Delving into Motion-Aware Matching for Monocular 3D Object Tracking

Kuan-Chih Huang[1] Ming-Hsuan Yang[1,2,3] Yi-Hsuan Tsai[2]

[1]University of California, Merced [2]Google [3]Yonsei University

Abstract

Recent advances of monocular 3D object detection facilitate the 3D multi-object tracking task based on low-cost camera sensors. In this paper, we find that the motion cue of objects along different time frames is critical in 3D multi-object tracking, which is less explored in existing monocular-based approaches. To this end, we propose MoMA-M3T, a framework that mainly consists of three motion-aware components. First, we represent the possible movement of an object related to all object tracklets in the feature space as its motion features. Then, we further model the historical object tracklet along the time frame in a spatial-temporal perspective via a motion transformer. Finally, we propose a motion-aware matching module to associate historical object tracklets and current observations as final tracking results. We conduct extensive experiments on the nuScenes and KITTI datasets to demonstrate that our MoMA-M3T achieves competitive performance against state-of-the-art methods. Moreover, the proposed tracker is flexible and can be easily plugged into existing image-based 3D object detectors without re-training. Code and models are available at https://github.com/kuanchihhuang/MoMA-M3T.

1. Introduction

3D Multi-Object Tracking (3D MOT) is a crucial problem for various applications like autonomous driving. Numerous LiDAR-based methods [49, 50] have achieved remarkable results thanks to powerful 3D object detectors [13, 23, 38, 39]. Due to the lower cost of camera sensors, some image-based 3D object detection approaches [4, 16, 20, 26, 32, 35] receive much attention and achieve promising performance, and thus enable 3D object tracking based on merely the camera.

One straightforward approach to deal with monocular 3D object tracking is to match object features in adjacent frames [24, 61] (see Figure 1(a)). Although significant progress has been made, these methods may still fail to capture multi-frame motion information of objects. To tackle the long-range dependency, another line of work [6, 15] predicts the object states from the historical observations, in which the predicted and observed states in the current frame are in the output space that explicitly contains the object information, *e.g.*, location and pose of the object (see Figure 1(b)). However, these approaches may suffer from noisy observations of object states predicted by the inaccurate monocular 3D object detector.

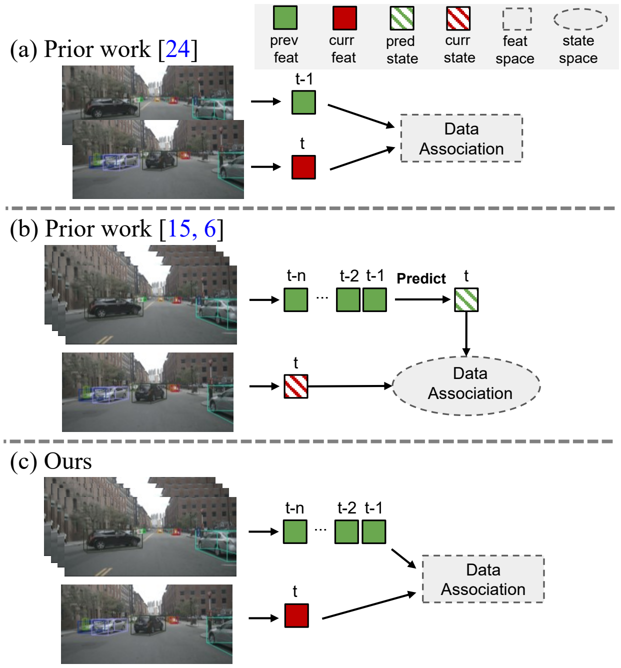

Figure 1. **Comparisons of different association methods in monocular 3D object tracking.** (a) Time3D [24] learns to match 3D object features in adjacent frames. (b) QD-3DT [15] and DEFT [6] utilize the object's previous features to predict their current states, and match with the observations in the output space. (c) Our approach directly aggregates the object's previous features and matches them with current observations in the feature space.

For the above-mentioned methods, one critical step is data association, in which the goal is to match observations across historical time frames and produce final tracking results. Therefore, in monocular 3D MOT, two main challenges are 1) how to obtain the long-range observations that can provide richer information for data association? 2) what are the better representations the algorithm utilizes as observations, in order to mitigate the problem of matching under noisy observations from the inaccurate monocular 3D detector? In this paper, we propose MoMA-M3T, a motion-aware matching approach for monocular 3D MOT, to handle these two challenges (see Figure 1(c)). Our main idea is to encode the multi-frame motion information of the object tracklets, i.e., their historically *relative* positions, into a *feature* space for data association, instead of encoding their *absolute* locations in the *output* space. To this end, the object movements encoded in the learned representations can be used for matching between tracklets and current object observations.

Specifically, MoMA-M3T consists of three main components: 1) we first use a motion encoder to encode the 3D object information, *e.g.*, relative position and object size/heading angle, into a motion-aware feature space; 2) Then, these encoded features also form a motion feature bank to record historical features, followed by a motion transformer module to generate spatial-temporal motion features as representations of object tracklets; 3) Finally, a motion-aware matching module to generate tracking results is introduced for data association between object observations and tracklets based on motion features. Moreover, our method that considers motion features enables the feasibility of applying learning strategies. We adopt a contrastive learning objective that samples several subsets of different object trajectories and learns better feature representations, *e.g.*, data points from the same trajectory but in augmented views are positive samples.

Extensive experiments on nuScenes [5] and KITTI [12] datasets demonstrate that our method achieves state-of-the-art performance based on monocular camera sensors. In addition, we show the benefit of our proposed components, including the usage of motion features, motion transformers, and motion-aware matching. More interestingly, we present the robustness of MoMA-M3T by plugging our learned modules with frozen weights into the same framework, but based on detection outputs from different 3D object detectors. Results show that our motion-aware approach generalizes well to various pre-trained detectors.

The main contributions of this work are as follows:

- We present MoMA-M3T, a framework that introduces motion features with a motion-aware matching mechanism for monocular 3D MOT.
- We propose a motion transformer module that captures the movement of object tracklets in a spatial-temporal perspective, enabling robust motion feature learning.
- Extensive experiments on nuScenes and KITTI datasets show that our method achieves competitive performance based on monocular sensors, with the flexibility to apply various pre-trained 3D detectors.

