Multi-Target Multi-Camera Tracking of Vehicles by Graph Auto-Encoder and Self-Supervised Camera Link Model

Hung-Min Hsu, Yizhou Wang, Jiarui Cai, Jenq-Neng Hwang
University of Washington
Seattle, WA, USA
{hmhsu, ywang26, jrcai, hwang}@uw.edu

Abstract

Multi-Target Multi-Camera Tracking (MTMCT) of vehicles is a challenging task in smart city related applications. The main challenge of MTMCT is how to accurately match the single-camera trajectories generated from different cameras and establish a complete global cross-camera trajectory for each target, i.e., the multi-camera trajectory matching problem. In this paper, we propose a novel framework to solve this problem using the self-supervised trajectory-based camera link model (CLM) with both appearance and topological features systematically extracted from a graph auto-encoder (GAE) network. Unlike most related works that represent the spatio-temporal relationships of multiple cameras with the laborious human-annotated CLM, we introduce a self-supervised CLM (SCLM) generation method that extracts the crucial multi-camera relationships among the vehicle trajectories passing through different cameras robustly and automatically. Moreover, we apply a GAE to encode topological information and appearance features to generate the topological embeddings. According to our experimental results, the proposed method achieves a new state-of-the-art on both CityFlow 2019 and CityFlow 2020 benchmarks with IDF1 of 77.21% *and* 55.56%*, respectively.*

1. Introduction

Multi-Target Multi-Camera Tracking (MTMCT) is the task of determining the trajectories of all objects of interest in a multi-camera system. It is a practical problem in the computer vision community, with various applications in the fields of intelligent transportation and smart city, *etc*. Usually, an MTMCT system is composed of two modules, *i.e*., single camera tracking (SCT) and inter-camera tracking (ICT). In the past decades, a great number of methods have been developed to track multiple targets under a single camera [47, 50, 53, 6, 56]. In this paper, we mainly focus on the cross-camera trajectory matching problem, i.e., ICT.

Despite its high application relevance, MTMCT remains a challenging task and a relatively unexplored territory in the deep learning community. Although some related key techniques, *i.e*., Multiple Object Tracking (MOT) [47, 50, 53] and object re-identification (ReID) [26, 48], are well-developed in recent years, tracking the objects across multiple cameras in a large city is still significantly challenging due to the following reasons: 1) the number of targets appearing in the entire camera network is unknown; 2) the number of cameras in which the target appears is also unknown.

There are two critical problems for multi-camera trajectory matching: 1) How many trajectories should be associated to establish a global trajectory since different targets appear in a different number of cameras? 2) The matched single-camera trajectory should obey the transition rule and mutually exclusive constraint (e.g., each single-camera trajectory can only be assigned to one global trajectory).

It is difficult for multi-camera trajectory matching to deal with these problems in a systematic way, therefore there are many works focused on using various constraints [31, 1, 52, 45]. Moreover, ICT can be accomplished using ReID with topological constraints to merge the SCT trajectories of the same identity in different cameras. Traditionally, most of the methods also formulate ICT as a graph clustering problem, where each SCT trajectory is referred to as a node in the graph. Therefore, the global ID for all the SCT trajectories from the same identity in different cameras can be established by grouping these nodes into the same cluster, based on the designed cost function to solve the graph optimization problem [20, 44, 31, 8], but none of them has successfully solve these issues by a deep learning framework.

ICT of vehicles is very challenging since vehicles can move across much more cameras than pedestrians in a fixed amount of time, moreover, the cross-camera vehicles usually occur in a very short time period, resulting in few image frames in each camera. In order to achieve better perfor-

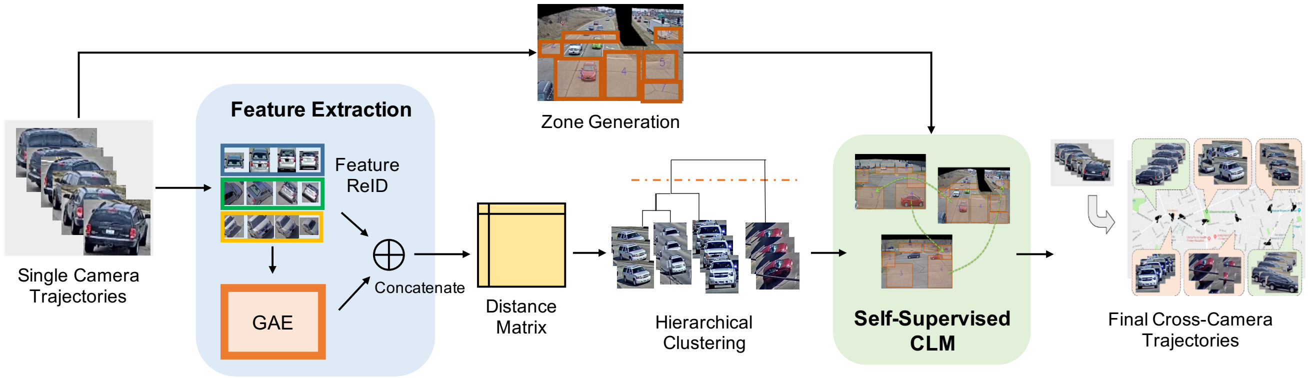

Figure 1. Illustration of the proposed MTMCT framework. First, single camera trajectories resulting from TSCT [25] is adopted as the input of our framework, and the features of each single camera trajectory is extracted by concatenating the ReID and GAE embeddings. Meanwhile, entry and exit zones are generated for each camera. In hierarchical clustering, the cross-camera trajectories with the high confidence scores are selected to generate the camera link model (CLM) from the entry and exit zones in the **self-supervised** manner. Finally, we use the generated CLM as the spatio-temporal constraint to obtain the final MTMCT results.

mance, ICT needs to consider more reliable spatio-temporal information such as camera link model (CLM) [24, 25, 23], which is either generated by human labeling or training data. Although CLM can provide superior performance, the exit/entry zones and zone links need to be labeled. [25] proposes a zone generation algorithm, however, the camera link construction still relies on the training data. In this work, we propose a framework to automatically generate the CLM.

