City-Scale Multi-Camera Vehicle Tracking by Semantic Attribute Parsing and Cross-Camera Tracklet Matching

Yuhang He, Jie Han, Wentao Yu, Xiaopeng Hong*, Xing Wei, and Yihong Gong
The Faculty of Electronic and Information Engineering, Xi'an Jiaotong University
{hyh1379478,hanjie1997,yu1034397129}@stu.xjtu.edu.cn
xingxjtu@gmail.com,{hongxiaopeng,ygong}@mail.xjtu.edu.cn

Abstract

This paper focuses on the Multi-Target Multi-Camera Tracking (MTMCT) task in a city-scale multi-camera network. As the trajectory of each target is naturally split into multiple sub-trajectories (namely local tracklets) in different cameras, the key issue of MTMCT is how to match local tracklets belonging to the same target across different cameras. To this end, we propose an efficient two-step MTMCT approach to robustly track vehicles in a camera network. It first generates all local tracklets and then matches the ones belonging to the same target across different cameras. More specifically, in the local tracklet generation phase, we follow the tracking-by-detection paradigm and link the detections to local tracklets by graph clustering. In the cross-camera tracklet matching phase, we first develop a spatial-temporal attention mechanism to produce robust tracklet representations. We then prune false matching candidates by traffic topology reasoning and match tracklets across cameras using the recently proposed TRACklet-to-Target Assignment (TRACTA) algorithm. The proposed method is evaluated on the City-Scale Multi-Camera Vehicle Tracking task at the 2020 AI City Challenge and achieves the second-best results.

1. Introduction

Multi-Target Multi-Camera Tracking (MTMCT) aims to locate the positions of interested targets, maintain their identity both within and across cameras and infer a complete trajectory for each target in a multi-camera network. It has a wide range of applications in customer-behavior analysis [30], auto-driving assisting [27] and etc. When the tracking targets are vehicles, the MTMCT has great application value for Intelligent Transportation System (ITS) and is a key component for city traffic management [45].

The MTMCT methods are mainly faced with the following two challenging problems: 1) How to generate high-quality sub-trajectories (namely local tracklets) for all the targets under each camera; 2) How to match the local tracklets in different cameras to generate an accurate, complete global trajectory for each target across all the cameras, *i.e.*, the cross-camera tracklet matching problem. The first problem is often referred as a single camera multi-target tracking problem and can be solved by the tracking-by-detection approaches [55, 25, 48, 56, 44, 53, 31]. To tackle the second problem, there are two more problems attached to match the local tracklets across different cameras. First, how to compute the affinity of different local tracklets in different cameras when there are dramatic variations in visual appearance and ambient environment under different viewpoints. Second, how to optimize the matching of local tracklets when the occurrence of each target in different cameras and the total number of targets in the camera network are both unknown.

There are methods [20, 11] propose to project regions of targets in synchronized frames to a reference plane and match the targets from different views by finding the targets that co-occupy the same locations in the reference plane. However, these methods rely on the overlapping field of views (FOV) of cameras views, and thus the application is limited in real-world scenarios. Research studies [52, 18, 6, 7] propose to match the local tracklets across different cameras without requiring the overlapping FOV of different cameras. They calculate the tracklet affinity by exploiting target trajectory temporal consistency [42], appearance features [38, 28], camera geometry [21, 18] and etc, and solve the cross-camera tracklet matching problem by hierarchical matching [51, 29], data association graph [7] or camera link models [19].

In this paper, we focus on the city-scale cross-camera vehicle tracking problem. As illustrated in Figure 1, to obtain a wide range of FOV and reduce the costs, the cameras are often placed far apart and their FOV are always non-overlapping. The target attributes such as appearance

*Corresponding author.

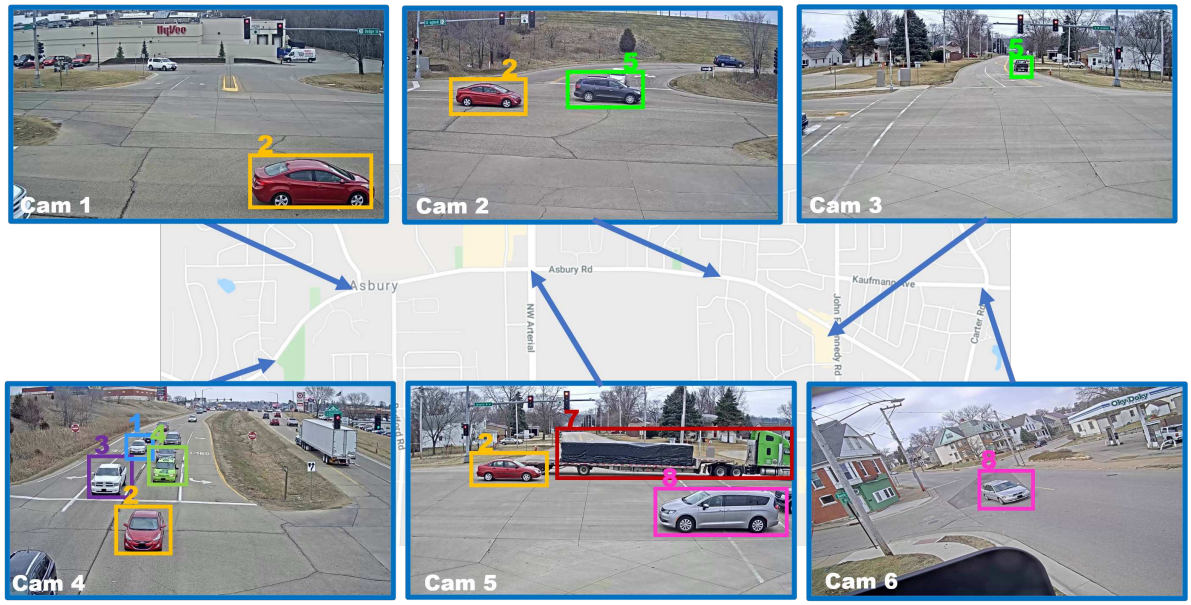

Figure 1: Illustration of city-scale multi-camera vehicle tracking. The number upon each bounding box denotes the ID label of each target and the arrows point to the city locations of the cameras. The pictures are from the dataset of the AI City challenge.

