Traffic-Aware Multi-Camera Tracking of Vehicles Based on ReID and Camera Link Model

Hung-Min Hsu
hmhsu@uw.edu
University of Washington
Seattle, Washington

Yizhou Wang
ywang26@uw.edu
University of Washington
Seattle, Washington

Jenq-Neng Hwang
hwang@uw.edu
University of Washington
Seattle, Washington

ABSTRACT
Multi-target multi-camera tracking (MTMCT), i.e., tracking multiple targets across multiple cameras, is a crucial technique for smart city applications. In this paper, we propose an effective and reliable MTMCT framework for vehicles, which consists of a traffic-aware single camera tracking (TSCT) algorithm, a trajectory-based camera link model (CLM) for vehicle re-identification (ReID), and a hierarchical clustering algorithm to obtain the cross camera vehicle trajectories. First, the TSCT, which jointly considers vehicle appearance, geometric features, and some common traffic scenarios, is proposed to track the vehicles in each camera separately. Second, the trajectory-based CLM is adopted to facilitate the relationship between each pair of adjacent cameras and add spatio-temporal constraints for the subsequent vehicle ReID with temporal attention. Third, the hierarchical clustering algorithm is used to merge the vehicle trajectories among all the cameras to obtain the final MTMCT results. Our proposed MTMCT is evaluated on the CityFlow dataset and achieves a new state-of-the-art performance with IDF1 of 74.93%.

KEYWORDS
• Computing methodologies → Tracking; Computer vision tasks; Neural networks.

1 INTRODUCTION
Due to the exponential growth of the deployed surveillance cameras with networking supports, the opportunity to take advantage of the rich information from the multi-camera systems is immense. Among the techniques for multi-camera systems, multi-target multi-camera tracking (MTMCT) is important for traffic flow optimization and anomaly detection. Basically, MTMCT consists of two sub-tasks: 1) Single camera tracking (SCT): Detection and tracking of the objects within each single camera; 2) Inter-camera tracking (ICT): Associations of the object trajectories across different cameras. Overall, MTMCT is aimed to obtain the trajectories of every object in the scene through all the cameras in the system.

However, MTMCT is a very challenging task due to unreliable object detection, heavy occlusion, low resolution, and varying lighting and viewing perspective conditions. Recently, many proposed works for MTMCT take into account target motion and human pose. Among these works, most of them are person based MTMCT, which only considers humans as the tracking targets. For vehicles, this task becomes more challenging because: 1) vehicles may stop for a long time at the traffic signs and continually be occluded among each other, which makes occlusion even more severe; 2) inter-class similarity for vehicles is higher because there may exist many different identities with the similar appearance.

To solve these two aforementioned challenges, we propose a novel MTMCT framework, which mainly consists of two innovations, i.e., traffic-aware single camera tracking (TSCT) and the trajectory-based camera link model (CLM).

First, TSCT is proposed to handle the long-term occlusions created in the traffic scenarios. Usually, there will be a large number of isolated and fragmented vehicle trajectories, created from a single camera multi-target tracker, in the center of the frames where vehicles do not enter or exit the camera’s field of view (FoV). For example, when a vehicle stops in front of a red traffic light, it can be partially or even fully occluded in the camera’s FoV for a long time. We would like to call this kind of zone as a traffic-aware zone. Here, we use the TrackletNet tracker (TNT) [36], which is a superior SCT method in intelligent transportation system application [11, 37, 38], as our single camera tracker. According to this condition, TSCT is proposed to find out the traffic-aware zones, where this kind of occlusion happens, by clustering the start and end nodes of all the resulting trajectories from the TNT. Vehicle Re-ID in the single camera is then implemented for these traffic-aware zones to connect these disconnected trajectories created in the traffic scenarios.

