Track without Appearance: Learn Box and Tracklet Embedding with Local and Global Motion Patterns for Vehicle Tracking

Gaoang Wang[1], Renshu Gu[2], Zuozhu Liu[1], Weijie Hu[3], Mingli Song[1], and Jenq-Neng Hwang[4]

[1]Zhejiang University, [2]Hangzhou Dianzi University, [3]Guangdong University of Petrochemical Technology, [4]University of Washington

{gaoangwang, zuozhuliu}@intl.zju.edu.cn, renshugu@hdu.edu.cn, huweijie@gdupt.edu.cn, brooksong@zju.edu.cn, hwang@uw.edu

Abstract

Vehicle tracking is an essential task in the multi-object tracking (MOT) field. A distinct characteristic in vehicle tracking is that the trajectories of vehicles are fairly smooth in both the world coordinate and the image coordinate. Hence, models that capture motion consistencies are of high necessity. However, tracking with the standalone motion-based trackers is quite challenging because targets could get lost easily due to limited information, detection error and occlusion. Leveraging appearance information to assist object re-identification could resolve this challenge to some extent. However, doing so requires extra computation while appearance information is sensitive to occlusion as well. In this paper, we try to explore the significance of motion patterns for vehicle tracking without appearance information. We propose a novel approach that tackles the association issue for long-term tracking with the exclusive fully-exploited motion information. We address the tracklet embedding issue with the proposed reconstruct-to-embed strategy based on deep graph convolutional neural networks (GCN). Comprehensive experiments on the KITTI-car tracking dataset and UA-Detrac dataset show that the proposed method, though without appearance information, could achieve competitive performance with the state-of-the-art (SOTA) trackers. The source code will be available at https://github.com/GaoangW/LGMTracker.

1. Introduction

Multi-object tracking (MOT) is an important topic in the computer vision and machine learning field. This technique is highly demanded in many tasks, such as traffic flow estimation, human behavior prediction and autonomous driving assistance [52, 51, 55, 25, 20]. From unsupervised rule-based [6, 7, 5, 22, 52] and optimization-based [66, 11, 62, 56, 32, 38, 37, 1, 24, 12] to deep learning-based trackers [13, 68, 41, 57, 40, 4, 67, 42, 63], significant progress of the MOT techniques has been made in the recent ten years. However, some critical challenges still remain. For example, occlusion is still one of the major issues. Without occlusion handling, the targets can easily get lost and identities may get switched. Other challenges, such as crowded scenarios, detection errors and camera motions, also have significant influences on a tracker's performance.

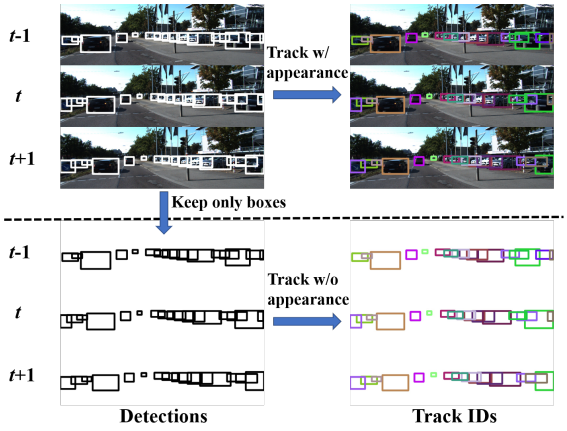

Figure 1. The top part shows tracking with appearance information, while the bottom shows tracking using detection boxes without employing appearance. Obviously, it is more challenging for tracking only based on motion information.

Appearance information is widely used for MOT and greatly improves performance. Appearance information is employed either in an association manner [52, 67, 14] or with regression-based approaches for joint learning of de-

tection and tracking [68, 41, 57, 4]. The assumption, as well as the attribution of its success, is that the same targets from adjacent frames should share similar appearance features. However, the appearance feature is still sensitive to occlusions and objects may have quite different appearance representations when they are occluded. Additionally, joint learning approaches require an extra computational cost.

Motion consistency is another cue that can be taken advantage of for MOT, especially for vehicle tracking scenarios. This is based on the assumption that the motions of objects usually follow fairly smooth patterns in both the world coordinate and the image coordinate. In particular, for objects that cannot change the orientation and speed rapidly, such as vehicles, motion consistency could play a pivot role for tracking. In addition, the motion feature, usually with the four bounding box parameters for each object, is simple and light, saving more computations than complex appearance features. As a result, mere motion trackers are still worth exploring. However, there are two main difficulties to establish deep motion-based models. First, motion itself can only provide limited information. As shown in Figure 1, after discarding all appearance information, it is highly challenging to associate the bounding boxes correctly even for humans when false positives and false negatives occur in the detection. Second, to alleviate the long-term occlusion issue, tracklet association is needed in deep motion-based models. Accurate association requires expressive tracklet embeddings that could be used to measure the similarity among different tracklets. However, learning such embeddings is very challenging as we need to capture temporal consistency as well. For example, tracklets of the same object might have different temporal lengths or do not share similar locations along time, leading to inferior embeddings in practice. Due to the aforementioned challenges, mere motion trackers usually cannot achieve comparable performance with models that adopt appearance information.

In this paper, we tackle the vehicle tracking problem only from the motion perspective. Without appearance information, we aim to explore how well a motion-based model can perform for the vehicle tracking task. A novel motion-based tracking approach, i.e., local-global motion (LGM) tracker, is proposed to exploit the motion consistency without using any appearance information after the detection. More specifically, without appearance information means: 1) NO further bounding box regression or refinement from the detector feature maps; 2) NO appearance information used for further association and re-identification. The flowchart of the proposed LGM tracker is shown in Figure 2. We model the MOT problem as a two-stage embedding task where both local and global motion consistencies are utilized. At the first stage, we aim at learning the box embedding based on deep graph convolutional neural networks (GCN) to associate boxes into tracklets. Since such local associations cannot capture the global track patterns, the occlusion issue is yet unaddressed. To break through such limitation, at the second stage, the tracklet embedding with global motion consistency is learned to further associate tracklets into tracks. To better model the tracklet embedding, a novel embedding strategy, *reconstruct-to-embed*, is proposed with the temporal gated convolution mechanism under an attention-based GCN.