features and motion patterns of the same target could be significantly different in different cameras. Moreover, as the occurrences of each target under different cameras are different and unknown (such as the target #2 occurs in cameras 1,2,4,5, the target #8 occurs in cameras 5,6, and targets 1,3,4,7 occur only once in the camera network), it is difficult to solve the tracklet matching problem and generate a complete global trajectory for each target across all the cameras. To tackle these problems, we provide an efficient two-step approach to track multiple vehicles in a city-scale multi-camera network, which first generates local tracklets for all the targets under each camera, respectively, and then connects these local tracklets across different cameras to generate a complete global trajectory for each target. More specifically, we first follow the tracking-by-detection paradigm to generate local tracklets for all the targets under each camera, respectively. Then we compute the affinity of local tracklets in different cameras by semantic attribute parsing, which produces a robust tracklet representation using a spatial-temporal attention mechanism and prunes false matching candidates by traffic topology reasoning. Taking the local tracklet affinity as input, the TRACklet-to-Target Assignment (TRACTA) algorithm [17] is exploited to solve the cross-camera tracklet matching problem, and the complete trajectory of each target across all the cameras is obtained by reconnecting the split local tracklets. Experimental evaluations on the City-Scale Multi-Camera Vehicle Tracking at the NVIDIA AI City Challenge 2020 demonstrate the superior of the proposed method.

In summary, the main contributions of this paper include:

- We provide an efficient two-step MTMCT method for city-scale multi-camera vehicle tracking.
- We propose the semantic attribute parsing for tracklet affinity measurement.
- We introduce a spatial-temporal attention mechanism to generate a robust representation for each target.

2. Related Work

2.1. Single-Camera Multi-Object Tracking

In recent years, there is a large literature on Single-Camera Multi-Object Tracking (SCMOT). Due to the rapid development of object detection techniques, the tracking-by-detection paradigm has become the mainstream for SCMOT task.

A large number of research studies use bipartite matching to tackle multiple object tracking problems [3, 9, 48, 40, 50]. Simple Online and Realtime Tracking (SORT) [3] uses Kalman Filter [22] and Hungarian algorithm [35] to tackle frame-to-frame prediction and association problems. To increase robustness against occlusions and misses, Wojke *et al.* [48] integrate appearance information for Simple Online and Realtime Tracking(SORT) through a pre-trained deep association metric. Instance Aware Tracker (IAT) [9] integrates single object tracking (SOT) algorithms for MOT,

which dynamically refreshes tracking models with a learned convolutional neural network. The method [39] proposes to use a structure of Recurrent Neural Networks (RNN) that integrates appearance, motion, and interaction cues to jointly learn a representation to improve tracking robustness. Xu *et al.* [50] propose a method of spatial-temporal relation networks (STRN) for similarity measurement between an object and a tracklet, which uses a spatial-temporal relation module to make object and tracklet features compatible. Shen *et al.* [40] propose a Tracklet Association Tracker (TAT) to directly learn the association results from features which uses a bi-level optimization formulation to unit feature learning and data association.

There are research studies [10, 14, 24] that tackle MOT problems based on finding the most likely tracking proposals. Han *et al.* [14] propose an algorithm named trajectory tracking to find the optimal state sequence which maximizes the joint state-observation probability. Kim *et al.* [24] use Multiple Hypothesis Tracking (MHT) to train appearance models for each track hypothesis on all detections from the entire track. The method [49] formulates learning a similarity function for data association as learning a policy for the Markov Decision Processes (MDPs), where a trajectory corresponds to a MDP. Recurrent Autoregressive Network (RAN) [12] uses an external memory to store previous input features of each trajectory, and uses an internal memory to learn long-term tracking information and associate detections by decision making.

There are also methods [54, 34, 44, 47] that use graph models to link detections (or tracklets) in the graph into trajectories. The method [34] builds different graph structures to generate local tracklets and uses a hierarchical correlation clustering (HCC) framework to get globally associated tracks. Tang *et al.* [44] propose a graph-based formulation that clusters and associates person hypotheses over time by tackling a minimum cost lifted multicut problem. TrackletNet Tracker (TNT) [47] uses a graph model to generate tracklets based on appearance similarity and spatial consistency and measure the similarity of two tracklets by the multi-scale TrackletNet.

2.2. Multi-Target Multi-Camera Tracking

Recent approaches in MTMCT problem follow the two-step paradigm: 1) generating local tracklets of all targets within each single camera [3, 50, 47, 10]; 2) matching local tracklets across all cameras [20, 51, 52, 4]. Xu *et al.* [51] propose a hierarchical composition approach to adaptively exploit multiple cues such as ground occupancy consistency, appearance similarity and motion coherence. Bredereck *et al.* [4] present a greedy matching algorithm to iteratively match local tracklets across different cameras.

There is a large literature on the assumption of overlapping Fields of Views (FOV) between cameras to tackle the MTMCT task [13, 23, 2]. The method [13] uses a generative model to estimate the probabilistic occupancy map (POM), and combines these probabilities with color and motion model to process trajectories. Berclaz *et al.* [2] reformulate the association step as a constrained flow optimization which results in a convex problem, and use the k-shortest paths (KSP) algorithm to solve the problem.

