Lipeng Gu[1]
Department of Automation,
School of Information Science and Technology,
Donghua University,
2999 Renmin North Road,
Songjiang District,
Shanghai 201620, China
e-mail: glp1224@163.com

Shaoyuan Sun
Department of Automation,
School of Information Science and Technology,
Donghua University,
2999 Renmin North Road,
Songjiang District,
Shanghai 201620, China
e-mail: shysun@dhu.edu.cn

Xunhua Liu
Department of Automation,
School of Information Science and Technology,
Donghua University,
2999 Renmin North Road,
Songjiang District,
Shanghai 201620, China
e-mail: xunhua_liu@163.com

Xiang Li
Department of Automation,
School of Information Science and Technology,
Donghua University,
2999 Renmin North Road,
Songjiang District,
Shanghai 201620, China
e-mail: xiangxlily@163.com

CenterTrack3D: Improved CenterTrack More Suitable for Three-Dimensional Objects

Compared with two-dimensional (2D) multi-object tracking (MOT) algorithms, three-dimensional (3D) multi-object tracking algorithms have more research significance and broad application prospects in the unmanned vehicles research field. Aiming at the problem of 3D multi-object detection and tracking, in this paper, the multi-object tracker CenterTrack, which focuses on 2D multi-object tracking task while ignoring object 3D information, is improved mainly from two aspects of detection and tracking, and the improved network is called CenterTrack3D. In terms of detection, CenterTrack3D uses the idea of attention mechanism to optimize the way that the previous-frame image and the heatmap of previous-frame tracklets are added to the current-frame image as input, and second convolutional layer of the hm output head is replaced by dynamic convolution layer, which further improves the ability to detect occluded objects. In terms of tracking, a cascaded data association algorithm based on 3D Kalman filter is proposed to make full use of the 3D information of objects in the image and increase the robustness of the 3D multi-object tracker. The experimental results show that, compared with the original CenterTrack and the existing 3D multi-object tracking methods, CenterTrack3D achieves 88.75% MOTA for cars and 59.40% MOTA for pedestrians and is very competitive on the KITTI tracking benchmark test set. [DOI: 10.1115/1.4050863]

Keywords: object detection, multi-object tracking, 3D Kalman filter, data association algorithm, artificial intelligence, vehicle autonomy

1 Introduction

In recent years, multi-object tracking has become a research hotspot in the field of computer vision. Multi-object tracking (MOT) aims to solve the problem of locating and tracking multiple objects in a given video sequence, and the number and categories of these objects are unknown [1]. With the rise of powerful deep-learning-based object detectors, multi-object tracking algorithms based on the tracking-by-detection mode have gradually emerged. They are mostly composed of three parts: object detector, object appearance feature modeling, and data association. These methods make full use of the ability of object detector based on deep network and associate the detected objects of interest through time by using the data association algorithm with the feature of objects as the measurement. The same objects in each frame finally form their own tracklets [2]. But these multi-object trackers based on best-performing detectors also have some inherent disadvantages. These trackers require a complex association strategy or a feature extraction and fusion algorithm which costs a lot of time and space.

Recently, there has been some work on combining object detection and tracking tasks into a unified network. The specific implementation method is to extend the existing object detector into a multi-object tracker, which has made great achievements in reducing the complexity of the multi-object trackers mentioned above. For example, Zhou et al. proposed a point-based multi-object tracker CenterTrack [3]. According to the center offset between adjacent frames predicted by CenterTrack, greedy matching algorithm is used to match objects detected in each frame with the closed unmatched object in the prior frame in descending order of confidence score. CenterTrack mainly builds on CenterNet by adding additional four input channels and two output channels, which are used to input the previous-frame image and the heatmap of previous-frame tracklets and predict the inter-frame center offset vector. Compared with CenterNet, CenterTrack only adds little time and space computational cost. In addition, because CenterNet can be easily expanded into a three-dimensional (3D) object detector [4], CenterTrack also has a certain 3D multi-object tracking ability after simple expansion.

In this paper, we focus on 3D multi-object detection and tracking, but CenterTrack's tracking strategy for 3D objects is completely consistent with two-dimensional (2D) objects, and no relevant optimizations have been made for 3D objects. To deal with this problem, we optimize detection and data association part of CenterTrack, referred to as CenterTrack3D. Our experiment was conducted on KITTI tracking benchmark dataset. CenterTrack3D performs better than the original CenterTrack and other published work in 3D multi-object tracking task and achieves 88.75% MOTA for cars and 59.40% MOTA for pedestrians on the KITTI tracking benchmark test set.

To summarize, our contributions are as follows:

(1) For 3D object detection, CenterTrack3D first uses the idea of attention mechanism to optimize the way in which the previous-frame image and the heatmap of previous-frame tracklets are added to the current-frame image as input. Then, the second convolution layer of the *hm* output head is replaced

[1]Corresponding author.

Manuscript received November 16, 2020; final manuscript received April 7, 2021; published online May 4, 2021. Assoc. Editor: Gaurav Pandey.

by dynamic convolution layer, which further improves detections of occluded objects, especially for pedestrians.

(2) For 3D multi-object tracking, the difference is that CenterTrack3D uses a cascaded data association algorithm based on the 3D Kalman filter to make full use of the object's 3D information to match same objects more accurately between adjacent frames. CenterTrack3D also mainly uses the greedy matching algorithm to match the closest objects between adjacent frames.

2 Related Work

2.1 Three-Dimensional Object Detection. Due to the development of deep-learning networks, excellent object detectors have developed rapidly in recent years [5–8]. 3D object detection also has gradually received widespread attention and has made good progress. CenterNet is not only an outstanding 2D object detector, but after simple expansion, it can predict the depth and the orientation to estimate the complete 3D bounding box by feeding only 2D images as input. There also are excellent 3D detectors which use images and point cloud [9,10]. MV3D combines the image and the top view and front view of point cloud to output the final 3D object detection results [10].

In this paper, we use CenterTrack as a 3D multi-object tracker, which builds on CenterNet and takes the prior tracking results as additional input. Our method optimizes the input mode of the previous-frame tracking results and replaces the ordinary convolution with dynamic convolution on the *hm* output head, which further recover occluded objects

2.2 Three-Dimensional Multi-Object Tracking Framework. In essence, 3D multi-object tracking (3DMOT) algorithm is almost the same as 2D multi-object (2DMOT) algorithm, but 3DMOT tracks objects of interest in the form of 3D bounding box. Nowadays, most of the mainstream 3DMOT stall follows the tracking-by-detection framework. AB3DMOT achieves good multi-object tracking results by tracking the accurate detections as input and using the improved SORT based on 3D Kalman filter as the matching algorithm [11,12]. JRMOT fuses 2D RGB image and 3D point cloud as input, estimates the 3D bounding box of objects of interest in each image, and then uses the joint probability association network to match objects of adjacent frames [13]. 3DT detects objects of interest in form of the 2D bounding box, estimates their complete 3D bounding boxes, and then utilizes 3D information and other cues to match objects [14].

