

# Multi-Target Multi-Camera Tracking of Vehicles Using Metadata-Aided Re-ID and Trajectory-Based Camera Link Model

Hung-Min Hsu<sup>✉</sup>, Member, IEEE, Jiarui Cai<sup>✉</sup>, Graduate Student Member, IEEE,  
Yizhou Wang<sup>✉</sup>, Graduate Student Member, IEEE, Jenq-Neng Hwang<sup>✉</sup>, Fellow, IEEE,  
and Kwang-Ju Kim<sup>✉</sup>, Associate Member, IEEE

**Abstract**—In this paper, we propose a novel framework for multi-target multi-camera tracking (MTMCT) of vehicles based on metadata-aided re-identification (MA-ReID) and the trajectory-based camera link model (TCLM). Given a video sequence and the corresponding frame-by-frame vehicle detections, we first address the isolated tracklets issue from single camera tracking (SCT) by the proposed traffic-aware single-camera tracking (TSCT). Then, after automatically constructing the TCLM, we solve MTMCT by the MA-ReID. The TCLM is generated from camera topological configuration to obtain the spatial and temporal information to improve the performance of MTMCT by reducing the candidate search of ReID. We also use the temporal attention model to create more discriminative embeddings of trajectories from each camera to achieve robust distance measures for vehicle ReID. Moreover, we train a metadata classifier for MTMCT to obtain the metadata feature, which is concatenated with the temporal attention based embeddings. Finally, the TCLM and hierarchical clustering are jointly applied for global ID assignment. The proposed method is evaluated on the CityFlow dataset, achieving IDF1 76.77%, which outperforms the state-of-the-art MTMCT methods.

**Index Terms**—MTMCT, multi-camera tracking, traffic-aware single camera tracking, trajectory-based camera link model, vehicle ReID, hierarchical clustering.

## I. INTRODUCTION

Due to the exponential growth of intelligent transportation systems, multi-target multi-camera tracking is becoming one of the important tasks. The purpose of MTMCT is to identify and locate targets in a multi-camera system. For

Manuscript received September 26, 2020; revised February 23, 2021 and March 28, 2021; accepted April 19, 2021. Date of publication May 17, 2021; date of current version May 26, 2021. This work was supported in part by the Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean Government (Development of ICT Convergence Technology for Daegu-GyeongBuk Regional Industry) under Grant 20ZD1100. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xiaolin Hu. (*Corresponding author: Hung-Min Hsu*)

Hung-Min Hsu, Jiarui Cai, Yizhou Wang, and Jenq-Neng Hwang are with the Department of Electrical and Computer Engineering, University of Washington at Seattle, Seattle, WA 98195 USA (e-mail: hmhsu@uw.edu; jcrai@uw.edu; ywang26@uw.edu; hhwang@uw.edu).

Kwang-Ju Kim is with the Daegu-Gyeongbuk Research Center, Electronics and Telecommunications Research Institute (ETRI), Daegu 42994, South Korea (e-mail: kwangju@etri.re.kr).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TIP.2021.3078124>, provided by the authors.

Digital Object Identifier 10.1109/TIP.2021.3078124

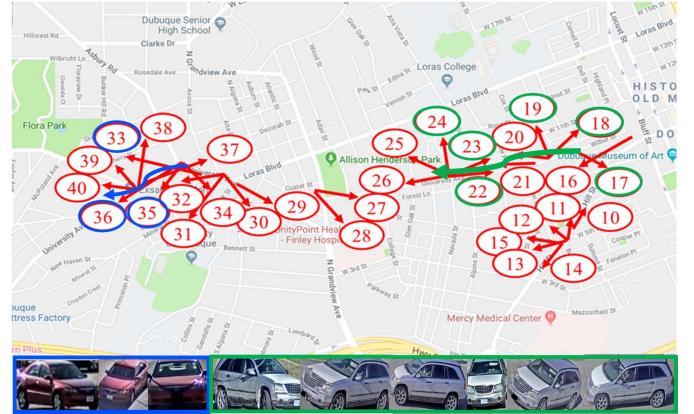


Fig. 1. Illustration for MTMCT of vehicles. Given any vehicle in videos recorded by several time-synchronized cameras with/without overlapping field of views (FoVs), the MTMCT task is aimed to track the same vehicle in all the cameras.

instance, Fig. 1 shows there are two vehicles tracked by an MTMCT system. However, there are some fundamental problems in detecting and tracking need to be solved to achieve this goal. Basically, most of the MTMCT systems are composed of two modules, i.e., single camera tracking (SCT) and inter-camera tracking (ICT). SCT aims to track the vehicle trajectories within a single camera. In terms of ICT, it is to re-identify the vehicle trajectories across multiple cameras by vehicle re-identification (ReID) [1].

In fact, MTMCT is a very complicated task since SCT and ReID are both challenging research topics. In SCT, the tracking-by-detection scheme has been proven effective in many works [2], [3]. Tracking-by-detection is composed of two steps: (i) object detection in each frame, (ii) trajectory generation by associating the corresponding detections across time. However, it is difficult to track vehicles within a single camera due to unreliable vehicle detection and heavy occlusion. For ReID, the different orientations of the same vehicle, the same car model from different vehicles, low resolution of video, and varying lighting conditions are all intractable problems. The poor performance of SCT and ReID can cause dramatic ID switching. If the trajectories are lost in SCT, ID switching will happen in MTMCT. On the other hand, the ReID model also needs the capability to identify the appearance of the same target in varied illuminations and

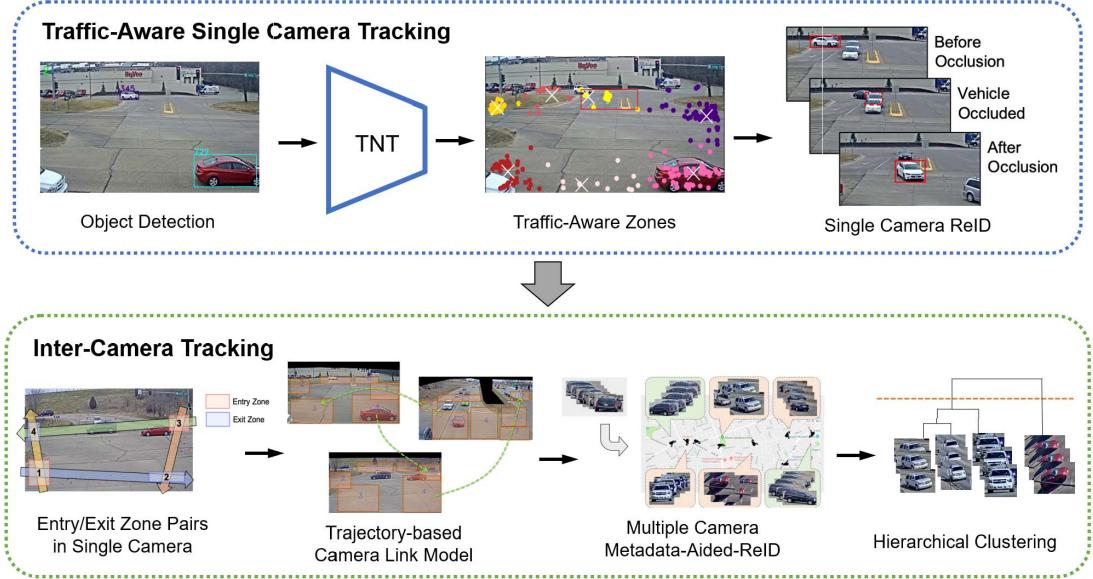


Fig. 2. The illustration of the proposed MTMCT framework. First, the TSCT is adopted as the single camera tracker to obtain SCT results so that entry, exit, and traffic-aware zones can be generated. Then, trajectory-based camera link model (TCLM) with transition time are automatically generated based on entry/exit zones and training data. Finally, metadata-aided ReID is applied to the solution space with the camera link constraint, and the hierarchical clustering is used for generating the final MTMCT results.

viewing angles in different cameras to avoid ID switching. To alleviate the mentioned issues, a more robust appearance feature is critically needed for SCT and ReID. To connect the vehicular tracks cross cameras, appearance feature based vehicle ReID is one of the most effective approaches. For vehicle ReID, some works focus on generating discriminative features by deep convolutional neural networks (CNNs). In most of the methods, the trained ReID model is used to extract effective embedding features, which can be used to estimate the similarity based on Euclidean distance between trajectories in the testing stage. To the best of our knowledge, there is no MTMCT system using the metadata information, such as brand, type and color of vehicles for ICT. Nonetheless, many vehicles are of the same brand, type and color but these vehicles are not the same identity. Thus, we assert that not only metadata information but also spatial and temporal information are all critical information for MTMCT.

In this paper, we propose an innovative framework for MTMCT of vehicles to solve all the mentioned issues. The flowchart of our proposed MTMCT system is shown in Fig. 2. First, we use a TrackletNet Tracker (TNT) [4] in the SCT, which has a superior performance in many intelligent transportation system applications [5], [6]. Then based on the appearance feature similarity and bounding box intersection-over-union (IOU) between consecutive frames, the detection results are associated into tracklets. For the neighboring disjoint tracklets, we estimate their similarity by a Siamese TrackletNet based on both appearance and temporal information. A graph model is then built with tracklets being treated as vertices and similarities between two tracklets as measured by the TrackletNet being treated as edge weights. Then the graph is partitioned into small groups, where each group can represent a unique vehicle ID and moving trajectory in

each camera. To enhance the performance of SCT, traffic-aware single camera tracking (TSCT) is further proposed to solve the long-term occlusion due to the traffic scenario. Then, we combine the temporal attention weighted clip-level deep appearance feature and metadata of the vehicle as the auxiliary information to establish a more representation feature for ReID across different cameras [7]. Basically, there are three metadata information: car type, brand and color. Finally, we propose the trajectory-based camera link model (TCLM) to obtain the spatial and temporal information to reduce the ambiguity of the different identities with the same car model. By exploiting TCLM, the dependency of well-embedded appearance can be mitigated based on these spatial and temporal constraints. To summarize, we claim the following contributions,

- The traffic-aware single camera tracking (TSCT) is proposed to achieve the best performance in the MTMCT task.
- Combining clip-level appearance feature and meta information to generate the feature of each trajectory for ReID across cameras.
- Trajectory-based camera link model (TCLM) are constructed to exploit the spatial and temporal information to enhance the performance of MTMCT.