The multi-camera trajectory matching problem is a global trajectory prediction problem, which is further formulated by us as a link prediction problem that has been well-studied in the recommendation system society. Therefore, we can solve this problem by exploiting the most effective solution of a link prediction Graph Neural Network (GNN).

In this paper, we perform data association in ICT by a learning-based framework, called self-supervised camera link model (SCLM), to automatically learn the spatio-temporal relationship in a multi-camera system. Inspired from the recommendation system, whose grouping recommendation concept [35] is very suitable for ICT since it also tries to cluster the nodes and there already exist robust graph neural networks that can be applied to solve our ICT problem. We also can use GNNs to generate the embedding for our multi-camera trajectory matching. Since GNNs can be used for solving the many-to-many matching scenario instead of the ReID concept that only use a Siamese network to train the embedding by one to one comparison. A GNN can consider multiple cameras at the same to make a more global decision instead of considering only two cameras at the same time. Therefore, in this paper, a graph auto-encoder (GAE) architecture is adopted to encode the topological information and appearance features together to generated the CLM. Unlike those likelihood encoding neural networks [24, 25], which only learn the pairwise similarity for data association in the single camera trajectories and cross-camera trajectories, the GAE can be used to cluster several single-camera trajectories to generate the final cross-camera trajectories. A GAE can learn the topological graph information from the convolutional neural network (CNN) features, i.e., global interactions among the trajectories can be considered. The flowchart of our proposed MTMCT system is shown in Fig. 1. Our experimental results show that the proposed framework can achieve the state-of-the-art performance in both CityFlow 2019 and CityFlow 2020, which illustrates the system's superior performance and generalizability.

To summarize, we claim the following contributions,

- A novel trajectory feature extractor that applies the graph auto-encoder (GAE) to fuse the topological features and original appearance features of each trajectory for ICT.
- The self-supervised camera link model (SCLM) is systematically established to enhance the performance of MTMCT by using the topological and temporal information.
- Our proposed method achieves the state-of-the-art performance (**the 1st place**) on both CityFlow 2019 and CityFlow 2020 benchmarks used in the CVPR AICity Challenges[1].

The rest of this paper is organized as follows. Section 2 reviews related works. Then, the framework of the proposed MTMCT system is described in Section 3. In Section 4, we evaluate the proposed method on the CityFlow 2019 and

[1] https://www.aicitychallenge.org/

CityFlow 2020 dataset [49] with ablation studies. Finally, the paper is concluded in Section 5.

2. Related Works

There is a large number of literatures on Multiple Object Tracking [47, 50, 53, 6, 56, 58, 47, 34]. Basically, MTMCT approaches can be grouped into two categories: graph based and non-graph based. To associate the local trajectories from SCT across different cameras, many approaches have been proposed, such as the Greedy Matching Association (GMA) method [4], and the Hierarchical Composition of Tracklet (HCT) framework [54]. Some researches exploit different information such as semantic attributes [55], appearance features [51, 57] and the motion patterns [20]. Moreover, some research works consider the camera topological configuration in MTMCT [32, 10, 38], which helps to match local tracklets between every two neighboring cameras. Methods in [4, 7] match single-camera trajectories between every pair of two adjacent cameras until trajectories across all the cameras are matched. Other methods [57, 27, 54] use greedy matching or hierarchical clustering to match all the trajectories across cameras iteratively. There are also research works use the Bayesian formulation to find a global solution for tracklet matching [55, 7]. [18] propose the Restricted Non-negative Matrix Factorization (RNMF) algorithm to compute the optimal assignment solution. On the other hand, candidate pruning using camera topology [27] and adaptive attribute selection [54] have also been proposed to reduce the search space in the matching process. Using camera topology to reduce the search space usually can achieve better performance than other methods [24, 25], however, the topological connectivity of camera network requires human labeling, which is not feasible for large scale practical application. In contrast, the proposed SCLM aims to establish the topological connectivity of camera network automatically (i.e., camera link) in a self-supervised manner.

The other category of research works in ICT regards the data association as a graph optimization problem, where each local trajectory is a node, and the values of edges represent the likelihood of the connected trajectories belonging to the same identity. The data association can then be formulated as an integer programming problem [2, 42] or, equivalently, minimum cost flow problem with either fixed costs based on distances [20, 44, 31, 8] along with motion attributes [14], and/or appearance attributes [9].

Moreover, after the GNN is first introduced in [43], several works have focused on improving the graph-structures. More specifically, by using different convolutional variants [5, 11, 28] as the input of GNNs, some approaches apply neural message passing [14] and extend the general GNN framework to customized graph networks. In a GNN, the nodes are generated by the initial features from a CNN, and the edges are trained by the node features and its neighbors in the graph. GNNs have achieved impressive performance in a wide variety of applications, including image captioning, action recognition [15], visual question answering [31], single object tracking [13] or single camera multiple object tracking [3].

However, to the best of our knowledge, there is no research using GNN for MTMCT. In this paper, we exploit the graph auto-encoder (GAE) [30] to encode the appearance features and graph information simultaneously so that more informative topological knowledge can be embedded than the previous methods. Through GAE training, topological information is embedded into these nodes and edges to form the representations of the entire graph. Hence, we can use the topological embedding trained by the GAE to construct CLM in a self-supervised way, and then apply the self-supervised learned CLM and topological embedding to generate the global ID assignment results.