2. Related Work

Monocular 3D Object Detection. Image-based 3D object detection has gained much attention recently due to the low-cost camera sensors. Numerous approaches [3, 7, 22, 33, 42, 45, 46, 55, 58] perceive 3D objects on the image plane by relying on geometric relationships [40, 41], such as object size [40], keypoints [25, 29], or depth uncertainty [30, 46]. To improve the 3D reasoning ability, several approaches [8, 10, 17, 47, 64] leverage depth information to facilitate object detection. In addition, some works [16, 26, 28, 48] focuses on developing multi-camera 3D object detection systems. These methods learn the bird-eye view representations of the surrounding scenarios by fusing information from multiple cameras. Instead of designing a powerful monocular 3D detector, our work targets on establishing a robust motion tracker to associate noisy monocular 3D observations.

Multi-Object Tracking. With the rapid advances of object detection, Multi-object tracking (MOT) [43, 53, 59, 61, 63] has been extensively explored in the 2D image space. Most state-of-the-art approaches adopt the tracking-by-detection paradigm [2, 52], which detects the objects first, followed by the tracking module that leverages different information such as visual appearances [6, 60] or motion cues [52, 59], to associate the object boxes.

Extending from 2D MOT techniques, existing 3D MOT approaches mainly rely on the high-quality LiDAR detector to track objects in 3D space. AB3DMOT [49] adopts a 3D IOU similarity metric, and the Kalman filter [18] to predict and update the state of objects. CenterPoint [54] adds a learnable velocity estimation head to replace the Kalman filter to perform tracking, while GNN3DMOT [50] and PTP [51] exploit graph neural networks to integrate the appearance and motion features from LiDAR and image information. Furthermore, to avoid handicraft or heuristic matching steps in the previous pipeline, SimTrack [31] introduces an end-to-end joint detection and tracking model to associate data implicitly. In this paper, different from relying on LiDAR signals as the above-mentioned methods, we develop the 3D MOT approach based on purely monocular cameras.

Monocular 3D Multi-Object Tracking. Compared with LiDAR-based 3D MOT, camera-based 3D MOT [11, 34, 61] is a challenging task due to the inaccurate object depth estimation. Early methods [43, 53, 62] mostly extend from 2D MOT frameworks to track 3D objects in the image plane, which may achieve undesirable performance since it cannot well capture the motion of objects. Furthermore, QD-3DT [15] and DEFT [6] jointly learn the objects' state and their Re-ID features, followed by an LSTM-based module for modeling the movement of objects. Recently, inspired by the success of transformer-based 2D MOT framework MOTR [56], MUTR3D [57] leverages 3D track queries to

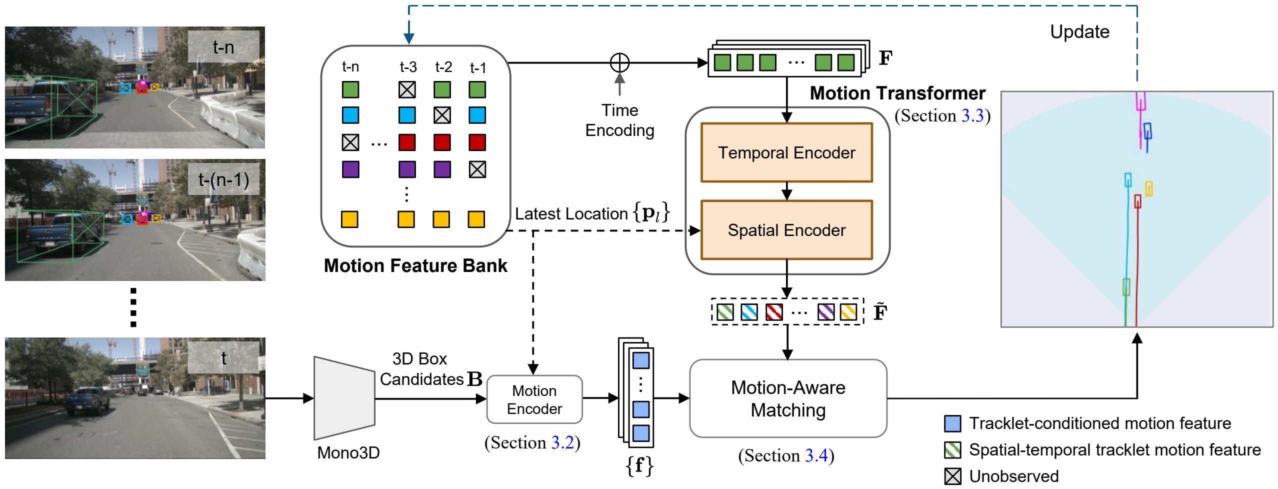

Figure 2. **Overall framework of the proposed MoMA-M3T.** At each timestamp, we leverage a monocular 3D object detector to generate 3D bounding box candidates $\mathbf{B}$. Then, we take the latest 3D positions $\{\mathbf{p}_l\}$ of tracklets to generate all possible movements for each tracklet-detection pair, followed by a motion encoder to extract the tracklet-conditioned motion features $\{\mathbf{f}\}$ (Section 3.2). On the other hand, a motion transformer module is applied to aggregate the motion cues $\mathbf{F}$ temporally and spatially from different timestamps, resulting in motion features $\tilde{\mathbf{F}}$ for each tracked object (Section 3.3). Finally, a motion-aware matching strategy is adopted to associate the learned motion features between tracklets and detections (Section 3.4).

associate objects based on the multi-camera detector [48]. On the other hand, Time3D [24] jointly learns the 3D detection and tracking from a monocular stream in an end-to-end manner, which utilizes the transformer to model the relationship between objects within adjacent frames.

However, less effort has been made to tackle one important problem of monocular 3D MOT, *i.e.*, matching inaccurate and noisy predictions in multi-frame observations. Thus, we focus on modeling object tracklets and detections with motion representations, while designing motion-aware modules to help the learning process, *e.g.*, motion transformer and motion-aware matching in the feature space.

3. Proposed Approach

3.1. Framework Overview

Given the detected 3D bounding box candidates $\mathbf{B}_t = \{\mathbf{b}_t\}$ at frame t from the monocular 3D object detectors [7, 45, 46], where $\mathbf{b} = (\mathbf{p}, \theta, h, w, l)$ denotes an object's 3D position $\mathbf{p} = (x, y, z)$, heading angle θ, and object size (h, w, l), we aim to perform online 3D MOT to find a set of tracklets $\mathbf{T}_t = \{\tau_t\}$. In this paper, we propose a motion-aware matching approach, MoMA-M3T, for monocular 3D MOT, following the tracking-by-detection paradigm to associate observations and object tracklets.

As shown in Figure 2, MoMA-M3T mainly consists of three modules: the motion encoder, the motion transformer, and the motion-aware matching module. At each timestamp t, we first utilize an encoder to generate possible motion-aware feature candidates based on the movement between observations and tracklets (Section 3.2). Then a motion transformer is exploited to aggregate motion representations of tracklets across different time frames in a spatial-temporal perspective (Section 3.3). Consequently, we learn the affinity matrix for identity matching based on the motion-aware representations of observations and tracklets with the motion-aware matching module (Section 3.4).

