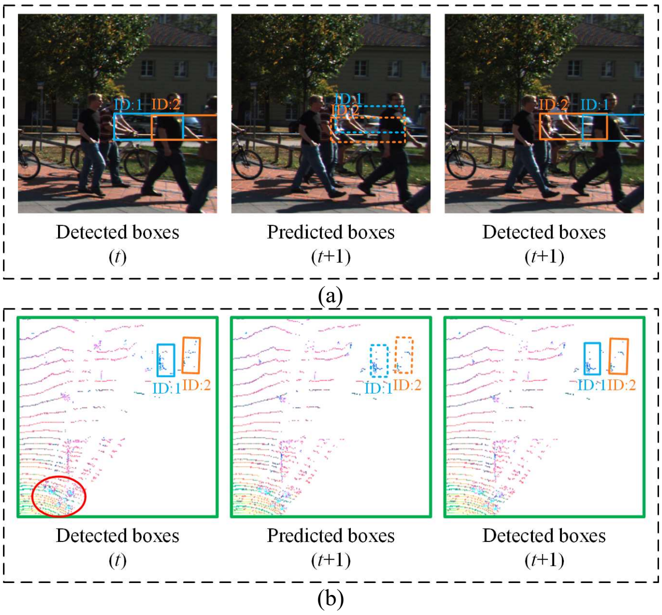
$$SUM_i = \sum_u \sum_v i(u, v) \quad (7)$$

$$SUM_d = \sum_u \sum_v d(u, v) \quad (8)$$

Specifically, it needs to be explained here why the weight is set not to emphasize channels with stronger features, but to achieve a balance. This is because the information in the three channels is complementary and all contribute to the accuracy improvement.

To use these features to construct a new feature image, we need to normalize them to 0-255 according to the rule of the color channel as follows

$$hid_{\max} = \max\{[h(u, v)r_h, i(u, v)r_i, d(u, v)r_d]\} \quad (9)$$



Detected boxes (t) Predicted boxes (t+1) Detected boxes (t+1)

(a)

Detected boxes (t) Predicted boxes (t+1) Detected boxes (t+1)

(b)

Fig. 6. Results of multi-object tracking in (a) color images and (b) HID images. (The bounding box of each object is represented by a different color box and marked with its own ID, the red circle in (b) shows the position of pedestrians in the HID image)

$$hid_{\min} = \min\{[h(u,v)r_h, i(u,v)r_i, d(u,v)r_d]\} \tag{10}$$

$$\bar{h}(u,v) = \left[\frac{h(u,v)r_h - hid_{\min}}{hid_{\max} - hid_{\min}} 255\right] \tag{11}$$

$$\bar{\imath}(u,v) = \left[\frac{i(u,v)r_i - hid_{\min}}{hid_{\max} - hid_{\min}} 255\right] \tag{12}$$

$$\bar{d}(u,v) = \left[\frac{d(u,v)r_d - hid_{\min}}{hid_{\max} - hid_{\min}} 255\right] \tag{13}$$

We obtain three channels: height $H = \{\bar{h}(u,v) | 1 \le u \le \tilde{m}, 1 \le v \le \tilde{n}\}$, intensity $I = \{\bar{\imath}(u,v) | 1 \le u \le \tilde{m}, 1 \le v \le \tilde{n}\}$, and density $D = \{\bar{d}(u,v) | 1 \le u \le \tilde{m}, 1 \le v \le \tilde{n}\}$, as shown in Fig. 5(a), (b), and (c). As a result, these three feature channels H, I, and D are merged into a new feature image, called the height-intensity-density (HID) image, as shown in Fig. 5(d).

Compared with the color image, the HID image is more robust and can effectively avoid the problems of object occlusion. Fig. 6 shows the tracking results in color images and HID images respectively. The color image-based tracking method is described in detail in Section V-B. In Fig. 6, the dotted line represents the predicted bounding box and the solid line represents the detected bounding box. As shown in Fig. 6 (a), although the objects are successfully detected at both the t sampling instant and the $t + 1$ sampling instant, the predicted bounding boxes drift during the tracking based on the color image. Therefore, the predicted objects and the detected objects at the $t + 1$ sampling instant match incorrectly, and the IDs of the two vehicles are exchanged. It can be seen obviously that the tracking drift is actually caused by the occlusion of pedestrians in the foreground in Fig. 6 (a), which is avoided in the HID image in Fig. 6 (b).

Compared with the color image, the HID image has some advantages. Background clutter leads to redundant features extracted in color images, while the top-down view of the HID image avoids complex information. Therefore, the object feature extracted from HID images is clearer, which ensures the accuracy of prediction. Object occlusion causes bounding boxes in the color image to overlap while bounding boxes in the HID image are independent of each other. Therefore, the objects in the HID image will not be affected by nearby objects, which improves the tracking accuracy.

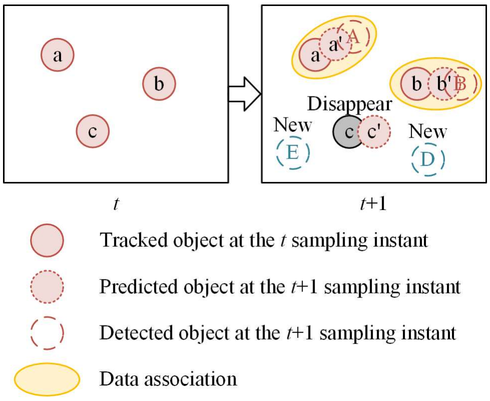

a
b
c
a
a'
A
b
b'
B
Disappear
New
E
c
c'
New
D
t
t+1
Tracked object at the t sampling instant
Predicted object at the t+1 sampling instant
Detected object at the t+1 sampling instant
Data association

Fig. 7. Process of the tracking-by-detection method.

IV. Multi-object Tracking

A. Problem Description

MOT is to correlate objects in consecutive sampling instants. Fig. 7 describes the tracking-by-detection paradigm. As can be observed, at the t sampling instant, there are 3 tracked objects a, b and c. At the $t + 1$ sampling instant, the position of tracked objects can be predicted by the tracker, and objects in the scene are also need to detected at the same time. Detected objects need to be associated with predicted objects at the $t + 1$ sampling instant. After data association, there are three kinds of outputs: matched predicted-detected objects, unmatched detected objects and unmatched predicted objects.

In this process, there are three main problems:

1) How to correlate objects correctly;
2) How to deal with unmatched predicted objects;
3) How to deal with unmatched detected objects.

To correlate objects correctly, we construct a HID image and develop a RKCF tracker to predict the objects in the HID images. For unmatched predicted objects, we build an object retention module. It checks whether the prediction is correct and retains correctly predicted objects to compensate for the missed detection. For unmatched detected objects, we build an object re-recognition module. It can not only compensate for the wrong prediction, but also deal with a tracked object which disappears temporarily and then reappears.