3. Proposed MTMCT Framework

3.1. Framework Overview

We first define the MTMCT task as follows. Assume there are V videos from V different cameras, then the global trajectory can be denoted as $\mathcal{T} = \{\Xi^1, \Xi^2, \cdots, \Xi^V\}$, where each element Ξ^i in $\mathcal{T}$ indicates local trajectory set that includes all trajectories of camera i. The local trajectory is defined as $\Xi_i = \{\xi_1^i, \xi_2^i, \cdots, \xi_j^i\}$, where i and j denote the index of camera and the index of trajectory in camera i, respectively.

As shown in Alg. 1, there are four steps in the proposed MTMCT framework: (1) Apply a tracking-by-detection method to generate SCT results. (2) Train a ReID model to extract the appearance feature of each trajectory. (3) Use appearance feature of each trajectory as a node to train a GAE to generate the trajectory of high confidence for establishing the trajectory-based camera link model (i.e., the spatial and temporal constraint) for ICT. (4) Use the feature of each trajectory and trajectory-based camera link model to generate the ICT results.

In an MTMCT system, the first step is to perform SCT, whose purpose is to produce Ξ_i. Since we mainly focus on ICT, here we use the traffic-aware single camera tracking (TSCT) [25] as our SCT tracker, which has been proved to achieve the stat-of-the-art performance on MTMCT of vehicles .

According to [25], the SCT results are used to obtain the entry/exit points of each trajectory which are the center point of the first/last detected bounding boxes of each trajectory. By using MeanShift clustering algorithm [12], the encompassing bounding boxes for each cluster can be generated as zones in the camera and the entry/exit zone densities to determine the type for each clustered zone can

Algorithm 1: The proposed MTMCT algorithm.

```
Input  : Detections set $\mathcal{D}$ from all $V$ cameras.
Output: Global ID for all trajectories within all $V$ cameras.
for camera $i$ to $V$ do
    $\Xi^i \leftarrow$ TSCT($\mathcal{D}$) [25] // generate trajectories set $\Xi^i = \{\xi_n^i\}$ of camera $i$
    $Z^i \leftarrow$ MeanShift($\Xi^i$) // generate zones $Z^i$ from trajectories $\Xi^i$
    Calculate the exit density $D_x$ and entry density $D_e$ to generate the exit zones and entry zones by Eq. (1) and Eq. (2);
    Train video-based ReID $\mathcal{A} \leftarrow$ TA-ReID($\mathcal{T}$);
    Train graph auto-encoder GAE($\mathcal{A}$);
    for trajectory $\xi_j^i$ from $\Xi^i$ do
        $\mathcal{A}(\xi_j^i) \leftarrow$ TA-ReID($\xi_j^i$) // extract appearance embedding using the Temporal Attention ReID model
        $\mathcal{G}(\xi_j^i) \leftarrow$ GAE($\xi_j^i$) // obtain topological features
        $\mathbf{f}(\xi_j^i) = \mathcal{A}(\xi_j^i) \oplus \mathcal{G}(\xi_j^i)$;
    end
end
Cross-camera trajectories $\leftarrow$ Hierarchical Clustering($\mathbf{f}(\Xi)$);
Use high confident cross-camera trajectories to establish $CLM$;
Generate Global IDs by selecting cross-camera trajectories with valid $CLM$ constraint;
```

be computed. The entry and exit zone densities in each zone (say the k-th) are defined as D_e and D_x,

$$D_e = \frac{N_{e,k}}{N_{e,k} + N_{x,k}},\ D_x = \frac{N_{x,k}}{N_{e,k} + N_{x,k}}. \quad (1)$$

where $N_{x,k}$ and $N_{e,k}$ denote the number of exit/entry points in the k-th MeanShift clustered zone, respectively. If the density of an entry or exit zone is higher than a threshold ρ_e or ρ_x, this zone is recognized as an entry or exit zone, respectively. Moreover, the number of clustered points in each zone needs to be over a specific threshold; otherwise, the zone will be removed.

$$Z = \begin{cases} \textit{entry zone} & \text{if } D_e > \rho_e, \\ \textit{exit zone} & \text{if } D_x > \rho_x, \\ \textit{don't care} & \text{otherwise.} \end{cases} \quad (2)$$

To extract the embedding features of the SCT generated trajectories, the video-based ReID training is adopted to learn the embedding, instead of using image-based ReID training since video-based ReID training has shown better performance due to its taking into account of the temporal information. More specifically, in terms of video-based ReID, we first use a ResNet-50 [16] CNN pre-trained on ImageNet as the backbone to extract frame-level appearance embedding, which is extracted from the 2048-dim fully-connected layer. As shown in Fig. 2, we then apply a Temporal Attention (TA) mechanism [25] to perform a weighted average on the frame-level features to create a clip-level embedding features. After that, an average pooling operation is applied to generate the trajectory-level embedding $\mathcal{A}$. The intuition is that the weighting of each frame in a trajectory should be different due to the different degree of occlusion. In the above appearance features extraction, we only consider triplet-loss based metric during the network training based on a pair of adjacent cameras, which is usually ineffective for a large-scale multi-camera system with a large number of vehicles. Thus, we further consider to use topological information by involving more trajectories from a group of adjacently connected cameras at the same time via the following graph auto-encoder (GAE) network mentioned in Section 3.2.