Our contributions are summarized as follows: 1) we tackle the vehicle tracking task from the motion perspective without using appearance information; 2) we propose a novel box and tracklet embedding method that can utilize both the local and global motion consistencies; 3) we evaluate the proposed LGM tracker on KITTI [18] and UA-Detrac [58] benchmark datasets and achieve competitive performance with the state-of-the-art (SOTA) trackers.

2. Related Work

2.1. Motion Models

Motion trackers [6, 44, 23, 64, 2, 17] without using appearance information are also studied in recent a few years. For such methods, some use fairly simple rules, like intersection-over-union (IOU) between adjacent frames as association [6]; some adopt particle filter framework in the tracking [44]; some apply recurrent neural networks (RNN) to learn the motion patterns [17, 2]. However, with limited information, good performance cannot be easily achieved by motion-based methods.

2.2. Graph Models

Conventional graph models [50, 39, 52, 28, 30, 11, 49, 59, 54, 56, 32, 37, 1, 24] are widely used in MOT for data association. Usually, detections or tracklets are adopted as graph nodes. Then the similarities among nodes are measured on the connected edges. The association is solved by optimizing the total cost or energy function. However, the similarity measure is usually based on hand-crafted feature fusion, requiring empirically setting a lot of hyper-parameters. Since graph neural networks (GNN) show great power recently, many approaches [8, 46, 61] adopt GNN for the association, rather than using the conventional graph models based on optimization. However, for most existing methods based on GNN, only the local association based on adjacent frames is considered. As a result, the long-term occlusion is still one major issue for GNN based trackers.

2.3. Joint Detection and Tracking

More recently, joint detection and tracking based methods have been drawn great attention [68, 41, 57, 40, 4]. Usually, the tracker takes sequential adjacent frames as input. Features are aggregated in different frames and bound-

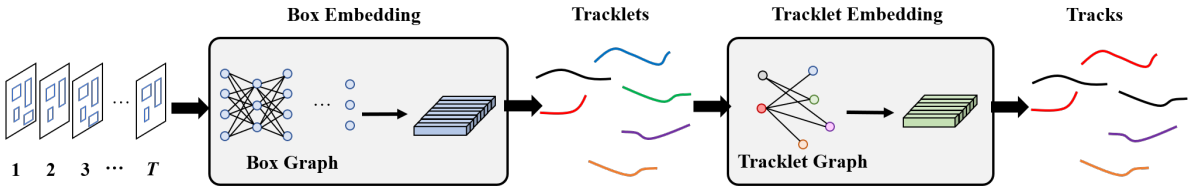

Figure 2. The flowchart of the local-global motion (LGM) tracker. The LGM tracker contains two modules, the box embedding module and the tracklet embedding module, which exploit the local and global motion patterns, respectively. Specifically, the box embedding module embeds input boxes and connects them into tracklets, and the tracklet embedding module aims to associate tracklets into tracks. Both modules are learned based on deep graph convolutional networks (GCN).

ing box regression is conducted with temporal information. More recently, several methods with visual transformers [48, 35] are also explored in the MOT field and achieve comparable results. For example, [48] proposes a baseline tracker via a transformer, which takes advantage of the query-key mechanism and introduces a set of learned object queries into the pipeline to enable detecting new-coming objects. [35] extends the DETR object detector [10] and achieves a seamless data association between frames in a new tracking-by-attention paradigm by encoder-decoder self-attention mechanisms. However, due to the heavy computational cost, the networks can only take a very limited number of frames as input. As a result, the global motion patterns are still not well utilized.

3. Method

As shown in Figure 2, we propose box and tracklet embedding based on GCN to learn both local and global motions in the LGM tracker. Boxes are locally connected to form the tracklets, followed by the tracklet association to further form the tracks. Details are demonstrated in the following sub-sections.

3.1. Box Motion Embedding

The box embedding is proposed to associate boxes into tracklets given the box graph with the connection between adjacent frames. Considering a temporal window, we build the box graph based on the adjacency among boxes. Specifically, denote $\boldsymbol{X}^0 \in \mathbb{R}^{N\times 4}$ as the input boxes with normalized box parameters x, y, w, h, where N is the total number of boxes inside the temporal window. Denote $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ as the adjacency matrix, where $\boldsymbol{A}_{ij} = 1$ if box i and box j are in the neighboring frames; otherwise set $\boldsymbol{A}_{ij} = 0$.

To learn both the structural and temporal relations among detections, inspired from [53], we stack L attention-guided GCN blocks. For the l-th GCN layer, the update rule is defined as follows,

$$\boldsymbol{X}^l = \text{ReLU}(\boldsymbol{D}^{l^{-1/2}}\hat{\boldsymbol{A}}^l\boldsymbol{D}^{l^{-1/2}}\boldsymbol{X}^{l-1}\boldsymbol{W}^l) + \boldsymbol{X}^{l-1}, \quad (1)$$

where $\boldsymbol{X}^{l-1}$ is the node embedding from the $(l-1)$-th layer, $\boldsymbol{W}^l$ is the convolution kernel, $\hat{\boldsymbol{A}}^l$ is the refined adjacency matrix and $\boldsymbol{D}^l$ is the diagonal node degree matrix with $\boldsymbol{D}^l_{ii} = \sum_{j=0}\hat{\boldsymbol{A}}^l_{ij}$.

Since most of the connections between adjacent frames among the nodes are from distinct objects, the aggregated information from different objects can have a negative effect on the embedding. As a result, we apply the attention mechanism to refine the adjacency matrix to deal with such an issue as follows,

$$\hat{\boldsymbol{A}}^l = (\boldsymbol{A} + \boldsymbol{I}) \odot \boldsymbol{X}^l_{att}, \quad (2)$$

where $\boldsymbol{X}^l_{att}$ is the self-attention feature, $\odot$ represents the elementwise multiplication, and $\boldsymbol{X}^l_{att}$ is defined as,

$$\boldsymbol{X}^l_{att} = \sigma(\text{ReLU}(f(\boldsymbol{X}^{l-1})\boldsymbol{W}^l_{att,1})\boldsymbol{W}^l_{att,2}), \quad (3)$$

where $\boldsymbol{W}_{att,1}$ and $\boldsymbol{W}_{att,2}$ are convolution kernels, σ is the Sigmoid activation function and f represents the operation of pairwise self dot product.