The data association strategies of these methods are often very complicated, or only consider the tracking of cars. In this paper, our approach uses 3D Kalman filter and 2D Kalman filter to estimate the state of prior objects' 3D bounding box and 2D bounding box in the current frame, and uses different metrics to track cars and pedestrians respectively. Specifically, we use Intersection-over-Union of 3D bounding box (3DIoU) as the first metric and the distance between center points of 2D bounding boxes as the second metric for cars and use the distance between center points of 2D bounding boxes as the only metric for pedestrians. In addition, we utilize 3D Kalman filter to track the missing prior pedestrians continuously.

3 Preliminaries

3.1 Framework. The overall framework of the multi-object tracker CenterTrack3D proposed in this paper is shown in Fig. 1. CenterTrack3D is improved by CenterTrack, which is developed based on CenterNet. Thus, CenterTrack3D and CenterTrack are all essentially an object detector with DLA-34 as the backbone network. There are eight parallel output heads, namely, *hm*, *wh*, *reg*, *dep*, *dim*, *rot*, *amodel_offset*, and *tracking*, where *tracking* head is used to predict the center offset between adjacent frames in original CenterTrack. The remaining seven heads locate objects of interest and predict 3D bounding box information in each frame. The information is used in the process of object feature modeling and data association to achieve continuous tracking of objects of interest in each frame in the form of 3D bounding box.

3.2 Detection. Assuming that at time t, the current-frame image is $I^{(t)}$, the previous-frame image is $I^{(t-1)}$, the heatmap of the previous-frame tracked objects is $H^{(t-1)}$, current-frame detections are $D^{(t)} = \{(s_i, c_i, p_i, box_i, \alpha_i, dim_i, loc_i, \theta_i)\}_{i=0}^{M-1}$, and previous-frame tracked objects are $T^{(t-1)} = \{(s_i, c_i, p_i, box_i, \alpha_i, dim_i, loc_i, \theta_i, id_i)\}_{i=0}^{N-1}$, where $s \in [0, 1]$ represents the detection confidence score, $c \in \{0, \ldots, C-1\}$ represents the category to which the object belongs, $p \in \mathbb{R}^2$ represents the object center point location, $box \in \mathbb{R}^4$ represents 2D bounding box, and $dim \in \mathbb{R}^3$ represents 3D bounding box, $\alpha \in [-\pi, \pi]$ represents the observation angle, $loc \in \mathbb{R}^3$ represents the coordinate position of the 3D object in the camera coordinate system, $\theta \in [-\pi, \pi]$ represents the rotation angle of the object around the y-axis in the camera coordinates, and $id \in \mathbb{Z}$ represents the unique identity of the object.

In order to enhance the temporal coherence of the detected objects in the continuous video sequence and improve the detection ability of the occluded objects in the crowded scene, the original CenterTrack takes the current-frame image $I^{(t)}$, the previous-frame image $I^{(t-1)}$ and the heatmap of the previous-frame tracked detections $H^{(t-1)}$ together as input, and loads them into the network. Where the heatmap of the previous-frame tracked detections $H^{(t-1)}$ is obtained by distributing the previous-frame tracklets $T^{(t-1)}$ on the single-channel heatmap through the Gaussian kernel function. In this way, the network is made to perceive changes in the scene, and the occluded object in the current frame can be detected based on the relevant information of the previous frame.

The original CenterTrack first passes the current frame $I^{(t)}$, the previous frame $I^{(t-1)}$ and $H^{(t-1)}$ through a convolutional layer, respectively, to obtain three $383 \times 1280 \times 16$ feature maps F^{cur_img}, F^{pre_img}, and F^{pre_hm}, and then F^{cur_img}, F^{pre_img}, and F^{pre_hm} uses a simple addition method for fusion (as shown by the dotted line in Fig. 2), as shown in Eqs. (1)–(4)

$$F^{cur\_img} = W^{cur\_img} \otimes I^{(t)} \tag{1}$$

$$F^{pre\_img} = W^{pre\_img} \otimes I^{(t-1)} \tag{2}$$

$$F^{pre\_hm} = W^{pre\_hm} \otimes H^{(t-1)} \tag{3}$$

$$F^{fuse} = F^{pre\_hm} + F^{pre\_img} + F^{cur\_img} \tag{4}$$

where W^{pre_hm}, W^{pre_img}, and W^{cur_img}, respectively, represent the corresponding convolution kernel of $I^{(t)}$, $I^{(t-1)}$, and $H^{(t-1)}$.

3.3 Loss Function. The *hm* output head of CenterTrack3D generates a low-resolution heatmap $\hat{Y} \in [0, 1]^{\frac{W}{R} \times \frac{H}{R} \times C}$, and the largest peak point in each 3×3 neighborhood of the heatmap is considered as an object center point $p \in \mathbb{R}^2$ to locate objects of interest. According to the center location, CenterTrack3D infers the object's required confidence, category, 2D bounding box, 3D bounding box, and other information on other output heads. We train on the eight output heads with the same loss function as CenterTrack [3]. The following mainly introduces the loss function of *hm* and *tracking* head, the loss functions of other heads will not be repeated in this paper.

The *hm* head is trained by using the focal loss function [15], to regress the object center location, as shown in Eq. (5):

$$L_k = \frac{1}{N} \sum_{xyc} \begin{cases} (1 - \hat{Y}_{xyc})^{\alpha} \log(\hat{Y}_{xyc}) & \text{if } Y_{xyc} = 1 \\ (1 - Y_{xyc})^{\beta} (\hat{Y}_{xyc})^{\alpha} \log(1 - \hat{Y}_{xyc}) & \text{otherwise} \end{cases} \tag{5}$$

where $Y \in [0, 1]^{\frac{W}{R} \times \frac{H}{R} \times C}$ represents the heatmap with ground-truth object center points, $\hat{Y} \in [0, 1]^{\frac{W}{R} \times \frac{H}{R} \times C}$ represents the predicted