Finally, we use the appearance embedding extracted from pair-wise video-based ReID and topological embedding from the group-camera based GAE to generate the final embedding, which can then be greedily grouped based on the distance matrix of all the connected camera associations by hierarchical clustering to generate the final cross-camera trajectories. During the training phase, the high-confidence cross-camera trajectories can be used for establishing the CLM (i.e., spatio-temporal constraints). We can then use the lower threshold in the online tracking stage to select the cross-camera trajectories from the greedy grouping with the spatio-temporal constraints from the learned SCLM (mentioned in Section 3.3).

3.2. Topological Embedding Extraction via GAE

MTMCT usually needs to incorporate a hierarchical clustering (grouping) algorithm for generating global IDs. However, it does not ensure the optimal solution. Therefore, we use a graph auto-encoder (GAE) [30] neural network to create discriminative topological embedding and assist the clustering of these trajectories from different cameras and thus produce the final MTMCT results (see Fig. 2). Basically, we use a graph convolutional network (GCN) [29] to construct the GAE for clustering, which is a multi-layer neural network that operates directly on a graph and induces topological embedding vectors of nodes based on properties of their neighborhood.

In order to generate the global trajectory, MTMCT can be treated as a correlation clustering problem, which can be

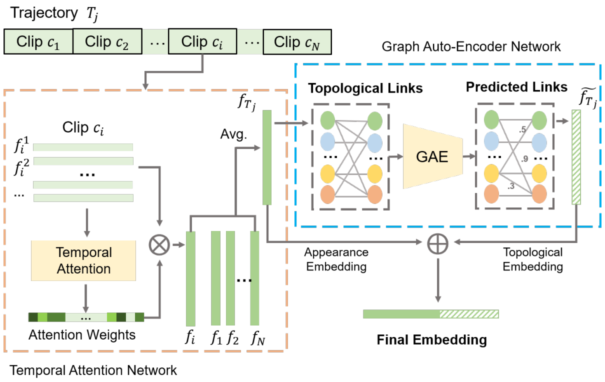

Figure 2. Illustration of the generation of the final embedding features. The appearance embedding is generated by a temporal attention architecture based on ResNet-50 as the backbone. The topological embedding is generated by the GAE network.

defined as the follows [36]:

$$\mathcal{X}^* = \arg\max_{\{x_{i,j}\}} \sum_{(i,j)\in E} w_{i,j} x_{i,j}, \qquad (3)$$
$$\text{s.t.,}\ \ x_{i,j} + x_{j,k} \leq 1 + x_{i,k}, \ \ \forall i,j,k \in V,$$

where the set $\mathcal{X}$ is the set of all possible combinations of assignments to the binary variables $x_{i,j}$, that should be set as 1 if two cross-camera trajectories i and j are created by the same vehicle identity. Thus, we maximize $\mathcal{X}^*$ based on rewarding the edges that associate multi-camera trajectories of the same vehicle and penalizing the edges that associate different vehicles, with Eq. (3) being used as the constraint to enforce the transitivity in the solution. In this section, we will illustrate how to use GAE and camera link models to solve the binary integer programming (BIP) problem.

A graph is represented as $G = (V, E)$, where $V(|V| = n)$ and E are the sets of nodes and edges, respectively. Referring ICT as a graph, the nodes denote the cross-camera trajectories and the edge represents the likelihood of the two corresponding nodes that belong to the same vehicle. Let $X \in \mathbb{R}^{n\times m}$ be a matrix containing all n nodes and each node is represented as m-dimensional feature vector, *i.e.*, the trajectory-level appearance embedding. Thus, each row $X_v \in \mathbb{R}^m$ is the feature vector for cross-camera trajectory v. In ICT scenario, we define an adjacency matrix A of G to indicate which trajectories represent the same vehicle. We apply two-layers GCN to construct the GAE and extract the topological information of higher order neighborhoods since one-layer GCN can only capture the topological information from immediate neighbors. In order to stack two-layers GCN, we need to compute a k-dimensional node feature matrix $L^{(1)} \in \mathbb{R}^{n\times k}$ as

$$L^{(1)} = \rho(\tilde{A}XW_0), \qquad (4)$$

where $\tilde{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ is the normalized symmetric adjacency matrix and D is the degree matrix D, where $D_{ii} = \sum_j A_{ij}$. $W_0 \in \mathbb{R}^{m\times k}$ denotes a weight matrix; ρ is a rectified linear activation unit. Finally, we can develop a GAE by calculating embedding Z as an encoder. The GAE can then be trained based on the reconstructed error. A reconstructed adjacency matrix $\hat{A}$ can thus be calculated from embedding Z as

$$\hat{A} = \sigma(ZZ^\top), \qquad (5)$$

where

$$Z = GCN(X, A) = \tilde{A}L^{(1)}W_j = \tilde{A}\rho(\tilde{A}XW_0)W_j. \qquad (6)$$

Note that ρ is the sigmoid function. Specifically, the optimization goal is to minimize the reconstruction error between the original adjacency matrix A and the reconstructed adjacency matrix $\hat{A}$, *i.e.*,

$$\arg\min ||\hat{A} - A||. \qquad (7)$$

In the training stage of GAE, the binary cross-entropy loss [30] is adopted as the loss function

$$\mathcal{L}_{Xent} = -\frac{1}{N}\sum_{j=1}^{N} y_j \log(\hat{y_j}) + (1 - y_j)\log(1 - \hat{y_j}), \qquad (8)$$