We use a combination of triplet loss $\mathcal{L}_{triplet}$ [45] and binary cross-entropy loss $\mathcal{L}_{xent}$ for training box embedding module. They are defined as follows,

$$\begin{aligned}\mathcal{L}_{triplet} &= \frac{1}{N}\sum_i^N[||\boldsymbol{X}_i^{out^a} - \boldsymbol{X}_i^{out^p}||_2^2 \\ &- ||\boldsymbol{X}_i^{out^a} - \boldsymbol{X}_i^{out^n}||_2^2 + \alpha]_+, \\ \mathcal{L}_{xent} &= \frac{1}{N}\sum_{ij}^N\{t_{ij}\log\boldsymbol{X}^L_{att,ij} \\ &+ (1-t_{ij})\log(1-\boldsymbol{X}^L_{att,ij})\},\end{aligned} \quad (4)$$

where $[\cdot]_+$ clamps the input value to be non-negative; $\boldsymbol{X}_i^{out^a}$, $\boldsymbol{X}_i^{out^p}$ and $\boldsymbol{X}_i^{out^n}$ represent the box embedding output from the anchor sample, positive sample and negative sample, respectively; α is the pre-defined margin; $\boldsymbol{X}^L_{att,ij}$ measures the similarity between node i and j in the last GCN layer, while t_{ij} is the binary label indicating the identity between node i and j. Therefore, the total loss for the box embedding training is defined as

$$\mathcal{L}_{box} = \mathcal{L}_{triplet} + \lambda_1\mathcal{L}_{xent}. \quad (5)$$

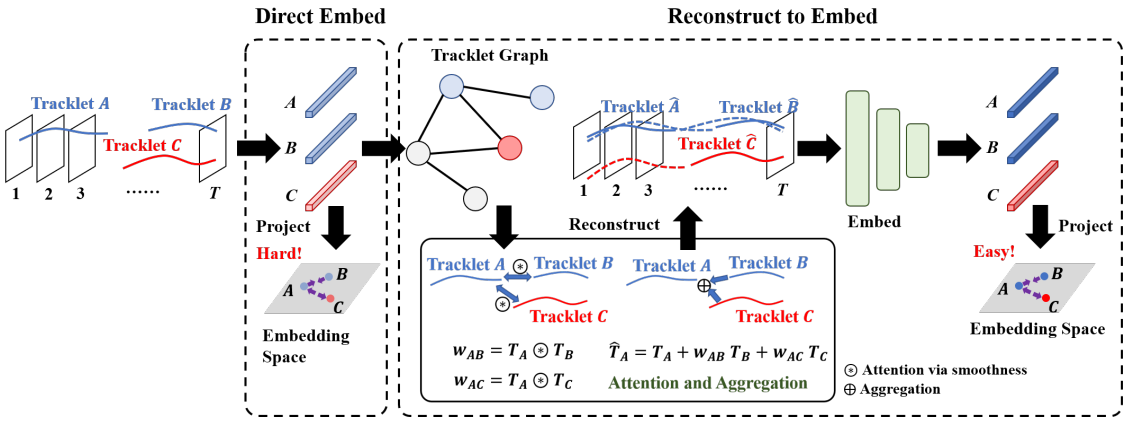

Figure 3. The figure shows the motivation of *reconstruct-to-embed* strategy in the tracklet embedding. The left part is the illustration of the difficulty of the direct tracklet embedding, while the right part shows the proposed indirect tracklet embedding strategy, i.e., *reconstruct-to-embed*. Attention-aggregation mechanism is employed based on the GCN framework in the tracklet reconstruction, followed by the final embedding.

3.2. Tracklet Motion Embedding

The goal of tracklet motion embedding is to explore the global motion patterns among tracklets for the association. However, tracklet embedding is not quite straightforward. Since tracklets from the same object exist in different frames and usually have different temporal lengths, it is very challenging to find a latent space to ensure that they share similar feature embeddings. As shown in the left part of Figure 3, the direct embedding is difficult to measure the similarity among tracklets. To alleviate the challenge in the tracklet embedding, we propose a novel embedding strategy, named as *reconstruct-to-embed*, following an *attention* and *reconstruction* mechanism based on GCN, as shown in the right part of Figure 3. The motivation behind this is simple. Take tracklet A for example. We calculate the attention from tracklet B and C based on the smoothness of temporal relations. Then based on the attention scores from B and C, we aggregate and reconstruct the latent trajectory $\hat{A}$. We use the same reconstruction strategy for B and C to generate $\hat{B}$ and $\hat{C}$. Compared with the original tracklets A and B, $\hat{A}$ and $\hat{B}$ have much more similar motion patterns after the reconstruction, which makes the embedding much easier than the situation in the direct embedding. Finally, embeddings are learned with one additional embed-head block with reconstructed tracklets.

The tracklet graph is defined as follows. Considering a temporal window, we denote $\boldsymbol{X}^0 \in \mathbb{R}^{N\times 4\times T}$ as the input tracklets with normalized box parameters x, y, w, h, where N is the number of tracklets and T is the temporal window size. Since tracklets usually have different temporal lengths, we pad zeros along the temporal dimension if the length is smaller than the temporal window T. We also define tracklet temporal occupancy matrix $\boldsymbol{M}^0 \in \mathbb{R}^{N\times 1\times T}$ as

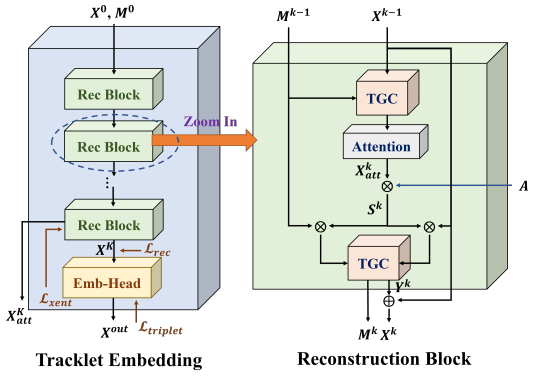

Figure 4. The overview architecture of the tracklet embedding module. The left part shows tracklet embedding with sequentially stacked reconstruction blocks and one embed-head block. Three losses, i.e., triplet loss $\mathcal{L}_{triplet}$, cross-entropy loss $\mathcal{L}_{xent}$ and reconstruction loss $\mathcal{L}_{rec}$, are combined in the training. The right part shows each basic reconstruction block based on GCN and temporal gated convolution (TGC) modules for temporal information extraction.

the mask input, where $\boldsymbol{M}^0_{it} = 0$ if no box exists for tracklet i in frame t due to missing detection or occlusion; otherwise $\boldsymbol{M}^0_{it} = 1$. Denote $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ as the adjacency matrix, where $\boldsymbol{A}_{ij} = 1$ if tracklet i and tracklet j do not have overlapping frames in the temporal window; otherwise $\boldsymbol{A}_{ij} = 0$ following the assumption that temporal overlapping tracklets cannot share the same ID.