Second, facing higher inter-class appearance similarity of distinct vehicles, trajectory-based CLM is further proposed to impose spatio-temporal constraints and reduce solution search space for the cross camera ReID. For two different vehicles with a very similar appearance, it is nearly impossible to re-identify them using a typical ReID method. However, taking advantage of the spatial and temporal constraints between a pair of adjacent connected cameras, we can easily filter out the vehicles that are not likely to
Figure 1: The illustration of our MTMCT framework. First, the traffic-aware single camera tracking (TSCT) is utilized to obtain SCT results for each camera. Second, CLMs with entry/exit zones and transition times are automatically generated. Third, vehicle ReID with temporal attention is implemented on the solution space from the CLM. Finally, the hierarchical clustering is involved for the final MTMCT results.

To summarize, we claim the following contributions:

- Propose a new MTMCT framework specifically designed for vehicles.
- Utilize a novel TSCT strategy to improve vehicle SCT results, considering common traffic scenarios.
- Create a trajectory-based CLM generation method that adds spatio-temporal constraints and reduces the solution space for cross camera vehicle ReID.
- Achieve a new state-of-the-art performance on the CityFlow dataset.
improve the performance of MTMCT, as evidenced in some works [11, 15, 34, 35]. For example, Lee et al. [15] use the CLM to estimate bidirectional transition time distribution in an unsupervised scheme for MTMCT. Tang et al. [34, 35] and Hsu et al. [11] also use the car speed to generate the transition time distribution for each connected pair of adjacently connected cameras. Therefore, the accuracy of MTMCT can be significantly improved by CLM by reducing the candidate set of matching. In this paper, we systematically generate the CLMs to assist the ReID in the MTMCT tasks.

**Appearance feature based vehicle ReID.** Vehicle ReID has attracted more research efforts in the past few years. VeRi-776 [17] and VeRi-Wild [20] are the most widely used benchmarks, which provide not only the high quality annotations but also the camera deployment geometry. In ReID, there are two types of appearance feature extraction methods. One is traditional handcrafted methods, such as SURF [1] and ORB [26]. The other is CNN-based methods [41, 44], which have been proven to achieve better performance than traditional handcrafted methods in the past years. Thus, recent methods [17, 19] focus on learning an embedding model that maps the samples into an embedding space. Liu et al. [17] propose a mixed difference network using a vehicle model and ID information to construct more robust feature embedding. Liu et al. [19] propose a “PROVID” ReID model by using visual feature, license plate and spatio-temporal information for the vehicle ReID task. Shen et al. [27] incorporate complex spatio-temporal information for effectively regularizing the ReID results by a two-stage framework. Based on a multi-view inference scheme. Zhou et al. [46] generate a global-view feature representation to improve the vehicle ReID. Huang et al. [12] utilize car key points to train orientation-based embedding for vehicle ReID. Tang et al. [33] propose a pose-aware multi-task re-identification (PAMTRI) framework by explicitly reasoning about vehicle pose, shape, color, and types. In this work, we propose a trajectory-based CLM to enhance the performance of vehicle ReID.

### 3 Proposed Method

There are four steps in our proposed method, as illustrated in Fig. 1. First, we apply TSCT after a single camera tracker TNT, as presented in Section 3.1. Then, the camera links are established based on the generated entry/exit zones, as illustrated in Section 3.2. Moreover, we utilize a vehicle ReID method by combining the temporal attention and batch sampling for inter-camera tracking, as explained in Section 3.3. Finally, we use the hierarchical clustering to merge the tracklets into object trajectories, as described in Section 3.4.

#### 3.1 Traffic-Aware Single Camera Tracking (TSCT)

In our MTMCT framework, the first step is SCT, where we implement the TrackletNet Tracker (TNT) [36]. A TNT is a tracklet-based graph clustering approach, where the vertices are generated based on detection association via CNN-based appearance feature and the intersection-over-union (IOU) in the two consecutive frames. While, the edge weights are estimated by a TrackletNet, which is a Siamese neural network trained to predict the likelihood of the two tracklets belonging to the same object. After the tracklet-based

graph is constructed, graph clustering [35] is applied to merge the tracklets of the same vehicle into one single trajectory.