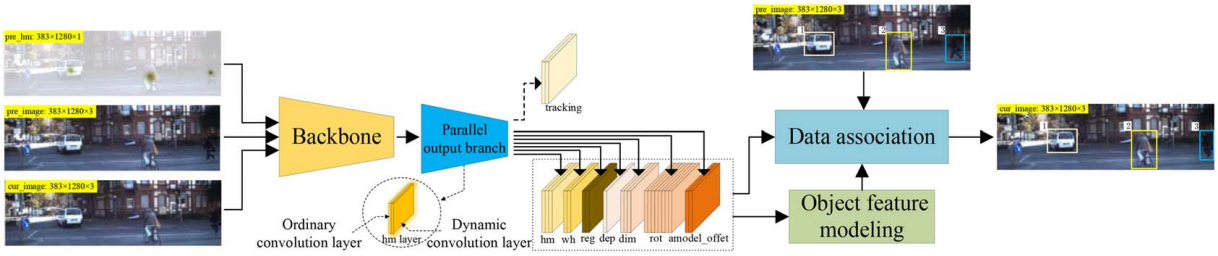

Fig. 1 Block diagram of CenterTrack3D

heatmap, N represents the number of objects in an image, and $\alpha = 2$ and $\beta = 4$ are the hyperparameters of focal loss. For each center point p of category c, we use a Gaussian kernel function $R_q(\{p_0, p_1, \cdots\})$ to render a Gaussian distribution peak point into $Y_{:,:,c}$, as shown in Eq. (6):

$$R_q(\{p_0, p_1, \cdots\}) = \max_i \exp\left(-\frac{(p_i - q)^2}{2\sigma_i^2}\right) \tag{6}$$

where p_i represents the center location of the i-th object in the image, $q \in \mathbb{R}^2$ represents the location on the heatmap Y, and the Gaussian kernel σ_i is related to the object size.

The *tracking* head is trained by using the L1 loss function, to regress the center offset between adjacent frames, as shown in Eq. (7)

$$L_{tracking} = \frac{1}{N}\sum_{i=1}^{N}\left|o_{p_i^{(t)}} - \left(p_i^{(t-1)} - p_i^{(t)}\right)\right| \tag{7}$$

where $p_i^{(t-1)}$ and $p_i^{(t)}$ are tracked ground-truth object points, $o_{p_i^{(t)}}$ is the offset of object $p_i^{(t)}$ between two adjacent frames.

3.4 Tracking. Assuming that the set of object center points in the current frame is $P^{(t)} = \{p_i\}_{i=0}^{M-1}$, and that in the previous frame is $P^{(t-1)} = \{p_i\}_{i=0}^{N-1}$. The original CenterTrack predicts a 2D offset vector set $O^{(t)} = \{o_i\}_{i=0}^{M-1} \in \mathbb{R}^{\frac{W}{R}\times\frac{H}{R}\times 2}$ in *tracking* head based on $P^{(t)} = \{p_i\}_{i=0}^{M-1}$. Through $P^{(t)} = \{p_i\}_{i=0}^{M-1}$ and $O^{(t)} = \{o_i\}_{i=0}^{M-1} \in \mathbb{R}^{\frac{W}{R}\times\frac{H}{R}\times 2}$, the current-frame center point location is estimated in the previous frame $I^{(t-1)}$, and the calculation is shown in Eq. (8)

$$\tilde{p}_i^{(t-1)} = o_i^{(t)} + p_i^{(t)} \tag{8}$$

The specific matching rules are as follows: (1) The set of current-frame detections $D^{(t)}$ is sorted in descending order according to the confidence score; (2) The distance $dist^{(t)} \in \mathbb{R}^{M\times N}$ between $\tilde{P}^{(t-1)} = \{\tilde{p}_i\}_{i=0}^{M-1}$ and $P^{(t-1)} = \{p_i\}_{i=0}^{N-1}$ is used as the measurement, and greedy algorithm is used to match $D^{(t)}$ with the closest $T^{(t-1)}$, where $T^{(t-1)}$ is tracklets in the previous frame. (3) If the object in the current frame has no unmatched previous-frame detection within the radius κ of its center point, the object is considered as a new tracklet, and a new ID is assigned to it. Where κ is related to the width and height of 2D bounding box, and the calculation equation is shown in Eq. (9)

$$\kappa = \sqrt{w^2 + h^2} \tag{9}$$

4 Work for Improvements

4.1 Network Improvements. As for occlusion, exposure and other environmental effects, even the appearance characteristics of the same object are different between two adjacent frames. Thus, the original network adds the tracking results of the previous frame to the current frame for training and inference, so as to detect objects of interest in the current frame more accurately. Considering that this direct addition method of information fusion cannot fully utilize the previous-frame image and tracklets, this paper introduces an attention mechanism to optimize the way in which the previous-frame image and the prior heatmap are added to the current-frame image as the network input [16], as shown by the solid line in Fig. 2. First, we calculate the corresponding attention maps of F^{pre_img}, F^{pre_hm}, and $|F^{pre\_img} - F^{cur\_img}|$, respectively, which are G^{pre_img}, G^{cur_img}, and G^{sub_img}, as shown in Eq. (10)

$$G^s = \sigma(W_{att}^s \otimes F^s)|_{s\in\{pre\_img, pre\_hm, sub\_img\}} \tag{10}$$

where $\otimes$ is a convolution operation and σ is a sigmoid activation function. W_{att}^s predicts its importance by learning the feature itself, and the sigmoid activation function maps the attention to the range of 0 to 1. Finally, the fusion is done using Eq. (11)

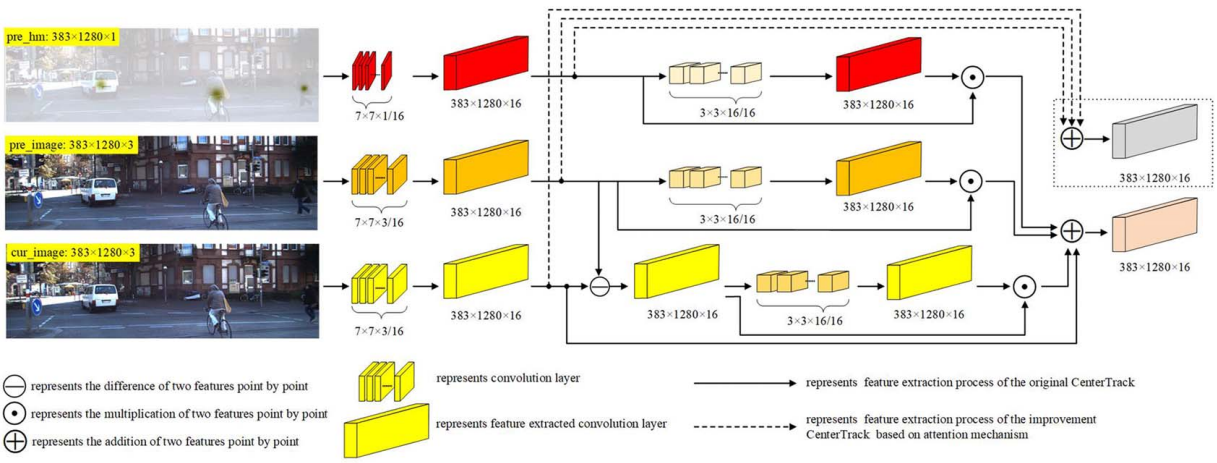

Fig. 2 Block diagram of network input mode

$$F^{fuse} = \frac{\sum_s^S G^s \odot F^s}{\sum_s^S G^s} + F^{cur\_img}, S \in \{pre\_img, pre\_hm, sub\_img\} \quad (11)$$

where $\odot$ represents elementwise multiplication.