B. Multi-object Detection from 3D Point Clouds

In our method, the Point-GNN is used to detect objects from the 3D color point cloud P_c at each sampling instant [7]. The network will output 3D bounding boxes and detection confidences of objects. Because we have obtained the geometric mapping relationship between the LRF and the camera in Section III-A, we can project the 3D bounding boxes into the imaging plane by using (1), and obtain the corresponding 2D bounding boxes in the image. In addition, we

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2023.3259999

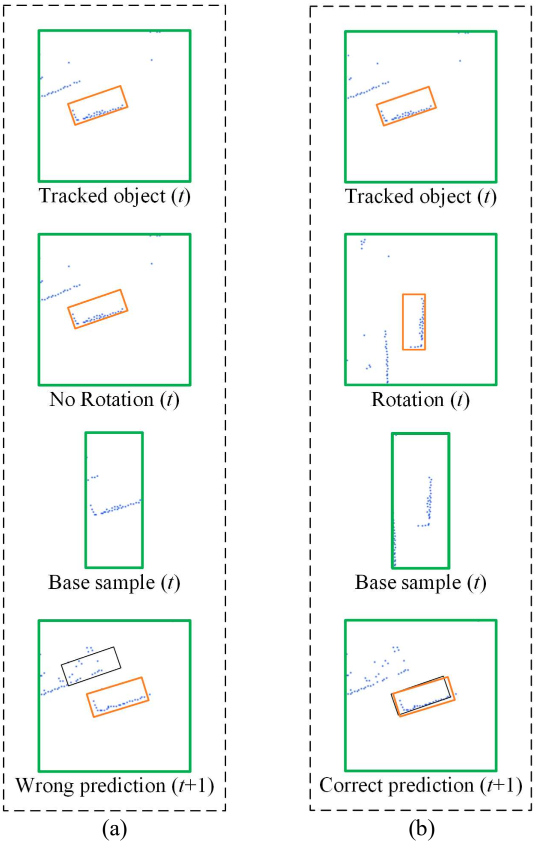


Fig. 8. Results of multi-object tracking (a) without rotation and (b) with rotation.

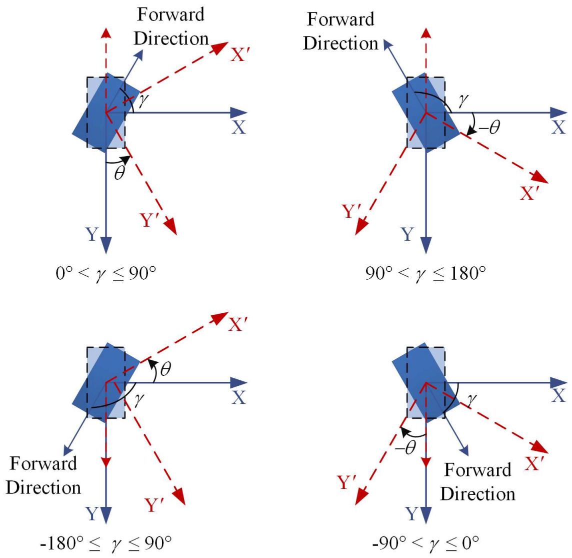


Fig. 9. HID image rotation.

can also project the 3D bounding boxes into the 2D grid on the xoy plane vertically, and obtain the corresponding 2D bounding boxes in the HID image. At the t sampling instant, the 2D bounding boxes of the detected objects in the HID image D_t are denoted by $B_t^D = \{\boldsymbol{b}_{ti}^D | 1 \le i \le n_t^D\}$.

C. Object Prediction Based on HID Images

The sequence of the HID images acquired at the different sampling instants t is denoted by $D = \{D_t | 1 \le t \le n_t\}$. The

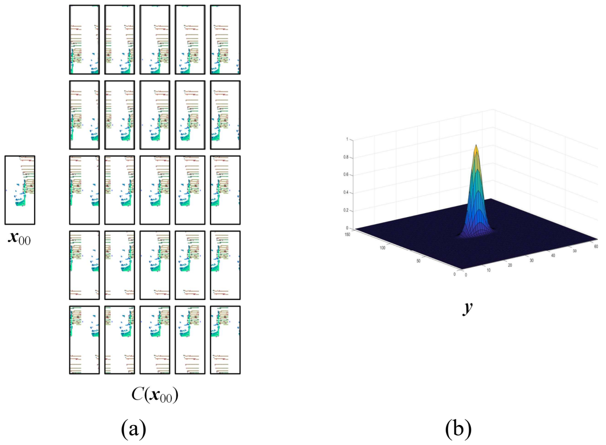


Fig. 10. Kernel regression model: (a) the circulant matrix and (b) regression output.

2D bounding boxes of the tracked objects in the HID image D_t are denoted by $B_t = \{\boldsymbol{b}_{ts} = (u_{ts}, v_{ts}, l_{ts}, w_{ts}, \gamma_{ts}) | 1 \le s \le n_t\}$, where (u_{ts}, v_{ts}), (l_{ts}, w_{ts}), and γ_{ts} are the center, size, and heading angle of the 2D bounding box in the HID pixel coordinate system. To achieve better object tracking effect in the HID images, we propose a rotation kernel correlation filter (RKCF), which includes three steps as follows.

1) Rotating the HID Image at the t Sampling Instant

As described in the kernel correlation filter [29], to train a classifier, we need to construct the training data which include a base sample and several virtual samples. The base sample is an image patch which has the same center as the 2D surrounding box of a tracked object. It is 3 times the size of 2D surrounding box of a tracked object. The virtual samples are obtained by cyclically shifting the base sample. The training data have a great effect on the accuracy of the classifier. If the 2D surrounding box is tilted, as shown in Fig. 8(a), the base sample may loss some key information of the tracked object, which will affect the virtual samples. In order to obtain a standard base sample, we hope that the 2D surrounding box has the same orientation as the image, as shown in Fig. 8(b).Thus, for each 2D surrounding box $\boldsymbol{b}_{ts}$, we should rotate the HID image D_t through θ_{ts} with (u_{ts}, v_{ts}) as the rotation center first, and then extract the patch around the 2D surrounding box $\boldsymbol{b}_{ts}$ from the rotated HID image $\overline{D}_t$ as the standard base sample. If $0 \le \gamma_{ts} \le \pi$, $\theta_{ts} = 90° - \gamma_{ts}$; if $-\pi \le \gamma_{ts} \le 0$, $\theta_{ts} = -90° - \gamma_{ts}$. If $\theta_{ts} > 0$, the rotation direction is anticlockwise; if $\theta_{ts} < 0$, the rotation direction is clockwise, as shown in Fig. 9.