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

After training the GAE, we can use embedding Z to decode the probability of connectivity for all nodes as the topological embedding $\mathcal{G}$ (*i.e.*, the edges of the graph), and concatenate the topological embedding $\mathcal{G}$ with the appearance embedding $\mathcal{A}$ as the final embedding F_{final},

$$F_{final} = \mathcal{G} \oplus \mathcal{A}. \qquad (9)$$

We can then use the final embedding features F_{final}, trajectory-based CLM and hierarchical clustering algorithm to produce the global IDs for MTMCT. Since the search space of ICT ReID is significantly reduced by transition time constraint imposed by the CLM, the Rank-1 accuracy will be close to 1. Therefore, we can greedily select the smallest pair-wise distance to merge the tracked vehicles across cameras. Furthermore, the temporal order between different tracked vehicles can also be used as a constraint to further reduce the search space of the ReID, since the orders of vehicles should be almost the same between two adjacently connected cameras. Finally, we can define a distance matrix $\mathbf{M}$,

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \cdots & \mathbf{M}_{1N} \\ \vdots & \ddots & \vdots \\ \mathbf{M}_{N1} & \cdots & \mathbf{M}_{NN} \end{bmatrix}, \qquad (10)$$

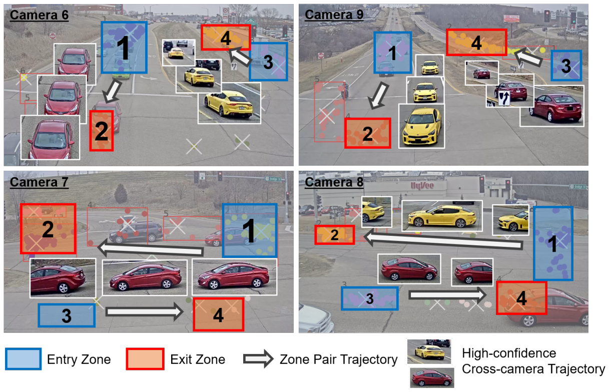

Figure 3. The illustration for self-supervised trajectory-based camera link model in a multi-camera system with four different views.

where

$$\mathbf{M}_{i,j} = \begin{cases} \text{dist}(\mathbf{f}(\xi_i), \mathbf{f}(\xi_j)) & \text{if valid camera link constraint,} \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

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

$$\begin{aligned} \text{dist}(\mathbf{f}(\xi_i), \mathbf{f}(\xi_j)) &= \text{dist}(\mathcal{A}(\xi_i) \oplus \mathcal{G}(\xi_i), \mathcal{A}(\xi_j) \oplus \mathcal{G}(\xi_j)) \\ &= \left\| F_{final}(\xi_i) - F_{final}(\xi_j) \right\|_2 . \end{aligned} \tag{12}$$

We can now start to merge these trajectories based on the distance, the process can be repeated until there is no valid transition pair or the minimum distance is larger than a threshold, and also the pairs which conflict with previously matched pairs can be removed.

3.3. Self-Supervised CLM Generation

The camera link model is a proven strategy for imposing the spatio-temporal constraints in an MTMCT task [24, 25]. A CLM commonly consists of the transition time distribution of adjacent cameras. Therefore, the ambiguity of the same car type of different identities can be eliminated. However, the drawback of the camera link model is that the transition time distribution of the camera link needs to be trained, enforcing the camera configuration of training data and testing data have to be the same, which prevents the practical use of the CLM in the real world scenarios. In order to overcome this drawback, we propose a self-supervised framework to generate the trajectory-based CLM [24], which establishes the transition distribution of each adjacently connected camera pairs, then use the upper bound and lower bound of transition time between these adjacently connected camera pairs as the time window, that can be used as a constraint to reduce the number of search candidates of associating cross-camera trajectories for ICT.

First of all, the CLM is composed of many linked camera pairs, and there are many exit/entry zones in each camera for camera connection. These exit/entry zones are generated by MeanShift clustering the exit/entry points of every SCT trajectory, then a rectangular bounding box is used to encompass these points of each cluster as the exit/entry zone. The exit point and entry point denote the position of the last and first frame of a trajectory in each camera between a linked camera pair, respectively. If one trajectory passing from an entry zone to an exit zone, we define the passed entry/exit zones as a zone pair trajectory. Therefore, there are many zone pair trajectories in each camera for representing different orientations of vehicle moving trajectories of this camera, since the driving orientation of a trajectory is commonly fixed based on the standard traffic rule (as shown in Fig. 3). The zone pair trajectories are used in the CLM so that the trajectory-based CLM between an adjacently connected camera pair can be defined as $L = (C^s, C^d)$. $C^s = \{P_i^s\}_{i=1}^m$ and $C^d = \{P_j^d\}_{j=1}^n$ denote the zone pair trajectories set in the source camera and destination camera, respectively.

Take Fig. 3 as an example, there are four cameras with overlapping FoVs, i.e., camera 6, camera 7, camera 8 and camera 9. The exit/entry zones are denoted as red/blue bounding boxes, respectively. The exit zone in Camera 6 (red bounding box) and the entry zone in Camera 7 (blue bounding box) indicate the camera connectivity. In this scenario, the vehicles exiting from the camera 6 will appear in the Camera 7 immediately, similarly for camera 8 to camera 9. Take the connectivity of camera 6 and camera 7 as an example, there is a transition camera pair $\text{Ł} = (C^6, C^7)$, where $C^6 = \{P_1^6, P_2^6\}$ and $C^7 = \{P_1^7, P_2^7\}$ have two zone pair trajectories, respectively. Therefore, P_1^6 and P_1^7 are one camera link while $P_1^6 = \{1, 2\}$ and $P_1^7 = \{1, 2\}$.

Therefore, C^s and C^d are used to enforce the spatio-temporal constraints, which specify that if there is a trajectory from P_i^s to P_j^d (*i.e.*, from one exit zone of the source camera to the other entry zone of the adjacently connected destination camera), the transition time of these two camera zones is bound by

$$\Delta t = t^s - t^d. \tag{13}$$

Here t^s and t^d are the time stamp the trajectory passing from source camera to destination camera, respectively. The camera link L is represented as a time window $(\Delta t_{\text{min}}, \Delta t_{\text{max}})$, thus only the transition time of vehicle trajectory pairs within the time window are considered for ICT. Moreover, there are more than one transition time constraints for each camera pair since the traffic is bidirectional.