In addition, since objects of interest in the video sequence are all shape-changing, and the receptive field of the ordinary convolutional layer is always rectangular, the ordinary convolutional layer cannot adapt to objects whose shapes often change. However, the dynamic convolution layer can add an offset to the position of each sampling point of the ordinary convolution kernel during the convolution operation, so as to obtain the receptive field that adapts to objects of different shapes. Therefore, in *hm* head of the network which predicts center points of objects of interest in images, the second convolutional layer is replaced with a dynamic convolutional layer, which can generate a heatmap with richer semantic information to better predict and locate objects of interest in the image, especially for pedestrians. The details are shown in the dotted ellipse in Fig. 1.

4.2 Association Algorithm Based on Three-Dimensional Kalman Filter. Since the association algorithm of original CenterTrack mostly focuses on 2D object and ignores 3D information and does not optimize for the 3D multi-object tracking task, this paper proposes a cascade association algorithm based on 3D Kalman filter to solve the problem that the original association algorithm does not make full use of the 3D information of detections. The algorithm flow is shown in Fig. 3.

As shown in Fig. 3, the cascade tracking algorithm based on 3D Kalman filter has three matching steps. First, we match the reliable detections in the current frame with tracklets in the previous frame, where the reliable detections represent high confidence detections. Second, we match the unmatched prior tracklets with objects with a lower confidence score in the current frame to alleviate the problem of missing detections in the current frame. If a low confidence detection in current frame is matched with an unmatched prior tracklet, it will be incorporated into existing trajectories, otherwise it will be considered as a false detection and abandoned. Third, we use 3D Kalman filter to continuously track tracklets that have not been matched in the previous frame, and recover some of objects that were missed in the current frame. Note that the *tracking* head is only used for training and not used in inference time, and 2D Kalman filter is used to predict the center offset between adjacent frames in the paper. This is because the experimental results show that *tracking* head has certain improvement on detections, and 2D Kalman filter can predict more accurate center offset than *tracking* head. The details are as follows:

(1) The current-frame detections $D^{(t)}$ is divided into two groups according to the confidence threshold $\vartheta = 0.4$, and the 3D Kalman filter and 2D Kalman filter are used to predict the state of previous-frame objects $T^{(t-1)} = \{(s_i^{t-1}, c_i^{t-1}, p_i^{t-1}, box_i^{t-1}, \alpha_i^{t-1}.dim_i^{t-1}, loc_i^{t-1}, \theta_i^{t-1})\}_{i=0}^{N-1}$ in the current frame and the predicted tracklets are $Pred_T^{(t-1)} = \{(s_i^{t-1}, c_i^{t-1}, p_i^{t-1}, box_i^{t-1}, \alpha_i^{t-1}.dim_i^{t-1}, loc_i^{t-1}, \theta_i^{t-1})\}_{i=0}^{N-1}$. We formulate the state of an object 3D bounding box as a 10-dimensional vector $X_1 = (x, y, z, \theta, l, w, h, \dot{x}, \dot{y}, \dot{z})$ in the vehicle camera coordinate system and the state of an object 2D bounding box as a seven-dimensional vector $X_2 = (u, v, a, r, \dot{u}, \dot{v}, \dot{a})$ in the image pixel coordinate system, where the additional variables $\dot{x}$, $\dot{y}$, $\dot{z}$ in vector X_1 represent the object velocity in the 3D space and the vector X_2 contains the bounding box center position (u, v), area of 2D bounding box a, aspect ratio r, and their respective velocities in image coordinates. The predicted states of 2D bounding box and 3D bounding box of previous-frame tracklets in the current frame are $\tilde{X}_1 = (x + \dot{x}, y + \dot{y}, z + \dot{z}, \theta, l, w, h, \dot{x}, \dot{y}, \dot{z})$ and $\tilde{X}_2 = (u + \dot{u}, v + \dot{v}, a + \dot{a}, r, \dot{u}, \dot{v}, \dot{a})$. In the following matching process, the 3DIoU refers to the Intersection-over-Union of three-dimensional volume between the predicted 3D bounding box state of previous-frame objects in the current frame and the 3D bounding box state of current-frame objects. The distance of center locations refers to the distance between the predicted center location state of previous-frame objects and center location of current-frame objects.

(2) The first matching (step 1): the matching process uses greedy algorithm. We use 3DIoU as the first measurement and the distance between center locations as the second measurement for cars. For example, when using 3DIoU as the measurement results in no suitable unmatched tracklets, distance between center locations will be used as the measurement. For pedestrians, distance between center locations is used as the only measurement. For the unmatched detection $unD_{>0.4}^{(t)}$ in the current frame, a 3D Kalman filter model, 2D Kalman filter model and a new tracklet will be created. In the paper, we initialize 3D Kalman filter model and 2D Kalman filter model of the unmatched detection $unD_{>0.4}^{(t)}$ with zero velocity for $\dot{x}$, $\dot{y}$, $\dot{z}$, $\dot{u}$, $\dot{v}$, $\dot{a}$.