2) Training the Classifier at the t Sampling Instant

Let $\boldsymbol{x}_{00} \in R^{m \times n}$ denote the based sample which is represented by color features. Without loss of generality, m and n are set as an odd number respectively. Let $\bar{m} = (m-1)/2$ and $\bar{n} = (n-1)/2$. Since the virtual samples are obtained by cyclically shifting the base sample $\boldsymbol{x}_{00}$, $\{\boldsymbol{x}_{ij} = P^i \boldsymbol{x}_{00} Q^j | -\bar{m} \le i \le \bar{m}, -\bar{n} \le j \le \bar{n}, i \ne 0, j \ne 0\}$ denotes the virtual samples, where P is a row cyclic shift operator (permutation matrix) and Q is a column cyclic shift operator. If $i > 0$, the base sample $\boldsymbol{x}_{00}$ shifts down i rows; if $i < 0$, the base sample $\boldsymbol{x}_{00}$ shifts up i rows. If $j > 0$, the base sample $\boldsymbol{x}_{00}$ shifts right j columns; if $j < 0$, the base sample $\boldsymbol{x}_{00}$ shifts left j columns. As a result, we can obtain the training samples $X = \{\boldsymbol{x}_{ij} = P^i\boldsymbol{x}_{00}Q^j \mid -\bar{m} \le i \le \bar{m}, -\bar{n} \le j \le \bar{n}\}$, which are rewritten as the circulant matrix $X = C(\boldsymbol{x}_{00}) \in R^{mm\times nn}$.

$$X = C(\boldsymbol{x}_{00}) = \begin{bmatrix} \boldsymbol{x}_{(-\bar{m})(-\bar{n})} & \boldsymbol{x}_{(-\bar{m})(-\bar{n}+1)} & \cdots & \boldsymbol{x}_{(-\bar{m})\bar{n}} \\ \boldsymbol{x}_{(-\bar{m}+1)(-\bar{n})} & \boldsymbol{x}_{(-\bar{m}+1)(-\bar{n}+1)} & \cdots & \boldsymbol{x}_{(-\bar{m}+1)\bar{n}} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x}_{\bar{m}(-\bar{n})} & \boldsymbol{x}_{\bar{m}(-\bar{n}+1)} & \cdots & \boldsymbol{x}_{\bar{m}\bar{n}} \end{bmatrix} \quad (14)$$

where C denotes the cyclic shift, as shown in Fig. 10(a).

Each sample $\boldsymbol{x}_{ij}$ corresponds to an output y_{ij}, which denotes the probability that the object moves i rows and j columns. The output y_{ij} obeys the 2D Gaussian distribution, as shown in Fig. 10. By using the samples $\boldsymbol{x}_{ij}$ and their outputs y_{ij}, we train the classifier, i.e. the kernel regression model

$$f(\boldsymbol{z}) = w^T\phi(z) = \sum_i\sum_j \alpha_{ij}\phi(\boldsymbol{x}_{ij})^T\phi(\boldsymbol{z}) \quad (15)$$

where $\boldsymbol{z}$ is a sample, $f(\boldsymbol{z})$ is its output (probability), and $\phi(\cdot)$ is the nonlinear mapping. The kernel regression model is solved by minimizing the squared error over the training samples $\boldsymbol{x}_{ij}$ and their regression outputs y_{ij} as follows

$$\min_{\boldsymbol{\alpha}} \sum_i\sum_j \left(f(\boldsymbol{x}_{ij}) - y_{ij}\right)^2 + \lambda\left\|\sum_i\sum_j \alpha_{ij}\phi(\boldsymbol{x}_{ij})\right\|^2 \quad (16)$$

where α_{ij} is an element of the parameter matrix $\boldsymbol{\alpha}$

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_{(-\bar{m})(-\bar{n})} & \alpha_{(-\bar{m})(-\bar{n}+1)} & \cdots & \alpha_{(-\bar{m})\bar{n}} \\ \alpha_{(-\bar{m}+1)(-\bar{n})} & \alpha_{(-\bar{m}+1)(-\bar{n}+1)} & \cdots & \alpha_{(-\bar{m}+1)\bar{n}} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{\bar{m}(-\bar{n})} & \alpha_{\bar{m}(-\bar{n}+1)} & \cdots & \alpha_{\bar{m}\bar{n}} \end{bmatrix},$$

y_{ij} is an element of the output matrix $\boldsymbol{y}$

$$\boldsymbol{y} = \begin{bmatrix} y_{(-\bar{m})(-\bar{n})} & y_{(-\bar{m})(-\bar{n}+1)} & \cdots & y_{(-\bar{m})\bar{n}} \\ y_{(-\bar{m}+1)(-\bar{n})} & y_{(-\bar{m}+1)(-\bar{n}+1)} & \cdots & y_{(-\bar{m}+1)\bar{n}} \\ \vdots & \vdots & \ddots & \vdots \\ y_{\bar{m}(-\bar{n})} & y_{\bar{m}(-\bar{n}+1)} & \cdots & y_{\bar{m}\bar{n}} \end{bmatrix},$$

and λ is a regularization parameter.

Since X is a block circulant matrix, the corresponding kernel matrix K is also a block circulant matrix. This contributes to the fast computation of $\boldsymbol{\alpha}$ by using the 2D Fourier transform as follows

$$\widehat{\boldsymbol{\alpha}} = \frac{1}{\widehat{\boldsymbol{k}}^{\boldsymbol{xx}} + \lambda} \odot \widehat{\boldsymbol{y}} \quad (17)$$

where $\boldsymbol{k}^{\boldsymbol{xx}}$ is the generator matrix of the block circulant matrix K and the hat ^ denotes the Fourier transform. The detailed derivation for 1D samples is in [29]. $\boldsymbol{k}^{\boldsymbol{xx}}$ is represented as

$$\boldsymbol{k}^{\boldsymbol{xx}} = \exp\,(-\frac{1}{\sigma^2}(\|\boldsymbol{x}_{00}\|^2 + \|\boldsymbol{x}_{00}\|^2 \\ -2\mathcal{F}^{-1}(\widehat{\boldsymbol{x}}_{00}^{*}\odot\widehat{\boldsymbol{x}}_{00}))) \quad (18)$$

where $\mathcal{F}^{-1}$ is the inverse Fourier transform and $\widehat{\boldsymbol{x}}_{00}^{*}$ is the complex-conjugate of $\widehat{\boldsymbol{x}}_{00}$, σ is the bandwidth of the Gaussian kernel function. By using $\boldsymbol{k}^{\boldsymbol{xx}}$ and $\boldsymbol{y}$, we can compute $\boldsymbol{\alpha}$ fast and obtain the kernel regression model.

3) *Predicting the Object at the t+1 Sampling Instant*

At the $t+1$ sampling instant, the 2D bounding box $\boldsymbol{b}_{ts}$ of each tracked object in the HID image D_t is also used on the HID image D_{t+1}. Similar to D_t, the HID image D_{t+1} is also rotated through θ_{ts} with (u_{ts}, v_{ts}) as the rotation center. Then, the image patch around the surrounding box $\boldsymbol{b}_{ts}$ is extracted from the rotated HID image $\bar{D}_{t+1}$ as the standard base sample $\boldsymbol{z}_{00}$. And, the virtual samples are obtained by cyclically shifting the base sample. As a result, we can obtain the testing samples $Z = \{\boldsymbol{z}_{ij} = P^i\boldsymbol{z}_{00}Q^j \mid -\bar{m} \le i \le \bar{m}, -\bar{n} \le j \le \bar{n}\}$.