There is another category of MTMCT works based on non-overlapping FOVs [46, 5, 28, 8, 38]. Cai *et al.* [5] and Lee *et al.* [28] exploit appearance cues to match local tracklets across cameras. Tesfaye *et al.* [46] propose a unified three-layer hierarchical framework based on the constrained dominant sets clustering (CDSC) technique to tackle tracking problems in non-overlapping cameras which uses the first two layers to solve within-camera tracking and exploits the third layer to match tracklets of the same object in all cameras in a simultaneous fashion. Chen *et al.* [6] use a piecewise major color spectrum histogram representation (PMCSHR) to match tracklet across multiple non-overlapping camera views. Cheng *et al.* [8] match local tracklets between every two cameras of interests.

There are also methods [18, 41, 26] that tackle MTMCT task based on graph models. Hofmann *et al.* [18] propose a maximum a posteriori (MAP) formulation to jointly model multi camera as well as temporal data association, and construct a constrained min-cost flow graph to track objects in 3D world space. Shitrit *et al.* [41] formulate multi-object tracking problem as a multi-commodity min-cost max-flow problem which involves a layered graph, where each possible spatial location has several grid cells, and one cell corresponds to one possible identity group. Leal *et al.* [26] define a graph structure which captures both temporal correlations between targets and spatial correlations enforced by camera configuration, and use Dantzig-Wolfe decomposition and branching to tackle the complex combinatorial optimization problem caused by the graph structure.

There are two relevant methods [17, 36] applicable to multi cameras with or without overlapping FOVs. MDA [36] enumerates all the tracklet matching hypotheses to nd the most likely hypothesis, and assigns tracklets in the same hypothesis to the same target. In the TRACklet-to-Target Assignment (TRACTA) [17], each tracklet is assigned to a unique target and the optimal assignment is computed by a restricted non-negative matrix factorization algorithm in[17].

3. Methodology

The proposed method contains two major modules: the local tracklet generation module and the cross-camera tracklet matching module. The input of the proposed method is M video sequence from M cameras. In the local tracklet generation module, we follow the tracking-by-detection paradigm to track target vehicles in each camera

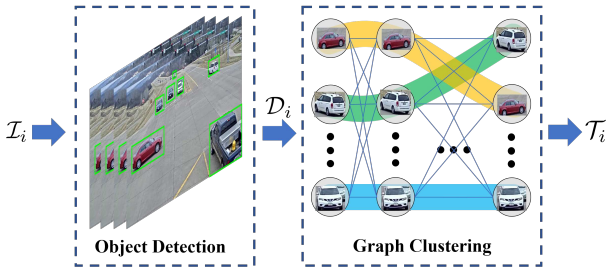

Figure 2: The pipeline of the local tracklet generation module, which first detects the targets in each frame and then links the detections to local tracklets by graph clustering.

and generate a local tracklet for each target. In the cross-camera tracklet matching module, we calculate the affinity of local tracklets in different cameras by semantic attribute parsing, and match the local tracklets belonging to the same target across different cameras by tracklet-to-target assignment. By reconnecting the split local tracklets of the same target in different cameras, each target obtains a complete trajectory across all the cameras. In the following, we provide detailed descriptions of the key techniques, *i.e.*, the local tracklet generation, the semantic attribute parsing and tracklet-to-target assignment, of the proposed framework.

3.1. Local Tracklet Generation

Given a video sequence from camera i, we aim to track targets robustly with target interaction and occlusion handling and maintain target identities across image frames. Due to the impressive progress of Deep Convolutional Neural Netowrk (DCNN), the DCNN based object detectors have achieved significant improvement. We employ the tracking-by-detection paradigm to track multiple targets in a single camera and generate a local tracklet for each target. As illustrated in Figure 2, for each camera i, the local tracklet generation module first detects the targets in each image frame and then links detections into local tracklets by graph clustering [47].

More specifically, we denote by $\mathcal{I}_i = \{\mathbf{I}_1, \mathbf{I}_2, ..., \mathbf{I}_T\}$ the input image sequence of camera i, where $\mathbf{I}_t$ is the image frame at time t and T is the length of the image sequence. Firstly, we detect targets in $\mathcal{I}_i$ using an object detector and denote by $\mathcal{D}_i = \{\mathcal{D}_i^1, \mathcal{D}_i^2, ..., \mathcal{D}_i^T\}$ the detection collection of the whole image sequence, where $\mathcal{D}_i^t$ is the detection set of the t-th image frame in camera i. Then we construct a weighted graph model $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ based on the detection collection $\mathcal{D}_i$, where each element $v_x \in \mathcal{V}$ in $\mathcal{V}$ denotes a detection observation in $\mathcal{D}_i$ and each element $e_{x,y} \in \mathcal{E}$ in $\mathcal{E}$ represents an edge between nodes x and y. Each edge $e_{x,y}$ is assigned with a connectivity cost $w_{x,y}$ that measures the cost of connecting nodes x and y:

$$w_{x,y} = \psi(v_x, v_y), \tag{1}$$

where $\psi(v_x, v_y)$ is the connectivity measurement algorithm in [47]. On this basis, the single camera multi-target tracking can be achieved by graph clustering, where nodes corresponding to the same target are clustered together, and each cluster corresponds to a local tracklet. We denote by $\mathcal{T}_i = \{\mathcal{T}_i^1, \mathcal{T}_i^2, ..., \mathcal{T}_i^{N_i}\}$ the local tracklet collection of camera i, where $\mathcal{T}_i^j$ is the j-th local tracklet in camera i and N_i is the tracklet number of $\mathcal{T}_i$.

3.2. Semantic Attribute Parsing

Taking the generated local tracklet set $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_M\}$ of the M cameras as input, we propose to calculate the affinity of local tracklets in different cameras by exploiting robust tracklet representations and prune infeasible matching candidates by traffic topology reasoning.

3.2.1 Robust Tracklet Representation

To measure the tracklet affinity in different cameras, we train a vehicle re-identification (ReID) model to produce a robust and discriminative representation for each input tracklet. As illustrated in Figure 3, we first extract the image-based appearance feature of each tracklet using an appearance feature extractor and then integrate the sequential features into a robust tracklet representation using a spatial-temporal attention mechanism.