The rest of this paper is organized as follows. Section II reviews related works. Then, we elaborate the proposed MTMCT system in Section III. In Section IV, we evaluate the proposed method on the CityFlow dataset [8]–[10] and compare it with the state-of-the-art methods. Finally, the paper is concluded in Section V.

## II. RELATED WORKS

A large amount of literature on person ReID and MTMCT have attracted growing attention in the past few years.

In addition, some works tackle vehicle ReID due to smart-city-related applications. In this section, we discuss the most relevant research works to the MTMCT tasks by the following two parts: overlapping field of view (FOVs) and non-overlapping FOVs.

### A. Overlapping FOVs

There are many research studies to solve the MTMCT tasks with overlapping FOVs between cameras. Fleuret *et al.* [11] use probabilistic occupancy map (POM) with color and motion attributes for MTMCT. Berclaz *et al.* [12] formulate the MTMCT task as an integer programming problem and deal with the problem by the  $k$ -shortest paths (KSP) algorithm. Moreover, some research works use graph modeling algorithms to solve the MTMCT problem. For example, Hofmann *et al.* [13] and Shitrit *et al.* [14] use a constrained min-cost flow graph approach to associate the detections frame-by-frame. Leal-Taixe *et al.* [15] formulate the MTMCT problem as a multi-commodity network flow problem and use the branch-and-price (B&P) algorithm to link detections into trajectories. Recently, the two-step approach for the MTMCT problem is becoming more and more popular. The main idea of the two-step approach is firstly to track all the targets within a single camera, then match the generated trajectories from every single camera to all the other cameras. Therefore, the first step is SCT, which is intensively studied in computer vision and pattern recognition communities [16]–[21]. Thanks to the great advances of convolution neural networks (CNNs), object detection techniques have been shown to achieve impressive performance in recent years, tracking-by-detection scheme [22]–[26] has thus become the mainstream approach for multiple object tracking. After SCT, the next step is to match local trajectories on different cameras. Hu *et al.* [27] and Eshel and Moses [28] associate the trajectories across cameras via an epipolar geometry constraint, which predicts the bounding boxes of those targets into a reference plane on the next frame and determines those targets with intersections in the reference plane as the same identity. Bredereck *et al.* [29] propose a Greedy Matching Association (GMA) method to iteratively associate single-camera trajectories to generate the cross-camera trajectories. Xu *et al.* [30] propose a Hierarchical Composition of Tracklet (HCT) framework to match local tracklets to the other cameras by using the appearance feature and the ground plane locations. Xu *et al.* [31] further propose the Spatio-Temporal Parsing (STP) structure, which prunes matching candidates of trajectories by using semantic attributes of targets, to solve the tracklet matching problem.

### B. Non-Overlapping FOVs

The other category of research works that aim at the MTMCT task with non-overlapping FOVs. Research studies in this category attempt to match single-camera trajectories across different non-overlapping FOV cameras by exploiting different information such as appearance feature [32], [33], motion pattern [13], and camera topological configuration [34]. For appearance cues, Cai and Medioni [33] use a Relative Appearance Context (RAC) to differentiate

adjacent targets. Chen *et al.* [35] exploit the Piecewise Major Color Spectrum Histogram Representation (PMCSHR) to estimate the similarity of targets in different views by generating a major color histogram for each target. Zhang *et al.* [36] use Convolutional Neural Networks (CNNs) to generate the feature representation for each target and propose a Feature Re-Ranking mechanism (FRR) to find correspondences among tracklets. Ristani and Tomasi [37] consider not only the CNN based appearance feature but also motion pattern. Moreover, they formulate the MTMCT task as a binary integer program problem and propose deep feature correlation clustering (DeepCC) approach to match the trajectories of a single camera to all the other cameras. Chen *et al.* [38] propose an Equalized Graph Model (EGM) to find the solution of trajectory assignment by defining a trajectory in each camera as a node in the graph and edges are the connections of trajectories. The final cross camera trajectory of each target is to find the min-cost flow from the source node to the sink one. Chen and Bhanu [39] propose social grouping in MTMCT, which uses Conditional Random Field (CRF) to match single-camera trajectories across different cameras by minimizing the unary and pairwise energy costs. Besides, Ye *et al.* [40] design a dynamic graph matching (DGM) framework for video-based person ReID, which utilizes a co-matching strategy to reduce the false matching in MTMCT.

Some research works consider the camera topology in MTMCT. For example, [41], [42] attempt to match local tracklets between every two neighboring cameras. Lee *et al.* [34], [43] present a fully unsupervised online learning approach, which efficiently integrates discriminative visual features by the proposed Two-Way Gaussian Mixture Model Fitting (2WGMMF) and context feature, to systematically build camera link model so as to match single-camera trajectories to the neighboring cameras. On the other hand, Wang *et al.* [44] propose consistent cross-view matching (CCM) framework to use the global camera network constraints for video-based person ReID, which also proves the effectiveness of using the camera topology for MTMCT.

## III. PROPOSED MTMCT FRAMEWORK

There are four steps in the proposed MTMCT framework: (1) Apply traffic-aware single camera tracking (TSCT) to generate SCT results. (2) Train a ReID model and metadata classifier to extract the appearance feature and metadata feature of each trajectory. (3) Establish the TCLM (i.e., the spatial and temporal constraint) for ICT. (4) Use the feature of each trajectory and TCLM to generate the ICT results. We show the notation table in Table I and the whole framework in Alg. 1.

### A. Traffic-Aware Single Camera Tracking

In an MTMCT system, the first step is to perform SCT. Before we illustrate our SCT approach, we provide the definition of the SCT problem first. The input of MTMCT is  $V$  videos from  $V$  cameras so that we can denote the global trajectory set as  $\mathcal{T} = \{\Xi^1, \Xi^2, \dots, \Xi^V\}$ . Each element  $\Xi^i$  in  $\mathcal{T}$  indicates local trajectory set, which includes all trajectories in camera  $i$ . We then define the local trajectory set as

TABLE I  
THE NOTATION DEFINITION

Symbol	Description
$c_i^t$	A candidate centroid $i$ at iteration $t$ .
$N(c_i^t)$	The neighborhood of samples within a given distance around $c_i$ .
$N_{e,k}$	Number of entry points in each zone.
$N_{x,k}$	Number of exit points in each zone.
$D_e$	Entry zone density.
$D_x$	Exit zone density.
$D_{ta}$	Traffic-aware zone density.
$\mathcal{M}(\xi_j)$	Metadata feature of a trajectory $\xi_j$ .
$C^s$	Zone pair trajectories set in the source camera.
$C^d$	Zone pair trajectories set in the destination camera.
$P_i^s$	$i$ th zone pair trajectory in $C^s$ .
$P_j^d$	$j$ th zone pair trajectory in $C^d$ .
$z_s$	Transition zone of $P_i^s$ .
$z_d$	Transition zone of $P_j^d$ .
$\alpha_z$	Overlapping ratio of the vehicle to zone $z$ .

### Algorithm 1 The Proposed MTMCT

---

**Input :** Detections set  $\mathcal{D}$  from all  $V$  cameras.  
**Output:** Global ID for all trajectories within all  $V$  cameras.

```

1 for camera  $i$  to  $V$  do
2    $\Xi^i \leftarrow \text{TNT}(\mathcal{D}_i)$  // generate trajectories
    set  $\Xi^i = \{\xi_n^i\}$  of camera  $i$ 
3    $Z^i \leftarrow \text{MeanShift}(\Xi^i)$  // generate zones  $Z^i$ 
    from trajectories  $\Xi^i$ 
4   Calculate the exit density  $D_x$  and entry density  $D_e$ 
    to classify the traffic-aware zone, exit zone and
    entry zone;
5   Merge the isolated trajectories in the traffic-aware
    zones;
6   for trajectory  $\xi_j^i$  from  $\Xi^i$  do
7      $\mathcal{A}(\xi_j^i) \leftarrow \text{TA-ReID}(\xi_j^i)$  // extract
       appearance embedding using
       the Temporal Attention ReID
       model
8      $\mathcal{M}(\xi_j^i) \leftarrow \text{MetadataClassifier}(\xi_j^i)$  // obtain
       metadata features
9      $f(\xi_j^i) = \mathcal{A}(\xi_j^i) \oplus \mathcal{M}(\xi_j^i)$ ;
10  end
11 end
12 Global IDs  $\leftarrow \text{HierarchicalClustering}(f(\Xi))$ ;
```

---

$\Xi^i = \{\xi_1^i, \xi_2^i, \dots, \xi_j^i\}$ , where  $i$  is the index of camera and  $j$  denotes the index of trajectory in camera  $i$ . Therefore, the purpose of SCT is to produce  $\Xi^i$ .

Overall, the procedure of the proposed traffic-aware single camera tracking (TSCT) is as follows: (1) Use TNT to generate trajectories set  $\Xi$  of each camera. (2) Use  $\Xi$  as the input of MeanShift to generate the zones. (3) Calculate the exit density  $D_x$ , entry density  $D_e$  and traffic-aware zone density  $D_{ta}$  to classify the traffic-aware zone, exit zone and entry zone. (4) Merge the isolated trajectories in the traffic-aware zones based on appearance feature similarity and IOU for trajectory refinement. We will illustrate the intuition and the detail of each step in the following paragraphs.