By using the kernel regression model (15), the prediction responses of all the testing samples are computed as

$$\widehat{\boldsymbol{f}}(Z) = \widehat{\boldsymbol{k}}^{\boldsymbol{xz}} \odot \widehat{\boldsymbol{\alpha}} \quad (19)$$

where $\boldsymbol{k}^{\boldsymbol{xz}} = \phi(\boldsymbol{z}_{00})^T\phi(X)^T$ is the generator matrix of the block circulant matrix $K^{\boldsymbol{z}}$.

Similar to (18), $\boldsymbol{k}^{\boldsymbol{xz}}$ is computed as

$$\boldsymbol{k}^{\boldsymbol{xz}} = \exp\,(-\frac{1}{\sigma^2}(\|\boldsymbol{x}_{00}\|^2 + \|\boldsymbol{z}_{00}\|^2 \\ -2\mathcal{F}^{-1}(\widehat{\boldsymbol{x}}_{00}^{*}\odot\widehat{\boldsymbol{z}}_{00}))) \quad (20)$$

$\boldsymbol{f}(Z) \in R^{m\times n}$ is a matrix, containing the output for all cyclic shifts of $\boldsymbol{z}_{00}$, i.e. the full prediction responses.

From the full prediction responses $\boldsymbol{f}(Z)$, we can find the testing sample $\boldsymbol{z}_{ab}$ that has the maximal response. The a rows and b columns that $\boldsymbol{z}_{00}$ cyclically shifts to $\boldsymbol{z}_{ab}$ represent the distances that the tracked object moves between two sampling instants in the rotated HID image $\bar{D}_{t+1}$. The tracked object that predicted by RKCF at the $t+1$ sampling instant is called the predicted object. In the original HID image D_{t+1}, the position of the predicted object is computed as

$$\begin{cases} u_{(t+1)s} = u_{ts} + a\cos\theta_{ts} - b\sin\theta_{ts} \\ v_{(t+1)s} = v_{ts} + b\cos\theta_{ts} + a\sin\theta_{ts} \end{cases} \quad (21)$$

The 2D bounding boxes of the predicted objects in the HID image D_{t+1} at the $t+1$ sampling instant are denoted by $B_{t+1}^P = \{\boldsymbol{b}_{(t+1)j}^P \mid 1 \le j \le n_{t+1}^P\}$.

D. Data Association

To assign the detected objects to the predicted objects, the assignment cost matrix $M_{t+1} = \{\boldsymbol{m}_{(t+1)i,j} \mid 1 \le i \le n_{t+1}^D, 1 \le j \le n_{t+1}^P\}$ is computed as the product of the prediction score and the intersection-over-union (IOU) distance between the 2D bounding box of each detected object and the 2D bounding boxes of all the predicted objects by using the Kuhn-Munkres (KM) method [30].

$$\boldsymbol{m}_{(t+1)i,j} = S_{(t+1)j}^P \frac{\boldsymbol{b}_{(t+1)i}^D \cap \boldsymbol{b}_{(t+1)j}^P}{\boldsymbol{b}_{(t+1)i}^D \cup \boldsymbol{b}_{(t+1)j}^P} \quad (22)$$

where $S_{(t+1)j}^P$ is the prediction score of the jth predicted object, which will be introduced in Section IV-E. The initial prediction score of each predicted object is set as 1.

After the KM assignment, there are three kinds of outputs: matched predicted-detected objects, unmatched predicted objects, and unmatched detected objects. The matched predicted-detected objects are used to update the tracked objects and continuously train and predict at the next sampling instant. The unmatched predicted objects are used for object retention. The unmatched detected objects are used for object re-recognition.

E. Object retention

To ensure the integrity of the tracking sequence, we propose an object retention module to identify the correctly predicted objects from the unmatched predicted objects and retain them.

Let $\bar{B}_{t+1}^{P}=\{\bar{\boldsymbol{b}}_{(t+1)j}^{P}|1\le j\le \bar{n}_{t+1}^{P}\}$ be the 2D bounding boxes of the unmatched predicted objects in the HID image D_{t+1} at the $t+1$ sampling instant. For $\bar{\boldsymbol{b}}_{(t+1)j}^{P}$, we find the 2D bounding box $\boldsymbol{b}_{ts}$ of the corresponding tracked object at the t sampling instant. Then, we compute the similarity s_j between the color image in $\bar{\boldsymbol{b}}_{(t+1)j}^{P}$ and the color image in $\boldsymbol{b}_{ts}$ to check if the prediction is correct by using the average hash algorithm.

We use color images instead of HID images to calculate the similarity. It is because the point cloud information has already been used in the object prediction module, and the use of the color image information has a multimodal effect.

For the color image I in each 2D bounding box, the processing steps are described as follows [31].

1) The color image I are resize to 16×16, and then transformed to the grayscale image G.

2) The average value of the grayscale image G is computed by

$$\mu_g=\frac{\sum_{u=1}^{16}\sum_{v=1}^{16}G(u,v)}{256} \tag{23}$$

3) If $G(u,v)\ge \mu_g$, its hash value is set as 1; if $G(u,v)<\mu_g$, its hash value is set as 0. Then, we obtain the 256-bit binary hash value of the image I.

Then, the similarity s_j between the color image in $\bar{\boldsymbol{b}}_{(t+1)j}^{P}$ and the color image in $\boldsymbol{b}_{ts}$ is computed as the hamming distance between their 256-bit binary hash values. If $s_j\ge T_s$, these two objects are the same, indicating that the prediction is correct. The correctly predicted object is retained to update the tracked object for continuous training and predicting at the next sampling instant. If not, it is determined that the prediction is wrong. The wrongly predicted object is added to the unsuccessfully tracked objects and waits for the object re-recognition.

It is always assumed that the detection at the t sampling instant is accurate. But the detection at the t sampling instant may actually be inaccurate detection, which leads to the inaccurate prediction at the $t+1$ sampling instant and the error accumulation in the process of continuous prediction. To filter out the impact of the wrong prediction, we propose a detection score and a prediction score for each predicted object.

In addition, we set a prediction score $S_{(t+1)j}^{P}$ as the weight of the predicted object. The initial prediction score of each correctly predicted object is set as 0.5. As the prediction time increases, the prediction error accumulates, so the prediction score is gradually reduced to ensure the accuracy of association:

$$S_{(t+1)j}^{P}=S_{tj}^{P}(1-\delta) \tag{24}$$

A correctly predicted object can be retained for up to 8 frames.

The prediction score in the object retention module ensures the integrity of the tracking sequence. The object retention module checks whether the prediction is correct and retains correctly predicted objects. Therefore, the prediction can still be continued in the case of missed detection, and the reliability of prediction results is ensured.

F. Object Re-recognition

To identify the correctly detected objects from the unmatched detected objects and find out new objects, reappearing objects and disappeared objects, we develop an object re-recognition module.