Appearance Feature Extractor. We adopt the ResNet architecture [16] as the backbone of our feature extractor and add a bottleneck layer before the classification layer, which has been proved effective for robust ReID feature learning [33, 32]. For each input image frame, the feature extractor outputs a 2048-D feature vector of the fully connected-layer before the classification layer.

During tracking, the objective function of the feature extractor consists of two parts: the cross-entropy loss for identity classification and the triplet loss for metric learning. The input of the cross-entropy loss is the last fully-connected layer of the ReID model whose node number is equal to the number of the identities H. For each input image i, we denote by y_i the ground-truth one-hot label of i and p_i the prediction vector of the classification layer. The objective of identity classification is to ensure the prediction vector p_i is equal (or close) to the ground-truth label y_i, and the cross-entropy loss function can be written as:

$$L_{xent}(\mathbf{I}_i) = -\sum_{u=1}^{H} log(p_i(u)) \cdot y_i(u), \tag{2}$$

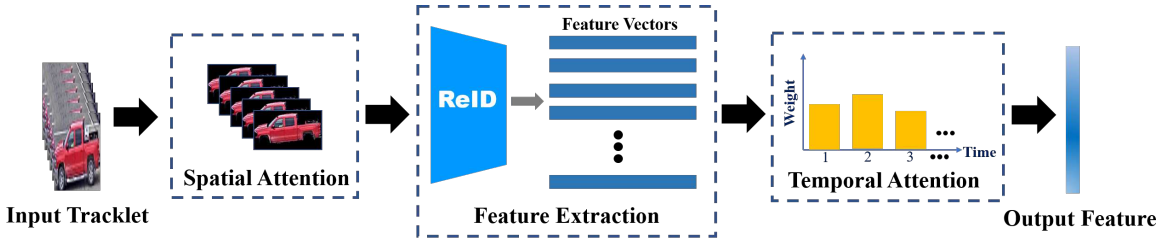

Figure 3: Illustration of the feature extraction procedure. Given an input local tracklet, the proposed method first exploits the spatial attention to mitigate the influence of the background and then integrates the extracted features into a final target representation using the temporal attention mechanism.

where H is the number of target identities and $log(\cdot)$ outputs the logarithm of the input scalar. To improve the robustness of the feature model, a label smooth mechanism [43] is exploited to prevent the over-fitting to the training identities, where the ground-truth one-hot label y_t is converted to a smooth label y'_i by:

$$y'_i(u) = \begin{cases} 1 - \frac{H-1}{H}\eta & \text{if } y_i(u) = 1, \\ \frac{\eta}{H} & \text{otherwise,} \end{cases} \tag{3}$$

where η is a small constant. On this basis, the cross-entropy loss with label smoothing can be written as:

$$L'_{xent}(\mathbf{I}_i) = -\sum_{u=1}^{H} log(p_i(u)) \cdot y'_i(u). \tag{4}$$

To constrain the obtained feature representation is robust and discriminative, *i.e.*, the feature distance of images from different targets is larger than the distance of the same targets, the triplet loss is adopted for discriminative feature learning. Given a constructed image triplet $\mathcal{I}_i = \{\mathbf{I}_i^a, \mathbf{I}_i^p, \mathbf{I}_i^n\}$ of image i, where $\mathbf{I}_i^a$ and $\mathbf{I}_i^p$ are images of the same target identity and $\mathbf{I}_i^n$ is an image with a different target identity. The objective of triplet loss is to constrain the feature distance between $\mathbf{I}_i^a$ and $\mathbf{I}_i^n$ is larger than the distance between $\mathbf{I}_i^a$ and $\mathbf{I}_i^p$ with a certain margin, which can be formulated as:

$$L_{trip}(\mathcal{I}_i) = \left[\|\varphi(\mathbf{I}_i^a) - \varphi(\mathbf{I}_i^p)\|_2 - \|\varphi(\mathbf{I}_i^a) - \varphi(\mathbf{I}_i^n)\|_2 + m\right]_+, \tag{5}$$

where $\varphi(\cdot)$ outputs the feature vector of the input image and m is the margin parameter.

The overall objective function is a combination of the triplet loss and the cross-entropy loss with label smoothing:

$$L = \sum_{i=1}^{O} L'_{xent}(\mathbf{I}_i) + \lambda L_{trip}(\mathcal{I}_i), \tag{6}$$

where O is the total number of training samples and λ is a weight parameter.

Spatial-Temporal Attention. To generate a robust representation of each tracklet, we propose a simple but efficient spatial-temporal attention mechanism to reduce the influence of background clutter and target occlusion in the tracklets. Each input tracklet $\mathcal{T}_i^j$ is composed of a set of aligned images of a target and we denote by $\mathbf{C}_{i,t}^j \in \mathbb{R}^{W \times H}$ the aligned image of $\mathcal{T}_i^j$ at time t, where W and H are the width and height of the aligned image, respectively.

The spatial attention mechanism aims to assign a low attention weight to the background region and a high attention weight to the target regions. Let $\mathbf{M}_{i,t}^j \in \mathbb{R}^{W \times H}$ be the attention map of $\mathbf{C}_{i,t}^j$, where each element $\mathbf{M}_{i,t}^j(u,v)$ in $\mathbf{M}_{i,t}^j$ denotes the attention weight of $\mathbf{C}_{i,t}^j(u,v)$. We then calculate the spatial weighted image $\mathbf{C}_{i,t}^{j^*}$ of $\mathbf{C}_{i,t}^j$ by:

$$\mathbf{C}_{i,t}^{j^*} = \mathbf{C}_{i,t}^j \odot \mathbf{M}_{i,t}^j, \tag{7}$$

where $\odot$ denotes the element-wise matrix multiplication. For the efficiency purpose, the attention map $\mathbf{M}_{i,t}^j$ in this paper is a binary mask that is calculated by the Mask-RCNN [15].