(3) The second matching (step 2): It mainly deals with the previous-frame unmatched tracklets $unT^{(t-1)}$. The detections $D_{\leq 0.4}^{(t)}$ whose confidence is less than or equal to 0.4 are matched with predicted tracklets $Pred\_unT^{(t-1)}$ which are not matched in step 1 process. For these $D_{\leq 0.4}^{(t)}$, they will be merged into the existing tracklets if it satisfies the following two conditions: (1) the center location distance between it and some objects in the previous frame is less than 500 pixels; (2) the 3DIoU between it and any object in the current frame is 0; and (3) it is not close to the edges of the image, which means predicted objects should be more

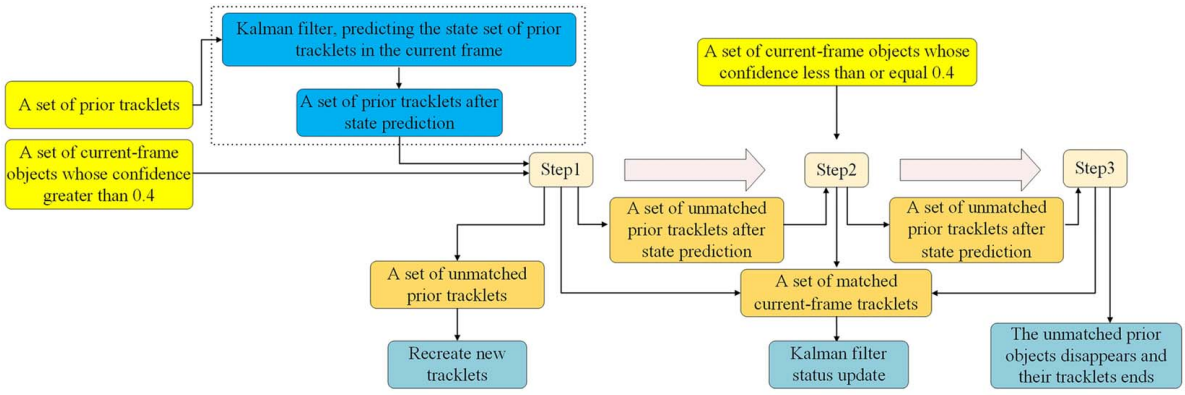

Fig. 3 Cascaded data association algorithm based on 3D Kalman filter

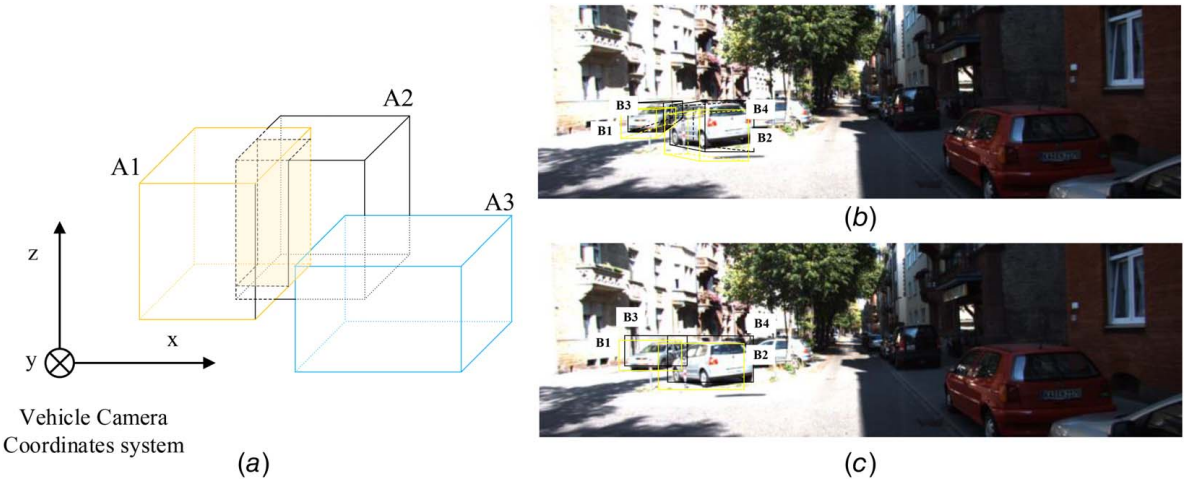

Fig. 4 Schematic diagram of 3D perspective scene: (*a*) schematic diagram of 3D bounding box, (*b*) schematic diagram of 3D multi-object tracking results, and (*c*) schematic diagram of 2D multi-object tracking results

than 20 pixels away from the edges of the image in this paper.

(4) The third matching (step 3): It mainly deals with prior tracklets that have not been matched in step 1 and step 2 process. For these $Pred\_unT^{(t-1)}$, it will be merged into the existing tracklets if it satisfies the following four conditions: (1) the 3DIoU between it and any object in the current frame is less than 0.3; (2) it is not close to the edges of the image; (3) Step 3 can only be used consecutively once for the same object; (4) it has been successfully matched at least three times before.

(5) For the matched detections in the current frame, their 3D Kalman filter and 2D Kalman filter status will be updated. Besides, prior tracklets which have not been matched in step 1, step 2, and step 3 process are considered to have disappeared in the current frame, and their tracklets are terminated.

In the data association algorithm, the 3DIoU is used as the first measurement for cars. This is because compared with IoU, 3DIoU has depth information and can better distinguish objects whose 2D bounding boxes which locate at different depths, as shown in Fig. 4. In Fig. 4(*a*), 2D bounding boxes of object A2 and object A3 overlap from the perspective of the camera's *z*-axis but locate at different depths, while object A1 and object A2 are all overlap in the form of 2D bounding box or 3D bounding box. In Figs. 4(*b*) and 4(*c*), bounding boxes B1 and bounding boxes B2 are the location of previous frame objects in the current frame, and bounding boxes B3 and bounding boxes B4 are the location of current frame objects, where Object B1 and Object B3 are the same object, and Object B2 and Object B4 are the same object. Object B1 and object B4 are almost nonintersecting when they are represented by 3D bounding box, but there is a certain area of overlap between object B1 and object B4 when they are represented by 2D bounding box.

In addition, the three-dimensional shape and position of cars between adjacent frames also change little, so it is suitable to use the 3D bounding box to describe its feature. However, for pedestrians, due to the swing of their arms and changes in their steps, the three-dimensional shape of pedestrians changes significantly between adjacent frames, so 3DIoU is not suitable as the measurement to match pedestrians.

5 Experiment and Result Analysis

5.1 Experimental Configuration and Dataset. The experimental configuration in this paper is shown in Table 1. The dataset uses the public KITTI tracking benchmark dataset, and its train set and test set have 21 and 29 video sequences, respectively. The dataset is collected by a camera mounted on a car, mainly to detect and track cars and pedestrians, and provide corresponding category, ID, 2D bounding box and 3D bounding box and other information. This paper only evaluates cars and pedestrians, so the eight different categories originally labeled in the dataset are merged into three categories. Specifically, Car and Van are merged into the "Car" category, and Pedestrian and Person are merged into the "Pedestrian" category. The "Cyclist" category is kept, and "Truck," "Tram," "DontCare" and "Misc" category is deleted. During the experiment, 21 video sequences in the train set were divided into train set and verification set at a ratio of 1:1. The validation set is used for ablation experiments to evaluate CenterTrack3D proposed in this paper.