The main idea of TSCT is to take advantage of traffic rules to refine the generated trajectories from SCT by merging the isolated trajectories, which are the trajectories that suddenly disappear or appear in the middle of the image. Here we use the TrackletNet Tracker (TNT) [4] as our SCT tracker. TNT is a tracklet based graph clustering model, where each vertex in the graph is a tracklet instead of a detection. In order to establish a tracklet based graph, we need to generate tracklets as vertices. In [4], tracklet generation is based on detection association. First, we use metric learning to train our CNN based appearance feature extractor. Then, the tracklets are generated based on the appearance similarity and the intersection-over-union (IOU) from all detections between two consecutive frames. The tracklets are represented as the vertices of the tracklet based graph. In terms of the edges of a tracklet based graph, the edge weights are estimated by a TrackletNet, which is a Siamese neural network trained to calculate likelihood of the two tracklets being from the same object. A TrackletNet combines both temporal and spatial features as input to estimate the likelihood. After tracklet based graph construction, clustering [45] is applied to merge tracklets from the same vehicle into one group.

The purpose of TSCT is to use traffic rules to improve the SCT results [46]. Therefore, the first step is to generate the traffic-aware zones. We notice that there are many isolated trajectories in the generated SCT results and most of them locating in the same place. It turns out that the vehicles need to wait for the stop sign or traffic light. Therefore, the window size for TNT to associate the detections results is not long enough to cover the necessary time for the traffic rules. To solve this problem, we propose an unsupervised solution to generate the traffic-aware zones, which can be exploited to reconnect these isolated trajectories to achieve trajectory refinement. In TSCT, we use MeanShift to produce these traffic-aware zones based on exit/entry point of each trajectory. Therefore, the procedure of zone generation can be divided into the following steps. First of all, we define the entry point as the first position  $P_{j,f}$  and the exit point as the last position  $P_{j,l}$  of the  $j$ th trajectory, then we apply these points as the input of MeanShift algorithm. The main idea here is to refer these points as nodes, which can be clustered to generate several clusters as zones. Since each generated cluster can be encompassed by a rectangular bounding box, which represents a zone. We can categorize these clustered zones into three types: exit zone, entry zone and traffic-aware zone. The use of exit zones and entry zones, as part of the TCLM, which will be discussed in the Section III-C. On the other hand, the traffic-aware zone is used to reconnect the isolated trajectories, which are trajectories terminated or initiated in this zone. Here we use MeanShift as the clustering approach, which works by seeking the candidates for centroids within a given region. Given a candidate centroid  $c_i$  at iteration  $t$ , the candidate is updated according to  $c_i^{t+1} = m(c_i^t)$ .  $N(c_i)$  is the neighborhood of samples within a given distance around  $c_i$  and  $m$  is the mean shift vector which is calculated for each centroid that indicates towards a region of the maximum increase in the density of nodes. The following equation effectively updates a centroid

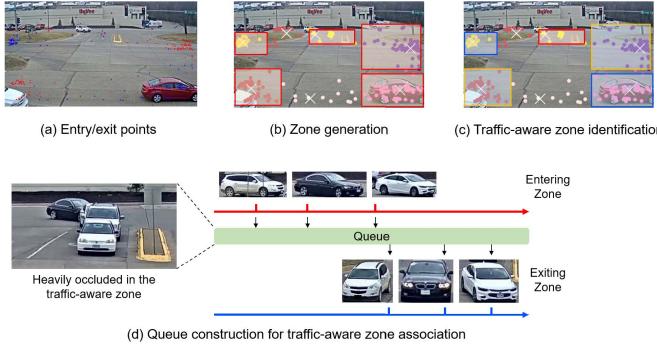


Fig. 3. Traffic-aware zone generation. (a) Exit/entry points of all trajectories are obtained from SCT (red: entry; blue: exit). (b) The generated zones from MeanShift. (c) The traffic-aware zones, entry zones and exit zones are identified based on traffic-aware zone density, entry zone density and exit zone density, respectively (red: traffic-aware zone; yellow: entry zone; blue: exit zone). (d) A queue for isolated trajectories is maintained to keep the ordering for following single camera ReID.

to be the mean of the samples among its neighborhood:

$$m(c_i^t) = \frac{\sum_{c_j \in N(c_i^t)} K(c_j - c_i^t) c_j}{\sum_{c_j \in N(c_i^t)} K(c_j - c_i^t)}, \quad (1)$$

$$K(c_j - c_i^t) = \exp\left(-\frac{\|c_j - c_i^t\|}{2\sigma^2}\right). \quad (2)$$

Here  $\sigma$  is the bandwidth of radial basis function kernel  $K$ , which is a parameter used to indicate the size of the region to search through.

After MeanShift, we generate the encompassing bounding boxes for each cluster as zones in the camera and compute the entry/exit zone density to determine the type for each clustered zone. The number of nodes in each zone needs to be over a specific threshold; otherwise, the zone will be removed. Thus, the entry and exit zone densities in each zone are defined as  $D_e$  and  $D_x$ , where

$$D_e = \frac{N_{e,k}}{N_{e,k} + N_{x,k}}, \quad D_x = \frac{N_{x,k}}{N_{e,k} + N_{x,k}}. \quad (3)$$

$N_{x,k}$  and  $N_{e,k}$  are the number of exit points and entry points in each zone, respectively. If the density of an entry or exit zone is higher than a threshold  $\rho_e$  or  $\rho_x$ , this zone will be recognized as an entry or exit zone, respectively. In terms of traffic-aware zone, the traffic-aware zone density  $D_{ta}$  is defined by

$$D_{ta} = 1 - \frac{|N_{e,k} - N_{x,k}|}{N_{e,k} + N_{x,k}}, \quad (4)$$

where  $D_{ta}$  needs to be above a threshold  $\rho_{ta}$ , then the zone will be designated as a traffic-aware zone.

$$Z = \begin{cases} \text{entry zone} & \text{if } D_e > \rho_e, \\ \text{exit zone} & \text{if } D_x > \rho_x, \\ \text{traffic-aware zone} & \text{if } D_{ta} > \rho_{ta}, \\ \text{don't care} & \text{otherwise.} \end{cases} \quad (5)$$

Finally, we trained a ReID model specifically for the isolated trajectories reconnection. The TSCT follows the First-In-First-Out (FIFO) strategy to merge trajectories, i.e., the TSCT takes into account the order and temporal constraint into the

tracklet grouping. An example of TSCT is illustrated in Fig. 3, where three vehicles are waiting for the traffic sign so that there are three exit points in the traffic-aware zone. Consequently, there are three new tracklets appearing in the traffic-aware zone due to the changes of traffic sign. We keep the temporal ordering of the three exit points and adopt the FIFO strategy to reconnect these three corresponding tracklets corresponding to these three exit points to the newly appeared tracklets. Specifically, we use FIFO and the order constraint (i.e., Fig. 7(b)) for the single camera ReID.

## B. Metadata-Aided ReID

Metadata-Aided ReID is to combine appearance embedding and metadata feature to generate the final embedding feature, then we can use the final embedding feature for ICT. In this section, we will illustrate the training procedures of our appearance ReID model and metadata feature.

After TNT tracking, we obtain the tracking results for each camera, which are the input data of ReID. In other words, we can take advantage of a sequence of images, instead of a single image, since video-based ReID can achieve better performance than image-based ReID. For frame-level feature extraction, we adopt the ResNet-50 [47] network pre-trained on ImageNet as our feature extractor, and the appearance feature of an object is obtained from the 2048-dim fully-connected layer. After frame-level feature extraction, temporal information can be further taken advantage of to establish a more discriminative feature. To this end, we use temporal attention (TA) mechanism [48] to perform a weighted average the frame-level feature and create a clip-level feature. The main idea is that some frames of the object might be highly occluded by other objects, then we want to lower the weight of these frames. We set a specific number of frames to do the TA to generate the clip-level feature. Finally, we add another average pooling layer for these clip-level feature  $f_c$  to generate the final tracklet-level feature.

There are two convolutional networks, which are spatial convolutional network and temporal convolutional network, used in the TA model. Assume the clip size is  $c$ , the spatial convolutional network is a 2D convolution operation to produce  $c \times 256$ -dim feature vectors, then we apply 1D temporal convolutional network with kernel size 3 to generate an attention vector for weighting the frame-level feature so that the clip-level feature  $f_c$  can be created.

For the network training, we adopt the metric learning strategy by using the batch sample (BS) [49] in the triplet generation. In terms of the loss function, there are two types of loss to be jointly optimized, which are BS triplet loss and cross-entropy loss. Thus, the final loss function of our network is a combined loss, described as follows:

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{BStri} + \lambda_2 \mathcal{L}_{Xent}. \quad (6)$$

First, the triplet loss is to minimize the feature distance of the same identity and maximize the feature distance of different identity pairs [50]. In this paper, we adopt BS triplet loss to calculate triplet loss in a minibatch  $B$ , which can be

defined as,

$$\mathcal{L}_{BStri}(\theta; \xi) = \sum_b \sum_{a \in B} l_{triplet}(a), \quad (7)$$

where

$$l_{triplet}(a) = \left[ m + \sum_{p \in P(a)} w_p D_{ap} - \sum_{n \in N(a)} w_n D_{an} \right]_+. \quad (8)$$

$m$  denotes the margin,  $D_{ap}$  and  $D_{an}$  represent the distances between the anchor sample  $a$  to the positive instance and negative instance, respectively.  $w_p$  and  $w_n$  are the weights of positive and negative instances, which are defined as follows,

$$\begin{aligned} w_p &= P(w_p == \text{multinomial}_{x \in P(a)}\{D_{ax}\}), \\ w_n &= P(w_n == \text{multinomial}_{x \in N(a)}\{D_{ax}\}), \end{aligned} \quad (9)$$

where  $P(a)$  and  $N(a)$  are positive and negative instances, respectively.

