CAMO-MOT: Combined Appearance-Motion Optimization for 3D Multi-Object Tracking With Camera-LiDAR Fusion

Li Wang, Xinyu Zhang, *Member, IEEE*, Wenyuan Qin, Xiaoyu Li, Jinghan Gao, Lei Yang, *Graduate Student Member, IEEE*, Zhiwei Li, Jun Li, Lei Zhu, Hong Wang, and Huaping Liu, *Senior Member, IEEE*

***Abstract*— 3D Multi-object tracking (MOT) ensures consistency during continuous dynamic detection, conducive to subsequent motion planning and navigation tasks in autonomous driving. However, camera-based methods suffer in the case of occlusions and it can be challenging to track the irregular motion of objects for LiDAR-based methods accurately. Some fusion methods work well but do not consider the untrustworthy issue of appearance features under occlusion. At the same time, the false detection problem also significantly affects tracking. As such, we propose a novel camera-LiDAR fusion 3D MOT framework based on Combined Appearance-Motion Optimization (CAMO-MOT), which uses both camera and LiDAR data and significantly reduces tracking failures caused by occlusion and false detection. For occlusion problems, we are the first to propose an occlusion head to select the best object appearance features multiple times effectively, reducing the influence of occlusions. To decrease the impact of false detection in tracking, we design a motion cost matrix based on confidence scores which improve the positioning and object prediction accuracy in 3D space. As existing multi-object tracking methods always evaluate each category separately and do not consider the mismatch between objects of different categories, we also propose to build a multi-category cost to implement multi-object tracking in multi-category scenes. A series of validation experiments are conducted on the KITTI and nuScenes tracking benchmarks. Our proposed method achieves state-of-the-art performance with 79.99% HOTA and the lowest identity switches (IDS) value (23 for Car and 137 for Pedestrian) among all multi-modal MOT methods on the KITTI test dataset. And our method achieves state-of-the-art performance among all algorithms on the nuScenes test dataset with 75.3% AMOTA.**

***Index Terms*— Multi-object tracking, camera-LiDAR fusion, autonomous driving, intelligent transportation systems.**

I. Introduction

3D MULTI-OBJECT tracking (MOT) can be used to extract continuous dynamic information from surrounding environments. And it can also obtain the total number of objects in a field of view and predict the next object state to improve system reliability. Therefore, 3D MOT has been a crucial module in autonomous driving [1], [2].

Conventional MOT algorithms first extract the appearance or motion information from acquired detector results for corresponding objects. The similarity to an existing trajectory state is then calculated using an association method to achieve dynamic tracking. Camera-based MOT methods [3], [4] are primarily applied to 2D data by using rich visual texture information to achieve stable tracking, even for irregular motion. However, when an object is occluded due to interference, its visual feature reliability is significantly reduced, as shown in Fig. 1(a). LiDAR-based MOT methods [5], [6] primarily adopt motion information of objects for data association, which makes these methods accurately track objects, even if objects are occluded. However, this lack of visual texture information can cause mismatches when objects move irregularly or rapidly, as shown in Fig. 1(b). Therefore, camera-LiDAR fusion methods [7], [8], [9] are proposed to integrate the strengths of both methods. According to the fusion position, these methods can be divided into front-end and back-end fusion. Front-end fusion means feature-level fusion, such as JMODT [10], mmMOT [8], etc, where mmMOT [8] transfers the texture information in the image works into the three-dimensional space by fusing deep convolutional features and point clouds features, which enhances the representation of objects in different dimensions, and then uses the fused

Manuscript received 14 May 2022; revised 18 November 2022, 17 January 2023, and 16 May 2023; accepted 8 June 2023. This work was supported in part by the National High Technology Research and Development Program of China under Grant 2018YFE0204300; in part by the National Natural Science Foundation of China under Grant 62273198 and Grant U1964203; in part by the China Postdoctoral Science Foundation under Grant 2021M691780; and in part by the State Key Laboratory of Robotics and Systems, Harbin Institute of Technology (HIT), under Grant SKLRS-2022-KF-12. The Associate Editor for this article was Y. I. Wu. *(Corresponding author: Xinyu Zhang.)*

Li Wang is with the State Key Laboratory of Automotive Safety and Energy and the School of Vehicle and Mobility, Tsinghua University, Beijing 100084, China, and also with the State Key Laboratory of Robotics and Systems, Harbin Institute of Technology (HIT), Harbin 150001, China (e-mail: wangli_thu@mail.tsinghua.edu.cn).

Xinyu Zhang, Wenyuan Qin, Lei Yang, Jun Li, and Hong Wang are with the State Key Laboratory of Automotive Safety and Energy and the School of Vehicle and Mobility, Tsinghua University, Beijing 100084, China (e-mail: xyzhang@tsinghua.edu.cn; qinwenyuan1996@163.com; yanglei20@mails.tsinghua.edu.cn; lj19580324@126.com; wangh@tsinghua.edu.cn).

Xiaoyu Li and Jinghan Gao are with the State Key Laboratory of Robotics and System, Harbin Institute of Technology (HIT), Harbin 150006, China (e-mail: 22s108236@stu.hit.edu.cn; 22s108222@stu.hit.edu.cn).

Zhiwei Li is with the College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China (e-mail: 2022500066@buct.edu.cn).

Lei Zhu is with Mogo Auto Intelligence and Telematics Information Technology Company Ltd., Beijing 100029, China (e-mail: Ray.zhu_china@outlook.com).

Huaping Liu is with the State Key Laboratory of Intelligent Technology and Systems and the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: hpliu@tsinghua.edu.cn).

Digital Object Identifier 10.1109/TITS.2023.3285651

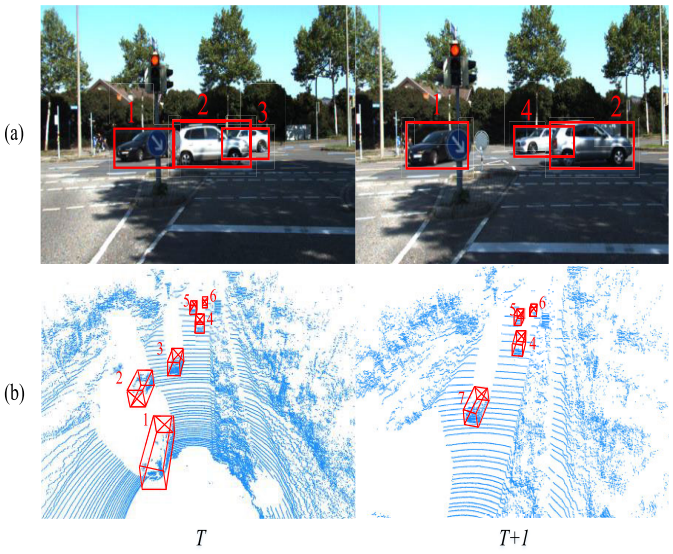


Fig. 1. Issues encounter in 2D and 3D MOT. (a) Inaccurate appearance features can occur in camera images when objects are occluded (ID changed from 3 to 4 at the moment $T+1$). (b) The presence of large inter-frame displacements in LiDAR data can also result in tracking failures when an existing object is considered a new object (ID changed from 3 to 7 at the moment $T+1$).

features between objects to construct an affinity network. Back-end fusion means fusion at the result level, such as DeepFusion-MOT [65], EagerMOT [9], etc. Among them, EagerMOT [9] first uses the 2D Intersection over Union to fuse the detection results sent by the 2D detector and the 3D detector and then associates the detection pairs via LiDAR association and Camera association to realize the back-end fusion process of the two modalities. Due to the difference between image and point cloud, feature alignment of the two modal data is quite difficult in front-end fusion methods [7], [8]. The back-end fusion method [9] utilizes the 2D and 3D detection results from the image and LiDAR to match. However, when the detector produces false detections and scene categories increase, tracking false objects and identity switches (IDS) between different categories often occur in back-end fusion.

To address these issues, we propose a novel multi-modal MOT framework called CAMO-MOT. Our method is based on the combined appearance-motion optimization, which effectively takes advantage of information from both images and point cloud information and solves the problems of object occlusion, tracker tracking false detection objects, and the association between different categories. CAMO-MOT consists of three main modules, including an optimal occlusion state-based object appearance module (O2S-OAM), a confidence score-based motion module (CS-MM), and a multi-category multi-modal fusion association module (M2-FAM).

We design O2S-OAM to identify object occlusion states by introducing a novel occlusion head, which identifies the degree of visualization of the current object by sampling the image area of each object and feeding it into the network. By identifying the occlusion situation of the object at each moment, the appearance feature with the best occlusion situation is selected to update the 2D features of the trajectory in the online tracking process, thereby enhancing the robustness of the appearance feature of the trajectory. The performance of the MOT algorithm depends mainly on the detector's accuracy. Still, the existing detectors [11], [12], [13], [14] all have the problem of false detection, so the existing trajectories often generate identity switches in the multi-object tracking algorithm due to interference from false detections. Therefore, the CS-MM is proposed to reduce the influence of false detection in the motion module. We utilize a robust and reasonable distance criterion—3D Generalized Intersection over Union ($gIoU_{3D}$) [15] to construct a motion cost matrix between detections and trajectories. Meanwhile considering the false detection objects usually have minor detection confidence, we set more significant costs for low-confidence detections based on the original $gIoU_{3D}$ cost matrix to reduce the tracked possibility. Currently, most multi-object tracking methods [6], [7], [10], [16], [17] only provide tracking under a single category. Although a few methods [9], [18] provide multi-category tracking results, there are identity switches between different categories due to the scene's complexity. To solve this problem, we propose the M2-FAM module to construct a multi-category cost, making the association only exist in the category itself.

Extensive experimental studies on the KITTI [19] and nuScenes tracking benchmarks [20] are implemented. On the KITTI tracking test dataset, our CAMO-MOT ranks first among all multi-modal MOT methods and achieves a +5.60% HOTA and a +2.56% MOTA compared with the famous EagerMOT [9] which is the state-of-the-art multi-modal method before. Furthermore, our method has the lowest IDS value (23 for Car and 137 for Pedestrian) compared with all other methods, which illustrates that our proposed method can maintain stable tracking in complex occlusion situations by better utilizing image and point cloud information. Furthermore, on the nuScenes tracking test dataset, our CAMO-MOT ranks **first** among all MOT methods and achieves a +1.20% AMOTA compared with BEVFusion [21], which ranks second and utilizes a much more powerful detector than our detector. Experimental results show that our CAMO-MOT performs excellently under different datasets and has strong portability. Usually, the quality of the detector has a very large impact on the performance of the tracker, however, our CAMO-MOT still achieves very competitive performance with a poor detector. We hope that CAMO-MOT can provide a simple but effective baseline algorithm.

The primary contributions of this study are as follows:

- We propose a novel camera-LiDAR fusion 3D MOT framework CAMO-MOT based on the combined appearance-motion optimization, which effectively integrates camera and LiDAR information to achieve stable 3D multi-object tracking for interfacing with many 3D detectors.
- We are the first to propose an occlusion head for state estimation, which addresses the effect of occlusion in 2D images by selecting the optimal appearance feature during tracking.
- We present a fusion association strategy that introduces a category cost to settle the interference problem for different categories in the tracking process.

II. Related Work

A. Tracking

Tracking can be divided into Single-Object tracking (SOT) [22], [23], [24], [25], [26] and Multi-Object tracking (MOT) [9], [17], [27], [28]. SOT uses the local information in each frame to track the given object in the beginning frame, whereas MOT does not have a priori knowledge of the object and usually uses the detection results sent by the detector to effectively associate all objects detected in each frame with the existing trajectory. However, both types of tracking algorithms should account for the impact of occlusion on tracked objects, such as SOT algorithms [29], [30] and MOT algorithm [31]. Likewise, both should consider perception and prediction, such as most MOT algorithms utilize Kalman filter to predict existing trajectories, SOT algorithm LiCaNet [32] extracts rich and complementary multi-modal data from LiDAR and camera sensors to deliver precise pixel-level joint perception and motion prediction in real-time. In this paper, we focus mostly on 2D and 3D MOT methods.