3.2. Motion Feature Generation

Unlike the previous work [24] that directly encodes the absolute positions for objects, we express them with motion representations based on their movement vectors along different time frames, which facilitates matching under inaccurate observations from the monocular 3D detector.

Motion Representation. Consider an object's two global positions $\mathbf{p}_a, \mathbf{p}_b \in \mathbb{R}^3$, the relative movement from b to a can be expressed as:

$$\mathbf{r}_{a|b} = \mathbf{p}_a - \mathbf{p}_b. \tag{1}$$

In addition, we define the motion state of an object at timestamp t as $\mathbf{s}_t = (\mathbf{r}, \theta, h, w, l)_t$ with its heading angle and size, where $\mathbf{r}$ indicates the position movement from the previous frame to the current frame. To obtain motion features of the object, we apply a motion encoder via a multi-layer perceptron (MLP) to describe state information:

$$\mathbf{f}_t = \mathrm{MLP}(\mathbf{s}_t) \in \mathbb{R}^{\mathrm{C}}, \tag{2}$$

where C is the feature dimension. As such, $\mathbf{f}_t$ can be used to express the motion features of any object at frame t.

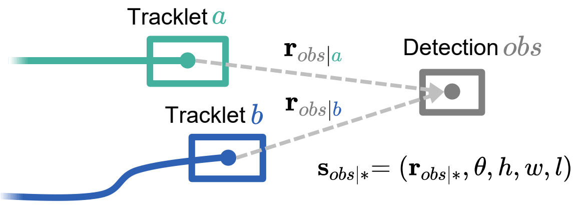

Figure 3. **Tracklet-conditioned motion state.** For any observation with estimated 3D position $\mathbf{p}_{obs}$, we calculate the relative movement $\mathbf{r}_{obs|*}$ to the latest position of any active tracklet, *e.g.*, a and b. With other detected object information, *i.e.*, heading angle θ and object size (h, w, l), we generate the object's tracklet-conditioned motion state $\mathbf{s}_{obs|*}$.

Tracklet-conditioned Motion Feature. For a single frame observation, since its previous location is undetermined before the tracking association process, we take the latest positions of all tracklets as their last locations to generate all possible motion features.

Specifically, consider M tracklets with their latest positions $\mathbf{P}_l = \{\mathbf{p}_l\}$ and N observations in the current frame with the estimated positions $\mathbf{P}_{obs} = \{\mathbf{p}_{obs}\}$, we can adopt (1) to calculate all possible movements between detections and tracklets as $\mathbf{r}_{\{obs|l\}} \in \mathbb{R}^{N\times M\times 3}$. Note that, if there is no tracklet, we set the relative movements of objects as zero. Next, we generate all candidate motion states and utilize (2) to extract all candidate motion features for the current observations, which are referred to as tracklet-conditioned motion features. We utilize an example shown in Figure 3 to illustrate the process of motion state generation.

Motion Feature Bank. We create a motion feature bank to maintain historical motion features $\mathbf{F}_{bank} \in \mathbb{R}^{N_{max}\times T_{max}\times C}$ and global 3D positions $\mathbf{P}_{bank} \in \mathbb{R}^{N_{max}\times T_{max}\times 3}$ for all tracklets, where N_{max} is the maximum number of tracked objects, and T_{max} denotes the maximum time length. After the tracking association process in each time frame (details introduced in the later section), we store the tracked objects' latest positions and their tracklet-conditioned motion features in the feature bank.

3.3. Motion Transformer

To capture the motion behavior of different tracked objects, inspired by the transformer's success in modeling sequential data, we propose a motion transformer to express the object's motion representations from a spatial-temporal perspective, which consists of three modules: time encoding, temporal encoder, and spatial encoder.

Input and Time Encoding. For the input of the transformer, we take the latest T-frame features of each tracklet from the motion feature bank. Considering the objects may be non-consecutive in certain frames due to the occlusion or undetected results (denoted as grey grids in the motion feature bank of Figure 2), we add a learnable time positional embedding to make the model aware of temporal cues. Specifically, we take the time differences between the historical and current frames, and then apply a learnable positional encoding to learn the temporal cues.

Temporal Encoder. To extract the temporal information for each tracklet, we exploit a transformer as the temporal encoder to model the object's motion representations along the temporal dimension. Specifically, considering the input motion feature $\mathbf{F} \in \mathbb{R}^{T\times C}$ of any tracklet from the motion feature bank $\mathbf{F}_{bank}$ along T frames, we prepend a learnable motion token $\mathbf{F}_m \in \mathbb{R}^{1\times C}$ to the sequence following BERT [9]. Then, the concatenated features are fed to the multi-head self-attention encoder layers:

$$\mathbf{Q} = [\mathbf{F}, \mathbf{F}_m]\mathbf{W}_q, \mathbf{K} = [\mathbf{F}, \mathbf{F}_m]\mathbf{W}_k, \mathbf{V} = [\mathbf{F}, \mathbf{F}_m]\mathbf{W}_v,$$
$$[\hat{\mathbf{F}}, \hat{\mathbf{F}}_m] = \mathrm{FFN}(\mathrm{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V})), \quad (3)$$

where $\mathbf{W}_*$ are learnable parameters for the temporal encoder, and $[\cdot, \cdot]$ means the feature concatenation operation. We use one linear layer followed by the ReLU activation to build our feed-forward network FFN (see [44] for details about self-attention layer MultiHead). Consequently, we output the learned motion token $\hat{\mathbf{F}}_m \in \mathbb{R}^{1\times C}$ to reflect motion representations of the tracked object, which is then sent to the subsequent spatial encoder module.

Spatial Encoder. We observe that the states of objects (*e.g.*, locations) may depend on other objects in the same scene. Thus, we exploit a spatial encoder after the temporal module to capture spatial dependencies among tracklets, including tracklets' states and their relationships.