The architecture of tracklet embedding is shown in Figure 4. We stack K reconstruction blocks to measure the long-term relations in both structural and temporal dimensions for reconstruction, and then followed by one embed-head block for the final embedding. As shown in Figure 4,

each reconstruction block updates the feature maps and soft masks based on the attention and temporal gated convolution (TGC) module. The goal of attention is to measure the similarities among tracklets. Good tracklet embeddings should have high similarities if they are from the same object. However, simple operations cannot well represent the similarities. If we take cosine similarity measure for example, the zero-padded motion features of two tracklets from the same object would be always zero due to the fact that two tracklets from the same object do not have overlapping frames. Inspired by the TGC module in the inpainting task [15, 65, 21], where missing values can be reconstructed with input masks, we can use TGC with the occupancy masks for tracklet extrapolation. Based on the extrapolated tracklets, the similarities are easier to measure and the temporal consistency among tracklets from the same object can be exploited. Specifically, the attention/similarity map of the k-th reconstruction block is calculated as follows,

$$\begin{aligned} \tilde{\boldsymbol{X}}^k, \tilde{\boldsymbol{M}}^k &= g(\boldsymbol{X}^{k-1}, \boldsymbol{M}^{k-1}), \\ \boldsymbol{X}^k_{att} &= \sigma(\text{ReLU}(f(\tilde{\boldsymbol{X}}^k)\boldsymbol{W}^k_{att,1})\boldsymbol{W}^k_{att,2}), \end{aligned} \tag{6}$$

where $\boldsymbol{X}^{k-1}$ and $\boldsymbol{M}^{k-1}$ are feature maps and masks from the previous block, g is the TGC module and f is the pairwise self dot product operation. Here, we calculate attention maps after the extrapolation of tracklets based on TGC module. The TGC module has multiple stacked TGC layers. To be specific, the feature map and mask of each TGC layer are updated as follows,

$$\begin{aligned} \boldsymbol{M}^{j+1} &= \sigma(\boldsymbol{M}^j * \boldsymbol{W}^j_M), \\ \boldsymbol{Y}^{j+1} &= \text{ReLU}(\boldsymbol{Y}^j * \boldsymbol{W}^j_Y) \odot \boldsymbol{M}^{j+1} + \boldsymbol{Y}^j, \end{aligned} \tag{7}$$

where $*$ represents the convolution operation, $\odot$ represents the dot product, $\boldsymbol{M}^j$ is the soft mask, $\boldsymbol{Y}^j$ is the feature mask, $\boldsymbol{W}^j_M$ and $\boldsymbol{W}^j_Y$ are the convolution kernel weights for the mask and feature maps, respectively.

Then we aggregate the attention maps based on the graph structure. The aggregation output $\boldsymbol{S}^k$ is represented as

$$\boldsymbol{S}^k = \boldsymbol{D}^{k^{-1/2}} \hat{\boldsymbol{A}}^k \boldsymbol{D}^{k^{-1/2}}, \tag{8}$$

where $\hat{\boldsymbol{A}}^k$ is with the same definition as Eq. (2), which recalculates the attention maps based on the adjacency matrix of the graph structure.

We aggregate the attention map $\boldsymbol{S}^k$ with $\boldsymbol{X}^{k-1}$ and $\boldsymbol{M}^{k-1}$, followed by a second TGC module to obtain the output of the k-th reconstruction block as follows,

$$\begin{aligned} \boldsymbol{Y}^k, \boldsymbol{M}^k &= g(\boldsymbol{S}^k\boldsymbol{X}^{k-1}, \boldsymbol{S}^k\boldsymbol{M}^{k-1}), \\ \boldsymbol{X}^k &= \boldsymbol{X}^{k-1} + \boldsymbol{Y}^k. \end{aligned} \tag{9}$$

Simply put, the second TGC module plays a role as a further transformation of node embeddings.

To obtain the final tracklet embedding, a feed-forward transformation with two dense layers is included in the embed-head block. The feed-forward transformation is defined as follows,

$$\boldsymbol{X}^{out} = \text{Norm}_{l_2}(\text{ReLU}(f(\boldsymbol{X}^K)\boldsymbol{W}_1)\boldsymbol{W}_2), \tag{10}$$

where Norm_{l_2} is the l_2 normalization, $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are the weights of the two dense layers.

We also adopt the triplet loss $\mathcal{L}_{triplet}$ and binary cross-entropy loss $\mathcal{L}_{xent}$ based on the output $\boldsymbol{X}^{out}$ for tracklet embedding training with similar equations to Eq. (4). In addition, a reconstruction loss $\mathcal{L}_{rec}$ is employed in the tracklet embedding. To be specific, the reconstruction loss is defined as follows,

$$\mathcal{L}_{rec} = ||\boldsymbol{M}^*(\boldsymbol{X}^K - \boldsymbol{X}^*)||_2, \tag{11}$$

where $\boldsymbol{X}^*$ is the ground truth track and $\boldsymbol{M}^*$ is the ground truth occupancy mask. Finally, the total loss for tracklet embedding training is defined as follows,

$$\mathcal{L}_{tracklet} = \mathcal{L}_{triplet} + \lambda_2\mathcal{L}_{xent} + \lambda_3\mathcal{L}_{rec}. \tag{12}$$

3.3. Inference

After training, the association is conducted based on the box and tracklet embedding in the inference stage. The temporal sliding window procedure is adopted with 50% overlapping frames. Within each temporal window, boxes are associated based on the embedding distances. Then the tracklets are associated with the bottom-up greedy method. 50% overlap is used to ensure the new boxes and tracklets can be matched to existing tracked objects.

4. Experiments

4.1. Datasets

Two vehicle tracking benchmark datasets, i.e., KITTI [18] and UA-Detrac [58], are used for validation.

KITTI. The KITTI car tracking benchmark consists of 21 training sequences and 29 testing sequences. Videos are captured at 10 FPS and contain large inter-frame motions. We only evaluate the tracking performance on the *car* category.