As mentioned above, the unsuccessfully tracked objects whose 2D bounding boxes before tracking failure are denoted by $B_f=\{\tilde{\boldsymbol{b}}_k|1\le k\le \tilde{n}\}$, include all the wrongly predicted objects. Let $\bar{B}_{t+1}^{D}=\{\bar{\boldsymbol{b}}_{(t+1)i}^{D}|1\le i\le \bar{n}_{t+1}^{D}\}$ be the 2D bounding boxes of the unmatched detected objects at the $t+1$ sampling instant. Then, we compute the appearance score s_{ik} between the color image in $\bar{\boldsymbol{b}}_{(t+1)i}^{D}$ and $\tilde{\boldsymbol{b}}_k$ by using the average hash algorithm, following the processing steps in Section IV-E.

If $s_{ik}\ge T_h$, the two objects in $\bar{\boldsymbol{b}}_{(t+1)i}^{D}$ and $\tilde{\boldsymbol{b}}_k$ are the same. If $T_h>s_{ik}\ge T_l$, we further use the spatial distance score l_{ik} between $\bar{\boldsymbol{b}}_{(t+1)i}^{D}$ and $\tilde{\boldsymbol{b}}_k$ to verify if the two objects in $\bar{\boldsymbol{b}}_{(t+1)i}^{D}$ and $\tilde{\boldsymbol{b}}_k$ are the same. If $l_{ik}\le T_d$, these two objects are the same. The threshold T_d is set to $T_d=l_m\times n_{ik}$, where l_m is the maximal distance that the object moves between the two successive frames and n_{ik} is the difference of the frame numbers between $\bar{\boldsymbol{b}}_{(t+1)i}^{D}$ and $\tilde{\boldsymbol{b}}_k$.

If the two objects in $\bar{\boldsymbol{b}}_{(t+1)i}^{D}$ and $\tilde{\boldsymbol{b}}_k$ are the same, the ith unmatched detected object in $\bar{\boldsymbol{b}}_{(t+1)i}^{D}$ is considered as a correctly detected object or a reappearing object. It is used to update the tracked object and is removed from unsuccessfully tracked objects. If not, a new tracked object is created for $\bar{\boldsymbol{b}}_{(t+1)i}^{D}$. In addition, if an unsuccessfully tracked object exists for more than 5 frames, we think that this object disappears permanently.

The combination of the appearance score and the spatial distance score in the object re-recognition module ensures the accuracy of association. The object re-recognition avoids creating new tracks repeatedly and improves the tracking performance. It can not only compensate for the wrong prediction but also deal with a tracked object which disappears temporarily and then reappears. It is effective for recovering a tracking sequence from tracking failure.

V. Experiments and Analysis

Our method is evaluated on the KITTI tracking data set [32]. This data set contains 50 sequences with 19103 frames, which have the ground truth. For the object prediction module, we set $\lambda=1\times 10^{-4}$ and $\sigma=0.5$. For the object retention module, we set $\delta=0.05$. For the object re-recognition module, we set $T_s=120$, $T_h=200$, $T_l=160$ and $l_m=25$. The hyperparameters are tuned via cross validation.

A. Metrics

1) 2D Evaluation Metrics

For KITTI 2D MOT, the experimental results are evaluated according to the recently proposed HOTA (Higher Order Tracking Accuracy) metrics [33]. The HOTA metric naturally decomposes into a family of sub-metrics which are able to separately measure different aspects of tracking. The sub-metrics are the detection accuracy score (DetA) and association accuracy score (AssA).

2) 3D Evaluation Metrics

For KITTI 3D MOT, the experimental results are evaluated

TABLE I
ANALYSIS OF WEIGHTED HID IMAGE

Input	HOTA ↑	DetA ↑	AssA ↑
Color Image	76.30%	75.31%	77.54%
Non-weighted HID image	77.55%	75.27%	80.15%
Weighted HID Image	**78.04%**	**75.31%**	**81.13%**

TABLE II
ANALYSIS OF RKCF

Method	HOTA↑	DetA↑	AssA↑
HID + KCF + Retain (KCF)	76.98%	75.12%	79.14%
HID + RKCF + Retain (RKCF)	**77.78%**	**75.15%**	**80.76%**

according to the metrics proposed in [24], including the multi-object tracking accuracy (MOTA), multi-object tracking precision (MOTP), averaged MOTA (AMOTA), and averaged MOTP (AMOTP).

B. Ablation Study

We perform ablation study on the KITTI-Car training set based on the 2D MOT evaluation metrics.

1) Performance of the weighted HID Image

In order to show the performance of our weighted HID image, we compare the effect of object tracking based on the weighted HID image with that based on the color image and the non-weighted HID image.

The color image-based tracking method uses a kernel correlation filter (KCF) to predict the objects in two sequential color images. Then, the predicted objects are associated with the detected objects by the Kuhn-Munkres algorithm. Furthermore, an object retention module and an object re-recognition module are added to overcome the object matching failure in the in-between frames.

In the comparative experiment, only the type of input image is different. The other parameters of the three methods are the same, which ensures the fairness of the comparison of tracking performance based on the color image and the HID image.

To show the performance of the HID image, we compare the effect of object tracking based on the color image with that based on the HID image. As can be observed in Table I, all the metrics of the method with the HID image are higher than those of the method with the color image. This shows that the HID image has a good effect on MOT, which effectively avoids the influence of object occlusion and improves the tracking accuracy compared with the color camera image.

To show the performance of the weighted HID image, we compare the effect of object tracking based on the weighted HID image with that based on the non-weighted HID image. As shown Table I, the HOTA and AssA metrics of the method with the weighted HID image are higher than those of the method with the non-weighted HID image. If the weight is not added, the effective information of the point cloud intensity will be hidden and the tracking accuracy will decrease. This shows that the weight has a good effect on MOT, which improves the tracking accuracy.

2) Performance of the RKCF

To show the performance of the RKCF, we compare the effects of object tracking by using the RKCF and original KCF. As can be observed in the difference between the experimental results of “HID + KCF + Retain (KCF)” and “HID + RKCF + Retain (RKCF)” in Table II, the RKCF is 0.8% higher than the KCF in terms of HOTA.

The rotation mechanism of RKCF makes the base sample more accurate and avoids the information loss in feature extraction, which effectively reduces the accumulated error compared to the original KCF and ensures the accuracy of prediction.

Under the mechanism of the object retention module, the prediction can still be continued for several steps, even if there is no corresponding detection. In the process of continuous prediction, the accumulated error of prediction will increase, resulting in tracking drift. Due to the rotation of RKCF, the base sample is more accurate, which effectively reduces the accumulated error compared to the original KCF and ensures the accuracy of prediction. On the contrary, the base sample obtained by KCF is incomplete, which leads to a rapid increase in the error of prediction until the prediction fails.