B. 2D Multi-Object Tracking

Existing image-based 2D MOT methods have benefited from the rapid development of detection algorithms [3], [33], [34]. Some methods [34], [35] utilize a tracking-by-detection framework, in which the tracker acquires the object region from the detector and then associates it with a trajectory using data association methods. This method often involves Kalman filters for IoUs or optical flow for matching prior to the deep learning era. These methods are simple and fast, but they fail very easily in some complex scenarios. Addictionally, LSSiam [36] incorporates local semantic information into visual object tracking and presents a focal logistic loss and an efficient online template updating strategy, which is a template-matching-based tracking strategy.

With the advent of deep learning, many trackers have employed deep appearance features for use in associating objects. For example, DeepSORT [37] employs an offline trained ReID model and Kalman filters to associate objects. The Deep Affinity Network [38] accepts two image frames as inputs, extracts appearance features under multiple perceptual object fields, and outputs a similarity score. Tracktor [34] uses the Faster R-CNN as a backbone to construct a regression head for object location in the next frame, using bounding boxes from the previous frame for tracking. JDE [35] adopts YOLOv3 [39] as a detector, adds a ReID branch to extract deep object features, and jointly trains them to improve algorithm performance. FairMOT [40] is similar to JDE in that Centernet [41] is utilized as a detector to improve algorithm performance further. ByteTrack [42] leverages YOLOX [43] as a detector and a Kalman filter combined with a confidence score to perform two associations, achieving the top rank for the MOT17 dataset. However, it does not utilize deep object features in images, which causes tracking failures in complex scenes. TransTrack [44] uses a joint detection and tracking framework and the DETR [45] architecture as a backbone. Each object detected in the previous frame is treated as a query and is then passed to the network for use in estimating the current state. In this way, the association method is used to complete the association tracks. MOTR [46] is a complete end-to-end multi-object tracking framework that utilizes video data for training. It establishes object associations in the network and considers both the appearance and loss of objects, achieving competitive results for the MOT16 dataset [47]. UMA [48] integrates tracking loss and metric learning losses into a triple network for multi-task learning, which effectively improves the computation efficiency and simplifies the training process. However, the performance of these algorithms is reduced significantly when the object is occluded, due to the introduction of information outside the object itself.

C. 3D Multi-Object Tracking

3D MOT [49], [50] has a similar structure to that of 2D MOT but spatial information, which dramatically improves tracking accuracy. Motion information has been leveraged for tracking in previous studies [51], [52] and is achieved by associating object state estimations in a previous frame with objects in the current frame. Some methods like [7] utilize an LSTM as an object state estimator to achieve tracking, but this approach often fails when objects are moving irregularly. [53] designs a novel affinity measurement function to associate objects using multiple types of features coming from LiDAR and camera. This method improves the data association process in which the coordinates of the nearest corner are used to determine the position of the associated object, whereas the geometric features collected by LiDAR are primarily used in the computation during inclement weather, but the effect is somewhat average. AB3DMOT [18] achieves excellent tracking performance using 3D Kalman filters for IoUs. Monocular cameras have also been used to estimate object distances and velocities for use as motion features in 3D MOT [49]. EagerMOT [9] fuses 3D and 2D detection results and then applies a Kalman filter to achieve the highest HOTA on the KITTI tracking benchmark at that time, but it often leads to wrong tracking in the case of occlusion. In addition, some methods [54], [55] have used hand-crafted point cloud features for 3D MOT. MmMOT [8] transfers the texture information from images into 3D spaces by fusing deep features and point clouds in different ways, to achieve various multi-modal MOT representations. JMODT [10] utilizes an attention mechanism to fuse point clouds and images, providing fused features to detectors and trackers for joint training. However, due to the sparsity of the point cloud, the information of the two modalities cannot be effectively aligned in the data processing stage, so the information of the two modalities cannot be fully utilized. DeepFusion [56] combines camera data with deep LiDAR features rather than raw points and distances. Two modules are proposed to improve performance by addressing the alignment of data augmentation and the physical alignment of feature fusion. PnPNet [57] uses an end-to-end 3D MOT framework to solve detection and tracking but does not utilize image information, since only point cloud data is included. However, none of these methods introduce

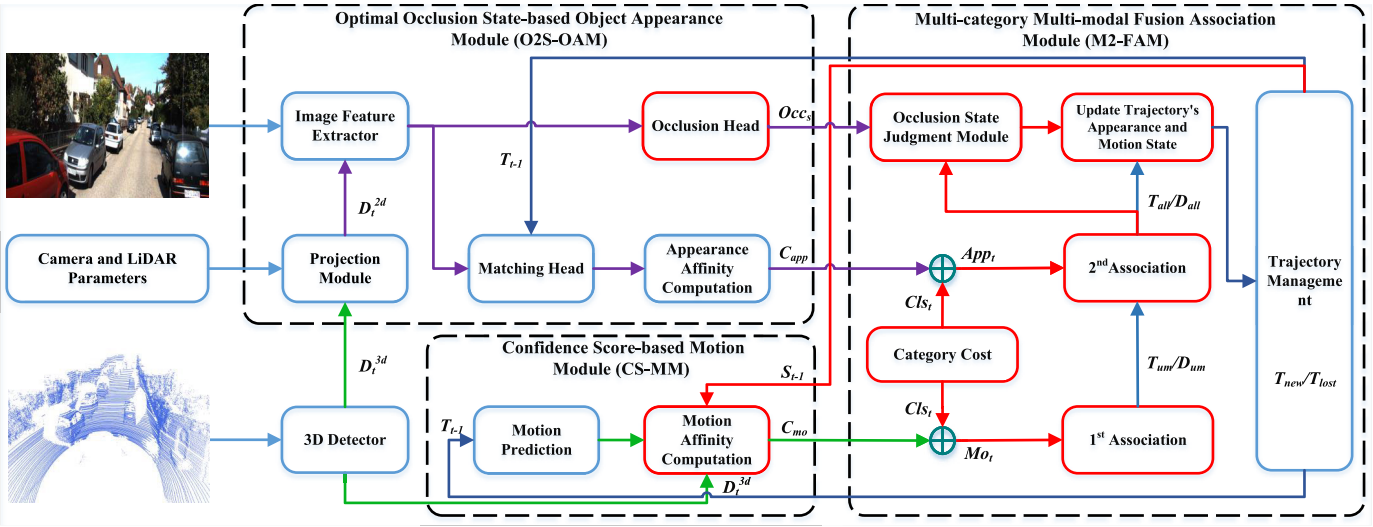


Fig. 2. An overview of our proposed CAMO-MOT framework during online tracking in which the following steps are implemented at each discrete time t. (I) The 3D detector inputs the point cloud to produce the 3D detection results D_t^{3d}, which are then passed to the CS-MM module. Simultaneously, we project D_t^{3d} onto the pixel plane of the camera to obtain the 2D detection results D_t^{2d} that corresponds to D_t^{3d}, which are then passed to the O2S-OAM module. (II) O2S-OAM is used to acquire deep object features from the image extractor at the frame t. Deep features are then passed to the occlusion head and used to estimate the occlusion state $Occ_s \in \{0, 1, 2, 3\}$ for each 2D detection result (D_t^{2d}). Following this step, the matching head outputs the appearance association matrix C_{app} based on the deep features for D_t^{2d} and the trajectories that are still alive after the previous frame $t-1$ have been processed (T_{t-1}). (III) CS-MM is used to predict the states of T_{t-1} at the current frame t and solve the motion association matrix C_{mo} using 3D Generalized Intersection over Union ($gIoU_{3D}$) distance, the 3D detection results D_t^{3d}, and S_{t-1} (the confidence scores of T_{t-1}). (IV) M2-FAM introduces the category cost Cls_t to C_{app} and C_{mo} which are used to obtain App_t and Mo_t. The first association is used to output detections and trajectories of various matching states based on Mo_t and a second association is then applied to the remaining unmatched detections D_{um} and unmatched trajectories T_{um} based on App_t. (V) Finally, the motion state of the matched trajectories T_{all} are all updated. However, we merely utilize the features of 2D bounding boxes with low occlusion ($Occ_s = 0$) in all matched detections to update the appearance state of the corresponding trajectories. The remaining unmatched detections D_{all} are considered to be new trajectories.

occlusion branches to improve 3D MOT. Furthermore, most methods only consider a single tracking category, which imposes certain restrictions on tasks in complex scenarios, such as cars and pedestrians being considered as the same object mistakenly.

In this study, we present a unique multi-modal MOT framework called CAMO-MOT that efficiently integrates image and point cloud through two associations in which two modalities are individually analyzed. We are the first to introduce an occlusion head to reduce the effects of occlusions and implement multi-category cost to improve tracking effects in multi-category object scenes.

III. CAMO-MOT

This section provides a comprehensive description of the proposed 3D MOT framework CAMO-MOT, which is based on a framework for tracking-by-detection. As shown in Fig. 2, the method comprises three basic modules: O2S-OAM, CS-MM, and M2-FAM.

A. Optimal Occlusion State-Based Object Appearance Module (O2S-OAM)

This section presents the details of O2S-OAM, mainly including the image feature extractor, the occlusion head, and the matching head. Fig. 3 shows the training process of the algorithm.

1) *Image Feature Extractor:* A modified DLA-34 [41] is utilized as the image feature extractor, as shown in Fig. 3. The image at the moment t is used as the input and outputs to the feature map $F_t = \{f_1, f_2, \ldots, f_M\}$, where M is the number of feature maps.

2) *Detection Head:* Both the occlusion head and the matching head directly extract the features of objects on the backbone network, and the detection task can just well train the backbone network to extract features. In addition, DEFT [17] and FairMOT [40] both show that the same backbone network is used for object detection and inter-frame correlation by sharing features can improve the efficiency and accuracy of each subtask, so we introduce a CenterNet-based detection head in the backbone network to obtain better network parameter weights. This is very beneficial for the matching head and occlusion head to share the same weights.

During inference, in order to obtain appropriate spatial alignment of the bounding boxes of different modalities of the same object, we employ the projection of the object's 3D bounding box onto the camera pixel plane as the object's 2D bounding box. The detecting head is not utilized throughout the inference process.

3) *Occlusion Head:* Algorithm performance is typically affected when objects are occluded. In this study, an occlusion head is introduced for the first time to identify occlusion states. To establish an association, a tracker is utilized to select optimal appearance features from multiple moments.