5.2 Evaluation Metrics. The metrics used to evaluate the multi-object tracking algorithm are shown in Table 2, where MOTA and MOTP evaluate the overall performance of the multi-object tracking algorithm, and mostly tracked (MT), mostly lost (ML), ID-switch (IDS), and fragmentations (FRAG) evaluate the tracker's efficiency in assigning the correct ID to the object [17].

Table 1 Experimental configuration

Item	CPU	Computing memory	GPU	System
Content	Intel i5-9400F	11GB	NVIDIA GTX 1080Ti	Ubuntu16.04

Table 2 Evaluation metrics used for multiple object tracking

Metrics	Better	Perfect	Description
MOTA	↑	100%	Multiple object tracking accuracy
MOTP	↑	100%	Multiple object tracking precision
MODA	↑	100%	Multiple object detection accuracy
MODP	↑	100%	Multiple object detection precision
MT	↑	100%	Mostly tracked targets
ML	↓	0	Mostly lost targets
IDS	↓	0	Identity switches
Frag	↓	0	Fragmentations

Table 3 Comparison of experimental results with other multi-object tracking algorithms on test set of KITTI tracking benchmark dataset ("car" class)

Method	Time	MOTA	MOTP	MODA	MODP	MT	ML	IDs	Frag
AB3DMOT [12]	0.47 ms	83.84%	85.24%	83.86%	88.25%	66.92%	11.38%	**9**	**224**
TuSimple [18]	60 ms	86.62%	83.97%	87.48%	87.38%	72.46%	6.77%	293	501
Quasi-Dense [19]	70 ms	85.76%	85.01%	86.03%	88.06%	69.08%	**3.08%**	93	617
JRMOT [13]	70 ms	85.70%	**85.48%**	85.98%	88.42%	71.85%	4.00%	98	372
CenterTrack [4]	60 ms	87.70%	85.13%	88.70%	**87.99%**	75.38%	3.69%	345	655
CenterTrack3D	65 ms	**88.75%**	85.05%	**89.20%**	87.89%	**77.85%**	4.00%	156	400

Note: Bold fonts are the best values for each measurement.

5.3 Training Process. The training process of CenterTrack3D is basically the same as that of CenterNet, using multi-task learning to train all prediction heads. This paper uses the model trained by CenterNet on the nuScenes dataset to fine-tune CenterTrack3D on the KITTI tracking benchmark dataset. The model trained by CenterNet on the nuScenes dataset is provided by authors of CenterTrack [3].

The main problem in training CenterTrack3D is to input the prior heatmap $H^{(t-1)}$ that simulates a real scene. At inference time, the heatmap may contain any number of missing objects, wrongly

Table 4 Comparison of experimental results with other multi-object tracking algorithms on test set of KITTI tracking benchmark dataset ("pedestrian" class)

Method	Time	MOTA	MOTP	MODA	MODP	MT	ML	IDs	Frag
AB3DMOT [11]	0.47 ms	39.26%	64.87%	40.37%	90.27%	16.84%	41.84%	170	940
TuSimple [17]	60 ms	58.15%	71.93%	58.74%	91.37%	30.58%	24.05%	**138**	818
Quasi-Dense [18]	70 ms	56.81%	**73.99%**	57.91%	91.75%	31.27%	**18.90%**	254	1121
JRMOT [12]	70 ms	46.33%	72.54%	47.82%	**91.78%**	23.37%	28.87%	345	1111
CenterTrack [3]	60 ms	56.72%	73.38%	57.56%	91.77%	34.36%	21.65%	194	884
CenterTrack3D	65 ms	**59.40%**	73.79%	**60.04%**	**91.78%**	**42.61%**	**18.90%**	150	**812**

Note: Bold fonts are the best values for each measurement.

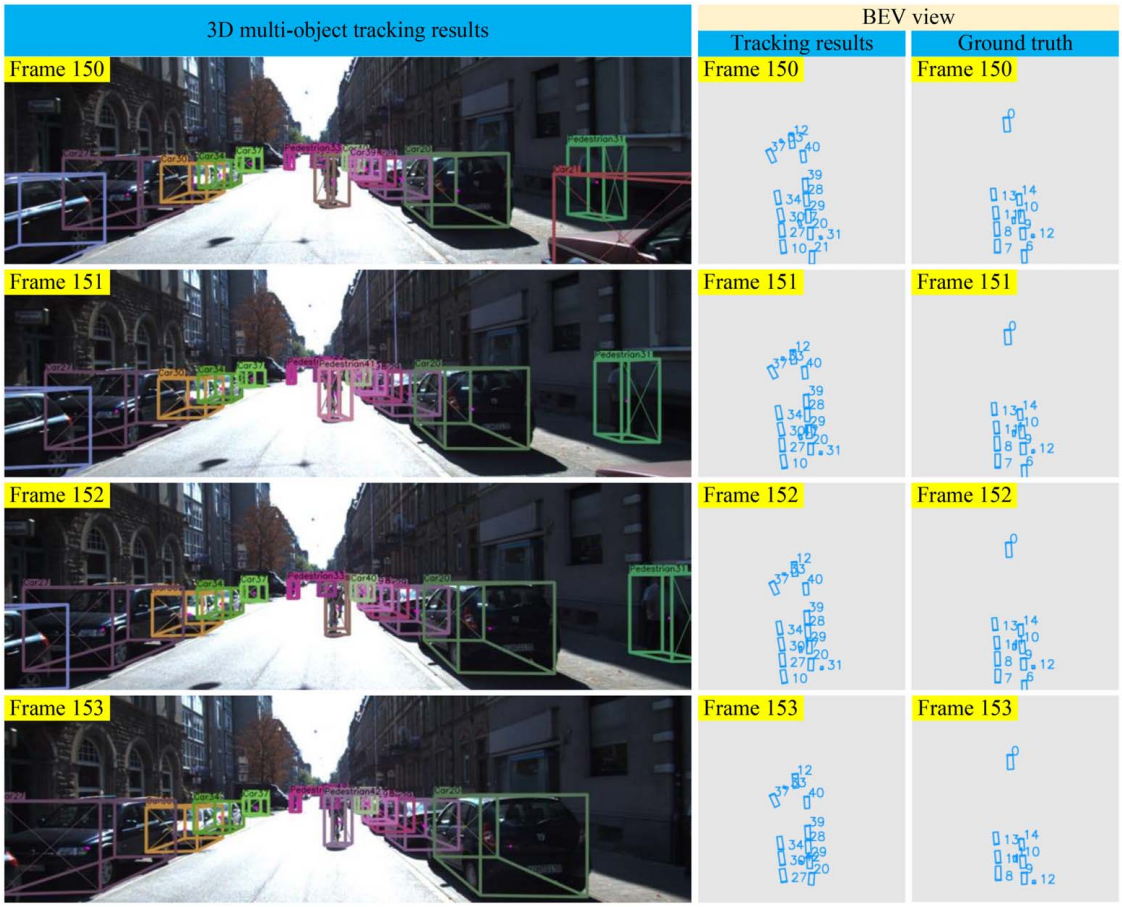

Fig. 5 3D Multi-object tracking results of four consecutive frames in KITTI tracking validation set