After using TNT to generate the SCT results, we observe that there are some vehicle ID switches due to some isolated trajectories, which usually appear while vehicles are waiting for a traffic light. Therefore, we propose TSCT to deal with this issue by generating the traffic-aware zones and performing single camera ReID to reconnect the isolated trajectories caused by the traffic scenarios.

First of all, all the traffic-aware zones within each camera are detected in an unsupervised manner based on the MeanShift clustering algorithm [4] applied on the collected entry/exit measurements. The procedure of traffic-aware zone generation is as follows: 1) use the first and the last positions of all the trajectories from the SCT results as zone nodes; 2) cluster the zone nodes into different groups by apply the MeanShift algorithm.

After the MeanShift, we define a rectangular zone for each group to bound the inside nodes. We denote $N_{e,k}$ as the number of entry nodes and $N_{x,k}$ as the number of exit nodes in the zone $k$. The traffic-aware density $D_{ta}$ is defined by

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

where $D_{ta}$ needs to be above a threshold $p_{ta}$, the zone will be designated as a traffic-aware zone, as shown in Fig. 2(c).

After the traffic-aware zones are generated, the next step is to reconnect the isolated trajectories in these zones. Here, we build a queue for each zone to store the ordering and appearance features of the vehicles that are interrupted in the traffic-aware zones. If there is a bounding box of an isolated trajectory that suddenly appears in the traffic-aware zone, we select a trajectory from the queue to compute the appearance similarity. The appearance feature is trained by single camera ReID, which is similar to the cross camera ReID, as explained in Section 3.3.

#### 3.2 Trajectory-based Camera Link Model

Our trajectory-based CLM can be divided into three steps: 1) entry/exit zones generation in each single camera; 2) vehicle trajectory classification according to entry-exit zone pairs in each single camera; 3) camera links and transition time estimation across different cameras. These three steps are described below.

**Entry/exit zones generation.** With the available routes and the detected entry/exit zones, we can estimate the relationship between each two zones. For a trajectory-based CLM, the procedure of the entry/exit zone generation is similar to the traffic-aware zone generation. The only difference is to calculate the entry and exit density to determine whether the generated zone is an entry/exit zone or not. The entry and exit densities are defined as $D_e$ and $D_x$, where

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

If the density of an entry or exit zone is higher than a threshold $p_e$ or $p_x$, this zone will be recognized as an entry or exit zone. There may be multiple entry/exit zones within one camera’s FoV.

**Vehicle trajectory classification by zone pairs.** After the zones are generated, all of the trajectories need to be classified according to entry-exit zone pairs. In our model, an entry-exit zone pair is used
to uniquely describe a trajectory, which is called a zone pair trajectory, as shown in Fig. 3(a). In this paper, we generate all possible camera links by using the training data of MTMCT to systematically generate the camera links instead of human labeling. In most cases, the straight and the right-turn trajectories can be described using different zone pairs being passed through. Sometimes the trajectory may not pass through the corresponding zone pair perfectly due to the viewing angle of the camera, i.e., also passes through the neighboring zones associated with the adjacently connected zone pair. In this case, measuring the distance (mismatch) between a tracked vehicle and a zone-pair trajectory is necessary. The distance can be calculated as
\[
dist(P, V) = \sum_{z \in P \cup V} |1(z \in P) - \alpha_z|, \tag{3}
\]
where \(P\) denotes the zone-pair trajectory and \(V\) is the actual zones gone through by the tracked vehicle, \(\alpha_z\) represents the overlapping 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 is also considered. Once the order in the tracked vehicle conflicts with the zone pair, the distance between a tracked vehicle and a zone-pair trajectory is set to infinity. Finally, the closest zone-pair trajectory with minimum distance is assigned by comparing the tracked vehicle with all the possible zone-pair trajectories within the camera. An example of distance calculation is shown in Fig. 3(b).