During training, image feature maps F_t, acquired by the image feature extractor using the ground truth of multi-object bounding boxes $B_t = \{b_1, b_2, \ldots, b_N\}$, are fed to the occlusion head, where N is the number of objects. This process

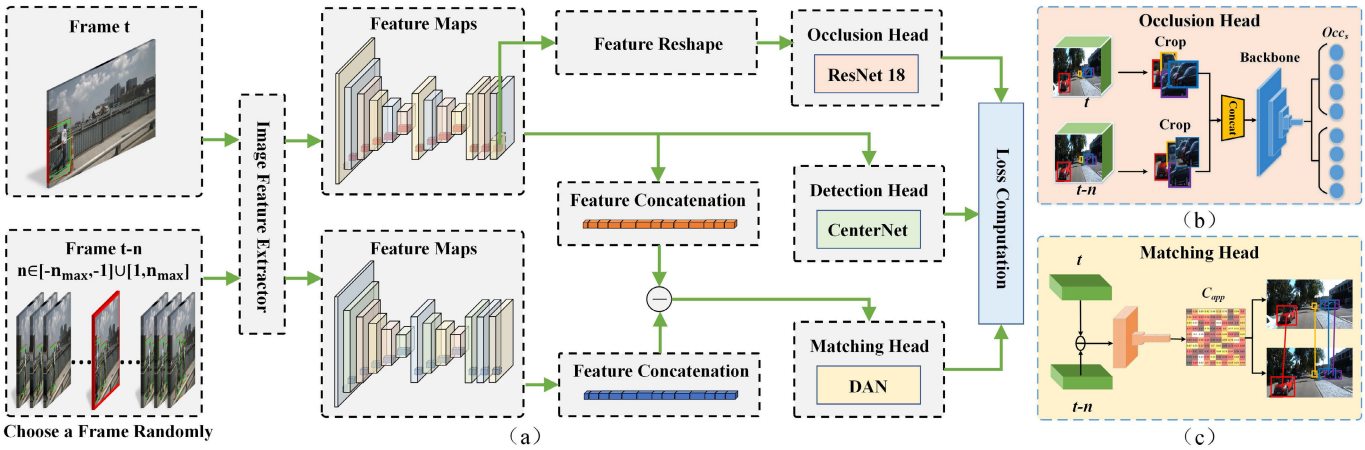


Fig. 3. The training network structure of the O2S-OAM module. (a) The overall structure. (b) The structure of the occlusion head. (c) The structure of the matching head. The following steps comprise the training process: (I) We use the current image frame and nearby frames at random as inputs. Concurrently, the locations of all ground-truth boxes on both images are recorded. (II) The multi-layer feature maps of the two images are obtained after passing through the image feature extractor. (III) The current frame's feature maps are directly sent into the detection head for 2D object detection. (IV) In the last stage of the feature maps, we first extract the depth features of each object based on the ground-truth bounding box coordinates of all objects in the current frame. We then resize the depth features and input the features into the occlusion head for occlusion status recognition. (V) According to the ground-truth coordinates of the center points of all objects in two frames, we obtain the corresponding depth features of all objects at each stage of feature maps. Then we aggregate the features of each object along the channel dimension and send it to the matching head to calculate the object affinity between the two frames. (VI) Finally, we add the losses of the three-part head network as the overall loss and then perform back-propagation and parameter update of the network.

is shown in Fig. 3. In order to improve the accuracy of the algorithm, the image area of the object is then bilinearly interpolated to 224×224 and sent to the occlusion head, which uses ResNet18 as its backbone, to identify occlusion states. In modern datasets [19], [20], the actual occlusion status of objects is usually directly given in the data annotation information in the form of an integer index among 0, 1, 2, and 3. We use the occlusion state of the object provided by the dataset as the ground truth label and then calculate the loss with the predicted value of the network. The term $Occ_s \in \{0, 1, 2, 3\}$ is then constructed for each object to determine its occlusion state, which is taken from the annotation information of the data set, and utilized for online tracking to achieve the optimal selection of appearance features.

During inference, the occlusion head is used to determine the occlusion state of the current frame's Det^{2D}. In the trajectory update module, F_t of the object closest to the current frame and with a low occlusion state is used as the appearance feature of the trajectory. Based on the matching head in the O2S-OAM module, each trajectory then uses its latest appearance features to perform appearance affinity computation with the detection results of the next frame.

4) *Matching Head:* Most existing association methods calculate similarity values by extracting ReID features used to associate objects. However, these methods do not adequately account for object relationships. As depicted in Fig. 3, we use an end-to-end association network to directly output an association matrix, which significantly enhances algorithm generality. This matching head is similar to that of prior studies [17], [38] and uses object feature embedding in the association of detection results between two frames, but the distinction is that we adopt the difference between features of the two frames as the input as the difference can better represent the degree of correlation of features. These embedded data are then extracted using the feature map acquired by the image feature extractor. The position of the object center in an original image of input size $W \times H$ is given by (x, y), which is then mapped to the m^{th} feature map of size $W_m \times H_m \times C_m$. This position is then shifted to $(\frac{x}{W}W_m, \frac{y}{H}H_m)$ during extraction of the C_m-dimensional vector o_m. Since the number of objects in each frame is not unique, N_{max} is set as the maximum number of detections per frame. The tensor $E_{t,t-n} \in R^{(C\times M)\times N_{max}\times N_{max}}$ is reconstructed by concatenating the differences between object features in frames t and $t-n$, and using a zero tensor to fill in any numbers less than N. A 1×1 convolutional neural network is adopted to produce the final association cost matrix $A_{t,t-n} \in R^{N_{max}\times N_{max}}$. Then, a row and a column are added to $A_{t,t-n}$ and the network's learnable parameters are utilized to augment the matrix with objects that are not associated (i.e., new objects entering or old objects leaving a scene [17], [38]). A $Softmax$ function is then used to process each row and column separately, to produce the final association matrices $\hat{A}_t, \hat{A}_{t-1} \in R^{(N_{max}+1)\times(N_{max}+1)}$. The averages of these matrices are used as the final affinity scores.

B. Confidence Score-Based Motion Module (CS-MM)

Following the steps of most MOT algorithms, we assume that the motion model of each object is CA (Constant Acceleration), thereby setting the state-transition matrix A as shown in Equation 3. The linear Kalman filter depicted in Equations 1 and 2 is then used to predict the motion states of T_{t-1} in the current frame t.

$$\hat{T}_t^j = AT_{t-1}^j, \tag{1}$$

$$\hat{P}_t^j = AP_{t-1}^j A^T + Q, \tag{2}$$

$$A = \begin{bmatrix} E_{3\times3} & \sigma E_{3\times3} & \frac{1}{2}\sigma^2 E_{3\times3} & \\ O_{3\times3} & E_{3\times3} & \sigma E_{3\times3} & O_{3\times n} \\ O_{3\times3} & O_{3\times3} & E_{3\times3} & \\ & O_{n\times9} & & E_{n\times n} \end{bmatrix}, \tag{3}$$

where T_{t-1}^{j} is the state of the j-th trajectory in T_{t-1} at the previous moment $t-1$, $\hat{T}_t^j$ is the estimated state of for T_{t-1}^{j} at the current moment t, P_{t-1}^{j} is the corresponding error covariance, $\hat{P}_t^j$ is the error covariance estimated for the current moment, Q is the covariance matrix for the state function, E and O represent the unit and zero matrices, σ is the interval time between two adjacent frames of the LiDAR scan, and $n = 13$ (13 represents position (x, y, z), velocity (vx, vy, vz), acceleration (ax, ay, az) and the other tracking information (length l, width w, height h, and rotation angle α) of an object in 3D space).

Generally speaking, an object with a low confidence score has a high false detection risk and a much lower probability of being associated. Objects with lower confidence scores have higher costs and are more challenging to track continuously. In this study, a confidence score is used to solve a cost matrix jointly. As shown in Equation 5, the motion cost matrix is divided by the predicted score of the trajectory obtained by Equation 6, in which trajectories with a lower state score exhibit higher losses when solving for optimal association pairs.

$$\begin{aligned} M_t^{i,j} &= 1 - gIoU_{3D}(D_t^{3d,i}, \hat{T}_t^j) \\ &= 2 - \frac{V(D_t^{3d,i} \cap \hat{T}_t^j)}{V(D_t^{3d,i} \cup \hat{T}_t^j)} - \frac{V(D_t^{3d,i} \cup \hat{T}_t^j)}{V_{hull}(D_t^{3d,i}, \hat{T}_t^j)} \end{aligned} \tag{4}$$

$$C_{mo} = \begin{bmatrix} M_t^{1,1}/\gamma_t^1 & \cdots & M_t^{1,N_t^T}/\gamma_t^{N_t^T} \\ \vdots & \ddots & \vdots \\ M_t^{N_t^D,1}/\gamma_t^1 & \cdots & M_t^{N_t^D,N_t^T}/\gamma_t^{N_t^T} \end{bmatrix}, \tag{5}$$

$$\gamma_t^\sigma = \begin{cases} 1, & c_{t-1}^\sigma = 0 \quad c_t^\sigma \neq 0 \\ \gamma_t^\sigma + \theta c_t^\sigma & otherwise \end{cases}, \tag{6}$$

where M_t is the cost matrix of the motion calculated with $gIoU_{3D}$ at moment t. $D_t^{3d,i} = \{x, y, z, w, h, l, \alpha, (vx, vy)\}$ is the i-th 3D detection bounding box containing the center position (x, y, z), and the 3D size(width, length, height) (w, h, l), and the rotation angle α, and the velocity (vx, vy) on the ground plane of the current frame t, note that whether velocity information is included or not depends on the dataset. $V(D_t^{3d,i} \cap \hat{T}_t^j)$, $V(D_t^{3d,i} \cup \hat{T}_t^j)$ are the intersection and union volumes of $D_{3d,t}^i$ and $\hat{T}_t^j$, respectively. $V_{hull}(D_t^{3d,i})$ is the volume of the convex hull computed by $D_t^{3d,i}$ and $\hat{T}_t^j$. N_t^D is the number of objects detected in the current frame. N_t^T is the number of trajectories at present. $c_t^\sigma \in [0, 1]$ is the detection confidence score. γ_t^σ is the state prediction score for the current trajectory (representing the degree of reliability). θ is the trajectory state decay factor. C_{mo} is the final output motion cost matrix of CS-MM.

C. Multi-Category Multi-Modal Fusion Association Module (M2-FAM)

A 3D motion cost matrix is first applied to estimate the position of the object in 3D space to compensate for the effects of occlusions in 2D space by predicting the 3D space position of the object at the last moment. An appearance cost matrix is then used to solve tracking failures caused by large inter-frame displacements and irregular motion of the object in 3D space. However, during the construction of these two cost matrixes in the online tracking, the affinity between every two objects is calculated, which makes the cost matrix mixed with many invalid cost calculations (between detections and trajectories of different categories). Interference between categories frequently results in tracking failures in MOT tasks, especially for densely distributed objects with multiple categories. To address these issues, we first introduce a category cost function during the inference phase so that objects are associated only within categories that improve algorithm robustness in complex multi-category scenes.

In this process, individual categories, are assigned different constants (e.g. 0, 1, 2) as their category indicators. We use Equation 7 to confirm whether the trajectory and detected categories are identical, and then use Equation 8 to calculate the category cost. It is important to note that a large gap (e.g. 10^5) is included between the constants to ensure no inter-class interference is present in subsequent associations. The resulting association framework is shown in Fig. 3. The motion cost matrix C_{mo} and the appearance cost matrix C_{app} are acquired from CS-MM and O2S-OAM, respectively. As shown in Equation 9 and 10, category cost is then separately added to the outputs Mo_t and App_t for subsequent associations.

$$Dis_t^{i,j} = D_t^i(cls) - T_{t-1}^j(cls), \tag{7}$$

$$Cls_t^{i,j} = \begin{cases} 10^5, & Dis_t^{i,j} \neq 0 \\ 0 & otherwise \end{cases}, \tag{8}$$

$$Mo_t = C_{mo} + Cls_t, \tag{9}$$

$$App_t = C_{app} + Cls_t, \tag{10}$$

where Dis_t is the difference between category indicators corresponding to all detections D_t and the all active trajectories T_{t-1} at moment t, cls is the index of the corresponding category, and Cls_t is category cost.

The Mo_t and App_t terms are acquired in the association phase. As shown in Equation 11, we first associate trajectories based on Mo_t to acquire the corresponding detections.

$$\begin{aligned} &\min \sum_{i=1}^{N_t^{tra}} \sum_{j=1}^{N_t^{det}} Mo_t^{i,j}, \\ &s.t. \quad Mo_t^{i,j} \leq \theta_{mo} \end{aligned} \tag{11}$$

We then use App_t to associate the remaining unmatched detections D_{um} and trajectories T_{um} after the first association, as shown in Equation 12.

$$\begin{aligned} &\min \sum_{i=1}^{N_{t'}^{tra}} \sum_{j=1}^{N_{t'}^{det}} App_t^{i,j}, \\ &s.t. \quad App_t^{i,j} \leq \theta_{App} \end{aligned} \tag{12}$$

where N_t^{det} and N_t^{tra} are the number of detections and trajectories at the moment t, respectively, $N_{t'}^{det}$ is the number of unassociated detections, $N_{t'}^{tra}$ is the number of unassociated trajectories, and θ_{mo} and θ_{App} are the maximum association costs for the motion and appearance modules, respectively.