TABLE III
ANALYSIS OF OBJECT RETENTION (RETAIN)
AND OBJECT RE-RECOGNITION (RECO)

Method	HOTA↑	DetA↑	AssA↑
HID + RKCF	74.85%	73.92%	76.06%
HID + RKCF + Retain	77.78%	75.15%	80.76%
HID + RKCF + Reco	76.19%	74.96%	77.71%
HID + RKCF + Retain + Reco	**78.04%**	**75.31%**	**81.13%**

3) Performance of the Object Retention Module and the Object Re-recognition Module

To show the performance of our object retention module and object re-recognition module, we set “HID + RKCF” as the baseline, and compare the effects of object tracking by adding the object retention module (HID + RKCF + Retain) and object re-recognition module (HID + RKCF + Reco).

As shown in Table III,“HID + RKCF + Retain” is 2.93% higher than “HID + RKCF” in terms of HOTA, proving that the object retention module is effective for continuously predicting objects and improving prediction reliability. “HID + RKCF + Reco” is 1.34% higher than “HID + RKCF” in terms of HOTA, proving that the object re-recognition module is effective for recovering a tracking sequence from tracking failure. “HID + RKCF + Retain + Reco” is 3.19% higher than “HID + RKCF” in terms of HOTA, proving the joint action of the two modules overcomes the object matching failure in the in-between frames and ensures athe integrity of the tracking sequence. In summary, the object retention module and the object re-recognition module can effectively improve tracking accuracy.

C. Performance of the detector

Since our method follows the tracking-by-detection paradigm, the detector performance also has a certain impact on the tracking results. We conducted two experiments on the KITTI-Car training set based on the 2D MOT evaluation metrics:

TABLE IV
CHANGING DETECTION METHOD WITH OUR TRACKING METHOD

Detection Method	HOTA ↑	DetA ↑	AssA ↑
SECOND	69.20%	59.31%	80.88%
Point-RCNN	75.02%	72.03%	78.32%
Point-GNN	**78.04%**	**75.31%**	**81.13%**

TABLE V
CHANGING TRACKING METHOD WITH POINT-GNN

Tracking Method	Detection Method	HOTA ↑	DetA ↑	AssA ↑
AB3DMOT	Point-GNN	75.11%	73.50%	77.03%
Our method		**78.04%**	**75.31%**	**81.13%**

TABLE VI
ANALYSIS OF GRID SIZE

Grid Size	HOTA ↑	DetA ↑	AssA ↑
0.05	77.65%	75.30%	80.32%
0.075	77.82%	**75.34%**	80.63%
0.1	**78.04%**	75.31%	**81.13%**
0.3	75.46%	75.00%	76.18%
0.5	74.13%	74.62%	73.95%
1	67.58%	73.98%	62.03%

TABLE VII
3D QUANTITATIVE COMPARISON

Method	AMOTA↑	AMOTP↑	MOTA↑	MOTP↑
AB3DMOT	44.26%	77.41%	83.35%	78.43%
FANTrack	40.03%	75.01%	74.30%	75.24%
mmMOT	33.08%	72.45%	74.07%	78.16%
Our method	**45.64%**	**79.68%**	**90.45%**	**81.44%**

1) Changing detection method with our tracking method to show the performance of Point-GNN detector

We adopt detection results of three different detectors on our tracking method respectively to show the performance of the Point-GNN detector. SECOND [34] is a one-stage voxel-based detection method and Point-RCNN [35] is a two-stage detection method based on point cloud segmentation. Point-GNN uses a graph neural network on the point cloud. As shown in Table IV, compared with other detectors, using the detection results of Point-GNN in our tracking method achieves the highest HOTA. We can find that Point-GNN achieves the best accuracy among the three methods.

2) Changing tracking method with Point-GNN to show the performance of our tracking method

There are no other MOT methods using the same detection method (Point-GNN). To make a fair comparison, we adopt the Point-GNN detection results on a generally recognized MOT baseline method AB3DMOT. Compared with the AB3DMOT method that uses Point-GNN, our method achieves higher HOTA, as shown in Table V. The experimental results show that our tracking method has better performance.

D. Analysis of the grid size

The performance of the HID image depends largely on the grid size. The larger the grid size, the smaller the size of the HID image and the fewer the details. The lack of HID image details leads to poor tracking accuracy. The smaller the grid size, the larger the size of the HID image, the more the details, and the slower the image processing. Due to the sparsity of point cloud, a too small grid size will not increase more details, while a too large size of the HID image will greatly increase the cost of image processing. Therefore, we need to find a proper resolution.

We add an experiment to determine which grid size is the most appropriate for tracking by changing the grid size on the KITTI-Car training set based on the 2D MOT evaluation metrics. As shown in Table VI, when the grid size is set as 0.1m, the HOTA and AssA reach the highest. At the grid size of 0.1m, our method achieves 20.8 FPS on the KITTI-Car test data set, which is suitable for autonomous driving. Therefore, the grid size is set to 0.1m for the construction of HID image in our method.

This grid size is not too large to lost detail or too small to increase time consumption, which balances tracking accuracy and time efficiency. At this grid size, the size of the HID image is appropriate, and the objects in the image are clear and easy to distinguish. Therefore, we can achieve the best performance. If the grid size is set smaller, not only the tracking accuracy will not improve, but also the real-time performance will not be satisfied.

E. Comparison Experiments

1) KITTI 3D MOT

In this experiment, we compare our method with some 3D MOT methods on the KITTI-Car validation set based on the 3D MOT evaluation metrics. Following the same experimental settings as AB3DMOT [24], we use sequences 1, 6, 8, 10, 12, 13, 14, 15, 16, 18, 19 as the validation set. As shown in Table VII, our method reaches the highest in all the 3D MOT metrics.

2) KITTI 2D MOT

In this experiment, we compare our method with other advanced methods on the KITTI-Car test set based on the 2D MOT evaluation metrics. Experimental results with different methods are shown in Table VIII. We divide these methods into supervised deep learning methods, unsupervised deep learning methods and traditional methods, which are currently published in the KITTI benchmark data set.

Our method achieves 20.8 FPS on the KITTI-Car test data set. The running time of our method is 0.048s, which is suitable for autonomous driving.

Among all published MOT methods, the HOTA of our method is only lower than two supervised deep learning methods, PC-TCNN and Permatrack. However, due to the complexity of the network, these two methods have longer running time and lower computational efficiency than our method. In addition, the training process of supervised deep learning methods relies on a large number of carefully labeled objects, and the labeling process is time-consuming and expensive. As a traditional method, our method does not require labeling.