The cross-entropy (Xent) loss [24] in the training is defined as follows,

$$\mathcal{L}_{Xent} = -\frac{1}{N} \sum_{j=1}^N y_j \log(\hat{y}_j), \quad (10)$$

where  $\hat{y}_j$  is the estimated probability of the probe object that belongs to object  $j$ , and  $y_j$  denotes the ground truth vector while  $N$  denotes the number of identities in training data.

**Metadata classification.** In the proposed Metadata-Aided ReID, vehicle type, brand and color are considered as vehicle metadata attributes. We adopt a 29-layer Light CNN framework [51], including small kernel sizes of convolution layers, network-in-network layers and residual blocks. This Light CNN architecture is used to reduce the parameter space to improve the performance of metadata classification. The max-feature-map (MFM) operation is an extension of maxout activation, which combines two feature maps and outputs element-wise maximum value. Finally, we use the output layer of this Light CNN framework as the metadata feature for vehicle ReID. Assume the metadata feature of trajectory  $j$  can be represented as  $\xi_j = \{a_1, \dots, a_n\}$ , and  $a_i$  denotes the output of the Light CNN for the  $i$ th frame of  $\xi_j$  (i.e., the probability distribution of metadata classification). Thus, the final metadata feature  $\mathcal{M}(\xi_j)$  of a trajectory  $\xi_j$  is defined as follows:

$$\mathcal{M}(\xi_j) = \frac{1}{n} \sum_{i=1}^n a_i. \quad (11)$$

In this work, we utilize data augmentation to achieve better results of metadata classification. First of all, we use the car pose algorithm to produce the 36 keypoints for each vehicle [52]. Based on the 36 vehicle keypoints, we can mark the four wheels as  $P_{front, left}$ ,  $P_{front, right}$ ,  $P_{back, left}$  and  $P_{back, right}$ . Then, the direction of a vehicle can be represented by a vector that is pointing from the center of the back axle to the center of the front axle,

$$\begin{aligned} \vec{r} &= ((P_{back, left} + P_{back, right})/2, \\ &\quad \times (P_{front, left} + P_{front, right})/2). \end{aligned} \quad (12)$$

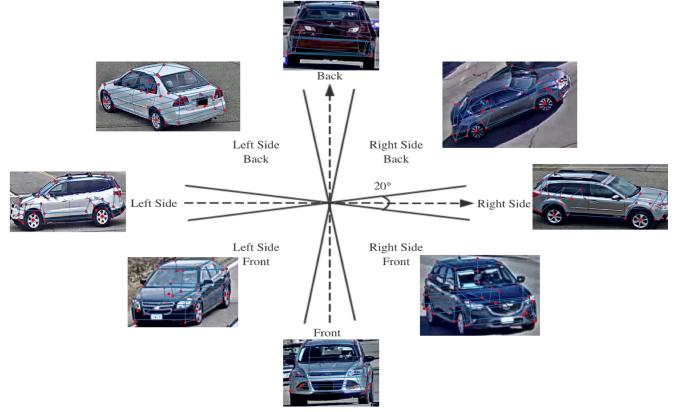


Fig. 4. Orientation view perspectives for data augmentation.



Fig. 5. Examples of vehicle keypoints detection and visibility estimation. The red, green, orange and purple outlines represent the front, back, left side and right side of vehicles, respectively. Notice that the yellow arrow represents the driving direction  $\vec{r}$ , and the data augmentation is based on the driving direction angle of  $\vec{r}$  to sample different car orientations for training.

Therefore, we can define the direction of a vehicle through the angle of  $\vec{r}$  with respect to the horizontal right-sided view perspective, i.e.,  $0^\circ$ . Fig. 4 shows that the 2D space is split into 8 regions which are calculated by

$$\mathcal{R} = \begin{cases} [\theta - \phi, \theta + \phi] & \text{if } \theta \in \{0^\circ, 90^\circ, 180^\circ, 270^\circ\} \\ [\theta, \theta + \omega] & \text{if } \theta \in \{10^\circ, 100^\circ, 190^\circ, 280^\circ\}. \end{cases} \quad (13)$$

Here the  $\phi$  and  $\omega$  are set as  $10^\circ$  and  $70^\circ$ , respectively. Fig. 5 shows an example of using vehicle keypoints to estimate the vehicle directions and visibility. In the data augmentation, we cluster all the images by these 8 different vehicle directions and enlarge the input size to be  $512 \times 512$ , the aspect ratio is retained by zero paddings. We believe the vehicle directions including the semantic meaning based on the visible surfaces can be exploited to diversify the variation of training data. We also slightly change the architecture of [51] by adding one network-in-network layer and extend the dimensions of the fully-connected layer from 256 to 2048. The classes of metadata and implementation details are illustrated in Section IV.

### C. Trajectory-Based Camera Link Model

Camera link model (CLM) is to exploit the camera topological configuration to generate the temporal constraint so as to improve the performance of vehicle ReID. The definition of the CLM is to connect adjacent cameras that are directly connected to each other. If no existing cameras need to be passed between the two cameras, the two cameras form a camera link. The link between two directly-connected cameras

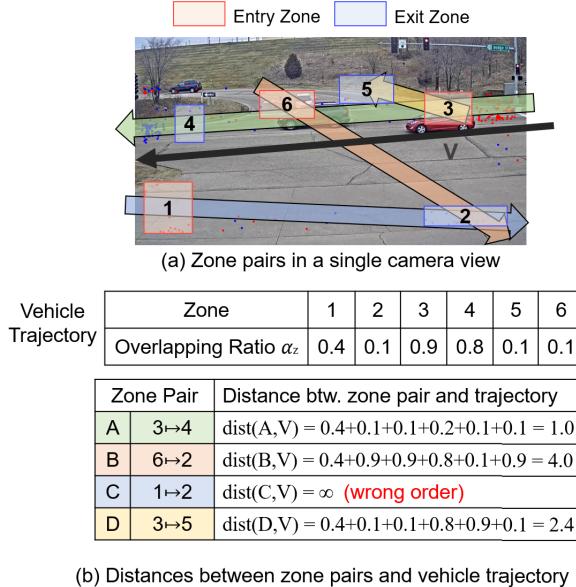


Fig. 6. An example of distance calculation between a vehicle trajectory (black) with respect to each entry-exit zone pair. Assume that there are four zone pairs (A, B, C, D) which are two entry and two exit zones in a single camera. If there is a vehicle passing through the four zones with the overlapping area ratio  $\alpha_z$  in the upper table in (b), the distances can be estimated by Eq. (14). Then, this vehicle trajectory can be classified into the zone pair with the smallest distance. In this case, it is zone pair A.

actually only connects one zone each in these two cameras. There may be multiple entry/exit zones within a camera's view. Due to road structures and traffic rules, the motion of the vehicles follows certain driving patterns. Thus, we group the SCT results into limited numbers of trajectories, then exploit the limited numbers of trajectories to establish trajectory-based camera link model (TCLM). It means that the CLM can be more specific into trajectory level to generate a more accurate transition time distribution of each camera link.

In order to generate TCLM, we need to distinguish trajectories that are determined by driving patterns. We define the zones of each camera so that the zone pair can be used to uniquely describe a trajectory (Fig. 6). The zones can be the intersection areas, the turning points of a road or the enter/exit areas of the camera's field-of-view. On the other hand, these zone pairs also need to be labeled to complete TCLM. In our work, we generate all possible zones and use the training data of MTMCT to automatically generate the camera links instead of human labeling. In most of the cases, the straight and the right-turn trajectories can be described using different zone pairs which are gone through by the passing vehicles. Sometimes the trajectory may not go through the corresponding prespecified zone pair perfectly due to the viewing angle of the camera. In this case, measuring the distance between a tracked vehicle and a zone pair is necessary. The distance can be calculated as,

$$\text{dist}(P, V) = \sum_{z \in P \cup V} |\mathbb{1}(z \in P) - \alpha_z|, \quad (14)$$

where  $P$  denotes the zone pair and  $V$  is the actual zones gone through by the tracked vehicle.  $\alpha_z$  represents the overlapping

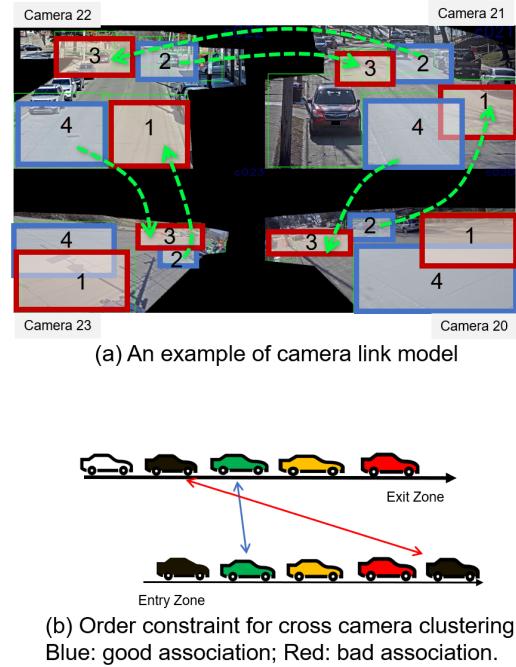


Fig. 7. Illustration for (a) trajectory-based camera link model, and (b) the order constraint. There are four cameras with overlapping FoVs: camera 20, camera 21, camera 22 and camera 23. The exit/entry zones are denoted as blue/red bounding boxes, respectively. In (a), the exit zone in Camera 21 (blue bounding box) and the entry zone in Camera 22 (red bounding box) indicate the camera connectivity. In this scenario, the vehicles exiting from the camera 21 will appear in the Camera 22 immediately, similarly for camera 22 to camera 23 and camera 20 to camera 21. Take the connectivity of camera 21 and camera 22 as an example, there is a transition camera link  $\mathcal{L}$  including  $C^{21}$  and  $C^{22}$ , where  $C^{21} = \{P_1^{21}, P_2^{21}\}$  and  $C^{22} = \{P_1^{22}, P_2^{22}\}$  have two zone pairs, respectively. Therefore,  $z_s = 2 \in P_1^{21}$  and  $z_d = 3 \in P_2^{22}$  are one transition zone pair while  $P_1^{21} = \{1, 2\}$  and  $P_2^{22} = \{3, 4\}$ . Another transition zone pair are  $z_s = 2 \in P_1^{22}$  and  $z_d = 3 \in P_2^{21}$  while  $P_1^{22} = \{1, 2\}$  and  $P_2^{21} = \{3, 4\}$ .

ratio of the vehicle to zone  $z$  (i.e., the overlapping area divided by the vehicle bounding box area). Furthermore, the order of the zones in the zone pair and the tracked vehicle are also considered. Once the order in the tracked vehicle conflicts with the zone pair, the distance between a tracked vehicle and a trajectory is set to infinity. Finally, the closest trajectory is assigned by comparing the tracked vehicle with all the possible zone pairs within the camera.