Because occlusion features that contain information other than the object itself are unreliable, appearance features and occlusion states are maintained for each moment of the tracking process. When updating the trajectory, we select the optimal appearance features among a lower occlusion state among multiple trajectory moments. These are assumed to be the final trajectory features to further improve algorithm robustness by eliminating the effects of occlusions.

D. Training

During training, pairs of images with an interval of n frames are used as inputs to the O2S-OAM, as shown in Fig. 3. This interval between image pairs is set as a random number of frames ($1 \leq n \leq n_{max}$) to promote robustness to temporary occlusions or missed detections [17], [58]. ResNet18 is adopted as the backbone for the occlusion head, with cross-entropy serving as its loss function $Ł_{occlusion}$. The specific form is as follows:

$$Ł_{occlusion} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{4} y_{ic} log(p_{ic}), \tag{13}$$

where y is the label and p is the probability vector output by the model.

In each training phase, we generate a ground truth association matrix M_g of size $(N_{max}+1)\times(N_{max}+1)$, consisting of entries $[i, j] \in 0, 1$. Values of 1 and 0 in the matrix indicate the object is associated when i and j are less than N_{max}, respectively. A value of 1 in the matrix (for i or j equal to $N_{max}+1$) indicates the object is lost or newly emerged and should not be associated.

Two frames are input and used to denote $Ł_{match_t}$ and $Ł_{match_{t-n}}$ as the associated matrices in frames t and $t-n$, respectively. Considering both should be symmetric, we apply an average as the final association loss function $Ł_{match}$:

$$Ł_{match} = \frac{Ł_{match_t} + Ł_{match_{t-n}}}{2(N_t^{gt} + N_{t-n}^{gt})}, \tag{14}$$

$$Ł_{match_t} = \sum_{i=1}^{N_{max}}\sum_{j=1}^{N_{max}+1} M_g(i, j) log(Softmax(\hat{A}(i, j))), \tag{15}$$

$$Ł_{match_{t-n}} = \sum_{i=1}^{N_{max}+1}\sum_{j=1}^{N_{max}} M_g(i, j) log(Softmax(\hat{A}(i, j))), \tag{16}$$

where N_t^{gt} and N_{t-n}^{gt} represent the number of objects at times t and $t-n$, respectively. The final loss function $Ł_{join}$ can then be expressed as follows:

$$Ł_{join} = \frac{1}{e^{\lambda_1}} Ł_{detect} + \frac{1}{e^{\lambda_2}}(Ł_{match} + Ł_{occlusion}) + \lambda_1 + \lambda_2, \tag{17}$$

where $Ł_{detect}$ is the CenterNet detection head loss and λ_1 and λ_2 are the learnable loss weights for detection and other branching tasks (e.g. occlusion head and matching head).

IV. Experiments

In order to fully verify the performance of our proposed CAMO-MOT algorithm, we evaluate it on the famous KITTI and nuScenes tracking datasets: (i) KITTI 2D MOT, (ii) KITTI 3D MOT, (iii) nuScenes 3D MOT. First, we introduce the two datasets and the evaluation metrics. Then, we described the implementation details of our method on each dataset. Finally, we presented comparative experiments with state-of-the-art methods on two benchmarks and adequate ablation studies. In addition, we also provide qualitative visualizations to illustrate the effectiveness.

KITTI: KITTI tracking benchmark [19] provides 21 training sequences and 29 test sequences, each sequence consists of hundreds of frames. The KITTI training set contains 8,008 frames, and the test set includes 11,095 frames. The input used in these experiments includes images, point clouds, and IMU/GPS data. KITTI provides the ground truth of the training set at a frequency of 10 Hz. For 2D MOT tracking benchmark, the result of the algorithm in the test set needs to be submitted to the KITTI official website, and HOTA [59] is the primary metric. For 3D MOT tracking benchmark, we follow the evaluation metric of [18], using AMOTA [18] as the primary metric.

Training sequences 1, 6, 8, 10, 12, 13, 14, 15, 16, 18, and 19 are applied as the validation set [9], [18], while other sequences are regarded as the training set. Our algorithm is applied to analyze cars and pedestrians, in agreement with the KITTI test set evaluation.

nuScenes: The nuScenes [20] tracking benchmark consists of 850 training sequences and 150 test sequences. Each sequence consists of approximately 40 frames (2Hz in 20 seconds), which further assesses the robustness of the MOT algorithm under various scenarios. The training set contains 34,149 frames, while the test set includes 6,008 frames. nuScenes samples keyframe (image, LiDAR, radar) at a rate of 2Hz and deliver annotation information for each keyframe. The keyframe frequency of 2Hz is not favorable to the precise prediction of the motion model and introduces a significant inter-frame displacement, posing a formidable design problem for the 3D MOT algorithm. For the 3D MOT tracking benchmark, the algorithm's performance on the test set must be evaluated using the official evaluator, which uses AMOTA [18] as the primary evaluation metric.

In contrast to KITTI, we segregate the validation set from the training set using the official script.[1] The validation set includes 150 scenes which are comprised of complex traffic and weather conditions, 6,019 frames, and 140k instances of object annotations. Our algorithm tracks seven categories specified by nuScenes.

Evaluation Metrics: For KITTI 2D tracking benchmark, we mainly follow the rules of CLEAR MOT [60] and HOTA [59] for evaluation. The Higher-Order Tracking Accuracy (HOTA) comprehensively evaluates the performance of the tracker considering the impact of different detection

[1] https://github.com/nutonomy/nuscenes-devkit/blob/master/python-sdk/nuscenes/utils/splits.py

thresholds on tracking. The HOTA is defined as:

$$\begin{cases} TPA(c) = \{k\}, \\ k \in \{TP \mid prID(k) = prID(c) \wedge gtID(k) = gtID(c)\}, \\ FNA(c) = \{k\}, \\ k \in \{TP \mid prID(k) \neq prID(c) \wedge gtID(k) = gtID(c)\} \\ \bigcup \{FN \mid gtID(k) = gtID(c)\}, \\ FPA(c) = \{k\}, \\ k \in \{TP \mid prID(k) = prID(c) \wedge gtID(k) \neq gtID(c)\} \\ \bigcup \{FP \mid prID(k) = prID(c)\}, \end{cases} \tag{18}$$

$$HOTA = \sqrt{\frac{\sum_{c \in TP} A(c)}{\mid TP \mid + \mid FN \mid + \mid FP \mid}}, \tag{19}$$

$$A(c) = \frac{\mid TPA(c) \mid}{\mid TPA(c) \mid + \mid FNA(c) \mid + \mid FPA(c) \mid}, \tag{20}$$

where TP means true positives, FN means false negatives, FP means false positives, TPA means true positive associations, FNA means false negative associations, FPA means false positive associations, prID is the predicted identity, and gtID is the actual identity.

CLEAR-MOT consists mostly of two exhaustive evaluation metrics: MOTA and MOTP. The Multi-Object Tracking Accuracy (MOTA) measures the overall tracking accuracy and is defined as:

$$MOTA = 1 - \frac{\sum_t (F_t + N_t^m + IDs_t)}{\sum_t G_t}, \tag{21}$$

where F_t, N_t^m, IDs_t and G_t represent false positives, the number of misses, mismatches, and ground truth, respectively.

The value of Multi-Object Tracking Precision (MOTP) is measured by the overlapping ratio between the estimated object and its ground truth and is defined as:

$$MOTP = \frac{\sum_t^i c_t^i}{\sum_t M_t}, \tag{22}$$

where c_t^i represents the distance between detection and its corresponding ground truth, and M_t is the number of all matches. Additionally, we employ five more metrics, MT (mostly tracked), ML (mostly lost), FP (false positives), FN (false negatives), and IDS (identity switches), aiming for a better comparison of the tracker performance.

For KITTI 3D MOT and nuScenes 3D MOT, we mainly follow the rules of averaged MOTA and MOTP (AMOTA and AMOTP [18]), which can completely evaluate the overall accuracy and robustness of the tracker under various recall thresholds. AMOTA is defined as:

$$AMOTA = \frac{1}{L} \sum_{r \in \left\{\frac{1}{L}, \frac{2}{L}, \cdots, 1\right\}} MOTA_r, \tag{23}$$

where $MOTA_r$ represents the MOTA computed at a specific recall value r. Since $MOTA_r$ has a strict upper bound of r, in order to get metrics ranging from 0% to 100%, $MOTA_r$ and AMOTA need to be scaled numerically. $sMOTA_r$ (scaled MOTA) and sAMOTA (scaled AMOTA) are defined as follows:

$$sMOTA_r = max(0, \frac{MOTA_r}{r}). \tag{24}$$

$$sAMOTA = \frac{1}{L} \sum_{r \in \left\{\frac{1}{L}, \frac{2}{L}, \cdots, 1\right\}} sMOTA_r. \tag{25}$$

Note that the AMOTA in the official nuScenes tracking benchmark is in fact the sAMOTA in [18].

A. Implementation Details

Our tracking method is implemented in Python under the Pytorch framework for all benchmarks and it is trained and evaluated on RTX 3090 GPUs. We employ a custom-written Python script for 3D visualization. During training, 13 feature maps are extracted from the modified DLA-34 embedded backbone [17]. We also transport the final feature map to the occlusion head. The data are augmented using random cropping, flipping, and scaling to increase robustness.

1) KITTI: During training, CAMO-MOT is trained for 100 epochs using the Adam [61] optimizer with an initial learning rate of $1.25e^{-4}$ and a batch size of 8. At 60 epochs, the learning rate is decreased by e^{-5}.

Our method can run at about 25 FPS (Frames Per Second) with an RTX 3090 GPU during inference. Hyperparameters are selected based on the highest HOTA scores found in the validation set. 3D detection results are filtered using Non-Maximum Suppression (NMS) with a threshold of 0.1 ($\theta_{nms} = 0.1$). We use $\theta_{mo} = 0.01$, and $\theta_{App} = 1.4$ as thresholds for the Hungarian algorithm to match motion and appearance matrices. A new trajectory will be created for the unmatched detection in the current frame. A trajectory is discarded if it has not been updated in the last 15 frames. For each trajectory's unmatched frames, the motion model's predicted trajectory state is appended to the result file.

2) nuScenes: During training, we adopt the same optimizer (Adam [61]) and batch size (8) as on the KITTI dataset. We train our method with a learning rate of $1.25e^{-4}$ for 60 epochs.

Hyperparameters are chosen based on the best AMOTA score identified in the validation set. We evaluate CAMO-MOT with a variety of 3D detectors. Confidence Score Filter (CSF) and NMS are applied to the input 3D detection boxes. Each object's 3D bounding box is projected onto all six cameras, and the 2D bounding box with the biggest projected area is used as the object's image feature. We use $\theta_{nms} = 0.08$ for all 3D detectors. The threshold for confidence score filter is detector-specific: $\theta_{CSF} = [0.03, 0.12, 0.14]$ for BEVFusion [21], FocalsConv [62], CenterPoint [28]. We use $\theta_{mo} = 0.02$, and $\theta_{App} = 1.4$ for all seven classes. Trajectory initialization is consistent with the scheme on the KITTI dataset. We adopt a mixed scheme of count-based and confidence-based [63] to discard trajectories, which means that a trajectory will be discarded when it has not been updated in the last 15 frames or the tracking score falls down the deletion threshold ($\theta_{del} = 0$). We output the motion model's updated trajectory state for matching frames in the trajectory.

We complete the trajectory state for each mismatch period and output up to 2 frames of the trajectory state predicted by the motion model during each mismatch period. Similar to SimpleTrack [64], the tracking score of the predicted state is equal to 0.05 times the score of the most recently updated trajectory state. We apply NMS with $\theta_{nms} = 0.08$ to reduce false positives to all output trajectory states.