UA-Detrac. The UA-DETRAC is a large-scale tracking dataset for vehicles. It comprises 100 videos that record around 10 hours of vehicle traffic. The recording is made in 24 different locations, and it includes a wide variety of common vehicle types and traffic conditions. Overall, the dataset contains about 140k video frames, 8,250 vehicles, and 1,210k bounding boxes.

4.2. Implementation Details

For the architecture of the proposed LGM tracker, we stack $L = 8$ GCN blocks and $K = 4$ reconstruction blocks

Method	HOTA (%) ↑	AssA (%) ↑	MOTA (%) ↑	MT (%) ↑	ML (%) ↓	IDS ↓	FRAG ↓
†⋆JRMOT [47]	69.6	66.9	85.1	70.9	4.6	271	273
†⋆AB3DMOT [60]	69.8	69.1	83.5	67.1	11.4	126	254
†⋆MOTSFusion [33]	68.7	66.2	84.2	72.8	2.9	415	569
†⋆mono3DT [26]	73.2	74.2	84.3	73.1	2.9	379	573
⋆MASS [27]	68.3	64.5	84.6	74.0	2.9	353	516
⋆TuSimple [11]	71.6	71.1	86.3	71.1	6.9	292	220
⋆SMAT [19]	71.9	72.1	83.6	62.8	6.0	198	294
⋆CenterTrack [68]	73.0	71.2	88.8	82.2	2.5	254	227
DCO-X [38]	46.5	38.7	66.2	38.3	14.5	955	708
SCEA [23]	56.1	52.2	74.9	53.7	12.3	**324**	317
MCMOT-CPD [31]	56.6	50.6	78.0	52.5	12.5	475	309
LGM (ours)	**73.1**	**72.3**	**87.6**	**85.1**	**2.5**	448	**164**

Table 1. Result on KITTI-car tracking testing set. From top to bottom, we divide SOTA methods into three categories (*3D trackers*, *2D trackers*, *mere motion 2D trackers*) according to different input information. † represents 3D tracking methods and ⋆ represents trackers using appearance information. The best result for each part is shown in red, blue and bold, respectively.

Method	PR-MOTA (%) ↑	PR-MOTP (%) ↑	PR-MT (%) ↑	PR-ML (%) ↓	PR-IDS ↓	PR-FRAG ↓
IHTLS [16]	11.1	36.8	13.8	19.9	953.6	3556.9
H2T [59]	12.4	35.7	14.8	19.4	852.2	1117.2
CMOT [3]	12.6	36.1	16.1	18.6	**285.3**	1516.8
GOG [43]	14.2	37.0	13.9	19.9	3334.6	3172.4
IOUT [6]	16.1	**37.0**	14.8	19.7	2308.1	3250.4
V-IOUT [7]	17.7	36.4	**17.4**	18.8	363.8	1123.5
FAMNet [13]	19.8	36.7	17.1	18.2	617.4	**970.2**
LGM (ours)	**22.5**	35.2	15.5	**10.1**	1563.5	3186.8

Table 2. Result on UA-Detrac testing set. The best performance is shown in bold type.

in the box and the tracklet embedding modules, respectively. For the TGC module in the tracklet embedding, we have 6 basic TGC layers. We use 17 frames and 65 frames as the temporal window for the box and tracklet embedding modules, respectively. The final embedding dimension for the box and tracklet is set to $D = 128$. The margin α for calculating the triplet loss is set to 0.2. For the loss combination, we simply set all λs in Eq. (5) and Eq. (12) to 1.

We use detection results from CenterNet [69, 68] and CompACT [9] as our input boxes for KITTI and UA-Detrac datasets, respectively. Both the box and tracklet embedding modules are trained with Adam optimizer [29] with an initial learning rate of 1e-3. We use a cosine annealing learning rate scheduler for the learning rate decay. The maximum step is set to 200000.

Data augmentation strategy is adopted for training the LGM tracker. For the box embedding module, the input boxes are pre-processed with random horizontal flips, randomly jittered sizes and positions. We also randomly add boxes as false positives and remove some ground truth boxes as false negatives. A similar augmentation strategy is used for training the tracklet embedding module. Besides that, we also randomly split the ground truth tracks into pieces of tracklets for the augmentation.

4.3. Evaluation Metrics

We use the default metrics defined by the benchmark datasets for the evaluation. The metrics include Higher Order Tracking Accuracy (HOTA) [34], Association Accuracy (AssA), Multiple Object Tracking Accuracy (MOTA) [36], ID F1 score (IDF1), Multiple Object Tracking Precision (MOTP) [36], the number of ID Switches (IDS), the percentage of Mostly Tracked targets (MT), the percentage of Mostly Lost targets (ML) and the total number of times a trajectory is Fragmented (FRAG). HOTA and AssA are newly defined in [34] and adopted as the main evaluation metrics for MOT in the KITTI benchmark dataset. For the UA-Detrac dataset, the metrics with PR-curve integrated are used, as defined in [58].

4.4. Main Results on Benchmark Datasets

KITTI-Car Tracking Benchmark. The tracking result on the KITTI-car testing dataset is shown in Table 1. From top to bottom, we divide SOTA methods into three categories i.e., †*3D trackers*, ⋆*2D trackers*, and *mere motion 2D trackers*, according to different input information.

Loss Combination	HOTA	AssA	MOTA
$\mathcal{L}_{triplet}$	75.7	75.0	88.0
$\mathcal{L}_{triplet} + \mathcal{L}_{xent}$	**76.9**	**77.2**	**89.0**

Table 3. Results of different loss combinations for box embedding module on the KITTI-car tracking validation set.

We achieve the best performance among the mere motion trackers, and the proposed LGM tracker is also very competitive to other SOTA methods. The result demonstrates the effectiveness of the proposed LGM tracker based on mere motion information.

UA-Detrac Benchmark. The tracking result on the UA-Detrac testing dataset with CompACT detections is shown in Table 2. We can see that the proposed LGM tracker outperforms most SOTA methods, including methods that use both appearance and motion information.