Considering the aggregated motion features from the temporal encoder only encode the local relative movement, we further introduce the absolute position of objects as global information. For M tracklets with their features $\{\hat{\mathbf{F}}_m\} \in \mathbb{R}^{M\times C}$ via (3) and latest locations $\{\mathbf{p}_l\} \in \mathbb{R}^{M\times 3}$ from $\mathbf{P}_{bank}$, we use two linear layers to encode the global positional features $\mathbf{X}_p = \mathrm{MLP}(\{\mathbf{p}_l\})$. Then, we incorporate the position and motion features into the transformer:

$$\mathbf{Q}^s{=}\{\hat{\mathbf{F}}_\mathrm{m}\}\mathbf{W}^\mathrm{s}_\mathrm{q}, \mathbf{K}^\mathrm{s}{=}[\mathbf{X}_\mathrm{p}, \{\hat{\mathbf{F}}_\mathrm{m}\}]\mathbf{W}^\mathrm{s}_\mathrm{k}, \mathbf{V}^\mathrm{s}{=}[\mathbf{X}_\mathrm{p}, \{\hat{\mathbf{F}}_\mathrm{m}\}]\mathbf{W}^\mathrm{s}_\mathrm{v},$$
$$\tilde{\mathbf{F}} = \mathrm{FFN}(\mathrm{MultiHead}(\mathbf{Q}^s, \mathbf{K}^s, \mathbf{V}^s)), \quad (4)$$

where $\mathbf{W}^s_*$ are learnable parameters for the spatial encoder. Finally, we output the spatially interacted features $\tilde{\mathbf{F}} \in \mathbb{R}^{M\times C}$ for tracklets to represent their final motion features, which can be used for the matching process.

3.4. Motion-Aware Matching Learning

After obtaining the motion features $\tilde{\mathbf{F}}$ for tracklets and the tracklet-conditioned motion features $\{\mathbf{f}\}$ for observations in the current frame, we aim to perform affinity learning to solve data association for detection-tracklet pairs.

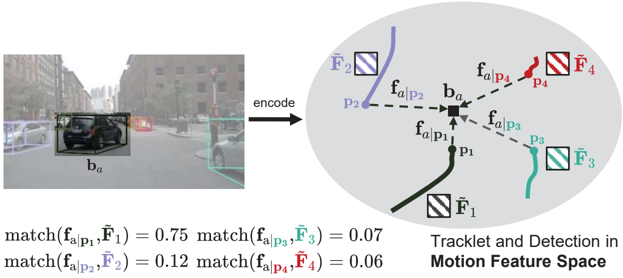
encode
$\mathbf{b}_a$
$\tilde{\mathbf{F}}_2$
$\tilde{\mathbf{F}}_4$
$\tilde{\mathbf{F}}_3$
$\tilde{\mathbf{F}}_1$
$\mathbf{f}_{a|\mathbf{p}_2}$
$\mathbf{f}_{a|\mathbf{p}_4}$
$\mathbf{f}_{a|\mathbf{p}_3}$
$\mathbf{f}_{a|\mathbf{p}_1}$
$\mathbf{p}_1$
$\mathbf{p}_2$
$\mathbf{p}_3$
$\mathbf{p}_4$
$\text{match}(\mathbf{f}_{a|\mathbf{p}_1},\tilde{\mathbf{F}}_1) = 0.75$ $\text{match}(\mathbf{f}_{a|\mathbf{p}_3},\tilde{\mathbf{F}}_3) = 0.07$
$\text{match}(\mathbf{f}_{a|\mathbf{p}_2},\tilde{\mathbf{F}}_2) = 0.12$ $\text{match}(\mathbf{f}_{a|\mathbf{p}_4},\tilde{\mathbf{F}}_4) = 0.06$
Tracklet and Detection in
Motion Feature Space

Figure 4. **Motion-aware matching.** For M tracklets with their motion features $\tilde{\mathbf{F}}$ and latest locations $\mathbf{p}$ ($M = 4$ for illustration), given any observation $\mathbf{b}_a$, we generate tracklet-conditioned motion features $\mathbf{f}_{a|\mathbf{p}}$ based on the representations described in Section 3.2. We use an MLP layer to predict a pairwise matching score, as the difference between detection's and tracklet's motion features.

Matching in Motion Feature Space. In Figure 4, consider M tracklets with their motion features $\tilde{\mathbf{F}} \in \mathbb{R}^{M \times C}$ and N detected objects $\{\mathbf{b}_i\}$ with their tracklet-conditioned motion features $\{\mathbf{f}_{i|\mathbf{p}}\} \in \mathbb{R}^{N \times M \times C}$ (based on the tracked objects' latest positions $\mathbf{p}$ as described in Section 3.2), our goal is to output a matching score between 0 and 1 to indicate whether any detection-tracklet pair has the same identity. We use an MLP layer to learn a mapping function with the input of the difference between detection's and tracklet's motion features, followed by a sigmoid function:

$$\mathbf{A}_{ij} = \text{Sigmoid}(\text{MLP}(\mathbf{f}_{\text{i}|\mathbf{p}_\text{j}} - \tilde{\mathbf{F}}_\text{j})), \tag{5}$$

where $\mathbf{A}_{ij}$ is the probability of the detection i and the tracklet j belonging to the same identity. We apply a binary focal loss FL [27] to learn the matching process:

$$\mathcal{L}_{match} = \frac{1}{N \cdot M} \sum_{i}^{N} \sum_{j}^{M} \text{FL}(\mathbf{A}_{\text{ij}}, \hat{\mathbf{A}}_{\text{ij}}), \tag{6}$$

where $\hat{\mathbf{A}}$ is the ground truth affinity value, *i.e.*, 1 or 0, depending on whether the pairs are the same object or not.

Contrastive Motion Feature Learning. Due to the occlusion and the inaccurate predictions, monocular 3D object detection results are generally noisy, which is challenging for the model to learn the motion pattern of the objects. To alleviate this, we propose a contrastive motion learning strategy that learns robust motion representations for each tracklet as illustrated in Figure 5.

Considering all tracked objects in a video, we randomly sample k subset of their trajectories (positions along different timestamps). Based on contrastive learning, the trajectory subsets sampled from the same tracklet should have similar motion representations, and the distinct trajectory subsets should have dissimilar representations. For all sampled trajectories, we can utilize the motion transformer described in Section 3.3 to encode their motion feature $\tilde{\mathbf{F}}$ and apply a contrastive loss [19] for representation learning:

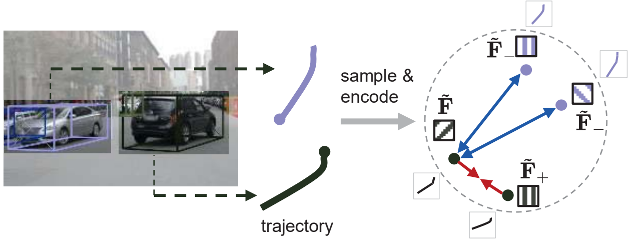

Figure 5. **Contrastive motion learning strategy.** We randomly sample the subsets of the trajectory for all objects and apply contrastive learning to construct a robust feature space, encouraging the motion features from the same trajectory to be similar ($\tilde{\mathbf{F}}$ and $\tilde{\mathbf{F}}_+$), and dissimilar to different trajectories ($\tilde{\mathbf{F}}$ and $\tilde{\mathbf{F}}_-$).