B. Comparative Evaluation

1) KITTI 2D MOT: A series of experiments are implemented on the KITTI tracking benchmark test set to evaluate our proposed CAMO-MOT algorithm, the results of which are provided in Tables I and II. As shown, our method ranks first place and surpasses state-of-the-art tracking methods with remarkable margins on the KITTI tracking benchmark test set.

Our proposed method compares with EagerMOT [9], the state-of-the-art multi-modal algorithm currently, and we adopt the identical detector to ensure competitive balance. Our method achieves a HOTA of 79.95%, a MOTA of 90.38%, and an IDS of 23 for the Car class, as shown in Table I. It also achieves a HOTA of 44.90%, a MOTA of 52.19%, and an IDS of 137 for the Pedestrian class, as shown in Table II. These represent HOTA increases of +5.56% and +5.52% and MOTA increases of +2.56% and +2.37% for the Car and Pedestrian classes, respectively, compared with EagerMOT [9]. These results indicate our method offers strong tracking capabilities, achieving the lowest IDS values among all methods. This suggests we can achieve stable tracking with less mismatching, dramatically improving reliability.

2) KITTI 3D MOT: Following the evaluation protocol by [18], we provide the 3D MOT results of our method on the KITTI dataset. As very few methods give the 3D MOT results, we our method compare with AB3DMOT [18], as shown in Table III. Our method achieves sAMOTA increases of +2.01% for the Car class and +13.98% for the Pedestrian class, illustrating the remarkable performance of our method.

3) nuScenes 3D MOT: We also evaluate our CAMO-MOT method on the nuScenes tracking benchmark test set. The results are shown in Table V, in which most state-of-the-art MOT methods are given. Our CAMO-MOT achieves the AMOTA of 75.3% and **ranks first** among all algorithms on the nuScenes tracking benchmark test set. On the test set, we fuse BEVFusion [21] and FocalsConv [62] according to the detection class and use the result as the 3D detector. Despite employing a lower detector than BEVFusion, we achieve an improved performance (+1.20% AMOTA than the second BEVFusion method).

Detector performance can also directly affect the tracking-by-detection framework. The universality of our CAMO-MOT method for different detectors is verified using a series of experiments involving multiple different detectors on both KITTI and nuScenes datasets, as shown in Table IV and Table VI. On the KITTI tracking validation set, we select four famous detectors, including SECOND [13], PVRCNN [78], PointRCNN [11] and PointGNN [79], to evaluate our method. Table IV shows that our method can achieve certain tracking performances on all detectors. When using the detected results from PointGNN [79], our method achieves the best accuracy with a HOTA of 80.74% for the Car class and 50.24% for the Pedestrian class. On the nuScenes tracking validation set, we employ multiple 3D detectors including CenterPoint [28], BEVFuison [21], FocalsConv [62] to test, as shown in Table VI. We also show some other methods to compare. Using the same detector (CenterPoint) as other methods, our method maintains strong performance in tracking. When using the fusion of BEVFuison [21] and FocalsConv [62] as the detector, our method can also achieve the best accuracy with 76.3% AMOTA. In addition, our method obtains minimal IDS without losing AMOTA precision, which illustrates that our method can prevent false matches to a large extent. These experiments effectively verify that our method can adapt to different detectors and demonstrate its universality.

C. Run-Time Discussion

Real-time performance is one of the most essential indicators for evaluating an algorithm. As shown in Table VII, we conduct a thorough analysis of the algorithm's execution time across several implementations.

On the KITTI dataset, the fusion method with occlusion head (AP+MO+OCC) runs at about 25FPS, which is faster than the majority of multi-modal front-end fusion algorithms, such as MmMOT [8] (4FPS), JRMOT (14FPS). However, it is inferior to EagerMOT [9] (90FPS) and DeepFusion-MOT [65] (110FPS), two multi-modal back-end fusion algorithms. On the test set, however, our fusion method outperforms EagerMOT and DeepFusion-MOT in HOTA accuracy by +5.56% and +4.49% for the car class, respectively. In addition, the LiDAR (MO)-only algorithm can run at 100FPS, on par with EagerMOT and DeepFusion-MOT in terms of speed, and on the validation set, it achieves HOTA accuracy comparable to the highest described in their works (−0.94% compared to DeepFusion-MOT [65] for the car class).

In addition, on the KITTI dataset, we examine the effect of the occluded head (OCC) on the algorithm's real-time performance. The fusion method with occlusion head (AP+MO+OCC) is indeed 3.2ms slower than the original fusion algorithm (AP+MO). Considering the accuracy boost that the occlusion head gives to the algorithm (+1.46% increase in HOTA), these time losses are relatively reasonable.

D. Ablation Studies

Ablation studies are also implemented to verify the effects of each modality on our proposed CAMO-MOT algorithm. Experiments involve images, point clouds, fusion methods, and fusion with an occlusion head applied to the KITTI validation set, and the results are shown in Table VII. In the case of the Car class, the fusion method (AP+MO) achieves a HOTA of +2.26%, a MOTA of +0.33%, and an IDS of −46 compared with the algorithm relying solely on point clouds (MO). It also achieves a HOTA of +19.13%, a MOTA of +11.07%, and an IDS of -17 compared with the algorithm relying solely on images (AP). Adding an occlusion head (AP+MO+OCC) produces a HOTA of +1.46%, a MOTA of +0.71%, and an IDS of −16 compared with the original

TABLE I

A COMPARISON OF EXISTING ALGORITHMS APPLIED TO THE KITTI TRACKING BENCHMARK TEST SET (CAR CLASS). METHODS UTILIZING THE SAME DETECTOR (POINTGNN) ARE LABELED BY "*", WHILE "DELTA" PROVIDES A COMPARISON BETWEEN OUR METHOD AND EAGERMOT [9]

Method	Modality	HOTA↑	MOTA↑	MOTP↑	MT↑	FN↓	IDS↓	ML↓	FP↓
JMODT [10]	LiDAR&Camera	70.73	85.35	85.37	77.39	1249	350	2.92	3438
EagerMOT* [9]	LiDAR&Camera	74.39	87.82	85.69	76.15	454	239	2.46	3497
DeepFusion-MOT [65]	LiDAR&Camera	75.46	84.63	85.02	68.61	601	84	9.08	4601
YONTD-MOT [66]	LiDAR&Camera	78.08	85.09	**86.98**	67.54	1188	42	7.08	3899
MSA-MOT [67]	LiDAR&Camera	78.52	88.01	85.45	86.77	2060	91	**1.23**	1974
LGM [16]	Camera	73.14	87.60	84.12	85.08	1568	448	2.46	**2249**
mono3DT [16]	Camera	73.16	84.28	85.45	73.08	745	379	2.92	4282
TripletTrack [68]	Camera	73.58	84.32	86.06	69.85	430	322	3.85	4642
DEFT [17]	Camera	74.23	88.38	84.46	84.31	1006	344	2.15	2647
Mono_3D_KF [69]	Camera	75.47	88.48	83.70	80.61	1045	162	4.15	2754
OC-SORT [31]	Camera	76.54	90.28	85.53	80.00	407	250	3.08	2685
PermaTrack [70]	Camera	78.03	**91.33**	85.65	**85.69**	**402**	258	2.62	2320
PC3T [6]	LiDAR	77.80	88.81	84.26	80.00	814	225	8.46	2810
AB3DMOT [18]	LiDAR	69.81	83.49	85.17	67.08	1060	126	11.38	4492
Ours*	LiDAR&Camera	**79.95**	90.38	85.00	84.61	962	**23**	7.38	2322
Delta	LiDAR&Camera	+5.56	+2.56	-0.69	+8.46	+508	-216	+4.92	-1175

TABLE II

A COMPARISON OF EXISTING ALGORITHMS ON THE KITTI TRACKING BENCHMARK TEST SET (PEDESTRIAN CLASS). METHODS USING THE SAME DETECTOR (POINTGNN) ARE LABELED BY "*", AND "DELTA" IS THE COMPARISON BETWEEN OUR METHOD AND EAGERMOT [9]

Method	Modality	HOTA↑	MOTA↑	MOTP↑	MT↑	FN↓	IDS↓	ML↓	FP↓
MSA-MOT [67]	LiDAR&Camera	44.73	47.86	64.35	33.68	4101	209	16.15	7761
EagerMOT* [9]	LiDAR&Camera	39.38	49.82	64.42	27.49	2161	496	24.05	8959
YONTD-MOT [66]	LiDAR&Camera	25.89	26.19	65.66	11.00	2882	1068	31.96	13137
TrackMPNN [71]	Camera	39.40	52.10	73.42	35.05	2758	626	18.90	7705
JCSTD [72]	Camera	39.44	43.42	71.72	18.56	**885**	236	34.36	11976
CenterTrack [73]	Camera	40.35	53.84	73.72	35.40	2201	425	21.31	8061
QD-3DT [74]	Camera	41.08	51.77	73.13	32.65	2084	717	19.24	8364
Quasi-Dense [75]	Camera	41.12	**55.55**	**73.70**	31.27	1309	487	19.24	8364
MDP [76]	Camera	42.76	47.02	70.17	25.77	2502	213	28.52	9550
TripletTrack [68]	Camera	42.77	50.08	73.84	24.05	798	323	29.21	10436
Mono_3D_KF [69]	Camera	42.87	45.44	69.06	33.68	3498	267	26.46	8865
NC2 [77]	LiDAR	44.30	44.18	65.68	**44.33**	6415	332	**13.06**	6176
AB3DMOT [18]	LiDAR	35.57	38.93	64.55	17.18	2135	259	41.24	11744
Ours*	LiDAR&Camera	**44.90**	52.19	64.50	35.40	2560	**137**	25.43	**1112**
Delta	LiDAR&Camera	+5.52	+2.37	+0.08	+7.91	+399	-359	+1.38	-7847

TABLE III

3D MOT EVALUATION ON THE KITTI VALIDATION SET. "DELTA" IS THE COMPARISON BETWEEN OUR METHOD AND AB3DMOT [18]. (AMOTA (AVERAGE MULTI-OBJECT TRACKING ACCURACY), AMOTP (AVERAGE MULTI-OBJECT TRACKING PRECISION), SAMOTA (SCALED AMOTA)

Method	Type	sAMOTA↑	AMOTA↑	AMOTP↑
AB3DMOT [18]	Car	93.28	45.43	77.41
Ours	Car	**95.29**	**48.04**	**81.48**
Delta	Car	+2.01	+2.61	+4.07
AB3DMOT [18]	Ped.	75.85	31.04	55.53
Ours	Ped.	**89.83**	**44.84**	**72.55**
Delta	Ped.	+13.98	+13.80	+17.02

TABLE IV

A UNIVERSALITY ASSESSMENT FOR OUR PROPOSED METHOD APPLIED TO THE VALIDATION SET IN THE KITTI TRACKING BENCHMARK

Detector	Type	HOTA↑	MOTA↑	IDS↓	FP↓	FN↓
SECOND [13]	Car	75.71	80.02	12	**237**	1425
PVRCNN [78]	Car	79.27	87.03	10	589	588
PointRCNN [11]	Car	77.99	86.31	9	680	558
PointGNN [79]	Car	**80.74**	**87.78**	**6**	544	**474**
SECOND [13]	Ped.	26.19	17.31	**8**	**130**	7955
PVRCNN [78]	Ped.	31.81	23.49	13	331	7144
PointRCNN [11]	Ped.	45.65	61.54	95	887	2792
PointGNN [79]	Ped.	**50.24**	**57.19**	88	2850	**1256**

fusion method (AP+MO). In the Pedestrian class, the fusion method (AP+MO) achieves a HOTA of +1.49%, a MOTA of +0.14%, and an IDS of −4 compared with point clouds (MO) and achieves a HOTA of +13.69%, a MOTA of +14.38%, and an IDS of -52 compared with images (AP). Adding an occlusion head (AP+MO+OCC) produces a HOTA of