The last step of TCLM generation is to estimate the transition time of camera links. After getting the trajectories and camera links, the transition of each camera link can be defined as  $\mathcal{L} = (C^s, C^d)$ , where  $C^s = \{P_i^s\}_{i=1}^m$  is the zone pair set in the source camera and  $C^d = \{P_j^d\}_{j=1}^n$  is the zone pair set in the destination camera. Each camera link can have more than one transition due to the bidirectional traffic. An example is shown in Fig. 7 to illustrate the concept of TCLM. We first define the transition zone pair  $z_s$  and  $z_d$ , such that  $z_s \in P_i^s (\forall P_i^s \in C^s)$  and  $z_d \in P_j^d (\forall P_j^d \in C^d)$ . Then, the transition can be imposed with the temporal constraint for both  $C^s$  and  $C^d$ . Given a camera link of a vehicle trajectory from  $P^s$  to  $P^d$  (i.e., from the source camera to the destination camera), the transition time is defined as

$$\Delta t = t^s - t^d, \quad (15)$$

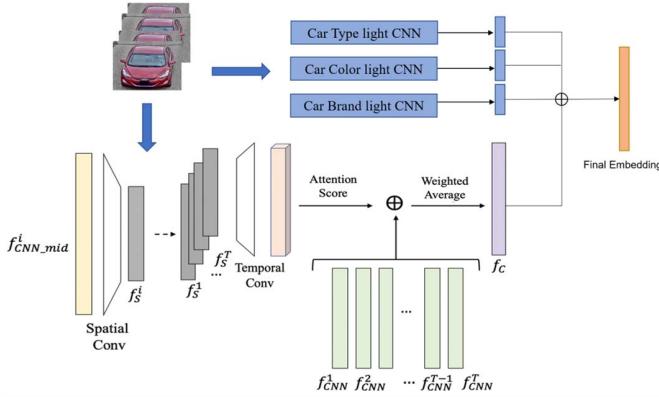


Fig. 8. Illustration for the final embedding features. The appearance feature is generated by a temporal attention architecture based on ResNet-50 as the backbone. Three metadata features are generated by three separately trained Light CNN networks. The final representation for global trajectory generation is formed by concatenating all these four features.

where  $t^s$  and  $t^d$  represent the time the tracks passing  $z_s$  and  $z_d$  respectively. Then, we can obtain a time window ( $\Delta t_{\min}$ ,  $\Delta t_{\max}$ ) for each camera link  $\mathcal{L}$ , so that only the tracked vehicle pairs with transition time within the time window are considered as valid. Thus, the search space of the ReID can be reduced by an appropriate time window.

#### D. Global Trajectory Generation

MTMCT can be referred as a correlation clustering problem to generate the global trajectory, which can be defined as the Binary Integer Program (BIP). Here, we describe the BIP as follows. Assume that there is a weighted graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ , where  $\mathcal{V}$  represents the single camera trajectory node set, the weight  $\mathcal{W}$  of the edge  $\mathcal{E}$  represents the corresponding correlation between the nodes.

$$\begin{aligned} X^* = \arg \max_{\{x_{i,j}\}} & \sum_{(i,j) \in \mathcal{E}} w_{i,j} x_{i,j}, \\ \text{s.t., } & x_{i,j} + x_{j,k} \leq 1 + x_{i,k}, \quad \forall i, j, k \in \mathcal{V}. \end{aligned} \quad (16)$$

The set  $X$  represents the set of all possible combinations of assignments to the binary variables  $x_{i,j}$ . Here  $w_{i,j}$  means the weight between two trajectories  $i, j$ , which can be obtained by calculating the distance of the two trajectories  $i, j$ . By rewarding edges that connect the same vehicle's multi-camera trajectories and penalizing edges that link to the different vehicles, we can maximize  $X^*$ . For example, the  $x_{i,j}$  of two multi-camera trajectories  $i, j$  should be assigned 1 if trajectories  $i, j$  indicate the same vehicle identity. Meanwhile, the constraints in Eq. (16) are used to enforce the transitivity in the solution.

Then, we can use the final embedding feature  $\mathbf{f}(\xi_i) = \mathcal{A}(\xi_i) \oplus \mathcal{M}(\xi_i)$ , TCLM and hierarchical clustering algorithm to produce the global IDs for MTMCT. The illustration of generating the final embedding is shown in Fig. 8, and the procedure of hierarchical clustering is in Fig. 9 and Alg. 2.  $\mathcal{A}(\xi_i)$  and  $\mathcal{M}(\xi_i)$  indicate the ReID feature and metadata feature, respectively.  $\oplus$  denotes the concatenation operator.

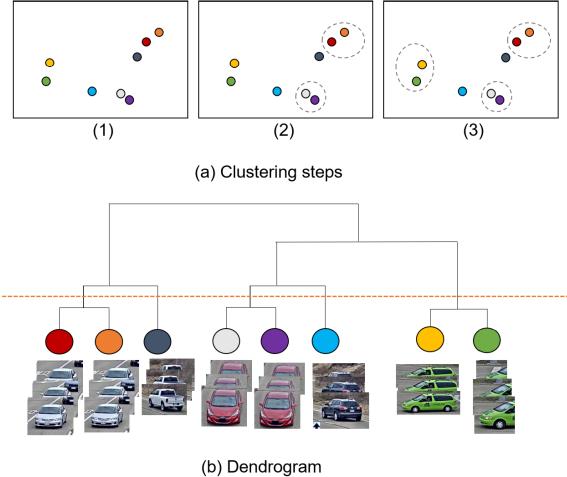


Fig. 9. The procedure of hierarchical clustering: (a) the clustering steps, (b) the dendrogram of clusters with different threshold levels.

---

#### Algorithm 2 Hierarchical Clustering

---

**Input :** Trajectories set  $\Xi = \{\xi_n\}$  from all cameras.  
**Output:** Global ID for all trajectories within all cameras.

```

1 Initialize distance matrix  $\mathbf{M}$  between each two
   trajectories  $\mathbf{M} \leftarrow \infty$ ;
2 for trajectory  $\xi_i$  from  $\Xi$  do
3   for trajectory  $\xi_j$  from  $\Xi$  do
4     | Calculate  $\mathbf{M}_{i,j}$  by Eq. (18).
5   end
6 end
7 Flatten and sort the upper triangular part of  $\mathbf{M}$  in
   ascending order:  $F \leftarrow \text{sort}([\mathbf{M}_{i,j}]_{i < j})$ ;
8  $\text{iter} \leftarrow 0$ ,  $\delta \leftarrow$  distance threshold;
9 while  $\text{iter} < \# \text{ of iterations}$  do
10   for  $m_{i,j}$  from  $F$  do
11     | if  $m_{i,j} < \delta$  and valid order constraint for  $\xi_i$ ,
         $\xi_j$  then
        |   | Assign the same global ID to  $\xi_i$  and  $\xi_j$ ;
12     | else
13     |   |  $m_{i,j} \leftarrow \infty$ ;
14     | end
15   end
16    $\text{iter} \leftarrow \text{iter} + 1$ ;
17 end
18 end
```

---

We can thus define a distance matrix

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \cdots & \mathbf{M}_{1N} \\ \vdots & \ddots & \vdots \\ \mathbf{M}_{N1} & \cdots & \mathbf{M}_{NN} \end{bmatrix}, \quad (17)$$

where

$$\mathbf{M}_{i,j} = \begin{cases} \text{dist}(\mathbf{f}(\xi_i), \mathbf{f}(\xi_j)) & \text{if valid camera link constraint,} \\ 0 & \text{otherwise,} \end{cases} \quad (18)$$

which represents all the distance between any two trajectories from two different cameras. Here,

$$\begin{aligned} \text{dist}(\mathbf{f}(\xi_i), \mathbf{f}(\xi_j)) &= \text{dist}(\mathcal{A}(\xi_i) \oplus \mathcal{M}(\xi_i), \mathcal{A}(\xi_j) \oplus \mathcal{M}(\xi_j)) \\ &= \|\mathcal{A}(\xi_i) \oplus \mathcal{M}(\xi_i) - \mathcal{A}(\xi_j) \oplus \mathcal{M}(\xi_j)\|_2. \end{aligned} \quad (19)$$

Moreover, the time window ( $\Delta t_{\min}, \Delta t_{\max}$ ) of each camera link is generated as the CLM constraint. Since the search space of ReID is reduced by the CLM constraint, the Rank-1 accuracy can be improved. By using hierarchical clustering, we greedily select the smallest pair-wise distance to merge the tracked vehicles cross cameras.