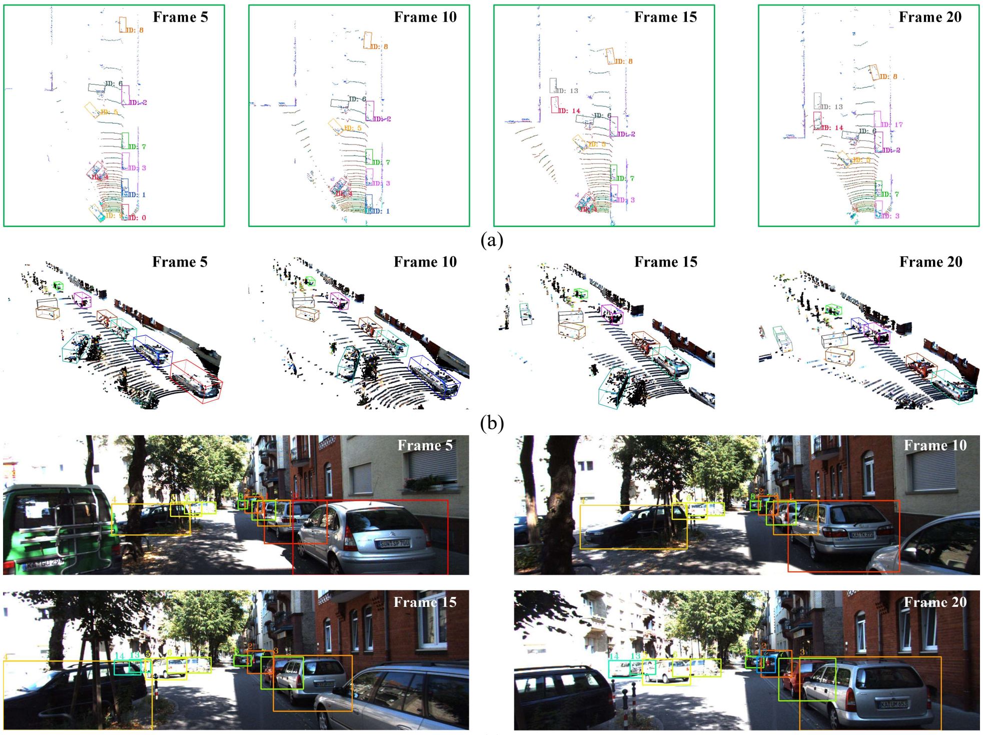
The other supervised deep learning methods have the same

TABLE VIII
2D QUANTITATIVE COMPARISON

Method	Category	HOTA↑	DetA↑	AssA↑	Time↓
PC-TCNN [15]	Supervised Deep Learning Method	**80.90%**	**78.46%**	**84.13%**	0.3s
Permatrack [14]		78.03%	78.29%	78.41%	0.1s
Mono3DT [11]		73.16%	72.73%	74.18%	0.03s
LGM [17]		73.14%	74.61%	72.31%	0.08s
CenterTrack [13]		73.02%	75.62%	71.20%	0.045s
QD-3DT [12]		72.77%	74.09%	72.19%	0.03s
TrackMPNN [18]		72.30%	74.69%	70.63%	0.05s
DiTMOT [16]		72.21%	71.09%	74.04%	0.08s
MOTSFusion [9]		68.74%	72.19%	66.16%	0.44s
SMAT [19]	Unsupervised Deep Learning Method	71.88%	72.13%	72.13%	0.1s
Visual-Spatial [20]		62.50%	61.10%	65.30%	-
Mono-3D-KF [25]	Traditional Method	75.47%	74.10%	77.63%	0.3s
EagerMOT [26]		74.39%	75.27%	74.16%	0.01s
AB3DMOT [24]		69.99%	71.13%	69.33%	0.0047s
MASS [27]		68.25%	72.92%	64.46%	0.01s
Our method		**75.90%**	75.20%	77.22%	0.048s

real-time performance as our method, but the accuracy is lower. CenterTrack and TrackMPNN can only associate detection frames between two consecutive frames, so it is not possible to re-match objects that have disappeared for a long time. In our method, on the other hand, the object retention module and the object re-recognition module are developed based on the color image, which overcomes the object matching failure in the in-between frames. QD-3DT and Mono3DT only use single-modal information, while our method takes full advantage of point cloud information and image information. We not only construct a novel weighted height-intensity-density (HID) image by the point cloud but also develop an object re-recognition module and an object retention module by using the color image, which makes the multi-modal information complement each other effectively.

Compared with unsupervised deep learning methods, our method achieves higher HOTA. This is because the labels of some unsupervised deep learning methods are generated from



(a)

(b)

(c)

Fig. 11. Qualitative results on the KITTI-Car test set (a) in the HID images, (b) in the 3D point clouds, and (c) in the color images.

the predictions of traditional tracking methods. If the predictions are incorrect and generate noise, the trained model will also be inaccurate, resulting in low accuracy of the tracking result. Moreover, our method ranks first among all the traditional methods in terms of HOTA.

In conclusion, The reason that our MOT method can achieve good performance is mainly due to the following aspects: 1) The HID image is constructed from the 3D point cloud, which effectively avoids the influences of object occlusion; 2) the object retention and the object re-recognition are developed based on the color image, which overcomes the object matching failure in the in-between frames; 3) the 3D point cloud and the color image are fully utilized, which makes the multi-modal information complement each other effectively.

In order to show the details of our MOT method, the qualitative results of the sequence 0000 from the KITTI tracking test set are shown in Fig. 11. The bounding box of each object is represented by a different color box and marked with its own ID. Our method is able to create an accurate trajectory in the 3D space and performs well even in a cluttered scene.

VI. Conclusion

In this paper, we propose a multi-object tracking framework based on the HID image with the multi-modal information. The main contributions of our method include the construction of the HID image, the RKCF based on the HID image, the object retention and the object re-recognition based on the color image. These technical features make our method accurate. Experimental results show that our method achieves competitive performance among the traditional multi-object tracking methods.

References

[1] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” *Acm Comput Surv*, vol. 38, no. 4, pp. 13, Dec. 2006.

[2] Y. Shan, B. Zheng, L. Chen, L. Chen and D. Chen, “A reinforcement learning-based adaptive path tracking approach for autonomous driving,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 10581-10595, Oct. 2020.

[3] G. Zhong, S. Niar, A. Prakash and T. Mitra, “Design of multiple-target tracking system on heterogeneous system-on-chip devices,” *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 4802-4812, Jun. 2016.

[4] J. Ji, A. Khajepour, W. W. Melek and Y. Huang, “Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 2, pp. 952-964, Feb. 2017.

[5] H. Yang, L. Shao, F. Zheng, L. Wang and Z. Song, “Recent advances and trends in visual tracking: A review,” *Neurocomputing*, vol. 74, no. 18, pp. 3823-3831, Aug. 2011.

[6] W. Hu, X. Xiao, D. Xie, T. Tan, and S. Maybank, “Traffic accident prediction using 3-D model-based vehicle tracking,” *IEEE Trans. Veh. Technol.*, vol. 53, no. 3, pp. 677-694, May. 2004.

[7] W. Shi and R. Rajkumar, “Point-GNN: Graph neural network for 3D object detection in a point cloud,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 14-19.

[8] W. Zhang, H. Zhou, S. Sun, Z. Wang, J. Shi and C. C. Loy, “Robust multi-modality multi-object tracking,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 2365-2374.

[9] J. Luiten, T. Fischer and B. Leibe, “Track to reconstruct and reconstruct to track,” *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1803-1810, April. 2020.

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

[11] H. N. Hu, Q. Z. Cai, D. Wang, L. Ji and F. Yu, “Joint monocular 3D vehicle detection and tracking,” in *Proc. IEEE Int. Conf. Comput. Vision.*, 2019.