Table 5 Ablation experiment results on the verification set of KITTI tracking validation set ("car" class)

O[a]	C1[b]	C2[c]	S1[d]	S2[e]	S3[f]	MOTA	MOTP	MODA	MODP	MT	ML	IDs	Frag
✓						88.27%	87.46%	88.90%	90.82%	85.97%	2.15%	68	158
	✓					88.01%	87.21%	89.27%	90.76%	85.97%	2.15%	138	227
		✓				88.38%	87.21%	88.81%	90.65%	87.05%	1.80%	47	137
	✓	✓				87.83%	**87.57%**	89.05%	**90.91%**	86.69%	**1.43%**	133	220
	✓	✓	✓			88.75%	**87.57%**	89.05%	**90.91%**	86.69%	**1.43%**	33	125
	✓	✓	✓	✓		89.15%	87.44%	89.42%	90.84%	87.76%	**1.43%**	**30**	107
	✓	✓	✓	✓	✓	**89.33%**	87.32%	**89.63%**	90.76%	**88.48%**	**1.43%**	33	**105**

Note: Bold fonts are the best values for each measurement.
[a]The original multi-object tracker CenterTrack.
[b]Contribution one which is that the previous-frame image and the heatmap of previous-frame tracklets are added to the current-frame image as the network input with the idea of the attention mechanism.
[c]Contribution two which is that The second convolutional layer of the *hm* output head is replaced with dynamic convolution layer.
[d]Step 1 in the cascaded data association algorithm based on 3D Kalman filter.
[e]Step 2 in the cascaded data association algorithm based on 3D Kalman filter.
[f]Step 3 in the cascaded data association algorithm based on 3D Kalman filter.

localized objects, or false positives, but these errors will not appear in the ground-truth annotations of the dataset. Therefore, this paper simulates these three errors during the training process. First, we perform local dithering by adding Gaussian noise at the center point of each tracked object in the previous frame, as shown in Eq. (12). Second, we randomly add false positives near the ground-truth center point with probability λ_{fp}. Third, we simulate false negatives by randomly discarding detections with probability λ_{fn}. These three augmentations can effectively enhance the robustness of CenterTrack3D

$$(x', y') = (x + r \times \lambda_{jt} \times w_i, y + r \times \lambda_{jt} \times h_i) \tag{12}$$

where (x, y) is the ground-truth object center point, (x', y') is the center point after adding Gaussian noise, r is sampled from a Gaussian distribution, (w, h) is the width and height of the corresponding ground-truth center point.

During training and testing, we keep the original input resolution 1280×384. The hyperparameters are set to $\lambda_{fp} = 0.1$ and $\lambda_{fn} = 0.2$, and the output confidence threshold is $\varsigma = 0.3$.

In addition, during training, $I^{(t-1)}$ and $H^{(t-1)}$ do not need to be sampled from time $t-1$, and randomly sample from all frames $k \in [M_f - t, M_f + t]$ to avoid overfitting, where $M_f = 3$ is a hyperparameter.

5.4 Main Results. This paper retrains CenterTrack3D on all train sets of the KITTI tracking benchmark data set and evaluates it on the test set. The results are shown in Tables 3 and 4, where the experimental results of the original CenterTrack are obtained by implementing a 3D multi-object tracking experiment on the test set. Note that the experimental results of the original CenterTrack on the KITTI website are obtained by implementing a 2D multi-object tracking experiment on the test set. The detection results are shown in Fig. 5.

On the test set, CenterTrack3D runs at 65 ms, yields 88.75% MOTA for cars and 59.40% MOTA for pedestrians. Compared with the original CenterTrack, most metrics have improved significantly. In addition, the results also show that the performance of CenterTrack3D is more competitive than the methods that have been published on the KITTI rankings (Tables 3 and 4). In summary, our proposed method has priority performance on 3D multi-object tracking tasks.

5.5 Ablation Experiment. The main contributions of this paper to the improvement of the original CenterTrack are as follows: (1) The previous-frame image and the heatmap of previous-frame tracklets are added to the current-frame image as the network input with the idea of the attention mechanism; (2) The second convolutional layer of the hm output head is replaced with dynamic convolution layer; and (3) The original data association algorithm is replaced with a cascaded data association algorithm based on 3D Kalman filter. The following ablation experiments are performed on these three contributions, and the experimental results are shown in Tables 5 and 6. Note that the experimental results are the best results of training in the last ten epochs.

It can be seen from Tables 5 and 6 that *C*1 and *C*2 (shown as in Tables 5 and 6) both can improve the original multi-object tracker

Table 6 Ablation experiment results on the verification set of KITTI tracking validation set ("pedestrian" class)

O[a]	C1[b]	C2[c]	S1[d]	S2[e]	S3[f]	MOTA	MOTP	MODA	MODP	MT	ML	IDs	Frag
✓						69.86%	78.52%	70.52%	**94.43%**	50.00%	15.48%	30	151
	✓					70.68%	77.98%	72.60%	94.37%	51.19%	11.90%	86	209
		✓				70.23%	**78.41%**	71.30%	94.42%	52.38%	15.48%	48	158
	✓	✓				70.63%	78.24%	71.77%	94.25%	58.33%	**10.71%**	51	171
	✓	✓	✓			71.01%	78.24%	71.77%	94.25%	58.33%	**10.71%**	34	159
	✓	✓	✓	✓		72.15%	78.06%	72.59%	94.08%	**60.71%**	11.90%	**20**	**134**
	✓	✓	✓	✓	✓	**72.21%**	78.02%	**72.78%**	94.07%	**60.71%**	11.90%	25	135

Note: Bold fonts are the best values for each measurement.
[a]The original multi-object tracker CenterTrack.
[b]Contribution one which is that the previous-frame image and the heatmap of previous-frame tracklets are added to the current-frame image as the network input with the idea of the attention mechanism.
[c]Contribution two which is that the second convolutional layer of the *hm* output head is replaced with dynamic convolution layer.
[d]Step 1 in the cascaded data association algorithm based on 3D Kalman filter.
[e]Step 2 in the cascaded data association algorithm based on 3D Kalman filter.
[f]Step 3 in the cascaded data association algorithm based on 3D Kalman filter.