$$\mathcal{L}_{\text{con}} = -\frac{1}{|\mathbf{N}_p|} \sum_{(i,j) \in \mathbf{N}_p} \log \frac{\exp(\tilde{\mathbf{F}}_i \cdot \tilde{\mathbf{F}}_j / \tau)}{\sum_{(i,k) \in \mathbf{N}_a} \exp(\tilde{\mathbf{F}}_i \cdot \tilde{\mathbf{F}}_k / \tau)}, \tag{7}$$

where τ is the temperature parameter, which is set to 0.1 in our implementation. $\mathbf{N}_p$ is the set of positive pairs sampled from the same trajectory, while negative pairs are in the set of $\mathbf{N}_a \backslash \mathbf{N}_p$, in which $\mathbf{N}_a$ contains all the samples. The process encourages the motion embeddings $\tilde{\mathbf{F}}_i$ and $\tilde{\mathbf{F}}_j$ from the same tracklet to be similar, while $\tilde{\mathbf{F}}_i$ and $\tilde{\mathbf{F}}_k$ from the different ones should be dissimilar.

Overall Objectives. The overall training loss of our network is defined as the summation of the matching loss and the contrastive loss: $\mathcal{L} = \mathcal{L}_{match} + \mathcal{L}_{con}$.

3.5. Online Inference and Feature Update

At each time frame t, after obtaining the 3D bounding box candidates from the monocular 3D object detector [7, 37, 45, 46], our modules first generate the motion features of tracklets through the motion transformer (Section 3.3), and tracklet-conditioned motion features of current object detections (Section 3.2). Using the affinity matrix for each detection-tracklet pair via (5), the Hungarian algorithm [21] is applied to match one-to-one pairs. If the matching score is larger than a threshold (*i.e.*, 0.5 in this paper), this pair is selected as the tracking result. The motion features of the matched detection are also updated in the motion feature bank, along with their global positions. Furthermore, we use the track rebirth strategy [1, 61] to retain unmatched tracklets until they are unmatched for 10 consecutive frames for handling the occlusion issue.

4. Experiments

4.1. Experimental Setups

Datasets. We evaluate our approach on nuScenes 3D MOT [5] and KITTI 3D MOT [12]. The nuScenes dataset

Method	Reference	AMOTA(%)↑	AMOTP(m)↓	MOTA(%)↑	MOTP(m)↓	MOTAR(%)↑	MT↑	ML↓
CenterTrack [61]	ECCV'20	4.6	1.543	4.3	0.753	23.1	573	5235
TraDeS [53]†	CVPR'21	5.9	1.49	-	-	-	-	-
PermaTrack [43]	ICCV'21	6.6	1.491	6.0	0.724	32.1	652	5065
DEFT [6]	CVPRw'21	17.7	1.564	15.6	0.770	48.4	1951	3232
QD-3DT [15]	PAMI'22	21.7	1.550	19.8	0.773	56.3	1893	2970
Time3D [24]†	CVPR'22	21.4	**1.36**	17.3	0.75	-	-	-
MoMA-M3T (Ours)	-	24.2	1.479	21.3	0.713	58.1	1968	3026
MoMA-M3T (Ours)‡	-	**28.5**	1.416	**24.6**	**0.695**	**62.3**	**2236**	**2642**

Table 1. **3D MOT performance on the nuScenes test set for the single-camera tracking setting**. The best results are highlighted in **bold**. † indicates the results reported in their papers. ‡ denotes using the detector [46] trained with a longer schedule and data augmentations.

contains 1000 real-world videos captured from six surrounding cameras with 7 object categories for the tracking task. The dataset is officially split into 700, 150, and 150 sequences for training, validation, and testing. We follow [15] to train our network on the keyframes and test on the *full frames* for monocular 3D object tracking, which has higher frame rates. The KITTI tracking dataset consists of 21 training and 29 testing scenes. As there is no official benchmark for the 3D tracking task on the KITTI dataset, we apply the metrics proposed in AB3DMOT [49] for evaluation. We follow [15] to divide the entire training set into a train set (13 scenes) and a validation set (8 scenes).

Evaluation Metrics. On the nuScenes dataset, we utilize the official benchmark protocol to report the average performance for all categories, including AMOTA, AMOTP, MOTA, MOTP, MOTAR, mostly tracked (MT), and mostly lost (ML). For the KITTI tracking dataset, we report the sAMOTA and AMOTA metrics [49] of the car category for 3D evaluation. We refer the readers to the supplementary material for more details.

Implementation Details. Our approach is implemented in Pytorch on an NVIDIA 3090 GPU. For training the proposed MoMA-M3T, we utilize the Adam optimizer for 100 epochs with batch size 128. The learning rate starts at 0.0001 and decays with a step of 0.5 decay rate every 20 epochs. In each mini-batch, we randomly sample 16 tracklets with $T = 6$ frames and 16 detections for training the identity association process. In addition, we randomly sample $k = 2$ subsets of each trajectory for motion contrastive learning, which results in 1 positive and $(16-1) \times k = 30$ negative samples for each trajectory. For the motion feature bank, we set $N_{max} = 50$ and $T_{max} = 10$ with the channel number $C = 128$. For nuScenes, we use PGD3D [46] as our main monocular 3D detector. For KITTI, we utilize MonoDLE [33] for fair comparisons with existing methods. We include more details in the supplementary material.

4.2. Main Results

Monocular 3D MOT on nuScenes. To evaluate the 3D tracking performance based on monocular sensors on the nuScenes dataset, we follow [6, 15, 24, 61] to consider tracking and recognizing 3D objects from different cameras independently, which refers to as the single-camera tracking setting. In Table 1, we report the tracking performance averaged of all categories on the nuScenes test set. Compared with other monocular 3D MOT methods, our approach achieves state-of-the-art results in most metrics. Specifically, compared to Time3D [24] using the detector with similar detection performance, i.e., 31.2 mAP (Time3D) *vs.* 30.1 mAP (ours) on the nuScenes test set, our tracking method performs better than Time3D by +2.8 AMOTA on average, which is the major metric in the benchmark.