TABLE V
A COMPARISON OF EXISTING ALGORITHMS APPLIED TO THE NUSCENES TRACKING BENCHMARK TEST SET

Method	Detector	Modality	AMOTA↑	FP↓	FN↓	IDS↓
SimpleTrack [64]	CenterPoint [28]	LiDAR	66.8	17514	23451	575
CenterPoint [28]	CenterPoint [28]	LiDAR	65.0	17355	24557	684
OGR3MOT [80]	CenterPoint [28]	LiDAR	65.6	17877	24013	**288**
TransFusion [81]	TransFuison [81]	LiDAR&Camera	71.8	16232	21846	944
BEVFusion [21]	BEVFusion-e [21]	LiDAR&Camera	74.1	19997	19395	506
EagerMOT [9]	CenterPoint [28]&Cascade R-CNN [82]	LiDAR&Camera	67.7	17705	24925	1156
CBMOT [63]	CenterPoint [28]&CenterTrack [73]	LiDAR&Camera	68.1	21604	22828	709
Ours	BEVFuison [21]&FocalsConv [62]	LiDAR&Camera	**75.3**	**17269**	**18192**	324

TABLE VI
A COMPARISON OF EXISTING ALGORITHMS APPLIED TO THE NUSCENES TRACKING BENCHMARK VAL SET. THE NUMBERS MARKED WITH [64] ARE FROM SIMPLETRACK [64]

Method	Detector	Modality	AMOTA↑	AMOTP↓	IDS↓
AB3DMOT [18], [64]	CenterPoint [28]	LiDAR	59.8	77.1	1570
SimpleTrack [64]	CenterPoint [28]	LiDAR	69.6	54.7	405
CenterPoint [28]	CenterPoint [28]	LiDAR	66.5	56.7	562
OGR3MOT [80]	CenterPoint [28]	LiDAR	69.3	62.7	262
CBMOT [63]	CenterPoint [28]&CenterTrack [73]	LiDAR&Camera	69.2	56.3	459
EagerMOT [9]	CenterPoint [28]&Cascade R-CNN [82]	LiDAR&Camera	71.2	56.9	899
Ours	CenterPoint [28]	LiDAR&Camera	71.9	**52.1**	393
Ours	BEVFuison [21]	LiDAR&Camera	76.0	56.1	243
Ours	FocalsConv [62]	LiDAR&Camera	75.3	52.7	337
Ours	BEVFuison [21]&FocalsConv [62]	LiDAR&Camera	**76.3**	52.7	**239**

TABLE VII
ABLATION STUDIES WITH VARYING OCCLUSION STATES AND APPEARANCE FEATURES IN THE KITTI TRACKING BENCHMARK. AP REFERS TO THE APPEARANCE MODULE; MO REFERS TO THE MOTION MODULE; OCC REFERS TO THE OCCLUSION MODULE; L REFERS TO LIDAR; C REFERS TO CAMERA

Type	Module	Modality	HOTA↑	MOTA↑	IDS↓	FPS↑
Car	AP	C	60.15	76.00	39	33
	MO	L	77.02	86.74	68	**100**
	AP+MO	L&C	79.28	87.07	22	28
	AP+MO+OCC	L&C	**80.74**	**87.78**	**6**	25
Ped.	AP	C	36.07	42.49	148	33
	MO	L	48.27	56.73	100	**100**
	AP+MO	L&C	49.76	56.87	96	28
	AP+MO+OCC	L&C	**50.24**	**57.19**	**88**	25

TABLE VIII
THE RESULTS OF ABLATION STUDIES INVOLVING APPEARANCE FEATURES IN THE KITTI VALIDATION SET. COMPARED VARIABLES INCLUDE THE OPTIMAL OCCLUSION (OCC), FEATURES IN THE LAST THREE FRAMES (LTF), AND THE METHOD FOR WEIGHTED JUDGING BY OCCLUSION (LTF+OCC)

Variable	Type	HOTA↑	MOTA↑	MOTP↑	IDS↓	FP↓	FN↓
OCC		**80.74**	**87.78**	**87.99**	**6**	**544**	**474**
LTF	Car	79.28	87.07	87.96	22	544	474
LTF+OCC		80.71	87.77	87.99	7	544	474
OCC		**50.24**	**57.19**	**66.64**	**88**	**2850**	**1256**
LTF	Ped.	49.76	56.89	66.62	96	2850	1256
LTF+OCC		50.23	57.15	66.62	88	2850	1256

+0.48%, a MOTA of +0.32%, and an IDS of −8 compared with the original fusion method (AP+MO). These experiments show that our fusion method achieves the best results and validates the role of the proposed occlusion head branch.

To verify that our chosen optimal occlusion features are valid, we separately calculate (1) the features based on optimal occlusion situations (OCC), (2) the average value of appearance features for the last three frames (LTF), (3) the weighted accumulation of appearance features for the last three frames according to the occlusion situation, as shown in Table VIII. The resulting occlusion weights are 1, 0.7, 0.3, and 0 according to the occlusion state (fully visible, partly occluded, largely occluded, fully occluded), respectively. It is evident that tracking effects are best in the case of OCC. It can be seen from the results that whether OCC or LTF+OCC, it is better than LTF, indicating that the occlusion head plays an important role in our method. When introducing the object appearance of multiple frames, interference of different degrees will be introduced due to the different occlusion states, so that the final appearance features of the object cannot fully represent the object itself, and the performance of LTF+OCC is not as good as that of OCC.

We also test algorithm performance under the two association rules of appearance followed by movement and then movement followed by appearance, as shown in Table IX. This MO→AP and AP→MO approach produces HOTA values of +0.35% and +0.55%, MOTA values of +0.18% and +0.38% and IDS values of −15 and −27 for the Car and Pedestrian categories, respectively. We analyze that the possible reason is that when there are objects with very similar appearances in the scene, an identity switch is likely to occur, resulting in a decrease in the metrics.

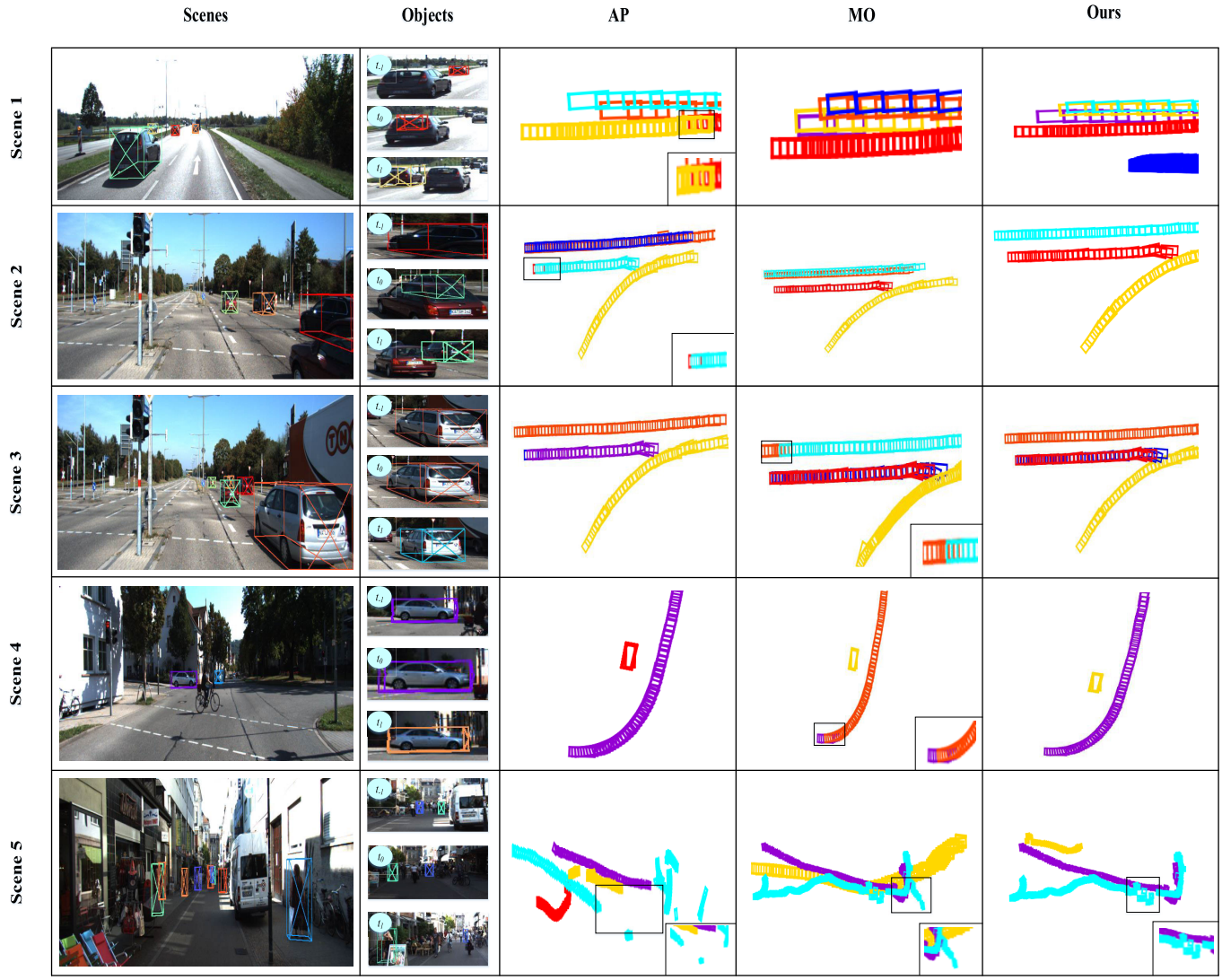


Fig. 4. Qualitative visualization of object trajectories with single and fusion modalities. AP and MO refer to the appearance module and motion module, respectively, while the "Scenes" and "Objects" columns provide corresponding image scenes and detected objects. The t_{-1}, t_0, and t_1 terms denote the previous, current, and next moments, respectively. The third, fourth, and fifth columns present the 2D object trajectories and colors mean the different detected objects. Our method achieves the lowest ID switches.

TABLE IX

THE RESULTS OF ABLATION STUDIES INVOLVING ASSOCIATION RULES. AP REFERS TO THE APPEARANCE MODULE AND MO REFERS TO THE MOTION MODULE

Variable	Type	HOTA↑	MOTA↑	MOTP↑	IDS↓	FP↓	FN↓
AP→ MO	Car	80.39	87.60	87.98	21	544	474
MO→ AP		**80.74**	**87.78**	**87.99**	**6**	**544**	**474**
AP→ MO	Ped.	49.69	56.81	66.59	115	2853	1259
MO→ AP		**50.24**	**57.19**	**66.64**	**88**	**2850**	**1256**

TABLE X

THE RESULTS OF ABLATION STUDIES INVOLVING CATEGORY COST. NCL MEANS NO CATEGORY COST AND CL MEANS CATEGORY COST

Variable	Type	HOTA↑	MOTA↑	MOTP↑	IDS↓	FP↓	FN↓
NCL	Car	79.81	87.70	87.99	13	544	474
CL		**80.74**	**87.78**	**87.99**	**6**	**544**	**474**
NCL	Ped	49.99	56.76	66.55	114	2853	1259
CL		**50.24**	**57.19**	**66.64**	**88**	**2850**	**1256**

The effectiveness of category cost during tracking is also evaluated experimentally. As shown in Table X, the HOTA of Car and Pedestrian classes in the method with category cost (CL) increases by +0.93% and +0.25% than the one with no category cost (NCL). This suggests our proposed method eliminates the possibility of associations between categories effectively, significantly increasing the tracking accuracy for Car and Pedestrian classes.

E. Visualization

The effectiveness of the fusion method is further evaluated using several qualitative visualization experiments on the KITTI dataset.