**Camera link model construction.** Given the locations of the cameras, we can obtain the routing information provided by the training data. The routing information contains all the links between every two adjacently connected cameras. If there is one link that connects two cameras without passing through another camera, we define them as a camera link. In other words, if all the routes from the training data between two cameras passed by at least one other camera, this link should not exist in our camera link model.

The camera link and the corresponding transition time of each camera pair can be defined as \(T = (C^s, C^d)\), where \(C^s = \{p_x^s\}_{i=1}^m\) is the zone pair trajectories set in the source camera and \(C^d = \{p_x^d\}_{j=1}^n\) is that set in the destination camera. Each camera pair can have more than one transition time due to the bi-directional traffic. In the camera pair with the overlapping view, \(T\) usually only contains one single zone pair trajectory, while for the non-overlapping view case, \(T\) can involve multiple zone pair trajectories. An example to explain the concept of our camera link model is shown in Fig. 4(a).

**Transition time estimation.** To estimate the transition time of each camera link \(T\), we first define the transition zones \(z_s\) and \(z_d\), such that \(z_s \in p_x^s (V p_x^s \in C^s)\) and \(z_d \in p_x^d (V p_x^d \in C^d)\). Then, the transition time can be applied as the temporal constraint for both \(C^s\) and \(C^d\). Given a camera link of a vehicle trajectory from \(p_x^s\) to \(p_x^d\), i.e., from source camera to destination camera, the transition time is defined as
\[
\Delta t = t^s - t^d, \tag{4}
\]
where \(t^s\) and \(t^d\) represent the time of the tracks passing \(z^s\) and \(z^d\), respectively. We can then obtain a time window \((\Delta_{\text{min}}, \Delta_{\text{max}})\) for each camera link \(T\) so that only the tracked vehicle pairs whose transition time are within the time window are considered as valid. Thus, the search space of the ReID can be greatly reduced by using the appropriate time window. Then, we can use the embedding feature from the ReID model, trajectory-based camera link model, and clustering algorithm to produce the global IDs for MTMCT.
3.3 Cross Camera ReID

After TSCT, the SCT results for each camera are available as the input data for MTMCT. Video-based ReID can achieve better performance than image-based ReID since it can take advantage of temporal continuity of a sequence of image frames instead of a single image. Therefore, an MTMCT system usually uses trajectory-level features for ReID. The feature extractor is based on the ResNet-50 [6], which is pre-trained on ImageNet, and the appearance feature of a vehicle is a 2048-dim vector output by the fully connected layer.

Once the frame-level features are extracted, we use temporal attention (TA) mechanism [5] to aggregate the frame-level features into the clip-level features. After that, we use the average pooling to generate trajectory-level features. In the TA modeling, there are two convolutional networks, one is a spatial convolutional network with 2D convolutions and the other is a temporal convolutional network with 1D convolution. Through training these two networks, an attention vector \( v_{\text{att}} \) can be obtained to weight the frame-level features \( f_{\text{frame}} \), resulting in the clip-level features, where

\[
f_{\text{clip}} = v_{\text{att}} \cdot f_{\text{frame}}. \tag{5}\]

In terms of network training, metric learning is adopted to enhance this vehicle ReID task. To train the model in a more efficient way, we adopt the batch sample (BS) scheme [13] in the triplet generation. Overall, in our final loss function, the BS triplet loss is combined with the cross-entropy loss to jointly exploit distance metric learning and identity classification,

\[
\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{BS}} + \lambda_2 \mathcal{L}_{\text{Xent}}. \tag{6}
\]