Furthermore, the orders between different tracked vehicles can be used as an additional constraint to further reduce the search space of the ReID. Due to the traffic rule or the road condition, the orders of vehicles should be almost the same. Take Fig. 7(b) as an example, we will remove the pairs which conflict with previously matched pairs. The process will be repeated until there is no valid transition pair or the minimum distance is larger than a threshold. Given two vehicle trajectories  $\xi_{src_1}, \xi_{src_2}$  in source camera and two vehicle trajectories  $\xi_{dst_1}, \xi_{dst_2}$  in destination camera, we define

$$\text{sign}(\xi_{src_1} - \xi_{src_2}) = \text{sign}(\xi_{dst_1} - \xi_{dst_2}), \quad (20)$$

i.e., the orders of tracked vehicles in source and destination camera should remain the same. In this case, the search space can be further reduced.

#### IV. EXPERIMENTS

##### A. CityFlow MTMCT Dataset and Implementation Details

In this section, we evaluate the proposed method using a benchmark city-scale MTMCT dataset CityFlow [8] and compare it with state-of-the-art MTMCT methods. To the best of our knowledge, CityFlow is the only one city-scale multiple camera vehicle tracking dataset, where the TCLM is established by the training data, and the cameras in the testing data are overlapped with the training data. There are 3.25 hours of videos in CityFlow, which are collected from 40 cameras across 10 intersections spanning over 2.5 miles of distance coverage in a mid-sized U.S. city. There are many different types of road scenarios in CityFlow (e.g., intersections, stretches of roadways, and highways). Moreover, CityFlow contains 229,680 bounding boxes for 666 vehicles, whose license plates are all obscured due to privacy concerns.

Our experimental environment is set as follows: the SCT is based on Python: 3.5.4 and Tensorflow: 1.4.0; the ReID model is realized using the PyTorch 0.3.1, Torchvision 0.2.0 and Python 2.7. One NVIDIA Quadro GV100 32GB GPU is used for our experiments. As for the data used for training, the details are listed as follows. (1) For SCT, we train the TNT using the training data of AI City Challenge 2018 dataset. (2) For ReID, the training data consists of two parts, i.e., the cropped vehicles from the training data of CityFlow and the training data of CityFlow-ReID dataset. (3) In terms of

metadata classifier, we train on the training data of CityFlow-ReID dataset. (4) The TCLM is trained on the training data of CityFlow dataset.

Some implementation details of the proposed methods, i.e., SCT and MA-ReID, are detailed here. First of all, as the baseline for TSCT we use the TNT, which is pretrained on the AI City Challenge 2018 dataset [47] that contains over 3.3K vehicles. The dimensionality of the appearance feature of TNT is 512, the time window of tracklet generation is set as 64 frames, and the batch size for training TNT is 32. We select Adam as the optimizer for TNT and the learning rate is from  $10^{-3}$  to  $10^{-5}$  with the weight decay 10 times in every 2000 steps.

Next, the parameter of MeanShift is the bandwidth of the radial basis function kernel, which is set as 250 for generating the appropriate area size of zones. According to our experiments, the area of the traffic-aware zone is supposed to be at least 150% of the average area of the vehicle bounding box. Moreover, the traffic-aware density  $\rho_{ta}$  of each zone is set as 0.8 to ensure this zone is a traffic-aware zone, otherwise the traffic-aware zone is not reliable and we rather do nothing. Furthermore,  $\rho_e$  and  $\rho_x$  are also set as 0.8 in our experiments. After generating the zones, we need to calculate the overlapping area of the bounding boxes of the vehicles and the traffic-aware zone to locate the vehicles passing through the traffic-aware zone. Then, the bounding box of the last position of a tracklet will be inserted into a queue. Once there are some new trajectories suddenly appear in the traffic-aware zone, we calculate the IOU of the bounding box of the first position and the first bounding box popping from the queue. Due to the traffic jam or the queuing of vehicles, the IOU can sometimes be very small. We thus set the IOU threshold very low so as to merge these isolated trajectories. The appearance feature is not reliable in the traffic-aware zone due to serious occlusions, therefore the First-In-First-Out (FIFO) and IOU constraints are more reliable than appearance similarity. Thus, the appearance similarity threshold between isolated trajectories is set as 0.4 for isolated trajectories reconnection. This value is not sensitive because FIFO strategy and IOU constraint have imposed high restriction for isolated trajectory reconnection. The SCT results are similar from similarity score 0.1 to 0.4, and there is no isolated trajectory pair matching for similarity score over 0.5.

In terms of MA-ReID, there are two architectures used for appearance feature extractor and metadata classifier, respectively. In appearance feature extraction, the temporal attention is applied for training the ReID model, and the trained 4-dim attention vector is used to generate the clip-level embedding based on average pooling to obtain the trajectory-level embedding. We train the ReID model for 800 epochs with batch size 32, the learning rate and weight decay are  $3 \times 10^{-4}$  and  $5 \times 10^{-4}$ , respectively. We adopt the ResNet-50 as the backbone network for both SCT ReID and ICT ReID, and the loss function is a combination of batch-sample (BS) loss and cross-entropy (Xent) loss. In our experiments, we resize the input image size as  $224 \times 224$  and ResNet-50 is pre-trained on ImageNet.

TABLE II  
METADATA CLASSIFICATION RESULTS ON CITYFLOW

Metadata	Accuracy
Type	0.9483
Brand	0.8801
Color	0.9400

Furthermore, the metadata classifier is trained on the training data of CityFlow ReID dataset, where 3204 low-resolution images are excluded, then the rest of the images are randomly split to Train/Validation/Test with ratio 0.7/0.1/0.2. In our experiments, we label the metadata of vehicle in the following categories: 1) Type: sedan, suv, minivan, pickup truck, hatchback and truck; 2) Brand: Dodge, Ford, Chevrolet, GMC, Honda, Chrysler, Jeep, Hyundai, Subaru, Toyota, Buick, KIA, Nissan, Volkswagen, Oldsmobile, BMW, Cadillac, Volvo, Pontiac, Mercury, Lexus, Saturn, Benz, Mazda, Scion, Mini, Lincoln, Audi, Mitsubishi and others; 3) Color: black, white, red, grey, silver, gold, blue, green and yellow. Table II shows that our classification accuracy in all the three metadata are over 88%. Therefore, the MTMCT can benefit from the accurate metadata information.

In MTMCT, the IDF1 score [53] is wildly used as evaluation metrics. Thus, IDF1 is adopted to rank the performance for each participant in the CityFlow dataset, which means to calculate the ratio of correctly identified vehicles over the average number of ground-truth and predicted vehicles. The definition of IDF1 is shown as follows:

$$\begin{aligned} IDP &= \frac{IDTP}{IDTP + IDFP}, \\ IDR &= \frac{IDTP}{IDTP + IDFN}, \\ IDF1 &= \frac{2IDTP}{2IDTP + IDFP + IDFN}, \end{aligned} \quad (21)$$

where IDFN, IDTN and IDTP are defined as follows

$$\begin{aligned} IDFN &= \sum_{\tau} \sum_{t \in T_{\tau}} m(\tau, \gamma_m(\tau), t, \Delta), \\ IDFP &= \sum_{\gamma} \sum_{t \in T_{\gamma}} m(\tau_m(\gamma), \gamma, t, \Delta), \\ IDTP &= \sum_{\tau} \text{len}(\tau) - IDFN = \sum_{\gamma} \text{len}(\gamma) - IDFP, \end{aligned} \quad (22)$$

where the ground truth trajectory is denoted as  $\tau$ ,  $\gamma_m(\tau)$  represents the best matches of the computed trajectory for  $\tau$ ;  $\gamma$  is the computed trajectory;  $\tau_m(\gamma)$  denotes the best match of ground truth trajectory for  $\gamma$ ;  $t$  indicates the frame index;  $\Delta$  is the IOU threshold that determines if computed bounding box matches the ground truth bounding box (here we set  $\Delta = 0.5$ );  $m(\cdot)$  represents a mismatch function which is set as 1 if there is a mismatch at  $t$ ; otherwise,  $m(\cdot)$  is set as 0.

### B. System Performance

According to [57]–[59], the spatio-temporal information can improve the performance of MTMCT. In this work, the TCLM is utilized to take advantage of spatio-temporal information to achieve the best performance. On the other

TABLE III  
MTMCT RESULTS COMPARISON ON CITYFLOW DATASET

Methods	IDF1
MOANA+BA [8]	0.3950
DeepSORT+BS [8]	0.4140
TC+BA [8]	0.4630
ZeroOne [54]	0.5987
DeepCC [37]	0.5660
LAAM [55]	0.6300
ANU [56]	0.6519
TrafficBrain [57]	0.6653
DDashcam [58]	0.6865
UWIPL [59]	0.7059
Ours(TSCT+TA)	0.7493
<b>Ours(TSCT+TA+META)</b>	<b>0.7677</b>

TABLE IV  
THE MTMCT PERFORMANCE FOR DIFFERENT DIMENSIONS OF THE PROPOSED METHOD ON CITYFLOW DATASET

Appearance Dimension	IDF1	IDP	IDR
512	0.6872	0.6641	0.7120
1024	0.7241	0.6928	<b>0.7525</b>
2048	<b>0.7677</b>	<b>0.8276</b>	0.7159



Fig. 10. Qualitative results of the proposed method on CityFlow dataset.