Although the exit/entry zones are generated by the SCT results, each trajectory may not pass through the exit/entry zones, since these MeanShift generated exit/entry zones are merely created by the clusters, which are impossible to cover all the potential trajectories. However, we still need to produce the zone-pair trajectory for all the trajectories to construct the camera links. To solve this issue, let us define the distance between a tracked vehicle and a zone-pair trajectory [24] as

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

where P represents the zone-pair trajectory; V denotes the passing zones of the tracked vehicle and α_z is the maximum IOU of the bounding boxes of the trajectory and the zone z. All of these zone-pair trajectories are directional, which means that the trajectory needs to possess the same driving orientation as the matched zone-pair trajectory. Otherwise, the distance will be set as infinity. Eventually, each trajectory in the camera can be assigned one zone-pair trajectory after associating it with all the possible zone-pair trajectories.

The final step is to train the trajectory-based camera link model in a self-supervised manner, which is shown in Fig. 3. As discussed previously, the camera links are commonly trained by the labeled training data, resulting in an obvious drawback that the cameras' configuration in the testing has to be the same as that of the training. In order to overcome this issue and avoid laborious ground truth labeling of cross-camera trajectories, we use video-based ReID across the exit-entry zones to systematically generate the high-confidence cross-camera trajectories, *i.e.*, we associate the trajectories if the similarity confidence score is higher than a specific threshold. Based on the high-confidence cross-camera trajectories, we can establish the camera links in a self-supervised manner without using pre-labelled training data pairs. We then enlarge the transition time windows of these camera links proportionally since this time windows are created by merely using a limited number of high confident cross-camera trajectories instead of sufficient cross-camera trajectories.

4. Experiments

4.1. Dataset

In this section, we show the performance of the proposed MTMCT method on the CityFlow 2019 dataset [49], which is the most representative city-scale benchmark for multi-camera multi-vehicle tracking, and compare our method with the state-of-the-art MTMCT approaches. CityFlow 2019, originally released for CVPR 2019 AICity Challenge, includes about 3.5 hours of videos from 40 cameras across 10 intersections in a mid-sized U.S. city, and the spanned distance is over 2.5 miles. Furthermore, there are 229,680 bounding boxes from 666 vehicles with obscured license plates due to the privacy issue. In addition, CityFlow 2019 dataset is further augmented with extra six cameras' videos as the new testing data, which is called CityFlow 2020 testing set, for CVPR 2020 AI City challenge workshop [37]. Thus, we also evaluate the proposed method on the CityFlow 2020 testing set for MTMCT evaluations.

4.2. Implementation Details

ICT implementation. The frame-level ReID appearance features are extracted by the ResNet-50 backbone pretrained by ImageNet with batch size of 32. The feature dimension is 2048 with the clip size of 4. The learning rate and weight decay are 3×10^{-4} and 5×10^{-4}, respectively. During the training of appearance feature, we resize the input image size as 224×224. Then, we use the TA-weighted trajectory-level ReID appearance features as the input of GAE. For GAE, we use Adam as the optimizer and the learning rate is set as 10^{-6}; we set the embedding size of the first convolution layer as 1024.

4.3. MTMCT Results

In MTMCT, the IDF1 score [41] is the mainstream evaluation metrics, which can be used to estimate the ratio of correctly identified vehicles over the average number of ground-truth and predicted vehicles, taking into account the ID switches in the tracking. Therefore, IDF1 is selected [37, 49] as the evaluation metric to rank the performance of the competing methods in both CityFlow 2019 and CityFlow 2020 benchmarks. The proposed SCLM scheme is shown to increase the performance of MTMCT and achieve the best performance comparing with all SOTA methods, based on our automatically generated camera links, which are constructed in a self-supervised manner without pre-labeled training data to create the transition time window for each camera link. According to Table 1, our experimental results show that our system not only can generate the CLM without training but also can improve the MTMCT tracking performance on CityFlow 2019 dataset by our GAE embedding.

In addition, we also evaluate the performance of GAE features by using the trained camera link model since the cameras in the CityFlow 2019 testing data have been included in the training data in CityFlow 2019 except for four overlapping field of view cameras. Table 1 and Table 3 show that the IDF1 of labeled and trained CLM is still better than the self-supervised camera link model (SCLM). Moreover, the results of our proposed method on CityFlow 2020 testing set, as shown in Table 2, also prove that the proposed SCLM can be generalized to deal with new cameras scenario since the cameras in CityFlow 2020 testing set are not overlapped with any of the cameras in CityFlow 2019 dataset. Finally, the Table 1 and Table 2 demonstrate that our method outperforms all the state-of-the-art methods, which achieves IDF1 77.21% and 55.56%, respectively. Fig. 4 shows some qualitative results, which indicate the proposed method can be generalized for different cam-