The objective of triplet loss is to minimize the feature distance of the same identity and maximize the feature distance of different identity pairs [8]. For the BS triplet loss, the objective is to calculate triplet loss \( \mathcal{L}_{\text{BS}}(\theta; \xi) \) in a minibatch defined as

\[
\mathcal{L}_{\text{BS}}(\theta; \xi) = \frac{1}{b} \sum_{a \in B} I_{\text{triplet}}(a), \tag{7}
\]

where

\[
I_{\text{triplet}}(a) = m + \sum_{p \in P(a)} w_p D_{ap} - \sum_{n \in N(a)} w_n D_{an}. \tag{8}
\]

Here, \( w_p \) and \( w_n \) are the weightings of positive and negative samples, \( m \) represents the margin, \( D_{ap} \) and \( D_{an} \) are the distances between the anchor sample to the positive sample and negative sample, respectively.

Based on the BS strategy, the weightings of positive and negative samples are defined as follows,

\[
w_p = \mathcal{P}(w_p \equiv \text{multinomial}_{x \in P(a)}(D_{ax})),
\]

\[
w_n = \mathcal{P}(w_n \equiv \text{multinomial}_{x \in N(a)}(D_{ax})), \tag{9}
\]

where \( x_p \) and \( x_n \) are positive and negative samples, respectively.
We perform data association between a pair of single camera trajectories corresponding to an adjacent exit/entry zone for multi-camera trajectories utilizing correlation clustering in each separate short time windows. The time window \((\Delta_{\text{min}}, \Delta_{\text{max}})\) is generated from CLM. In each window, 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. Therefore, MTMCT can be referred as a correlation clustering problem. Then the issue is to partition the node set into subsets. The single camera trajectories of the same identity should belong to the same subset. Edges in the same subsets accumulate high positive correlations, while edges in the different subsets accumulate high negative correlations. The problem can thus be defined as the following Binary Integer Program (BIP),

\[
X^* = \arg \max_{\mathbf{x}_{i,j}} \sum_{(i,j) \in \mathcal{E}} w_{i,j} x_{i,j},
\]

where \(\mathbf{x}_{i,j}\) is the assignment vector, \(w_{i,j}\) is the weight of edge \((i,j)\), and \(x_{i,j}\) is the binary variable indicating whether \((i,j)\) is an assigned edge. The constraints in Eq. 11 enforce the transitivity in the solution.

To reduce the computational complexity, we use hierarchical clustering in our method. Fig. 5 illustrates how to cluster the single camera trajectories into cross camera trajectories. Since the search space of ReID is reduced by the transition time constraint, the Rank-1 accuracy will be close to 1. Therefore, we can greedily select the smallest pair-wise distance to merge the tracked vehicles cross cameras. Furthermore, the order between different tracked vehicles can be used as a constraint to further reduce the search space of the ReID. Due to the traffic scenarios or the road conditions, the orders of vehicles should be almost the same. Take Fig. 4(b) as an example, we will remove the pairs whose orders conflict with those of previously matched pairs. The process will repeat until there is no valid transition pair, or the minimum distance is larger than a threshold.

### 3.4 Hierarchical Clustering

We perform data association between a pair of single camera trajectories corresponding to an adjacent exit/entry zone for multi-camera trajectories utilizing correlation clustering in each separate short time windows. The time window \((\Delta_{\text{min}}, \Delta_{\text{max}})\) is generated from CLM. In each window, 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. Therefore, MTMCT can be referred as a correlation clustering problem. Then the issue is to partition the node set into subsets. The single camera trajectories of the same identity should belong to the same subset. Edges in the same subsets accumulate high positive correlations, while edges in the different subsets accumulate high negative correlations. The problem can thus be defined as the following Binary Integer Program (BIP),

\[
X^* = \arg \max_{\mathbf{x}_{i,j}} \sum_{(i,j) \in \mathcal{E}} w_{i,j} x_{i,j},
\]

where \(\mathbf{x}_{i,j}\) is the assignment vector, \(w_{i,j}\) is the weight of edge \((i,j)\), and \(x_{i,j}\) is the binary variable indicating whether \((i,j)\) is an assigned edge. The constraints in Eq. 11 enforce the transitivity in the solution.