CenterTack, especially MOTA. In addition, step 1, step 2, and step 3 in the cascaded data association algorithm based on 3D Kalman filter all have a great effect on the improvement of the 3D multi-object tracking, which verifies the 3DIoU is more suitable for matching cars than distance between center points and continuous tracking of missing prior tracklets can improve the missing pedestrians. In brief, the improvement of network input mode and *hm* output head improve detections in each frame, and the 3D Kalman filter-based cascaded data association algorithm proposed in this paper is particularly suitable for solving the problem of 3D object association.

In addition, it can be seen from the ablation experiments in Tables 5 and 6 that some improvement steps seem to decrease some experimental metrics a little. For example, improvements of $C1$ and $C2$ will decrease some metrics a little, such as IDs and Frag. This is because these two improvements reduce the accuracy of the center point offset between two adjacent frames predicted by *tracking* head. This problem can be solved by our proposed cascaded data association algorithm based on 3D Kalman filter. However, the second and third steps of the cascade matching strategy will decrease metrics MOTP and MODP a little, because these two matching steps will add a small number of false detections to existing tracking trajectories. This is worth it, because MOTP and MODP have only decreased by less than 0.2%, but other metrics have improved significantly. For example, MOTA, MODA, and MT of "Car" category has increased by more than 0.6%, and MOTA, MODA, and MT of pedestrians has increased by more than 1%.

6 Conclusion and Future Work

This paper proposes an efficient and simple 3D multi-object tracking algorithm, CenterTrack3D, to solve the problem that CenterTrack ignores the 3D information of objects in the 3D multi-object tracking task. CenterTrack3D makes full use of the 3D information to match the same object more accurately between adjacent frames and can track missed tracklets in the current frame continuously. From the experimental results, the method proposed in this paper is very competitive on the KITTI tracking benchmark dataset. However, since this paper does not use laser point cloud data, only RGB images are used to estimate the depth of the object and predict the 3D bounding box of objects of interest in each frame of image. Therefore, we can study the fusion of laser point cloud and RGB image to further improve 3D multi-object detection and tracking in the future work.

Acknowledgment

We are grateful for the KITTI dataset produced jointly by Karlsruher Institut für Technologie and Toyota American Institute of technology.

Conflict of Interest

There are no conflicts of interest.

References

[1] Gioele, C., Francisco, L. S., Siham, T., Luigi, T., and Francisco, H., 2020, "Deep Learning in Video Multi-Object Tracking: A Survey," Neurocomputing, **381**(C), pp. 61–88.

[2] Punchihewa, Y. G., Vo, B. T., Vo, B. N., and Kim, D. Y., 2018, "Multiple Object Tracking in Unknown Backgrounds With Labeled Random Finite Sets," IEEE Trans. Signal Process., **66**(11), pp. 3040–3055.

[3] Zhou, X., Koltun, V., and Krhenbühl, P., 2020, "Tracking Objects as Points," European Conference on Computer Vision, Springer, Cham, pp. 474–490.

[4] Zhou, X., Wang, D., and Krhenbühl, P., 2019, "Objects as Points," preprint arXiv:1904.07850v1.

[5] Ren, S., He, K., Girshick, R., and Sun, J., 2015, "Faster r-cnn: Towards Real-Time Object Detection With Region Proposal Networks," IEEE Trans. Pattern Anal. Mach. Intell., **39**(6), pp. 1137–1149.

[6] Redmon, J., and Farhadi, A., 2018, "Yolov3: An Incremental Improvement," preprint arXiv:1804.02767.

[7] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., and Berg, A. C., 2016, "Ssd: Single Shot Multibox Detector," The 14th European Conference on Computer Vision, Springer, Cham, pp. 21–37.

[8] Bochkovskiy, A., Wang, C. Y., and Liao, H. Y. M., 2020, "YOLOv4: Optimal Speed and Accuracy of Object Detection," preprint arXiv:2004.10934.

[9] Ku, J., Mozifian, M., Lee, J., Harakeh, A., and Waslander, S. L., 2018, "Joint 3d Proposal Generation and Object Detection From View Aggregation," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Spain, pp. 1–8.

[10] Chen, X., Ma, H., Wan, J., Li, B., and Xia, T., 2017, "Multi-View 3D Object Detection Network for Autonomous Driving," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, pp. 1907–1915.

[11] Bewley, A., Ge, Z., Ott, L., Ramos, F., and Upcroft, B., 2016, "Simple Online and Realtime Tracking," 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, pp. 3464–3468.

[12] Weng, X., Wang, J., Held, D., and Kitani, K., 2020, "AB3DMOT: A Baseline for 3D Multi-Object Tracking and New Evaluation Metrics," preprint arXiv:2008.08063.

[13] Shenoi, A., Patel, M., Gwak, J., Goebel, P., Sadeghian, A., Rezatofighi, H., and Savarese, S., 2020, "JRMOT: A Real-Time 3D Multi-Object Tracker and a New Large-Scale Dataset," preprint arXiv:2002.08397.

[14] Hu, H. N., Cai, Q. Z., Wang, D., Lin, J., Sun, M., Krahenbuhl, P., and Yu, F., 2019, "Joint Monocular 3D Vehicle Detection and Tracking," Proceedings of the IEEE International Conference on Computer Vision, Seoul, South Korea, pp. 5390–5399.

[15] Lin, T. Y., Goyal, P., Girshick, R., He, K., and Dollár, P., 2017, "Focal Loss for Dense Object Detection," Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, pp. 2980–2988.

[16] Zhang, W., Zhou, H., Sun, S., Wang, Z., Shi, J., and Loy, C. C., 2019, "Robust Multi-Modality Multi-Object Tracking," Proceedings of the IEEE International Conference on Computer Vision, Seoul, South Korea, pp. 2365–2374.

[17] Bernardin, K., and Stiefelhagen, R., 2008, "Evaluating Multiple Object Tracking Performance: the CLEAR MOT Metrics," EURASIP J. Image Video Process., **2008**, pp. 1–10.

[18] Choi, W., 2015, "Near-online Multi-Target Tracking With Aggregated Local Flow Descriptor," Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, pp. 3029–3037.

[19] Pang, J., Qiu, L., Chen, H., Li, Q., Darrell, T., and Yu, F., 2020, "Quasi-Dense Similarity Learning for Multiple Object Tracking," preprint arXiv:2006.06664.