hand, some systems merely use the data association graph for MTMCT. Table III shows that our performance can achieve the state-of-the-art on the CityFlow. Meanwhile, we compare our method with locality aware appearance metric (LAAM) [55], which is the state-of-the-art approach on the DukeMTMC dataset. Although DukeMTMC dataset is no longer available for MTMCT, we still can compare the performance of the proposed method with the state-of-the-art approach from DukeMTMC dataset. LAAM improves DeepCC [37] by using training the model on intra-camera and inter-camera metrics. The temporal windows of LAAM for SCT and ICT of are 500 frames and 2400 frames; they also use ResNet-50 pre-trained on ImageNet. The results of our proposed method are shown in Table III. Our method outperforms all the state-of-the-art methods, which achieves 76.77% in IDF1. Table IV shows the ablation study of different dimension appearance features. The experimental results show that 2048 feature dimensions can achieve the best performance, while a smaller size of dimension decreases the performance. Fig. 10 shows

TABLE V  
THE FLOPS OF THE PROPOSED METHOD

Total params	49,199,169
Total memory	109.68MB
Total FLOPs	4.14G FLOPs

TABLE VI  
THE MTMCT PERFORMANCE FOR DIFFERENT COMBINATIONS OF THE PROPOSED METHOD

TNT	TSCT	TA	META	CLM	IDF1	IDP	IDR
✓					0.1583	0.4418	0.0959
✓		✓			0.5237	0.6816	0.4221
✓				✓	0.5776	0.5918	0.5779
✓			✓		0.7059	0.6912	<b>0.7211</b>
✓	✓	✓			0.7493	0.8071	0.6918
✓	✓	✓	✓	✓	<b>0.7677</b>	<b>0.8276</b>	0.7159

TABLE VII  
SCT RESULTS ON CITYFLOW

Methods	IDF1	MOTA	MOTP	Recall	MT
DeepSORT [60]	79.5%	68.9%	65.5%	69.2%	756
TC [45]	79.7%	70.3%	65.6%	70.4%	895
MOANA [61]	72.8%	67.0%	65.9%	68.0%	980
<b>TSCT</b>	<b>90.7%</b>	<b>83.4%</b>	<b>75.2%</b>	<b>87.1%</b>	<b>1698</b>

some of qualitative results, which proves our method can be generalized for different cameras and vehicles. Moreover, the computational complexity of our method is shown in Table V.

### C. Ablation Study

The ablation study of our method is shown in Table VI. There are three components in our framework: SCT, vehicle ReID and TCLM. In terms of SCT, our SCT consists of TNT and TSCT. For vehicle ReID, TA denotes temporal attention and META denotes the use of metadata feature for ReID. The experimental results illustrate how the proposed components enhancing robustness. In Table VI, when replacing TNT with the TSCT, IDF1 based on the TA feature increases by 4.3% on MTMCT. One significant improvement is to alter the average pooling by TA. Thus, the TSCT and TA are necessary components in our system based on the experimental results. Then, another significant improvement shown in the ablation studies is TCLM. Because not only the transition time constraint between cameras is considered for data association in TCLM but also the spatial information is taken into account at the same time. Since the appearance variance is small in the vehicle ReID, the TCLM can provide significant improvement. Finally, metadata information can further improve the IDF1 by almost 2%.

### D. SCT Performance

Table VII shows the performance of our approach comparing with those of state-of-the-art SCT methods [45], [60], [61] in MTMCT. DeepSORT [60] is a Kalman-filter-based online tracking method. Tracklet Clustering (TC) [45] is an offline method, which is the winner of the AI

City Challenge Workshop at CVPR 2018 [62]. MOANA [61] is the state-of-the-art approach on the MOT Challenge 2015 3D benchmark. In CityFlow dataset, there are three available public detection results (i.e., SSD512 [63], YOLOv3 [64] and Faster R-CNN [65]). Since the SSD512 [63] was reported to have the best performance [8], therefore we also use SSD512 as our detector directly in our experiments. In our experiments, the adopted metrics of SCT are IDF1 score (IDF1), Multiple Object Tracking Accuracy (MOTA), Multiple Object Tracking Precision (MOTP), Recall and the mostly tracked targets (MT), which are the standard metric for SCT. Based on our experimental results, the proposed TSCT achieves the best performance.

## V. CONCLUSION

In this paper, we propose a novel framework for Multi-Target Multi-Camera tracking (MTMCT), which includes traffic-aware single-camera tracking (TSCT), Metadata-Aided Re-Identification (MA-ReID) and also trajectory-based camera link model (TCLM). From our experiments, the proposed method is efficient, effective and robust, which achieves the state-of-the-art performance IDF1 76.77% in CityFlow dataset for city-scale MTMCT of vehicles.

## REFERENCES

- [1] Z. Tang *et al.*, “PAMTRI: Pose-aware multi-task learning for vehicle re-identification using highly randomized synthetic data,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 211–220.
- [2] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, “3D traffic scene understanding from movable platforms,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 1012–1025, May 2014.
- [3] H. Zhang, A. Geiger, and R. Urtasun, “Understanding high-level semantics by modeling traffic patterns,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3056–3063.
- [4] G. Wang, Y. Wang, H. Zhang, R. Gu, and J.-N. Hwang, “Exploit the connectivity: Multi-object tracking with TrackletNet,” 2018, *arXiv:1811.07258*. [Online]. Available: <http://arxiv.org/abs/1811.07258>
- [5] G. Wang, X. Yuan, A. Zheng, H.-M. Hsu, and J.-N. Hwang, “Anomaly candidate identification and starting time estimation of vehicles from traffic videos,” in *Proc. CVPR Workshops*, 2019, pp. 382–390.
- [6] Y. Wang, Y.-T. Huang, and J.-N. Hwang, “Monocular visual object 3D localization in road scenes,” in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 917–925.
- [7] T.-W. Huang, J. Cai, H. Yang, H.-M. Hsu, and J.-N. Hwang, “Multi-view vehicle re-identification using temporal attention model and metadata re-ranking,” in *Proc. AI City Challenge Workshop, IEEE/CVF Comput. Vis. Pattern Recognit. (CVPR) Conf.*, Long Beach, CA, USA, Jun. 2019, pp. 1–9.
- [8] Z. Tang *et al.*, “CityFlow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8797–8806.
- [9] M. Naphade *et al.*, “The 2019 AI city challenge,” in *Proc. CVPR Workshops*, 2019, pp. 452–460.
- [10] M. Naphade *et al.*, “The 4th AI city challenge,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 626–627.
- [11] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, “Multicamera people tracking with a probabilistic occupancy map,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 267–282, Feb. 2008.
- [12] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, “Multiple object tracking using K-shortest paths optimization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1806–1819, Sep. 2011.
- [13] M. Hofmann, D. Wolf, and G. Rigoll, “Hypergraphs for joint multi-view reconstruction and multi-object tracking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3650–3657.
- [14] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua, “Multi-commodity network flow for tracking multiple people,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1614–1627, Aug. 2014.