The results of a tracking method using single and fusion modalities are provided in Fig. 4. In scenes 1 and 2, the tracked object is considered a new entity due to occlusions in the movement process. However, our method allows for stable tracking without occlusion effects. In scene 3, the tracker

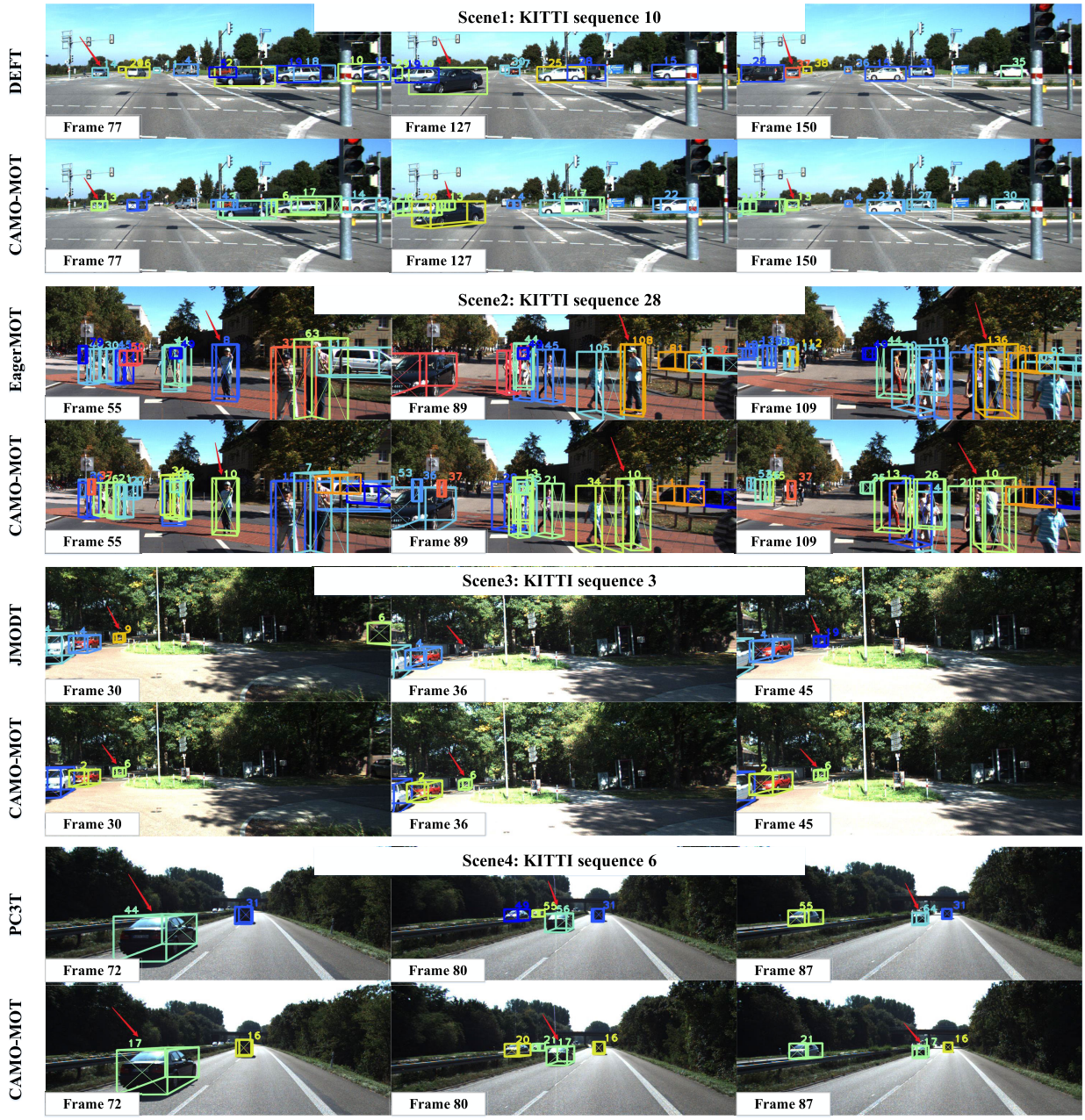


Fig. 5. Qualitative results comparison between CAMO-MOT and DEFT [17], EagerMOT [9], JMODT [10], PC3T [6] on KITTI tracking benchmark. Each pair of rows shows the comparison of the results for one sequence. The color of the boxes represents the identity of the tracks. Red arrows point at tracking failures (identity switches). Under occlusion situations, wrong associations occur: DEFT (ID 14→ID 37), EagerMOT (ID 8→ID 136). When the illumination changes, JMODT shows the wrong ID switch (ID 9→ID 19). When the object is accelerating, ID 44→ID 64 occurs in PC3T. However, our CAMO-MOT can still keep stable tracking in all these situations.

considers the object a new object when it suddenly accelerates. In scene 4, the object is considered new because of a sudden turn. None of these effects are observed using our method. In scene 5, the object moves irregularly and is also occluded by other objects. Although our method can not resolve this situation entirely, it provides obvious improvement compared with single modalities.

We also provide qualitative results comparison between our CAMO-MOT and DEFT [17], EagerMOT [9], JMODT [10], PC3T [6] on KITTI tracking benchmark, as shown in Fig. 5. Compared with DEFT and EagerMOT in occlusion, they both have identity switches, but our method can achieve stable tracking. PC3T utilizes only point cloud data and also has identity switches when the object continuously accelerates, but our method is still stable. When illumination changes, the multi-modal fusion method JMODT recognizes the object as a new one while our method still maintains stable tracking.

The above comparisons illustrate that our method can utilize the information of the camera and LiDAR to achieve stable tracking effectively, even in severe occlusions.

V. Conclusion

A new multi-modal 3D MOT framework CAMO-MOT based on the combined appearance-motion optimization is

proposed to fuse camera and LiDAR information and achieve stable tracking effectively. An occlusion head is designed to identify object occlusion states and select optimal appearance features to reduce the effects of occlusions. We also propose a 3D motion module based on confidence scores to eliminate false detections. This study represents the first attempt at introducing category cost to ensure objects are only related within the same category. Our method achieves the state-of-the-art performance among multi-modal MOT algorithms applied to the KITTI tracking benchmark and the state-of-the-art performance among all MOT algorithms applied to the nuScenes tracking benchmark. In addition, it achieves the lowest IDS among all algorithms on the KITTI test set and the top ten algorithms on the nuScenes test set, indicating remarkable safety and stability.

ACKNOWLEDGMENT

The authors would like to thank LetPub (www.letpub.com) for linguistic assistance and pre-submission expert review.

REFERENCES

[1] S. Xu et al., "System and experiments of model-driven motion planning and control for autonomous vehicles," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 9, pp. 5975–5988, Sep. 2022.

[2] Z. Cao et al., "Highway exiting planner for automated vehicles using reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 990–1000, Feb. 2021.

[3] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3464–3468.

[4] K. Fang, Y. Xiang, X. Li, and S. Savarese, "Recurrent autoregressive networks for online multi-object tracking," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 466–475.

[5] V. Vaquero, I. del Pino, F. Moreno-Noguer, J. Solà, A. Sanfeliu, and J. Andrade-Cetto, "Dual-branch CNNs for vehicle detection and tracking on LiDAR data," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 6942–6953, Nov. 2021.

[6] H. Wu, W. Han, C. Wen, X. Li, and C. Wang, "3D multi-object tracking in point clouds based on prediction confidence-guided data association," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5668–5677, Jun. 2022.

[7] H. Hu et al., "Joint monocular 3D vehicle detection and tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5389–5398.

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

[9] A. Kim, A. Ošep, and L. Leal-Taixé, "EagerMOT: 3D multi-object tracking via sensor fusion," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 11315–11321.

[10] K. Huang and Q. Hao, "Joint multi-object detection and tracking with camera-LiDAR fusion for autonomous driving," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 6983–6989.

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

[12] T. Xie et al., "Poly-PC: A polyhedral network for multiple point cloud tasks at once," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 1233–1243.

[13] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12689–12697.

[14] T. Xie et al., "FARP-Net: Local-global feature aggregation and relation-aware proposals for 3D object detection," *IEEE Trans. Multimedia*, early access, May 11, 2023, doi: 10.1109/TMM.2023.3275366.

[15] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 658–666.

[16] G. Wang, R. Gu, Z. Liu, W. Hu, M. Song, and J. Hwang, "Track without appearance: Learn box and tracklet embedding with local and global motion patterns for vehicle tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9856–9866.

[17] M. Chaabane, P. Zhang, J. R. Beveridge, and S. O'Hara, "DEFT: Detection embeddings for tracking," 2021, *arXiv:2102.02267*.

[18] X. Weng, J. Wang, D. Held, and K. Kitani, "3D multi-object tracking: A baseline and new evaluation metrics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 10359–10366.

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

[20] H. Caesar et al., "NuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11618–11628.

[21] Z. Liu et al., "BEVFusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," 2022, *arXiv:2205.13542*.

[22] J. Shen, Y. Liu, X. Dong, X. Lu, F. S. Khan, and S. Hoi, "Distilled Siamese networks for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 8896–8909, Dec. 2022.

[23] X. Dong, J. Shen, D. Wu, K. Guo, X. Jin, and F. Porikli, "Quadruplet network with one-shot learning for fast visual object tracking," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3516–3527, Jul. 2019.

[24] X. Dong, J. Shen, W. Wang, L. Shao, H. Ling, and F. Porikli, "Dynamical hyperparameter optimization via deep reinforcement learning in tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1515–1529, May 2021.

[25] J. Shen, X. Tang, X. Dong, and L. Shao, "Visual object tracking by hierarchical attention Siamese network," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3068–3080, Jul. 2020.

[26] Y. Zhang, B. Ma, J. Wu, L. Huang, and J. Shen, "Capturing relevant context for visual tracking," *IEEE Trans. Multimedia*, vol. 23, pp. 4232–4244, 2021.

[27] J. Shen, D. Yu, L. Deng, and X. Dong, "Fast online tracking with detection refinement," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 162–173, Jan. 2018.

[28] T. Yin, X. Zhou, and P. Krähenbühl, "Center-based 3D object detection and tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11779–11788.

[29] J. Shen, Z. Liang, J. Liu, H. Sun, L. Shao, and D. Tao, "Multiobject tracking by submodular optimization," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 1990–2001, Jun. 2019.

[30] X. Dong, J. Shen, D. Yu, W. Wang, J. Liu, and H. Huang, "Occlusion-aware real-time object tracking," *IEEE Trans. Multimedia*, vol. 19, no. 4, pp. 763–771, Apr. 2017.

[31] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, "Observation-centric SORT: Rethinking SORT for robust multi-object tracking," 2022, *arXiv:2203.14360*.

[32] Y. H. Khalil and H. T. Mouftah, "LiCaNet: Further enhancement of joint perception and motion prediction based on multi-modal fusion," *IEEE Open J. Intell. Transp. Syst.*, vol. 3, pp. 222–235, 2022.

[33] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3701–3710.

[34] P. Bergmann, T. Meinhardt, and L. Leal-Taixé, "Tracking without bells and whistles," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 941–951.

[35] Y. Li, A. Hilton, and J. Illingworth, "Towards reliable real-time multiview tracking," in *Proc. IEEE Workshop Multi-Object Tracking*, Jul. 2001, pp. 43–50.

[36] Z. Liang and J. Shen, "Local semantic Siamese networks for fast tracking," *IEEE Trans. Image Process.*, vol. 29, pp. 3351–3364, 2020.

[37] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3645–3649.

[38] S. Sun, N. Akhtar, H. Song, A. Mian, and M. Shah, "Deep affinity network for multiple object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 104–119, Jan. 2021.

[39] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[40] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "FairMOT: On the fairness of detection and re-identification in multiple object tracking," *Int. J. Comput. Vis.*, vol. 129, no. 11, pp. 3069–3087, Sep. 2021, doi: 10.1007/s11263-021-01513-4.

[41] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.07850*.

[42] Y. Zhang et al., "ByteTrack: Multi-object tracking by associating every detection box," 2021, *arXiv:2110.06864*.

[43] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.

[44] P. Sun et al., "TransTrack: Multiple object tracking with transformer," 2020, *arXiv:2012.15460*.