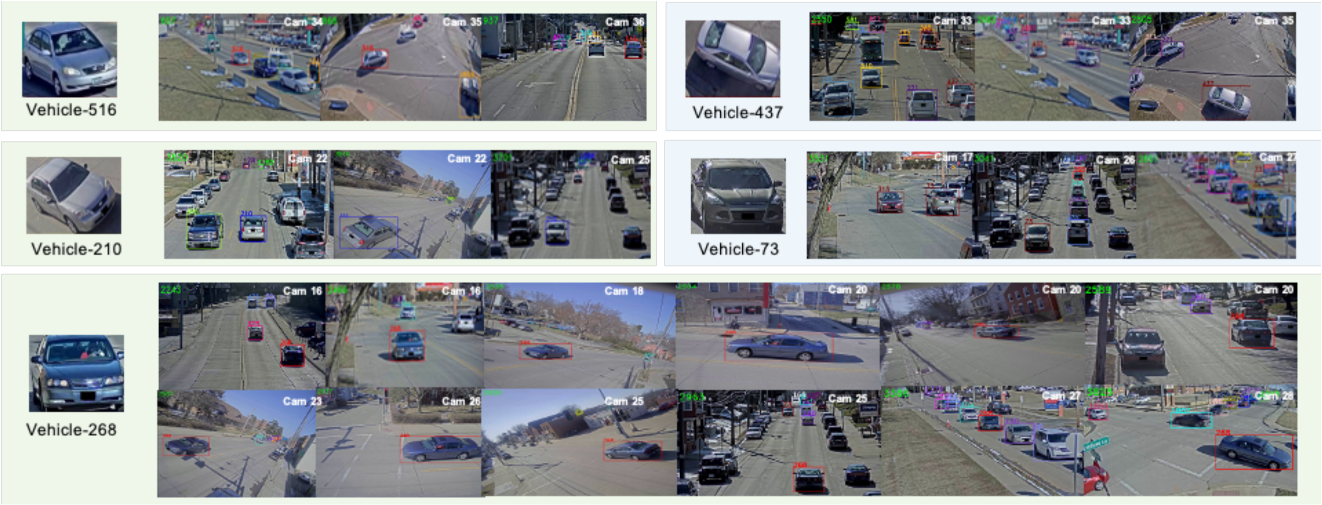

Figure 4. Qualitative results of the proposed MTMCT method.

Methods	IDF1
MOANA+BA [49]	0.3950
DeepSORT+BS [49]	0.4140
TC+BA [49]	0.4630
ZeroOne [46]	0.5987
DeepCC [42]	0.5660
LAAM [22]	0.6300
ANU [21]	0.6519
TrafficBrain [19]	0.6653
DDashcam [33]	0.6865
UWIPL [24]	0.7059
TSCT+TA [25]	0.7493
Ours (SCLM+GAE)	**0.7547**
Ours (CLM+GAE)	**0.7721**

Table 1. Results comparison on the CityFlow 2019 benchmark.

Methods	IDF1
UMD_RC [39]	0.1245
TRACTA [17]	0.4400
ELECTRICITY [40]	0.4585
Ours (SCLM+GAE)	**0.5556**

Table 2. Results comparison on the CityFlow 2020 benchmark.

eras and vehicles.

4.4. Ablation Study

In this subsection, we show the ablation study of the proposed method. In Table 3, there are two major components to be justified in the MTMCT of vehicles, i.e., the GAE features and the temporal attention (TA) based appearance feature. As shown in Table 3, when replacing TA with the GAE, IDF1 based on the GAE features increases by 3.26% and 2.28% on SCLM and CLM, respectively. The experimental results show that the proposed method is able to achieve the best IDF1. Even in SCLM setting, we can see that the IDF1 of the proposed method with SCLM is better than the state-of-the-art method with CLM (i.e., using supervisedly trained camera link models). Therefore, the proposed SCLM is more suitable for real-world applications since it is impossible to supervisedly train the CLMs for every intelligent transportation system in the world.

	SCT+ICT	IDF1	IDP	IDR
CLM	TNT+TA	0.7059	0.6912	0.7211
	TSCT+TA	0.7493	0.8071	0.6918
	TSCT+GAE	**0.7721**	**0.8243**	**0.7262**
SCLM (self-supervised)	TNT+TA	0.6861	0.6966	0.6760
	TSCT+TA	0.7221	0.7896	0.6653
	TSCT+GAE	**0.7547**	**0.8130**	**0.7042**

Table 3. The MTMCT performance for different combinations of the proposed method on CityFlow 2019 dataset.

5. Conclusion

In this paper, we propose a novel framework for Multi-Target Multi-Camera Tracking (MTMCT) based on self-supervised camera link model (SCLM) and graph auto-encoder (GAE). From our experiments, the proposed method is efficient, effective and robust, with achieved IDF1 77.21% and 55.56% tracking performance on CityFlow 2019 and CityFlow 2020 benchmarks achieving the state-of-the-art performance in the MTMCT of vehicles task.

References

[1] Alexandre Alahi, Laurent Jacques, Yannick Boursier, and Pierre Vandergheynst. Sparsity driven people localization with a heterogeneous network of cameras. *Journal of Mathematical Imaging and Vision*, 41(1):39–58, 2011.

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

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

[5] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[6] Jiarui Cai, Yizhou Wang, Haotian Zhang, Hung-Min Hsu, Chengqian Ma, and Jenq-Neng Hwang. Ia-mot: Instance-aware multi-object tracking with motion consistency. *BMTT Challenge Workshop, IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[7] 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.

[8] 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, 2016.

[9] Xiaojing Chen and Bir Bhanu. Integrating social grouping for multitarget tracking across cameras in a crf model. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(11):2382–2394, 2016.

[10] 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.

[11] 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.

[12] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.

[13] Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc Van Gool. A mobile vision system for robust multi-person tracking. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[14] 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.

[15] Davi Frossard and Raquel Urtasun. End-to-end learning of multi-sensor 3d tracking by detection. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 635–642. IEEE, 2018.

[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, Jie Han, Wentao Yu, Xiaopeng Hong, Xing Wei, and Yihong Gong. City-scale multi-camera vehicle tracking by semantic attribute parsing and cross-camera tracklet matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 576–577, 2020.