Method	Input	sAMOTA↑	AMOTA↑
QD-3DT [15]	Mono	39.92	11.86
CenterTrack [61]	Mono	42.28	11.37
MonoDLE [33]*	Mono	46.16	13.00
MoMA-M3T (Ours)	Mono	**47.17**	**16.12**

Table 2. **3D MOT performance on the KITTI validation set for the Car category at 0.25 IoU threshold** with the evaluation metric proposed in [49]. * indicates using AB3DMOT as the tracker. All results are reproduced by ourselves based on their official codes and trained on the same data. We utilize the same detector as MonoDLE for fair comparisons.

Monocular 3D MOT on KITTI. In Table 2, we report the 3D MOT performance for the car category on the KITTI tracking dataset with the evaluation metric proposed in [49] compared with different monocular-based methods, including QD3DT [15], CenterTrack [61], and MonoDLE [33] with the AB3DMOT tracker [49]. All baselines are reproduced by ourselves based on the official source codes and trained under the same settings. Overall, our approach achieves favorable performance against several monocular-based methods. Specifically, compared to MonoDLE with the AB3DMOT tracker, our MoMA-M3T with the same detector obtains an improvement of +1.01 in sAMOTA and +3.12 in AMOTA, which validates the effectiveness of our motion tracker.

Runtime Speed. We measure the inference speed of our motion tracker on a single NVIDIA 3090 GPU for processing the nuScenes validation set with batch size 1. Our tracker runs at 33.3 FPS on average.

	Representation	Matching Space	AMOTA↑	AMOTP↓
(a)	Global	Output	27.8	1.498
(b)	Global	Feature	28.8	1.460
(c)	Motion	Output	28.7	1.470
(d)	Motion	Feature	30.7	1.436
(e)	Motion	Feature†	**31.1**	**1.432**

Table 3. **Analysis of the importance of different representations and matching space** on the nuScenes validation set. † denotes using contrastive learning strategy in Section 3.4.

	Ablation	AMOTA↑	AMOTP↓	MOTA↑
(a)	Baseline	27.1	1.465	23.4
(b)	w/o Temporal encoder	29.5	1.447	25.5
(c)	w/o Spatial encoder	30.1	1.435	26.0
(d)	w/o Global positional feature	30.7	1.436	26.9
(e)	Motion Transformer→LSTM	29.7	1.440	26.3
(f)	Full model	**31.1**	**1.432**	**27.1**

Table 4. **Analysis of different components in the proposed motion transformer** using the nuScenes validation set. See Section 4.3 for details.

4.3. Ablation Study and Analysis

Importance of motion representations and feature space for matching. In Table 3, we show the effectiveness of learning motion representations in a feature space for matching: (1) We represent tracklets and observations in the global coordinate by normalizing their 3D positions based on the ego-vehicle position, which is the scene-centric representation. We denote it as the global representation compared with our motion representation. (2) Instead of matching in the feature space, we may associate tracklets and observations based on the distance between their output states (*e.g.*, object position, heading angle, and size), which resembles the practice in the Kalman filter [18].

In Table 3, we observe from (a) → (c) and (b) → (d) that our motion representations are aware of object movements and thus help model training to achieve better performance. Furthermore, from (a) → (b) and (c) → (d), we validate that matching in the feature space mitigates the potential noises in object states, which is important for monocular 3D MOT since the observations from the visual detector can be inaccurate. In addition, benefiting from matching in the feature space, (e) shows that the proposed contrastive loss learns more robust representations to further boost performance.

Effectiveness of each component in motion transformer. In Table 4, we further investigate the effectiveness of each design in our motion transformer: mainly including the temporal encoder, spatial encoder, and the global positional feature in the spatial encoder.

We show that each proposed module brings performance improvement. First, the baseline (a), without the whole proposed motion transformer, achieves undesirable performance (27.1 in AMOTA). In addition, comparing (b) with the full model (f), temporal learning provides the most improvement (+1.6 in AMOTA) since historical cues can help the model capture motion information. We also show the effect of using the spatial encoder and the global positional feature in the spatial encoder. Results comparing (c)(d) to (f) show the importance of capturing the spatial interaction between different tracklets, as well as the awareness of 3D location to model the spatial interaction. Finally, we replace the proposed transformer architecture with the classical LSTM model (e) to show the effectiveness of the spatial-temporal modeling from our motion transformer.

Detector	Method	AMOTA↑	AMOTP↓
FCOS3D [45]	KF3D	23.4	1.502
	LSTM	23.8	1.500
	Ours	**26.0**	**1.447**
EPro-PnP [7]	KF3D	25.8	1.482
	LSTM	27.0	1.470
	Ours	**29.7**	**1.418**

Table 5. **Analysis of different motion modules** on the nuScenes validation set. We evaluate performance with different detectors using the same model checkpoint of trackers without re-training.

Method	AMOTA↑	AMOTP↓	MOTA↑	MOTP↓	MOTAR↑
MUTR3D [57]†	27.0	1.494	24.5	0.709	64.3
CC-3DT [11]◇	41.0	1.274	35.7	**0.676**	69.0
MoMA-M3T (Ours)†	36.0	1.349	31.1	0.700	68.4
MoMA-M3T (Ours)†*	41.5	1.278	**36.8**	0.701	71.0
MoMA-M3T (Ours)◇	**42.5**	**1.240**	36.1	0.681	**71.1**

Table 6. **3D MOT performance on the nuScenes test set for the multi-camera tracking setting.** † and ◇ denote using DETR3D [48] and BEVFormer [26] as the detector with the ResNet101 [14] backbone, respectively. * indicates the detector trained with a V2-99 backbone provided by [36].

Robustness analysis on monocular 3D object detectors. To show the robustness of our tracker under various 3D detection outputs, we evaluate our motion tracker using different front-view-based monocular 3D object detectors, including FCOS3D [45] and EPro-PnP [7]. We perform tracking on these detectors using the same model without re-training to demonstrate the generalization ability of our motion tracker.

In Table 5, we compare our motion modules with LSTM [15] and 3D Kalman filter [18] that predict and match objects' state in the output space. We show that our MoMA-M3T achieves better performance across different detectors. One of the main reasons is that matching in the feature space can perform more robustly than that in the output space, in which our method is less sensitive to noises from outputs of various 3D object detectors.