[45] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," 2020, *arXiv:2010.04159*.

[46] F. Zeng, B. Dong, Y. Zhang, T. Wang, X. Zhang, and Y. Wei, "MOTR: End-to-end multiple-object tracking with transformer," 2021, *arXiv:2105.03247*.

[47] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," 2016, *arXiv:1603.00831*.

[48] J. Yin, W. Wang, Q. Meng, R. Yang, and J. Shen, "A unified object motion and affinity model for online multi-object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 6768–6777.

[49] S. Scheidegger, J. Benjaminsson, E. Rosenberg, A. Krishnan, and K. Granström, "Mono-camera 3D multi-object tracking using deep learning detections and PMBM filtering," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 433–440.

[50] Y. Wang, K. Kitani, and X. Weng, "Joint object detection and multi-object tracking with graph neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 13708–13715.

[51] A. Shenoi et al., "JRMOT: A real-time 3D multi-object tracker and a new large-scale dataset," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 10335–10342.

[52] M. Sualeh and G.-W. Kim, "Dynamic multi-LiDAR based multiple object detection and tracking," *Sensors*, vol. 19, no. 6, p. 1474, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/6/1474

[53] M. P. Muresan and S. Nedevschi, "Multi-object tracking of 3D cuboids using aggregated features," in *Proc. IEEE 15th Int. Conf. Intell. Comput. Commun. Process. (ICCP)*, Sep. 2019, pp. 11–18.

[54] J. Choi, S. Ulbrich, B. Lichte, and M. Maurer, "Multi-target tracking using a 3D-LiDAR sensor for autonomous vehicles," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 881–886.

[55] S. Song, Z. Xiang, and J. Liu, "Object tracking with 3D LiDAR via multi-task sparse learning," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Aug. 2015, pp. 2603–2608.

[56] Y. Li et al., "DeepFusion: LiDAR-camera deep fusion for multi-modal 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 17161–17170.

[57] M. Liang et al., "PnPNet: End-to-end perception and prediction with tracking in the loop," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11550–11559.

[58] M. Chaabane, L. Gueguen, A. Trabelsi, R. Beveridge, and S. O'Hara, "End-to-end learning improves static object geo-localization in monocular video," 2021, *arXiv:2004.05232*.

[59] J. Luiten et al., "HOTA: A higher order metric for evaluating multi-object tracking," *Int. J. Comput. Vis.*, vol. 129, no. 2, pp. 548–578, Oct. 2020, doi: 10.1007/s11263-020-01375-2.

[60] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image Video Process.*, vol. 2008, pp. 1–10, 2008.

[61] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017, *arXiv:1412.6980*.

[62] Y. Chen, Y. Li, X. Zhang, J. Sun, and J. Jia, "Focal sparse convolutional networks for 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 5428–5437.

[63] N. Benbarka, J. Schröder, and A. Zell, "Score refinement for confidence-based 3D multi-object tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Jun. 2021, pp. 8083–8090.

[64] Z. Pang, Z. Li, and N. Wang, "SimpleTrack: Understanding and rethinking 3D multi-object tracking," 2021, *arXiv:2111.09621*.

[65] X. Wang, C. Fu, Z. Li, Y. Lai, and J. He, "DeepFusionMOT: A 3D multi-object tracking framework based on camera-LiDAR fusion with deep association," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 8260–8267, Jul. 2022.

[66] X. Wang, J. He, C. Fu, T. Meng, and M. Huang, "You only need two detectors to achieve multi-modal 3D multi-object tracking," 2023, *arXiv:2304.08709*.

[67] Z. Zhu, J. Nie, H. Wu, Z. He, and M. Gao, "MSA-MOT: Multi-stage association for 3D multimodality multi-object tracking," *Sensors*, vol. 22, no. 22, p. 8650, Nov. 2022.

[68] N. Marinello, M. Proesmans, and L. Van Gool, "TripletTrack: 3D object tracking using triplet embeddings and LSTM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 4499–4509.

[69] A. Reich and H.-J. Wuensche, "Monocular 3D multi-object tracking with an EKF approach for long-term stable tracks," in *Proc. IEEE 24th Int. Conf. Inf. Fusion (FUSION)*, Nov. 2021, pp. 1–7.

[70] P. Tokmakov, J. Li, W. Burgard, and A. Gaidon, "Learning to track with object permanence," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10840–10849.

[71] 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, *arXiv:2101.04206*.

[72] W. Tian, M. Lauer, and L. Chen, "Online multi-object tracking using joint domain information in traffic scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 374–384, Jan. 2020.

[73] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," 2020, *arXiv:2004.01177*.

[74] H. Hu, Y. Yang, T. Fischer, T. Darrell, F. Yu, and M. Sun, "Monocular quasi-dense 3D object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 1992–2008, Feb. 2023.

[75] J. Pang et al., "Quasi-dense similarity learning for multiple object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 164–173.

[76] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4705–4713.

[77] C. Jiang, Z. Wang, S. Tan, and H. Liang, "A new adaptive noise covariance matrices estimation and filtering method: Application to multi-object tracking," 2021, *arXiv:2112.12082*.

[78] S. Shi et al., "PV-RCNN: Point-voxel feature set abstraction for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10526–10535.

[79] W. Shi and R. Rajkumar, "Point-GNN: Graph neural network for 3D object detection in a point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1708–1716.

[80] J. Zaech, A. Liniger, D. Dai, M. Danelljan, and L. Van Gool, "Learnable online graph representations for 3D multi-object tracking," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 5103–5110, Apr. 2022.

[81] X. Bai et al., "TransFusion: Robust LiDAR-camera fusion for 3D object detection with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1080–1089.

[82] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.

Li Wang was born in Shangqiu, Henan, China, in 1990. He received the Ph.D. degree in mechatronic engineering from the State Key Laboratory of Robotics and System, Harbin Institute of Technology, in 2020.

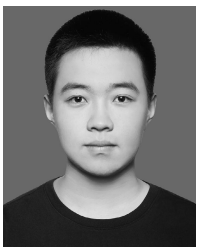
He was a Visiting Scholar with the Nanyang Technology of University, Singapore, for two years. He is currently a Post-Doctoral Fellow with the State Key Laboratory of Automotive Safety and Energy and the School of Vehicle and Mobility, Tsinghua University. He is the author of more than 30 SCI/EI articles. His research interests include autonomous driving perception, 3D robot vision, and multi-modal fusion.

Xinyu Zhang (Member, IEEE) was born in Huining, Gansu, China. He received the B.E. degree from the School of Vehicle and Mobility, Tsinghua University, in 2001.

He was a Visiting Scholar with the University of Cambridge. He is currently a Researcher with the School of Vehicle and Mobility and the Head of the Mengshi Intelligent Vehicle Team, Tsinghua University. He is the author of more than 30 SCI/EI articles. His research interests include intelligent driving and multimodal information fusion.

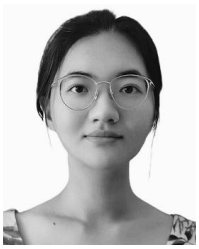
Wenyuan Qin was born in Lüliang, Shanxi, China, in 1996. He received the bachelor's degree in measurement and control technology and instruments from the Taiyuan University of Technology, China, in 2019. He is currently pursuing the joint training master's degree in vehicle engineering with the Beijing Institute of Technology, China, with a focus on autonomous driving.

Since April. 2021, he has been interning with the Autonomous Driving Laboratory, Tsinghua University, responsible for the development of deep learning based on 3D multi-object tracking for smart vehicles.



Xiaoyu Li was born in Shenyang, Liaoning, China, in 2000. He received the bachelor's degree from the Faculty of Robot Science and Engineering, Northeastern University, China, in 2022. He is currently pursuing the master's degree in mechatronic engineering with the State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, with a focus on autonomous driving.

Since December. 2021, he has been interning with the Autonomous Driving Laboratory, Tsinghua University, responsible for the development of deep learning based on 3D multi-object tracking for smart vehicles.



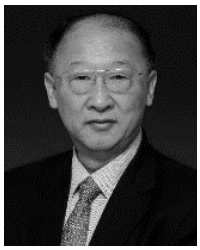
Jinghan Gao was born in Harbin, Heilongjiang, China. She received the bachelor's degree in engineering from Jilin University in 2020. She is currently pursuing the master's degree in mechatronics engineering with the State Key Laboratory for Robotics and Systems, Harbin Institute of Technology. Her research interests include autonomous driving. Since 2022, she has been an Intern with the Autonomous Driving Laboratory, Tsinghua University, where she is responsible for the development of 3D multi-object tracking based on deep learning for smart vehicles.

Lei Yang (Graduate Student Member, IEEE) was born in Datong, Shanxi, China, in 1993. He received the master's degree in robotics from Beihang University in 2018. He is currently pursuing the Ph.D. degree with the School of Vehicle and Mobility, Tsinghua University.

From 2018 to 2020, he was an Algorithm Researcher with the Autonomous Driving Research and Development Department, JD.COM. His research interests include computer vision, autonomous driving, and environmental perception.

Zhiwei Li was born in Lüliang, Shanxi, China. He received the Ph.D. degree from the China University of Mining and Technology, Beijing, in 2020.

He is currently an Assistant Professor with the College of Information Science and Technology, Beijing University of Chemical Technology. His research interests include autonomous driving and multi-modal information fusion. He has been engaged in data processing based on deep learning for more than ten years, published many SCI/EI articles, and participated in many international conferences.



Jun Li was born in Jilin, China, in 1958. He received the Ph.D. degree in internal-combustion engineering from the Jilin University of Technology in 1989.

In 1989, he joined China FAW Group Corporation. He is currently a Professor with the School of Vehicle and Mobility, Tsinghua University. In these years, he has presided over the product development and technological innovation of large-scale automobile companies in China. He is the author of more than 98 articles. He has many scientific research achievements in the fields of automotive powertrain, new energy vehicles, and intelligent connected vehicles. He serves as the Chairperson of the China Society of Automotive Engineers (SAE). In 2013, he was awarded an academician of Chinese Academy of Engineering (CAE) for contributions to vehicle engineering.

Lei Zhu received the M.B.A. degree from Tsinghua University.

He is currently the Founder and the CEO of Mogo Auto Intelligence and Telematics Information Technology Company Ltd., once served as the Chairperson of Baidu Vertical Search Technical Committee, the Vice President of Didi, and the CTO of Bitauto. He is also a Young Creative Talent of the Innovation Research Institute, University of South China. He mainly engaged in the research and development of core technologies, such as onboard OS, AI cloud, and V2X (vehicle to everything) systems, creating the industry-leading "Vehicle-Road-Cloud Integration" autonomous driving solution and achieving the commercialization of autonomous driving through vehicle intelligence, road intelligence, and AI cloud featuring collaboration in a holistic way.

Hong Wang received the Ph.D. degree from the Beijing Institute of Technology, China, in 2015.

From 2015 to 2019, she was a Research Associate of mechanical and mechatronics engineering with the University of Waterloo. She is currently a Research Associate Professor with Tsinghua University. She has published over 40 papers on top international journals, such as IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and IEEE TRANSACTIONS ON MECHATRONICS. Her research interests include the risk assessment and crash mitigation-based decision making during critical driving scenarios, ethical decision making for autonomous vehicles, component sizing, modeling of hybrid powertrains and intelligent control strategies design for hybrid electric vehicles, and intelligent control theory and application. She served as an Associate Editor for the 2019 Intelligent Vehicles Symposium held in Paris, France.

Huaping Liu (Senior Member, IEEE) received the Ph.D. degree in computer science and technology from Tsinghua University, Beijing, China, in 2004.

He is currently an Associate Professor with the Department of Computer Science and Technology, Tsinghua University. His research interests include robot perception and learning. He was a Senior Program Committee Member of the International Joint Conference on Artificial Intelligence in 2018. He was a recipient of the Andy Chi Best Paper Award in 2017. He was the Area Chair for Robotics Science and Systems in 2018.