- [15] L. Leal-Taixe, G. Pons-Moll, and B. Rosenhahn, "Branch-and-price global optimization for multi-view multi-target tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1987–1994.
- [16] F. Yang, H. Lu, and M.-H. Yang, "Robust superpixel tracking," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1639–1651, Apr. 2014.
- [17] X. Zhao, Y. Fu, and Y. Liu, "Human motion tracking by temporal-spatial local Gaussian process experts," *IEEE Trans. Image Process.*, vol. 20, no. 4, pp. 1141–1151, Apr. 2011.
- [18] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3074–3082.
- [19] X. Lan, S. Zhang, P. C. Yuen, and R. Chellappa, "Learning common and feature-specific patterns: A novel multiple-sparse-representation-based tracker," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 2022–2037, Apr. 2018.
- [20] J. Cai, Y. Wang, H. Zhang, H.-M. Hsu, C. Ma, and J.-N. Hwang, "IA-MOT: Instance-aware multi-object tracking with motion consistency," in *Proc. BMTT Challenge Workshop, IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020. [Online]. Available: <https://motchallenge.net/workshops/bmtt2020/>
- [21] H. Zhang, Y. Wang, J. Cai, H.-M. Hsu, H. Ji, and J.-N. Hwang, "Lifts: Lidar and monocular image fusion for multi-object tracking and segmentation," in *Proc. BMTT Challenge Workshop, IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020. [Online]. Available: <https://motchallenge.net/workshops/bmtt2020/>
- [22] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, "Poi: Multiple object tracking with high performance detection and appearance feature," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 36–42.
- [23] H. Jiang, J. Wang, Y. Gong, N. Rong, Z. Chai, and N. Zheng, "Online multi-target tracking with unified handling of complex scenarios," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3464–3477, Nov. 2015.
- [24] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M.-H. Yang, "Online multi-object tracking with dual matching attention networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 366–382.
- [25] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3539–3548.
- [26] C. Liu, R. Yao, S. H. Rezatofighi, I. Reid, and Q. Shi, "Model-free tracker for multiple objects using joint appearance and motion inference," *IEEE Trans. Image Process.*, vol. 29, pp. 277–288, 2020.
- [27] W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou, and S. Maybank, "Principal axis-based correspondence between multiple cameras for people tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 663–671, Apr. 2006.
- [28] R. Eshel and Y. Moses, "Homography based multiple camera detection and tracking of people in a dense crowd," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
- [29] M. Bredereck, X. Jiang, M. Körner, and J. Denzler, "Data association for multi-object tracking-by-detection in multi-camera networks," in *Proc. 6th Int. Conf. Distrib. Smart Cameras (ICDSC)*, 2012, pp. 1–6.
- [30] Y. Xu, X. Liu, Y. Liu, and S.-C. Zhu, "Multi-view people tracking via hierarchical trajectory composition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4256–4265.
- [31] Y. Xu, X. Liu, L. Qin, and S.-C. Zhu, "Cross-view people tracking by scene-centered spatio-temporal parsing," in *Proc. AAAI*, 2017, pp. 4299–4305.
- [32] Y. T. Tesfaye, E. Zemene, A. Prati, M. Pelillo, and M. Shah, "Multi-target tracking in multiple non-overlapping cameras using constrained dominant sets," 2017, *arXiv:1706.06196*. [Online]. Available: <http://arxiv.org/abs/1706.06196>
- [33] Y. Cai and G. Medioni, "Exploring context information for inter-camera multiple target tracking," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2014, pp. 761–768.
- [34] Y.-G. Lee, Z. Tang, and J.-N. Hwang, "Online-learning-based human tracking across non-overlapping cameras," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2870–2883, Oct. 2018.
- [35] W. Chen, L. Cao, X. Chen, and K. Huang, "A novel solution for multi-camera object tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 2329–2333.
- [36] Z. Zhang, J. Wu, X. Zhang, and C. Zhang, "Multi-target, multi-camera tracking by hierarchical clustering: Recent progress on DukeMTMC project," 2017, *arXiv:1712.09531*. [Online]. Available: <http://arxiv.org/abs/1712.09531>
- [37] E. Ristani and C. Tomasi, "Features for multi-target multi-camera tracking and re-identification," 2018, *arXiv:1803.10859*. [Online]. Available: <http://arxiv.org/abs/1803.10859>
- [38] W. Chen, L. Cao, X. Chen, and K. Huang, "An equalized global graph model-based approach for multicamera object tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 11, pp. 2367–2381, Nov. 2017.
- [39] X. Chen and B. Bhanu, "Integrating social grouping for multitarget tracking across cameras in a CRF model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 11, pp. 2382–2394, Nov. 2017.
- [40] M. Ye, J. Li, A. J. Ma, L. Zheng, and P. C. Yuen, "Dynamic graph co-matching for unsupervised video-based person re-identification," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 2976–2990, Jun. 2019.
- [41] D. Cheng, Y. Gong, J. Wang, Q. Hou, and N. Zheng, "Part-aware trajectories association across non-overlapping uncalibrated cameras," *Neurocomputing*, vol. 230, pp. 30–39, Mar. 2017.
- [42] W. Nie *et al.*, "Single/cross-camera multiple-person tracking by graph matching," *Neurocomputing*, vol. 139, pp. 220–232, Sep. 2014.
- [43] Y.-G. Lee, Z. Tang, J.-N. Hwang, and Z. Fang, "Inter-camera tracking based on fully unsupervised online learning," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 2607–2611.
- [44] X. Wang, R. Panda, M. Liu, Y. Wang, and A. K. Roy-Chowdhury, "Exploiting global camera network constraints for unsupervised video person re-identification," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Dec. 9, 2020, doi: [10.1109/TCSVT.2020.3043444](https://doi.org/10.1109/TCSVT.2020.3043444).
- [45] Z. Tang, G. Wang, H. Xiao, A. Zheng, and J.-N. Hwang, "Single-camera and inter-camera vehicle tracking and 3D speed estimation based on fusion of visual and semantic features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 108–115.
- [46] H.-M. Hsu, Y. Wang, and J.-N. Hwang, "Traffic-aware multi-camera tracking of vehicles based on ReID and camera link model," 2020, *arXiv:2008.09785*. [Online]. Available: <http://arxiv.org/abs/2008.09785>
- [47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [48] J. Gao and R. Nevatia, "Revisiting temporal modeling for video-based person ReID," 2018, *arXiv:1805.02104*. [Online]. Available: <http://arxiv.org/abs/1805.02104>
- [49] R. Kumar, E. Weill, F. Aghdasian, and P. Sriram, "Vehicle re-identification: An efficient baseline using triplet embedding," 2019, *arXiv:1901.01015*. [Online]. Available: <http://arxiv.org/abs/1901.01015>
- [50] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," 2017, *arXiv:1703.07737*. [Online]. Available: <http://arxiv.org/abs/1703.07737>
- [51] X. Wu, R. He, Z. Sun, and T. Tan, "A light CNN for deep face representation with noisy labels," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 11, pp. 2884–2896, Nov. 2018.
- [52] J. A. Ansari, S. Sharma, A. Majumdar, J. K. Murthy, and K. M. Krishna, "The earth ain't flat: Monocular reconstruction of vehicles on steep and graded roads from a moving camera," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 8404–8410.
- [53] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 17–35.
- [54] X. Tan *et al.*, "Multi-camera vehicle tracking and re-identification based on visual and spatial-temporal features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2019, pp. 275–284.
- [55] Y. Hou, L. Zheng, Z. Wang, and S. Wang, "Locality aware appearance metric for multi-target multi-camera tracking," 2019, *arXiv:1911.12037*. [Online]. Available: <http://arxiv.org/abs/1911.12037>
- [56] Y. Hou, H. Du, and L. Zheng, "A locality aware city-scale multi-camera vehicle tracking system," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2019, pp. 167–174.
- [57] Z. He, Y. Lei, S. Bai, and W. Wu, "Multi-camera vehicle tracking with powerful visual features and spatial-temporal cue," in *Proc. CVPR Workshops*, 2019, pp. 203–212.
- [58] P. Li *et al.*, "Spatio-temporal consistency and hierarchical matching for multi-target multi-camera vehicle tracking," in *Proc. CVPR Workshops*, 2019, pp. 222–230.
- [59] H.-M. Hsu, T.-W. Huang, G. Wang, J. Cai, Z. Lei, and J.-N. Hwang, "Multi-camera tracking of vehicles based on deep features re-id and trajectory-based camera link models," in *Proc. AI City Challenge Workshop, IEEE/CVF Comput. Vis. Pattern Recognit. (CVPR) Conf.*, Long Beach, CA, USA, Jun. 2019, pp. 416–424.
- [60] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3645–3649.

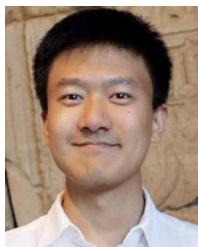
- [61] Z. Tang and J.-N. Hwang, "MOANA: An online learned adaptive appearance model for robust multiple object tracking in 3D," *IEEE Access*, vol. 7, pp. 31934–31945, 2019.
- [62] M. Naphade *et al.*, "The 2018 NVIDIA AI city challenge," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 53–60.
- [63] W. Liu *et al.*, "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 21–37.
- [64] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: <http://arxiv.org/abs/1804.02767>
- [65] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.



**Hung-Min Hsu** (Member, IEEE) received the Ph.D. degree from National Taiwan University in 2017. From 2017 to 2018, he was a Postdoctoral Fellow with the Research Center for Information Technology Innovation, Academia Sinica. In summer 2018, he joined the Department of Electrical and Computer Engineering (ECE), University of Washington, as a Postdoctoral Researcher. His research interests include natural language processing, recommendation systems, and computer vision. He has won three CVPR AI City Challenge Awards in 2019 and two CVPR Multi-Object Tracking and Segmentation Challenge Awards in 2020.



**Jiarui Cai** (Graduate Student Member, IEEE) received the B.S. degree in electrical and information engineering from the Beijing University of Posts and Telecommunications in 2017. She is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Washington at Seattle, Seattle. Her research interests include computer vision and deep learning.



**Yizhou Wang** (Graduate Student Member, IEEE) received the B.Eng. degree in automation from Northwestern Polytechnical University in 2016 and the M.S. degree in electrical engineering from Columbia University, advised by Prof. Shih-Fu Chang, in 2018. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Washington, advised by Prof. Jenq-Neng Hwang. His research interests include autonomous driving, computer vision, deep learning, and cross-modal learning.



**Jenq-Neng Hwang** (Fellow, IEEE) received the B.S. and M.S. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1981 and 1983, respectively, and the Ph.D. degree from the University of Southern California. In summer 1989, he joined the Department of Electrical and Computer Engineering (ECE), University of Washington at Seattle, where he has been a Full Professor since 1999. He has served as the Associate Chair for Research from 2003 to 2005 and 2011 to 2015. He also served as the Associate Chair for Global Affairs from 2015 to 2020. He is the Founder and the Co-Director of the Information Processing Laboratory, which has won CVPR AI City Challenges Awards in the past years. He has written more than 380 journals, conference papers, and book chapters in the areas of machine learning, multimedia signal processing, and multimedia system integration and networking, including an authored textbook *Multimedia Networking: From Theory to Practice* (Cambridge University Press). He has close working relationship with the industry on multimedia signal processing and multimedia networking. He is a Founding Member of the Multimedia Signal Processing Technical Committee of IEEE Signal Processing Society and was the Society's Representative to IEEE Neural Network Council from 1996 to 2000. He is a member of the Multimedia Technical Committee (MMTC) of IEEE Communication Society and the Multimedia Signal Processing Technical Committee (MMSP TC) of IEEE Signal Processing Society. He received the 1995 IEEE Signal Processing Society's Best Journal Paper Award. He has served as the Program Co-Chair for IEEE ICME 2016 and was the Program Co-Chair for ICASSP 1998 and ISCAS 2009. He has served as an Associate Editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON NEURAL NETWORKS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON IMAGE PROCESSING, and IEEE Signal Processing Magazine (SPM). He is on the Editorial Board of ZTE Communications, ETRI, IJDMB, and JSPS journals.



**Kwang-Ju Kim** (Associate Member, IEEE) received the B.S. degree in electronics engineering from Kyungpook National University (KNU), Daegu, South Korea, in 2010, the M.S. degree in electrical engineering from the Pohang Institute of Science and Technology (Postech), Pohang, South Korea, in 2013, and the Ph.D. degree in electronics engineering from KNU in 2020. From 2013 to 2015, he was a Researcher with General Electric Ultrasound Korea (GEUK). Since 2015, he has been with the Electronics and Telecommunications Research Institute (ETRI). His major research interests include computer vision, pattern recognition, and video surveillance.