Then we feed the spatial weighted images to our feature extractor $\varphi(\cdot)$ to obtain the appearance feature of each image. To mitigate the influence of ineffective features when targets are mostly occluded or disappear, we propose to assign a low temporal attention weight to partially occluded targets and assign a high weight to full-body captured targets by:

$$w_{i,t}^j = \frac{||\mathbf{M}_{i,t}^j||_2}{\sum_{t \in \pi_i^j} ||\mathbf{M}_{i,t}^j||_2}, \tag{8}$$

where π_i^j is the time index set of tracklet $\mathcal{T}_i^j$. The robust feature representation $\mathbf{f}_i^j$ of the local tracklet $\mathcal{T}_i^j$ can be obtained by:

$$\mathbf{f}_i^j = \sum_{t \in \pi_i^j} \varphi(\mathbf{C}_{i,t}^{j^*}) \cdot w_{i,t}^j \,. \tag{9}$$

3.2.2 Traffic Topology Reasoning

In a real-world traffic network, the movements of vehicles are following the traffic rules and limited by the traffic topology. This allows us to model the movement of the traffic flows and prune infeasible tracklet matching candidates in different cameras by traffic topology reasoning.

Figure 4 (a) depicts a T-junction traffic topology of three cameras, where the green and yellow arrows point to the movement of two-direction traffic flows, respectively. As the movements of vehicles are usually consistent and the vehicles in different traffic flows (pointed by the green and yellow arrows, respectively) are disjunctive, infeasible local tracklet matching candidates (such as tracklets whose moving direction are opposite) can be pruned by exploiting the traffic topology. More specifically, for each camera i, we construct a traffic flow set $\mathcal{F}_i = \{\mathcal{F}_i^l, \mathcal{F}_i^r, \mathcal{F}_i^u, \mathcal{F}_i^d\}$ that collects the local tracklets in different moving directions, where $\mathcal{F}_i^l, \mathcal{F}_i^r, \mathcal{F}_i^u, \mathcal{F}_i^d$ are the tracklet collection for left, right, up, down directions, respectively. Based on the traffic topology structure, we conduct a topology matrix $\mathbf{B}_{i,j} \in \{0,1\}^{4\times4}$ for each pair of cameras i and j, where $\mathbf{B}_{i,j}(u,v) = 1$ denotes the u-th direction in camera i and the v-th direction in camera j are connected and $\mathbf{B}_{i,j}(u,v) = 0$ denotes the disconnection. As illustrated in Figure 4 (b), according to the connection between each camera pair, we estimate the traffic topology of the camera network. On this basis, we prune the matching candidates between disconnected camera pair, which both improves the cross-camera tracklet matching efficiency and accuracy.

Moreover, as there is no overlapping FOV between different camera views, each vehicle will not appear in multiple cameras at the same time. We further narrow the matching candidates by exploiting the during time of each tracklet, where tracklets in different cameras with overlapping in time are neglected in matching.

3.3. Tracklet-to-Target Assignment

In this section, the TRACklet-to-Target Assignment (TRACTA) algorithm in [17] is employed to match local tracklets across different cameras and the split local tracklets are reconnected into a complete trajectory for each target across all the cameras.

More specifically, given the generated local tracklets in the M cameras, *i.e.*, $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_M\}$, we aim to assign a unique Target ID (TID) label to each tracklet. Let N_i be the number of tracklets in $\mathcal{T}_i$ and $N = \sum_{i=1}^{M} N_i$ is the total number of trackets in $\mathcal{T}_i$. For each pair of cameras i and j, we calculate the pairwise similarity matrix $\mathbf{S}_{i,j} \in [0,1]^{N_i \times N_j}$ of tracklet set $\mathcal{T}_i$ and $\mathcal{T}_j$, where each element $\mathbf{S}_{i,j}(u,v)$ in $\mathbf{S}_{i,j}$ denotes the similarity of tracklet $\mathcal{T}_i^u$ and tracklet $\mathcal{T}_j^v$. The similarity $\mathbf{S}_{i,j}(u,v)$ of tracklets

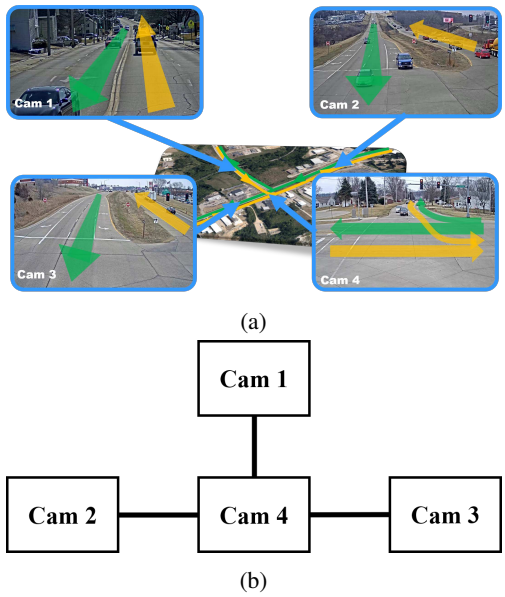

Figure 4: Illustration of the traffic topology reasoning, where (a) shows the traffic flows in a T-junction and (b) shows the traffic topology of the four-camera network in (a).

$\mathcal{T}_i^u$ and $\mathcal{T}_j^v$ is calculated by:

$$\mathbf{S}_{i,j}(u,v) = \begin{cases} exp(-||\mathbf{f}_i^u - \mathbf{f}_j^v||_2) & \text{if } \delta(\mathcal{T}_i^u, \mathcal{T}_j^v) = 1, \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

where $\delta(\mathcal{T}_i^u, \mathcal{T}_j^v) = 1$ denotes the tracklets $\mathcal{T}_i^u$ and $\mathcal{T}_j^v$ are connected in the traffic topology, $exp(\cdot)$ denotes the exponential function, and $\mathbf{f}_i^u$ and $\mathbf{f}_j^v$ denote the extracted tracklet representation of tracklets $\mathcal{T}_i^u$ and $\mathcal{T}_j^v$, respectively, using (9). The full tracklet similarity matrix $\mathbf{S} \in [0,1]^{N\times N}$ is constructed by the pairwise tracklet similarity matrices $\mathbf{S}_{i,j}, \forall\, i,j = 1, ..., M$.