Multi-camera 3D MOT on nuScenes. While our method focuses on the monocular setting, it is also applicable to

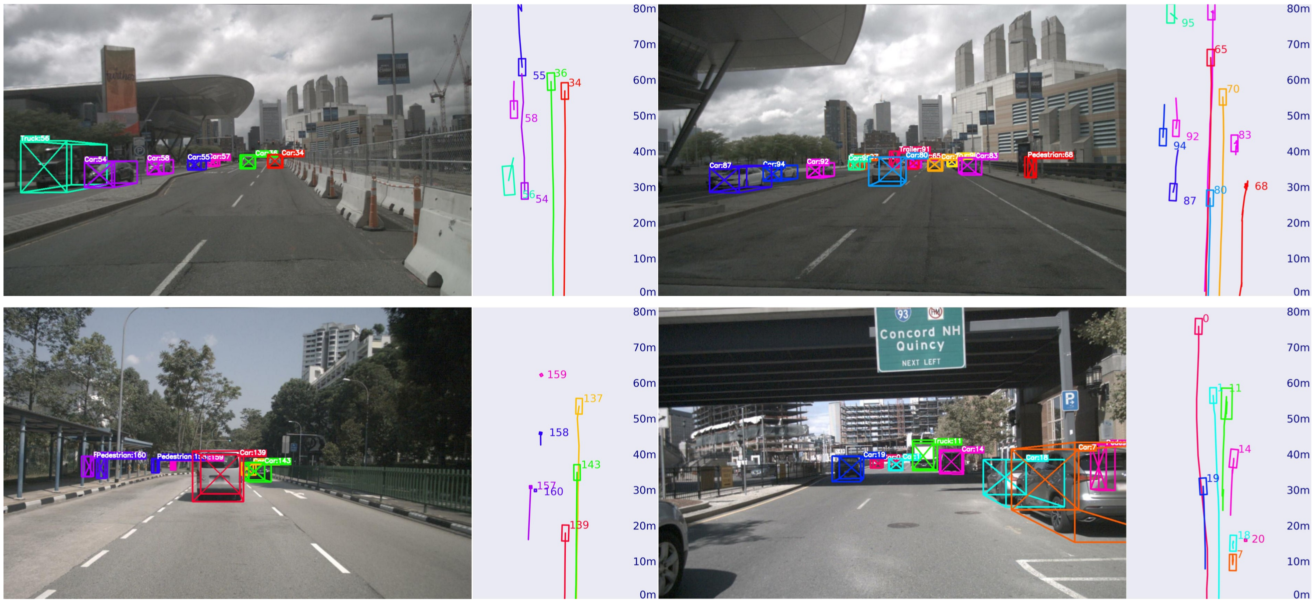

Figure 6. **Qualitative visualization on the nuScenes validation set.** We provide some examples of tracking results on the image view for the current frame (left) and the trajectories in the bird's eye view (right) for 15 historical frames. We utilize different colors and numbers to represent the different objects' identities. Best viewed in color and zoomed in.

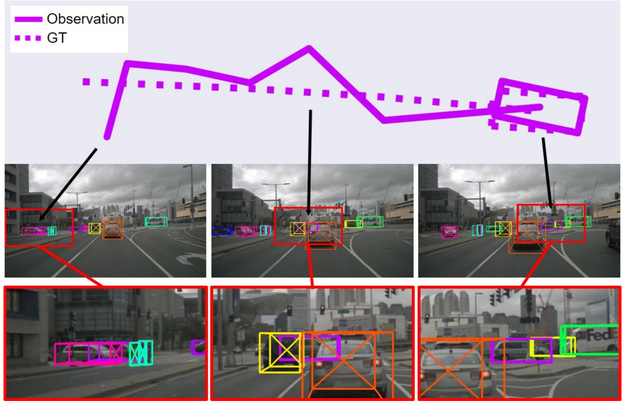

Figure 7. **Example results of handling inaccurate 3D observations.** The noisy 3D detection results (*i.e.*, the magenta boxes) are often caused by occlusion and inaccurate predictions. The solid and dotted lines denote the observed and ground truth trajectories on the bird's eye view plane. Our motion tracker is able to track objects even when the observations are not accurate enough.

multi-camera detection systems [26,48] that simultaneously recognize objects for all cameras, which can boost performance by filtering duplicate detections and benefiting tracking across cameras. We conduct experiments under the multi-camera tracking setting and present the tracking results on the nuScenes test dataset in Table 6. We show that our MoMA-M3T obtains significant improvements of +9.0 in AMOTA compared to MUTR3D [57] with the same detector (*i.e.*, DETR3D [48]), which indicates the effectiveness of our approach. Moreover, compared with CC-3DT [11] using the same detector (*i.e.*, BEVFormer [26]), our approach obtains +1.5 in AMOTA. This validates the effectiveness of adopting our method in various settings.

4.4. Qualitative Results

We show qualitative examples on the nuScenes validation set in Figure 6 to illustrate that our motion tracker can track objects across various scenarios. Also, we provide a representative example in Figure 7 to show that our motion-aware tracker can track objects well, even under inaccurate observations caused by occlusion and inaccurate depth estimation from the monocular 3D object detector. More qualitative results are included in the supplementary material.

5. Conclusions

In this paper, we present MoMA-M3T, a motion-aware matching strategy for monocular 3D MOT. We represent the motion information for tracklets with their relative movements, followed by a motion transformer to model the motion cues from a spatio-temporal perspective. Consequently, a motion-aware matching module is applied to match tracklets and current observations based on their motion features. Extensive experiments on the nuScenes and KITTI datasets demonstrate that MoMA-M3T achieves state-of-the-art performance and is compatible to integrate with existing monocular 3D object detectors without the need of finetuning our tracker.

References

[1] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixé. Tracking without bells and whistles. In *ICCV*, 2019. 5

[2] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *ICIP*, 2016. 2

[3] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *ICCV*, 2019. 2

[4] Garrick Brazil, Gerard Pons-Moll, Xiaoming Liu, and Bernt Schiele. Kinematic 3d object detection in monocular video. In *ECCV*, 2020. 1

[5] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 2, 5

[6] Mohamed Chaabane, Peter Zhang, Ross Beveridge, and Stephen O'Hara. Deft: Detection embeddings for tracking. In *CVPR Workshops*, 2021. 1, 2, 6

[7] Hansheng Chen, Pichao Wang, Fan Wang, Wei Tian, Lu Xiong, and Hao Li. Epro-pnp: Generalized end-to-end probabilistic perspective-n-points for monocular object pose estimation. In *CVPR*, 2022. 2, 3, 5, 7

[8] Yi-Nan Chen, Hang Dai, and Yong Ding. Pseudo-stereo for monocular 3d object detection in autonomous driving. In *CVPR*, 2022. 2

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 4

[10] Mingyu Ding, Yuqi Huo, Hongwei Yi, Zhe Wang, Jianping Shi, Zhiwu Lu, and Ping Luo. Learning depth-guided convolutions for monocular 3d object detection. In *CVPR*, 2020. 2

[11] Tobias Fischer, Yung-Hsu Yang, Suryansh Kumar, Min Sun, and Fisher Yu. Cc-3dt: Panoramic 3d object tracking via cross-camera fusion. In *CoRL*, 2022. 2, 7, 8