[12] H. N. Hu, Y. H. Yang, T. Fischer, T. Darrell, F. Yu and M. Sun, “Monocular Quasi-Dense 3D Object Tracking,” 2021, [online] Available: https://arxiv.org/abs/2103.07351.

[13] X. Zhou, V. Koltun and P. Krähenbühl, “Tracking objects as points,” in *Proc. Euro. Conf. Comput. Vision.*, 2020, pp. 474-490.

[14] P. Tokmakov, J. Li, W. Burgard and A. Gaidon, “Learning to Track with Object Permanence,” in *Proc. IEEE Int. Conf. Comput. Vision.*, 2021, pp. 10840-10849.

[15] H. Wu, Q. Li, C. Wen, X. Li, X. Fan, and C. Wang, “Tracklet proposal network for multi-object tracking on point clouds,” in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 1165-1171.

[16] S. Wang, P. Cai, L. Wang and M. Liu, “DiTNet: End-to-end 3D object detection and track ID assignment in spatio-temporal world,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3397-3404, Apr. 2021.

[17] G. Wang, R. Gu, Z. Liu, W. Hu, M. Song and J. N. Hwang, "Track without Appearance: Learn Box and Tracklet Embedding with Local and Global Motion Patterns for Vehicle Tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 9856-9866.

[18] A. Rangesh, P. Maheshwari, M. Gebre, S. Mhatre, V. Ramezani and M. M. Trivedi, “TrackMPNN: A Message Passing Graph Neural Architecture for Multi-Object Tracking,” 2021, [online] Available: https://arxiv.org/abs/2101.04206.

[19] N. F. Gonzalez, A. Ospina and P. Calvez. “Smat: Smart multiple affinity metrics for multiple object tracking,” in *Proc. Int. Conf. Image. Anal. Recognit*, 2020, pp. 48-62.

[20] F. Bastani, S. He and S. Madden, " Self-Supervised Multi-Object Tracking with Cross-Input Consistency," 2021, [online] Available: https://arxiv.org/abs/2111.05943.

[21] S. Karthik, A. Prabhu and V. Gandhi, “Simple Unsupervised Multi-Object Tracking,” 2020, [online] Available: https://arxiv.org/abs/2006.02609.

[22] R. Hostettler and P. M. Djurić, “Vehicle tracking based on fusion of magnetometer and accelerometer sensor measurements with particle filtering,” *IEEE Trans. Veh. Technol.*, vol. 64, no. 11, pp. 4917-4928, Nov. 2015.

[23] J. An, B. Choi, H. Kim and E. Kim, “A new contour-based approach to moving object detection and tracking using a low-end three-dimensional laser scanner,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7392-7405, Aug. 2019.

[24] X. Weng, J. Wang, D. Held, K. Kitani. “3D multi-object tracking: A baseline and new evaluation metrics,” in *Proc. IEEE Int. Conf. Intell. Rob. Syst.*, 2020, pp. 10359-10366.

[25] A. Reich and H. J. Wuensche, “Monocular 3D Multi-Object Tracking with an EKF Approach for Long-Term Stable Tracks,” in *Proc. IEEE Int. Conf. Information Fusion.*, 2021, pp. 1-7.

[26] A. Kim, A. Ošep and L. Leal-Taixé, “EagerMOT: 3D Multi-Object Tracking via Sensor Fusion,” in *Proc. IEEE Int. Conf. Robotics Automation.*, 2021, pp. 11315-11321.

[27] H. Karunasekera, H. Wang and H. Zhang, "Multiple Object Tracking With Attention to Appearance, Structure, Motion and Size," *IEEE Access*, vol. 7, pp. 104423-104434, July, 2019.

[28] Y. An, B. Li, H. Hu and X. Zhou, “Building an omnidirectional 3-D color laser ranging system through a novel calibration method,” *IEEE Trans. Ind. Electron.*, vol. 66, no. 11, pp. 8821-8831, Nov. 2019.

[29] J. F. Henriques, R. Caseiro, P. Martins and J. Batista, “High-speed tracking with kernelized correlation filters,” *IEEE Trans. Pattern Anal.*, vol. 37, no. 3, pp. 583-596, Mar. 2015.

[30] J. Munkres, “Algorithms for assignment and transportation problems,” *Soc. Ind. Appl. Math.*, vol.5, no.1, Mar. 1957.

[31] S. Farisa and D. Kurniadi. “Average hashing for perceptual image similarity in mobile phone application,” *Telemat. Inform.*, vol.4, no.1, pp. 12-18, Mar. 2016.

[32] A. Geiger, P. Lenz, C. Stiller and R. Urtasun, “Vision meets robotics: The KITTI dataset,” *Int. J. Rob. Res.*, vol. 32, no. 11, pp. 1231–1237, Sept. 2013.

[33] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Lealtaixé and B. Leibe, “HOTA: A higher order metric for evaluating multi-object tracking,” *Int. J. Comput. Vis.*, vol. 129, pp. 548–578, Oct. 2020.

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

[35] 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.*, 2019, pp. 770-779.

Yi An (Member, IEEE) received the B.S. degree in automation and the M.S. and Ph.D. degrees in control theory and control engineering from the Dalian University of Technology, Dalian, China, in 2001, 2004, and 2011, respectively.

From 2007 to 2011, he was a Lecturer with the School of Control Science and Engineering, Dalian University of Technology, where he has been an Associate Professor, since 2012. He has also been an Associate Professor with the School of Electrical Engineering, Xinjiang University, Urumqi, China, since 2021. His research interests include point cloud data processing, sensing and perception, information fusion, robot vision, and intelligent robot.

Jialin Wu received the B.S. degree in automation from the Dalian University of Technology, Dalian, China, in 2020, where she is currently pursuing the M.S. degree with the School of Control Science and Engineering.

Her research interests include point cloud data processing and object tracking.

Yunhao Cui received the M.S. degree in mechanical design and theory from Northeastern University, Shenyang, China, in 2014. He is currently pursuing the Ph.D. degree with the School of Mechanical Engineering, Dalian University of Technology, Dalian, China.

His research interests include point cloud data processing, intelligent mechanical equipment, and 3-D environment perception.

Huosheng Hu (Life Senior Member, IEEE) received the M.Sc. degree in industrial automation from Central South University, Changsha, China, in 1982, and the Ph.D. degree in robotics from the University of Oxford, Oxford, U.K., in 1993.

He is currently a Professor with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, U.K., where he is leading the Robotics Research Group. He has authored or coauthored more than 420 articles. His current research interests include robotics, human–robot interaction, embedded systems, mechatronics, and pervasive computing.

Prof. Hu is a founding member of the IEEE Robotics and Automation Society Technical Committee on Networked Robots, a fellow of the Institution of Engineering and Technology, and a senior member of the Association for Computing Machinery. He currently serves as an Editor-in-Chief of the *International Journal of Automation and Computing* and the online *Robotics* journal and an Executive Editor of the *International Journal of Mechatronics and Automation*.