4.5. Qualitative Results

Occlusion Handling. We show some qualitative examples of the proposed LGM tracker against CenterTrack [68] about occlusion handling on the KITTI-car testing dataset in Figure 5. Each color represents a distinct tracked object with the tracking ID on the top of the bounding box. The first two rows are results of CenterTrack and LGM tracker from sequence 0011, respectively. For the LGM tracker, three cars with a red circle drawn on the figure are occluded by a car in frame 268 and then associated to the correct labels in frame 274 after they reappear, while the CenterTrack fails in the association. The last two rows show another example from sequence 0018, where two cars in the red circle are correctly associated in frame 43 for the LGM tracker, while CenterTrack fails again. These two examples demonstrate the robustness of the proposed tracker's occlusion handling strategy.

Reconstruction Analysis on the Tracklet. To further illustrate the *reconstruct-to-embed* strategy in the tracklet embedding module, we also provide one visualization example, as shown in Figure 6. For Figure 6 (a), we plot two tracklets A and B from the same object in red and blue colors, respectively. The ground truth trajectory is displayed in black. Figure 6 (b) shows the reconstructed tracks, $\hat{A}$ and $\hat{B}$, after the last reconstruction block. Based on the visualization, it is obvious that $\hat{A}$ and $\hat{B}$ have much higher similarities than A and B, which makes the embedding much easier.

4.6. Ablation Study

Study of Loss Combination for Box Embedding. The study of loss combination for box embedding on the KITTI-car validation set with the same data split defined from [68] is shown in Table 3, where the first row only uses triplet loss in the training while the second row uses both triplet loss and binary cross-entropy loss. The same tracklet embedding module is used for both cases. We can see that with a combination of the two losses, the performance is better.

Loss Combination	HOTA	AssA	MOTA
$\mathcal{L}_{triplet}$	-	-	-
$\mathcal{L}_{triplet} + \mathcal{L}_{xent}$	-	-	-
$\mathcal{L}_{triplet} + \mathcal{L}_{rec}$	76.3	76.5	88.2
$\mathcal{L}_{triplet} + \mathcal{L}_{xent} + \mathcal{L}_{rec}$	**76.9**	**77.2**	**89.0**

Table 4. Results of different loss combinations for tracklet embedding module on the KITTI-car tracking validation set. Results for the first two combinations are not available since the training does not converge.

Module	HOTA	AssA	MOTA
Box	75.2	74.3	88.0
Box+Tracklet	**76.9**	**77.2**	**89.0**

Table 5. Results of different modules on the KITTI-car tracking validation set.

Study of Loss Combination for Tracklet Embedding. To better understand the importance of each loss term in the tracklet embedding module, we try four different loss combinations in the training and then evaluate the KITTI-car validation set. For the first two trials, with standalone triplet loss $\mathcal{L}_{triplet}$ or with a combination of triplet loss and binary cross-entropy loss $\mathcal{L}_{triplet} + \mathcal{L}_{xent}$, the model fails to converge. This demonstrates the necessity of the reconstruction loss in the training and it further proves the effectiveness of *reconstruct-to-embed* strategy illustrated in Figure 3. As shown in the last two rows of Table 4, the performance improves when introducing the binary cross-entropy loss $\mathcal{L}_{xent}$, which shows the importance of the supervision on the attention mechanism.

Analysis on Functionalities of Box and Tracklet Embedding. We also test the functionalities of the box and tracklet embedding modules on the KITTI-car validation set. The result is shown in Table 5. We can see that with box embedding alone we can achieve tolerable results, yet the tracklet association is not exploited. From the second row of the table, there is further improvement in the tracking performance with the tracklet embedding module added. This example demonstrates the importance of both box and tracklet embedding.

4.7. Generalization to Pedestrian Tracking

Although pedestrian tracking is not the focus of this paper, we still report the results for pedestrian tracking on the MOT17 testing set, as shown in Table 6, where the top three methods are widely used SOTA methods with both appearance and other clues while the bottom two only use motion clues for tracking. Pedestrian tracking is more challenging

Figure 5. Examples of occlusion handling. Each row shows three frames from the same sequence. Each tracked vehicle is represented in a unique color. The number on the bounding box is the tracking ID.

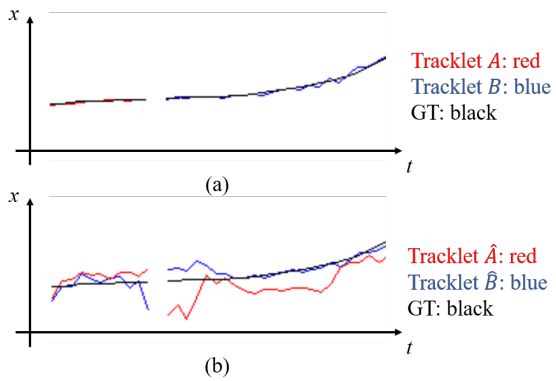

Figure 6. Visualization example of tracklet reconstruction. (a) shows two input tracklets, A and B. Both A and B are from the same object with the ground truth trajectory shown in black. (b) shows the reconstructed $\hat{A}$ and $\hat{B}$ after the last reconstruction block.

than vehicle tracking for motion-based trackers since the motion consistency assumption is not always the truth. Due to such challenges, we can still achieve comparable results using the proposed motion tracker, demonstrating the generalization ability to pedestrian tracking.

5. Conclusion

In this paper, we propose a novel tracker with motion consistency without looking at the appearance for the vehicle tracking task. Two modules, box and tracklet embedding, are designed to model both local and global motion information based on deep convolutional networks. We evaluate the proposed method on two vehicle tracking datasets, i.e., KITTI-car tracking benchmark and UA-Detrac benchmark and achieve competitive results with mere motion information. We also visualize the tracking results and the reconstructed trackets in the tracklet embedding module. This further proves the effectiveness of the proposed *reconstruct-to-embed* strategy. Several ablation studies are conducted to show the importance of each module and the losses in the model training. In future work, we plan to to incorporate appearance information in both the box and tracklet embedding modules for further improvement.

Method	MOTA	IDF1	MOTP
Tracktor++ [4]	56.3	55.1	78.8
TrctrD17 [63]	53.7	53.8	77.2
CenterTrack [68]	61.5	59.6	78.9
IOU Tracker	45.5	39.4	76.9
LGM (ours)	**56.0**	**55.6**	**78.0**

Table 6. Pedestrian tracking result on MOT17 dataset, where top three methods use appearance information while the bottom two methods only employ motion features.