As illustrated in Figure 5, different from the most existing tracklet-to-tracklet matching methods [7, 52, 19] that directly determine the matching or not matching of two tracklets, the TRACTA proposes to infer the matching of tracklets according to the assignment from tracklets to targets, where tracklets assigned to the same target are matched together and tracklets assigned to different targets are not matched. Compared with the tracklet-to-tracklet matching paradigm, the tracklet-to-target assignment has the following advantages: 1) When the number of tracklets for different targets is unknown and the occurrence of each target in different cameras is uncertain, it is difficult to determine how many tracklets should be matched with a specific tracklet. On the contrary, in TRACTA, each tracklet should be assigned to a unique target with no doubt. 2) The solution

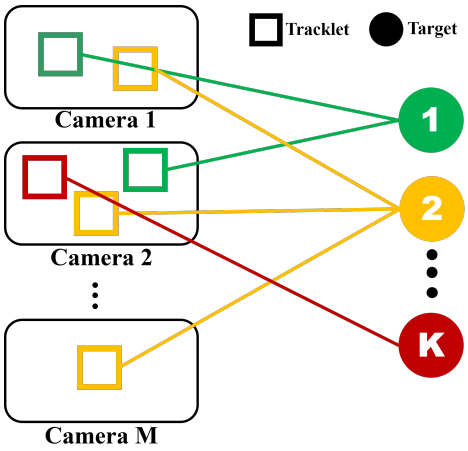

Figure 5: Illustration of the tracklet-to-target assignment algorithm, where the bounding box and circle denote the local tracklet and target identity, respectively, with the connection lines for the assignment of tracklets to targets.

spaces of the tracklet-to-tracklet matching and the TRACTA are $N \times N$ and $N \times K$ dimension, respectively, where N and K are the number of tracklets and targets, respectively. Since $K \ll N$ when there are multiple tracklets correspond to a target, solving the assignment problem is more efficient than the matching problem.

To optimize the assignment of local tracklets, we denote by $\mathbf{A}_i \in [0,1]^{N_i \times K}$ the assignment matrix from N_i tracklets to K targets in camera i. Each element $\mathbf{A}(u,v)$ in $\mathbf{A}$ denotes the probability of assigning tracklet $\mathcal{T}_i^u$ to target v. We estimate the total number of targets K by the target number estimation algorithm in TRACTA and conduct the full tracklet-to-target assignment matrix $\mathbf{A} \in [0,1]^{N \times K}$ with assignment matrices $\mathbf{A}_i, \forall\, i = 1, ..., M$ from M cameras.

The high similarity of two tracklets indicates that they might be assigned to the same target, *i.e.*, $\mathbf{S}(u,v) \rightarrow 1 \Rightarrow \mathbf{A}(u,:)\mathbf{A}(u,:)^T = 1$, and a low similarity is on the contrary, *i.e.*, $\mathbf{S}(u,v) \rightarrow 0 \Rightarrow \mathbf{A}(u,:)\mathbf{A}(u,:)^T = 0$. Thus, the $\mathbf{A}\mathbf{A}^T$ is an approximation of $\mathbf{S}$, *i.e.*, $\mathbf{A}\mathbf{A}^T \rightarrow \mathbf{S}$. On this basis, the tracklet-to-target assignment problem can be formulated as the following problem:

$$\mathbf{A}^* = \underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{S} - \mathbf{A}\mathbf{A}^T\|_2, \tag{11}$$

$$\text{s.t}\ \ \mathbf{A}\mathbf{I}_1 = \mathbf{I}_2, \tag{12}$$

where $\mathbf{I}_1$ and $\mathbf{I}_2$ are all-one vectors of dimension K and N, respectively, and Eq. (12) constrains the assignment from tracklets to targets is a mapping ($\forall u, \sum \mathbf{A}(u,:) = 1$).

The optimal solution of Eq (11) is obtained by the Restricted Non-negative Matrix Factorization (RNMF) in [17], and each tracklet u is assigned to a unique target v by finding the most likely target identity:

$$v = \underset{v}{\operatorname{argmax}}\ \mathbf{A}^*(u,v). \tag{13}$$

Based on the tracklet-to-target assignment results, we then generate global trajectory set $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_M\}$ by reconnecting the split local tracklets assigned to the same target across all the cameras, where $\mathcal{G}_i$ denotes the global trajectory set of camera i. We overview our tracking framework in Algorithm 1.

Algorithm 1 Tracking algorithm of the proposed method

Input: Image sequences collected from M cameras $\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_M\}$.
Output: Global trajectory set $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_M\}$.

1: **for** camera $i = 1 : M$ **do**
2: Generate local tracklet set $\mathcal{T}_i$ using single camera multi-object tracking technique.
3: **end for**
4: Generate robust representation $\mathbf{f}_i^j$ of each tracklet $\mathcal{I}_i^j$ using Eq. (9) and prune infeasible matching candidates by traffic topology reasoning.
5: Construct tracklet similarity matrix $\mathbf{S}$ using Eq. (10).
6: Compute tracklet-to-target assignment matrix $\mathbf{A}^*$ by optimizing Eq. (11).
7: Generate global trajectory set $\mathcal{G}$ according to $\mathbf{A}^*$.

4. Experiment

We participate in the track 3: the City-Scale Multi-Camera Vehicle Tracking task, of the 4-th AI City Challenge [1]. As described in [1], the benchmark dataset contains 215.03 minutes of video sequences collected from 46 cameras in a mid-sized U.S. city. The dataset contains 6 scenarios, including intersections, stretches of roadways and highways, where three of them are used for training, two are used for validation and the remaining one is used for testing. In total, there are nearly 300K bounding boxes annotated for 880 distinct annotated vehicle identities where each vehicle is at least captured by two cameras. The resolution of each video is at least 960p and the majority of the videos have a frame rate of 10 FPS.