[18] 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.

[19] Zhiqun He, Yu Lei, Shuai Bai, and Wei Wu. Multi-camera vehicle tracking with powerful visual features and spatial-temporal cue. In *Proc. CVPR Workshops*, pages 203–212, 2019.

[20] 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.

[21] Yunzhong Hou, Heming Du, and Liang Zheng. A locality aware city-scale multi-camera vehicle tracking system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 167–174, 2019.

[22] Yunzhong Hou, Liang Zheng, Zhongdao Wang, and Shengjin Wang. Locality aware appearance metric for multi-target multi-camera tracking. *arXiv preprint arXiv:1911.12037*, 2019.

[23] Hung-Min Hsu, Jiarui Cai, Yizhou Wang, Jenq-Neng Hwang, and Kwang-Ju Kim. Multi-target multi-camera tracking of vehicles using metadata-aided re-id and trajectory-based camera link model. *IEEE Transactions on Image Processing*, 30:5198–5210, 2021.

[24] 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.

[25] Hung-Min Hsu, Yizhou Wang, and Jenq-Neng Hwang. Traffic-aware multi-camera tracking of vehicles based on reid and camera link model. *arXiv preprint arXiv:2008.09785*, 2020.

[26] Tsung-Wei Huang, Jiarui Cai, Hao Yang, Hung-Min Hsu, and Jenq-Neng Hwang. Multi-view vehicle re-identification using temporal attention model and metadata re-ranking. In *AI City Challenge Workshop, IEEE/CVF Computer Vision and Pattern Recognition (CVPR) Conference, Long Beach, California*, 2019.

[27] 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 *Proceedings of the 26th ACM international conference on Multimedia*, pages 1457–1465, 2018.

[28] Margret Keuper, Siyu Tang, Bjoern Andres, Thomas Brox, and Bernt Schiele. Motion segmentation & multiple object tracking by correlation co-clustering. *IEEE transactions on pattern analysis and machine intelligence*, 42(1):140–153, 2018.

[29] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[30] Thomas N Kipf and Max Welling. Variational graph autoencoders. *arXiv preprint arXiv:1611.07308*, 2016.

[31] 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.

[32] 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.

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

[34] 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.

[35] Qi Liu, Ruobing Xie, Lei Chen, Shukai Liu, Ke Tu, Peng Cui, Bo Zhang, and Leyu Lin. Graph neural network for tag ranking in tag-enhanced video recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2613–2620, 2020.

[36] Charles Morefield. Application of 0-1 integer programming to multitarget tracking problems. *IEEE Transactions on Automatic Control*, 22(3):302–312, 1977.

[37] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Xiaodong Yang, Liang Zheng, Anuj Sharma, Rama Chellappa, and Pranamesh Chakraborty. The 4th ai city challenge. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, page 2665–2674, June 2020.

[38] Weizhi Nie, Anan Liu, Yuting Su, Huanbo Luan, Zhaoxuan Yang, Liujuan Cao, and Rongrong Ji. Single/cross-camera multiple-person tracking by graph matching. *Neurocomputing*, 139:220–232, 2014.

[39] Neehar Peri, Pirazh Khorramshahi, Sai Saketh Rambhatla, Vineet Shenoy, Saumya Rawat, Jun-Cheng Chen, and Rama Chellappa. Towards real-time systems for vehicle re-identification, multi-camera tracking, and anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 622–623, 2020.

[40] Yijun Qian, Lijun Yu, Wenhe Liu, and Alexander G Hauptmann. Electricity: An efficient multi-camera vehicle tracking system for intelligent city. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 588–589, 2020.

[41] 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.

[42] Ergys Ristani and Carlo Tomasi. Features for multi-target multi-camera tracking and re-identification. *arXiv preprint arXiv:1803.10859*, 2018.

[43] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

[44] 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.

[45] 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, 2017.

[46] Xiao Tan, Zhigang Wang, Minyue Jiang, Xipeng Yang, Jian Wang, Yuan Gao, Xiangbo Su, Xiaoqing Ye, Yuchen Yuan, Dongliang He, et al. Multi-camera vehicle tracking and re-identification based on visual and spatial-temporal features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 275–284, 2019.

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

[48] Zheng Tang, Milind Naphade, Stan Birchfield, Jonathan Tremblay, William Hodge, Ratnesh Kumar, Shuo Wang, and Xiaodong Yang. Pamtri: Pose-aware multi-task learning for vehicle re-identification using highly randomized synthetic data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 211–220, 2019.

[49] 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.

[50] Zheng Tang, Gaoang Wang, Hao Xiao, Aotian Zheng, and Jenq-Neng Hwang. Single-camera and inter-camera vehicle tracking and 3d speed estimation based on fusion of visual and semantic features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 108–115, 2018.

[51] 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.

[52] Yonatan Tariku Tesfaye, Eyasu Zemene, Andrea Prati, Marcello Pelillo, and Mubarak Shah. Multi-target tracking in multiple non-overlapping cameras using fast-constrained dominant sets. *International Journal of Computer Vision*, 127(9):1303–1320, 2019.

[53] 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.

[54] 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.

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

[56] Haotian Zhang, Yizhou Wang, Jiarui Cai, Hung-Min Hsu, Haorui Ji, and Jenq-Neng Hwang. Lifts: Lidar and monocular image fusion for multi-object tracking and segmentation. *BMTT Challenge Workshop, IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[57] Zhimeng Zhang, Jianan Wu, Xuan Zhang, and Chi Zhang. Multi-target, multi-camera tracking by hierarchical clustering: Recent progress on dukemtmc project. *arXiv preprint arXiv:1712.09531*, 2017.

[58] 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.