Acknowledgement This work is supported by National Natural Science Foundation of China (U20B2066, 61976186), the Major Scientific Research Project of Zhejiang Lab (No. 2019KD0AC01), the Fundamental Research Funds for the Central Universities, Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies, Guangdong Provincial Characteristic Innovation Natural Science Projects of Colleges and Universities, China (2019ktscx110) and Guangdong Provincial Special Funding projects for Introducing Innovation Team and Industry University Research Cooperation, China (2019C002001).

References

[1] Anton Andriyenko, Konrad Schindler, and Stefan Roth. Discrete-continuous optimization for multi-target tracking. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1926–1933. IEEE, 2012. 1, 2

[2] Maryam Babaee, Zimu Li, and Gerhard Rigoll. Occlusion handling in tracking multiple people using rnn. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2715–2719. IEEE, 2018. 2

[3] Seung-Hwan Bae and Kuk-Jin Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1218–1225, 2014. 6

[4] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *Proceedings of the IEEE international conference on computer vision*, pages 941–951, 2019. 1, 2, 8

[5] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468. IEEE, 2016. 1

[6] Erik Bochinski, Volker Eiselein, and Thomas Sikora. High-speed tracking-by-detection without using image information. In *International Workshop on Traffic and Street Surveillance for Safety and Security at IEEE AVSS 2017*, Lecce, Italy, Aug. 2017. 1, 2, 6

[7] Erik Bochinski, Tobias Senst, and Thomas Sikora. Extending iou based multi-object tracking by visual information. In *IEEE International Conference on Advanced Video and Signals-based Surveillance*, pages 441–446, Auckland, New Zealand, Nov. 2018. 1, 6

[8] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6247–6257, 2020. 2

[9] Zhaowei Cai, Mohammad Saberian, and Nuno Vasconcelos. Learning complexity-aware cascades for deep pedestrian detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3361–3369, 2015. 6

[10] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 3

[11] Wongun Choi. Near-online multi-target tracking with aggregated local flow descriptor. In *Proceedings of the IEEE international conference on computer vision*, pages 3029–3037, 2015. 1, 2, 6

[12] Chun-Te Chu, Jenq-Neng Hwang, Hung-I Pai, and Kung-Ming Lan. Tracking human under occlusion based on adaptive multiple kernels with projected gradients. *IEEE Transactions on Multimedia*, 15(7):1602–1615, 2013. 1

[13] Peng Chu and Haibin Ling. Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6172–6181, 2019. 1, 6

[14] Meng-Che Chuang, Jenq-Neng Hwang, Kresimir Williams, and Richard Towler. Tracking live fish from low-contrast and low-frame-rate stereo videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(1):167–179, 2014. 1

[15] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *International conference on machine learning*, pages 933–941. PMLR, 2017. 5

[16] Caglayan Dicle, Octavia I Camps, and Mario Sznaier. The way they move: Tracking multiple targets with similar appearance. In *Proceedings of the IEEE international conference on computer vision*, pages 2304–2311, 2013. 6

[17] Kuan Fang, Yu Xiang, Xiaocheng Li, and Silvio Savarese. Recurrent autoregressive networks for online multi-object tracking. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 466–475. IEEE, 2018. 2

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 2, 5

[19] Nicolas Franco Gonzalez, Andres Ospina, and Philippe Calvez. Smat: Smart multiple affinity metrics for multiple object tracking. In *International Conference on Image Analysis and Recognition*, pages 48–62. Springer, 2020. 6

[20] Renshu Gu, Gaoang Wang, and Jenq-Neng Hwang. Efficient multi-person hierarchical 3d pose estimation for autonomous driving. In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 163–168. IEEE, 2019. 1

[21] Renshu Gu, Gaoang Wang, and Jenq-Neng Hwang. Exploring severe occlusion: Multi-person 3d pose estimation with gated convolution. *arXiv preprint arXiv:2011.00184*, 2020. 5

[22] Gültekin Gündüz and Tankut Acarman. Efficient multi-object tracking by strong associations on temporal window. *IEEE Transactions on Intelligent Vehicles*, 4(3):447–455, 2019. 1

[23] Ju Hong Yoon, Chang-Ryeol Lee, Ming-Hsuan Yang, and Kuk-Jin Yoon. Online multi-object tracking via structural constraint event aggregation. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pages 1392–1400, 2016. 2, 6

[24] Andrea Hornakova, Roberto Henschel, Bodo Rosenhahn, and Paul Swoboda. Lifted disjoint paths with application in multiple object tracking. *arXiv preprint arXiv:2006.14550*, 2020. 1, 2

[25] Hung-Min Hsu, Tsung-Wei Huang, Gaoang Wang, Jiarui Cai, Zhichao Lei, and Jenq-Neng Hwang. Multi-camera tracking of vehicles based on deep features re-id and trajectory-based camera link models. In *AI City Challenge Workshop, IEEE/CVF Computer Vision and Pattern Recognition (CVPR) Conference, Long Beach, California*, 2019. 1

[26] Hou-Ning Hu, Qi-Zhi Cai, Dequan Wang, Ji Lin, Min Sun, Philipp Krahenbuhl, Trevor Darrell, and Fisher Yu. Joint

monocular 3d vehicle detection and tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5390–5399, 2019. 6

[27] Hasith Karunasekera, Han Wang, and Handuo Zhang. Multiple object tracking with attention to appearance, structure, motion and size. *IEEE Access*, 7:104423–104434, 2019. 6

[28] Margret Keuper, Siyu Tang, Yu Zhongjie, Bjoern Andres, Thomas Brox, and Bernt Schiele. A multi-cut formulation for joint segmentation and tracking of multiple objects. *arXiv preprint arXiv:1607.06317*, 2016. 2

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[30] Ratnesh Kumar, Guillaume Charpiat, and Monique Thonnat. Multiple object tracking by efficient graph partitioning. In *Asian Conference on Computer Vision*, pages 445–460. Springer, 2014. 2

[31] Byungjae Lee, Enkhbayar Erdenee, Songguo Jin, Mi Young Nam, Young Giu Jung, and Phill Kyu Rhee. Multi-class multi-object tracking using changing point detection. In *European Conference on Computer Vision*, pages 68–83. Springer, 2016. 6

[32] Philip Lenz, Andreas Geiger, and Raquel Urtasun. Followme: Efficient online min-cost flow tracking with bounded memory and computation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4364–4372, 2015. 1, 2

[33] Jonathon Luiten, Tobias Fischer, and Bastian Leibe. Track to reconstruct and reconstruct to track. *IEEE Robotics and Automation Letters*, 5(2):1803–1810, 2020. 6