4.1. Implementation Details

In the local tracklet generation module, the Mask-RCNN [15] is employed to detect vehicle in each image frame, where the NMS and confidence are fixed to 0.3 and 0.8, respectively. The parameter setting of the graph clustering follows to the parameters in [47]. In the semantic attribute parsing, the feature extractor is trained on the training data of the track2 and track3 of the AI City Challenge, and the weight parameter λ is set to 0.9 for training. For each input image, the input image is first resized to

256×128 and fed to the feature extractor model, and then the feature extractor outputs a 2048-D feature vector of the input image. In the tracklet-to-target assignment module, as the cameras are placed along with a roadway, we employ the TRACTA algorithm to match tracklets in adjacent cameras for computational efficiency.

4.2. Experimental Results

The widely adopted tracking evaluation metric IDF1 [37] is used for performance evaluation, which measures the trajectory consistency in the camera network. The final ranking result on the testing sequence is shown in Table 1, where our result is in bold. We can see that, the proposed method achieves the second-best result and significantly outperforms most of the competitive methods by a large margin.

Table 1: Comparison results with other teams

Rank	Team ID	IDF1 (%)
1	92	45.85
2	**11**	**44.00**
3	63	34.83
4	111	34.11
5	72	12.48
6	75	6.20
7	30	4.52
8	31	3.87

Moreover, to investigate the effectiveness of different components of the proposed method, an ablation experiment result is conducted in Table 2, where the *baseline* outputs the multi-camera tracking results based on greedy matching, *ST* denotes the spatial-temporal attention, *TT* denotes the traffic topology reasoning and *TRACTA* denotes the tracklet-to-target assignment. As shown in Table 2, the proposed method significantly improves the tracking performance with more than 10% improvement on IDF1.

Table 2: Ablation study

Method	IDF1 (%)	IDP (%)	IDR (%)
baseline	31.28	23.29	35.12
baseline+ST	34.51	29.54	41.50
baseline+ST+TT	38.61	47.19	32.80
baseline+ST+TT+TRACTA	44.00	53.63	37.31

5. Conclusion

In this paper, we propose a two-step approach for city-scale multi-camera vehicle tracking. The proposed method consists of two major steps: local tracklet generation and the cross-camera tracklet matching. Firstly, in local tracklet generation, we follow the tracking-by-detection paradigm and generate a local tracklet for each target by graph clustering. Secondly, taking the local tracklets from different cameras as input, the cross-camera tracklet matching step aims to match the local tracklets belonging to the same target across different cameras and produce a complete trajectory for each target across all the cameras. The proposed method is evaluated on the City-Scale Multi-Camera Vehicle Tracking task in the 2020 AI City Challenge and achieves the second-best result.

Acknowledgement

The authors express deep gratitude to National Key R&D Program (Grand No. 2019YFB1312000) and National Major Project of China (Grant No. 2017YFC0803905).

References

[1] Ai city challenge 2020 official website. `https://www.aicitychallenge.org`. Accessed: 2020-01-07.

[2] Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. Multiple object tracking using k-shortest paths optimization. *IEEE transactions on pattern analysis and machine intelligence*, 33(9):1806–1819, 2011.

[3] 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.

[4] Michael Bredereck, Xiaoyan Jiang, Marco Körner, and Joachim Denzler. Data association for multi-object tracking-by-detection in multi-camera networks. In *Distributed Smart Cameras (ICDSC), 2012 Sixth International Conference on*, pages 1–6. IEEE, 2012.

[5] Yinghao Cai and Gerard Medioni. Exploring context information for inter-camera multiple target tracking. In *IEEE Winter Conference on Applications of Computer Vision*, pages 761–768. IEEE, 2014.

[6] Weihua Chen, Lijun Cao, Xiaotang Chen, and Kaiqi Huang. A novel solution for multi-camera object tracking. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 2329–2333. IEEE, 2014.

[7] Weihua Chen, Lijun Cao, Xiaotang Chen, and Kaiqi Huang. An equalized global graph model-based approach for multi-camera object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(11):2367–2381, 2017.

[8] De Cheng, Yihong Gong, Jinjun Wang, Qiqi Hou, and Nanning Zheng. Part-aware trajectories association across non-overlapping uncalibrated cameras. *Neurocomputing*, 230:30–39, 2017.

[9] Peng Chu, Heng Fan, Chiu C Tan, and Haibin Ling. Online multi-object tracking with instance-aware tracker and dynamic model refreshment. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 161–170. IEEE, 2019.

[10] Ingemar J. Cox and Sunita L. Hingorani. An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on pattern analysis and machine intelligence*, 18(2):138–150, 1996.

[11] Ran Eshel and Yael Moses. Homography based multiple camera detection and tracking of people in a dense crowd. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[12] 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.

[13] Francois Fleuret, Jerome Berclaz, Richard Lengagne, and Pascal Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):267–282, 2007.

[14] Mei Han, Wei Xu, Hai Tao, and Yihong Gong. An algorithm for multiple object trajectory tracking. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. IEEE, 2004.

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] Yuhang He, Xing Wei, Xiaopeng Hong, Weiwei Shi, and Yihong Gong. Multi-target multi-camera tracking by tracklet-to-target assignment. *IEEE Transactions on Image Processing*, 29:5191–5205, 2020.

[18] Martin Hofmann, Daniel Wolf, and Gerhard Rigoll. Hypergraphs for joint multi-view reconstruction and multi-object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3650–3657, 2013.

[19] 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.

[20] Weiming Hu, Min Hu, Xue Zhou, Tieniu Tan, Jianguang Lou, and Steve Maybank. Principal axis-based correspondence between multiple cameras for people tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):663–671, 2006.