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 2, 5

[13] Chenhang He, Hui Zeng, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. Structure aware single-stage 3d object detection from point cloud. In *CVPR*, 2020. 1

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 7

[15] Hou-Ning Hu, Yung-Hsu Yang, Tobias Fischer, Trevor Darrell, Fisher Yu, and Min Sun. Monocular quasi-dense 3d object tracking. *TPAMI*, 2022. 1, 2, 6, 7

[16] Junjie Huang, Guan Huang, Zheng Zhu, Ye Yun, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 1, 2

[17] Kuan-Chih Huang, Tsung-Han Wu, Hung-Ting Su, and Winston H. Hsu. Monodtr: Monocular 3d object detection with depth-aware transformer. In *CVPR*, 2022. 2

[18] Rudolph Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 1960. 2, 7

[19] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020. 5

[20] Jason Ku, Alex D Pon, and Steven L Waslander. Monocular 3d object detection leveraging accurate proposals and shape reconstruction. In *CVPR*, 2019. 1

[21] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 1955. 5

[22] Abhinav Kumar, Garrick Brazil, and Xiaoming Liu. Groomed-nms: Grouped mathematically differentiable nms for monocular 3d object detection. In *CVPR*, 2021. 2

[23] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, 2019. 1

[24] Peixuan Li and Jieyu Jin. Time3d: End-to-end joint monocular 3d object detection and tracking for autonomous driving. In *CVPR*, 2022. 1, 3, 6

[25] Peixuan Li, Huaici Zhao, Pengfei Liu, and Feidao Cao. Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving. In *ECCV*, 2020. 2

[26] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *ECCV*, 2022. 1, 2, 7, 8

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *TPAMI*, 2018. 5

[28] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *ECCV*, 2022. 2

[29] Zechen Liu, Zizhang Wu, and Roland Tóth. SMOKE: Single-stage monocular 3d object detection via keypoint estimation. In *CVPR Workshops*, 2020. 2

[30] Yan Lu, Xinzhu Ma, Lei Yang, Tianzhu Zhang, Yating Liu, Qi Chu, Junjie Yan, and Wanli Ouyang. Geometry uncertainty projection network for monocular 3d object detection. In *ICCV*, 2021. 2

[31] Chenxu Luo, Xiaodong Yang, and Alan Yuille. Exploring simple 3d multi-object tracking for autonomous driving. In *ICCV*, 2021. 2

[32] Xinzhu Ma, Zhihui Wang, Haojie Li, Pengbo Zhang, Xin Fan, and Wanli Ouyang. Accurate monocular object detection via color-embedded 3d reconstruction for autonomous driving. In *ICCV*, 2019. 1

[33] Xinzhu Ma, Yinmin Zhang, Dan Xu, Dongzhan Zhou, Shuai Yi, Haojie Li, and Wanli Ouyang. Delving into localization errors for monocular 3d object detection. In *CVPR*, 2021. 2, 6

[34] Nicola Marinello1, Marc Proesmans, and Luc Van Gool. Triplettrack: 3d object tracking using triplet embeddings and lstm. In *CVPR Workshops*, 2022. 2

[35] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *CVPR*, 2017. 1

[36] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *ICCV*, 2021. 7

[37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 5

[38] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *CVPR*, 2020. 1

[39] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, 2019. 1

[40] Xuepeng Shi, Qi Ye, Xiaozhi Chen, Chuangrong Chen, Zhixiang Chen, and TaeKyun Kim. Geometry-based distance decomposition for monocular 3d object detection. In *ICCV*, 2021. 2

[41] Andrea Simonelli, Samuel Rota Bulò, Lorenzo Porzi, Elisa Ricci, and Peter Kontschieder. Towards generalization across depth for monocular 3d object detection. In *ECCV*, 2020. 2

[42] Andrea Simonelli, Samuel Rota Rota Bulò, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *ICCV*, 2019. 2

[43] Pavel Tokmakov, Jie Li, Wolfram Burgard, and Adrien Gaidon. Learning to track with object permanence. In *ICCV*, 2021. 2, 6

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 4

[45] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. FCOS3D: Fully convolutional one-stage monocular 3d object detection. In *ICCV Workshops*, 2021. 2, 3, 5, 7

[46] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Probabilistic and geometric depth: Detecting objects in perspective. In *CoRL*, 2021. 2, 3, 5, 6

[47] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *CVPR*, 2019. 2

[48] Yue Wang, Vitor Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, , and Justin M. Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *CoRL*, 2021. 2, 3, 7, 8

[49] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3D Multi-Object Tracking: A Baseline and New Evaluation Metrics. In *IROS*, 2020. 1, 2, 6

[50] Xinshuo Weng, Yongxin Wang, Yunze Man, and Kris Kitani. Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning. In *CVPR*, 2020. 1, 2

[51] Xinshuo Weng, Ye Yuan, and Kris Kitani. PTP: Parallelized Tracking and Prediction with Graph Neural Networks and Diversity Sampling. *Robotics and Automation Letters*, 2021. 2

[52] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, 2017. 2

[53] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to detect and segment: An online multi-object tracker. In *CVPR*, 2021. 2, 6

[54] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. In *CVPR*, 2021. 2

[55] Ming Liu Yuxuan Liu, Yuan Yixuan. Ground-aware monocular 3d object detection for autonomous driving. In *ICRA*, 2021. 2

[56] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-to-end multiple-object tracking with transformer. In *ECCV*, 2022. 2

[57] Tianyuan Zhang, Xuanyao Chen, Yue Wang, Yilun Wang, and Hang Zhao. Mutr3d: A multi-camera tracking framework via 3d-to-2d queries. In *CVPR Workshops*, 2022. 2, 7, 8

[58] Yunpeng Zhang, Jiwen Lu, and Jie Zhou. Objects are different: Flexible monocular 3d object detection. In *CVPR*, 2021. 2

[59] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *ECCV*, 2022. 2

[60] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *IJCV*, 2021. 2

[61] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *ECCV*, 2020. 1, 2, 5, 6

[62] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2

[63] Xingyi Zhou, Tianwei Yin, Vladlen Koltun, and Philipp Krähenbühl. Global tracking transformers. In *CVPR*, 2022. 2

[64] Zhikang Zou, Xiaoqing Ye, Liang Du, Xianhui Cheng, Xiao Tan, Li Zhang, Jianfeng Feng, Xiangyang Xue, and Errui Ding. The devil is in the task: Exploiting reciprocal appearance-localization features for monocular 3d object detection. In *ICCV*, 2021. 2