[34] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, pages 1–31, 2020. 6

[35] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *arXiv preprint arXiv:2101.02702*, 2021. 3

[36] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016. 6

[37] Anton Milan, Stefan Roth, and Konrad Schindler. Continuous energy minimization for multitarget tracking. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):58–72, 2013. 1, 2

[38] Anton Milan, Konrad Schindler, and Stefan Roth. Detection- and trajectory-level exclusion in multiple object tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3682–3689, 2013. 1, 6

[39] Anton Milan, Konrad Schindler, and Stefan Roth. Multi-target tracking by discrete-continuous energy minimization. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2054–2068, 2016. 2

[40] Bo Pang, Yizhuo Li, Yifan Zhang, Muchen Li, and Cewu Lu. Tubetk: Adopting tubes to track multi-object in a one-step training model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6308–6318, 2020. 1, 2

[41] Jinlong Peng, Changan Wang, Fangbin Wan, Yang Wu, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In *European Conference on Computer Vision*, pages 145–161. Springer, 2020. 1, 2

[42] Jinlong Peng, Tao Wang, Weiyao Lin, Jian Wang, John See, Shilei Wen, and Erui Ding. Tpm: Multiple object tracking with tracklet-plane matching. *Pattern Recognition*, page 107480, 2020. 1

[43] Hamed Pirsiavash, Deva Ramanan, and Charless C Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR 2011*, pages 1201–1208. IEEE, 2011. 6

[44] Ricardo Sanchez-Matilla, Fabio Poiesi, and Andrea Cavallaro. Online multi-target tracking with strong and weak detections. In *European Conference on Computer Vision*, pages 84–99. Springer, 2016. 2

[45] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 3

[46] Chaobing Shan, Chunbo Wei, Bing Deng, Jianqiang Huang, Xian-Sheng Hua, Xiaoliang Cheng, and Kewei Liang. Fgagt: Flow-guided adaptive graph tracking. *arXiv preprint arXiv:2010.09015*, 2020. 2

[47] Abhijeet Shenoi, Mihir Patel, JunYoung Gwak, Patrick Goebel, Amir Sadeghian, Hamid Rezatofighi, Roberto Martin-Martin, and Silvio Savarese. Jrmot: A real-time 3d multi-object tracker and a new large-scale dataset. *arXiv preprint arXiv:2002.08397*, 2020. 6

[48] Peize Sun, Yi Jiang, Rufeng Zhang, Enze Xie, Jinkun Cao, Xinting Hu, Tao Kong, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple-object tracking with transformer. *arXiv preprint arXiv:2012.15460*, 2020. 3

[49] Siyu Tang, Bjoern Andres, Miykhaylo Andriluka, and Bernt Schiele. Subgraph decomposition for multi-target tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5033–5041, 2015. 2

[50] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person reidentification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3539–3548, 2017. 2

[51] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and Jenq-Neng Hwang. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8797–8806, 2019. 1

[52] Zheng Tang, Gaoang Wang, Hao Xiao, Aotian Zheng, and Jenq-Neng Hwang. Single-camera and inter-camera vehicle tracking and 3d speed estimation based on fusion of visual and semantic features. In *CVPR Workshop (CVPRW) on the AI City Challenge*, 2018. 1, 2

[53] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 3

[54] Gaoang Wang, Yizhou Wang, Haotian Zhang, Renshu Gu, and Jenq-Neng Hwang. Exploit the connectivity: Multi-object tracking with trackletnet. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 482–490, 2019. 2

[55] Gaoang Wang, Xinyu Yuan, Aotian Zhang, Hung-Min Hsu, and Jenq-Neng Hwang. Anomaly candidate identification and starting time estimation of vehicles from traffic videos. In *AI City Challenge Workshop, IEEE/CVF Computer Vision and Pattern Recognition (CVPR) Conference, Long Beach, California*, 2019. 1

[56] Shaofei Wang and Charless C Fowlkes. Learning optimal parameters for multi-target tracking with contextual interactions. *International journal of computer vision*, 122(3):484–501, 2017. 1, 2

[57] Yongxin Wang, Xinshuo Weng, and Kris Kitani. Joint detection and multi-object tracking with graph neural networks. *arXiv preprint arXiv:2006.13164*, 2020. 1, 2

[58] Longyin Wen, Dawei Du, Zhaowei Cai, Zhen Lei, Ming-Ching Chang, Honggang Qi, Jongwoo Lim, Ming-Hsuan Yang, and Siwei Lyu. Ua-detrac: A new benchmark and protocol for multi-object detection and tracking. *Computer Vision and Image Understanding*, 193:102907, 2020. 2, 5, 6

[59] Longyin Wen, Wenbo Li, Junjie Yan, Zhen Lei, Dong Yi, and Stan Z Li. Multiple target tracking based on undirected hierarchical relation hypergraph. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1282–1289, 2014. 2, 6

[60] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3d multi-object tracking: A baseline and new evaluation metrics. *arXiv preprint arXiv:1907.03961*, 2020. 6

[61] Xinshuo Weng, Yongxin Wang, Yunze Man, and Kris M Kitani. Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6499–6508, 2020. 2

[62] Yu Xiang, Alexandre Alahi, and Silvio Savarese. Learning to track: Online multi-object tracking by decision making. In *Proceedings of the IEEE international conference on computer vision*, pages 4705–4713, 2015. 1

[63] Yihong Xu, Aljosa Osep, Yutong Ban, Radu Horaud, Laura Leal-Taixé, and Xavier Alameda-Pineda. How to train your deep multi-object tracker. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6787–6796, 2020. 1, 8

[64] Ju Hong Yoon, Ming-Hsuan Yang, Jongwoo Lim, and Kuk-Jin Yoon. Bayesian multi-object tracking using motion context from multiple objects. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 33–40. IEEE, 2015. 2

[65] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4471–4480, 2019. 5

[66] Li Zhang, Yuan Li, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 1

[67] Yang Zhang, Hao Sheng, Yubin Wu, Shuai Wang, Weifeng Lyu, Wei Ke, and Zhang Xiong. Long-term tracking with deep tracklet association. *IEEE Transactions on Image Processing*, 29:6694–6706, 2020. 1

[68] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. *arXiv preprint arXiv:2004.01177*, 2020. 1, 2, 6, 7, 8

[69] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 6