[21] Na Jiang, SiChen Bai, Yue Xu, Chang Xing, Zhong Zhou, and Wei Wu. Online inter-camera trajectory association exploiting person re-identification and camera topology. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 1457–1465. ACM, 2018.

[22] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[23] Sohaib Khan, Omar Javed, and Mubarak Shah. Tracking in uncalibrated cameras with overlapping field of view. In *2nd IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, volume 5, 2001.

[24] Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James M Rehg. Multiple hypothesis tracking revisited. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4696–4704, 2015.

[25] Chanho Kim, Fuxin Li, and James Rehg. Multi-object tracking with neural gating using bilinear lstm. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 200–215, 2018.

[26] Laura Leal-Taixe, Gerard Pons-Moll, and Bodo Rosenhahn. Branch-and-price global optimization for multi-view multi-target tracking. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1987–1994. IEEE, 2012.

[27] Kuan-Hui Lee and Jenq-Neng Hwang. On-road pedestrian tracking across multiple driving recorders. *IEEE Transactions on Multimedia*, 17(9):1429–1438, 2015.

[28] Young-Gun Lee, Zheng Tang, and Jenq-Neng Hwang. Online-learning-based human tracking across non-overlapping cameras. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10):2870–2883, 2017.

[29] Peilun Li, Guozhen Li, Zhangxi Yan, Youzeng Li, Meiqi Lu, Pengfei Xu, Yang Gu, Bing Bai, Yifei Zhang, and Di-Di Chuxing. Spatio-temporal consistency and hierarchical matching for multi-target multi-camera vehicle tracking. In *Proc. CVPR Workshops*, pages 222–230, 2019.

[30] Daniele Liciotti, Marco Contigiani, Emanuele Frontoni, Adriano Mancini, Primo Zingaretti, and Valerio Placidi. Shopper analytics: A customer activity recognition system using a distributed rgb-d camera network. In *International workshop on video analytics for audience measurement in retail and digital signage*, pages 146–157. Springer, 2014.

[31] Chongyu Liu, Rui Yao, S Hamid Rezatofighi, Ian Reid, and Qinfeng Shi. Model-free tracker for multiple objects using joint appearance and motion inference. *IEEE Transactions on Image Processing*, 29:277–288, 2019.

[32] Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. Bag of tricks and a strong baseline for deep person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[33] H. Luo, W. Jiang, Y. Gu, F. Liu, X. Liao, S. Lai, and J. Gu. A strong baseline and batch normalization neck for deep person re-identification. *IEEE Transactions on Multimedia*, pages 1–1, 2019.

[34] Liqian Ma, Siyu Tang, Michael J Black, and Luc Van Gool. Customized multi-person tracker. In *Asian Conference on Computer Vision*, pages 612–628. Springer, 2018.

[35] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957.

[36] Aubrey B Poore. Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking. *Computational Optimization and Applications*, 3(1):27–57, 1994.

[37] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, pages 17–35. Springer, 2016.

[38] Ergys Ristani and Carlo Tomasi. Features for multi-target multi-camera tracking and re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6036–6046, 2018.

[39] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. *arXiv preprint arXiv:1701.01909*, 4(5):6, 2017.

[40] Han Shen, Lichao Huang, Chang Huang, and Wei Xu. Tracklet association tracker: An end-to-end learning-based association approach for multi-object tracking. *arXiv preprint arXiv:1808.01562*, 2018.

[41] Horesh Ben Shitrit, Jérôme Berclaz, François Fleuret, and Pascal Fua. Multi-commodity network flow for tracking multiple people. *IEEE transactions on pattern analysis and machine intelligence*, 36(8):1614–1627, 2013.

[42] KA Shiva Kumar, KR Ramakrishnan, and GN Rathna. Distributed person of interest tracking in camera networks. In *Proceedings of the 11th International Conference on Distributed Smart Cameras*, pages 131–137. ACM, 2017.

[43] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[44] 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.

[45] 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.

[46] Yonatan Tariku Tesfaye, Eyasu Zemene, Andrea Prati, Marcello Pelillo, and Mubarak Shah. Multi-target tracking in multiple non-overlapping cameras using constrained dominant sets. *arXiv preprint arXiv:1706.06196*, 2017.

[47] 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.

[48] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *Image Processing (ICIP), 2017 IEEE International Conference on*, pages 3645–3649. IEEE, 2017.

[49] 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.

[50] Jiarui Xu, Yue Cao, Zheng Zhang, and Han Hu. Spatial-temporal relation networks for multi-object tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3988–3998, 2019.

[51] Yuanlu Xu, Xiaobai Liu, Yang Liu, and Song-Chun Zhu. Multi-view people tracking via hierarchical trajectory composition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4256–4265, 2016.

[52] Yuanlu Xu, Xiaobai Liu, Lei Qin, and Song-Chun Zhu. Cross-view people tracking by scene-centered spatio-temporal parsing. In *Processing of AAAI Conference on Artificial Intelligence*, pages 4299–4305, 2017.

[53] Fengwei Yu, Wenbo Li, Quanquan Li, Yu Liu, Xiaohua Shi, and Junjie Yan. Poi: Multiple object tracking with high performance detection and appearance feature. In *European Conference on Computer Vision*, pages 36–42. Springer, 2016.

[54] Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *Computer Vision–ECCV 2012*, pages 343–356. Springer, 2012.

[55] Bineng Zhong, Bing Bai, Jun Li, Yulun Zhang, and Yun Fu. Hierarchical tracking by reinforcement learning-based searching and coarse-to-fine verifying. *IEEE Transactions on Image Processing*, 28(5):2331–2341, 2018.

[56] Ji Zhu, Hua Yang, Nian Liu, Minyoung Kim, Wenjun Zhang, and Ming-Hsuan Yang. Online multi-object tracking with dual matching attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 366–382, 2018.