To reduce the computational complexity, we use hierarchical clustering in our method. Fig. 5 illustrates how to cluster the single camera trajectories into cross camera trajectories. Since the search space of ReID is reduced by the transition time constraint, the Rank-1 accuracy will be close to 1. Therefore, we can greedily select the smallest pair-wise distance to merge the tracked vehicles cross cameras. Furthermore, the order between different tracked vehicles can be used as a constraint to further reduce the search space of the ReID. Due to the traffic scenarios or the road conditions, the orders of vehicles should be almost the same. Take Fig. 4(b) as an example, we will remove the pairs whose orders conflict with those of previously matched pairs. The process will repeat until there is no valid transition pair, or the minimum distance is larger than a threshold.
highways. The length of the training videos is 58.43 minutes, while testing videos are 136.60 minutes in length. There are five scenarios in the CityFlow dataset, three of the scenarios are used for training, and the remaining two are used for testing. In total, the dataset contains 229,680 bounding boxes for 666 distinct annotated vehicle identities. The resolution is at least 960p and the frame rate is 10 FPS. Moreover, the license plates are blocked out in advance due to the privacy issues, i.e., the license plate information is not allowed to be used in the MTMCT.

In terms of implementation details, we use the dataset of the AI City Challenge 2018 [21] for training the TNT in the SCT. Since there are over 3.3K vehicles in the AI City Challenge 2018 dataset, which contains much richer information than the training set in the benchmark dataset. For both feature extraction of single camera ReID and cross camera ReID, we use ResNet-50 as the backbone network, trained with the combination of BS Triplet loss and Xent loss. In TNT, the dimension of the appearance feature is 512, the time window size is 64, and the batch size is 32. The optimizer used for training TrackletNet is Adam and the learning rate is from $10^{-3}$ to $10^{-5}$ for decreasing 10 times in every 2000 steps.

In terms of the single and cross camera ReID, the temporal attention model is trained with tracklet features from ground truth image ReID features. Same as UWIPL [11], we select the 4 as the clip length for each tracklet. The learning rate is $3 \times 10^{-4}$, the weight decay is $5 \times 10^{-4}$, the batch size is 32, and the network is totally trained for 800 epochs. The similarity estimation of the ReID features is calculated by the Euclidean distance. The input images are resized to $224 \times 224$. We use ResNet-50 pre-trained on ImageNet as the backbone for our model.

For MTMCT, we use IDF1, IDP, and IDR as evaluation metrics. IDF1 [24] is used to rank the performance of each team in the CityFlow dataset. It calculates the ratio of correctly identified detections over the average number of ground truth and computed detections. More specifically, false negative ID (IDFN), true negative ID (IDTN) and true positive ID (IDTP) counts are all used to compute the identification precision (IDP), the identification recall (IDR), and the corresponding F1 score IDF1.

$$IDP = \frac{IDTP}{IDTP + IDFP},$$
$$IDR = \frac{IDTP}{IDTP + IDFP},$$
$$IDF1 = \frac{2IDTP}{2IDTP + IDFP + IDFN}. \tag{12}$$

The definition of IDFN, IDTN and IDTP are as following,

$$IDFN = \sum_{\tau} \sum_{t \in T_{\tau}} \sum_{\gamma \in T_{\gamma}} m(\tau, y_{m(\tau)}, t, \Delta),$$
$$IDFP = \sum_{\tau} \sum_{t \in T_{\tau}} m(\tau, y, t, \Delta),$$
$$IDTP = \sum_{\tau} \sum_{\gamma \in T_{\gamma}} \text{len}(\tau) - IDFN = \sum_{\gamma} \text{len}(\gamma) - IDFP. \tag{13}$$

where $\tau$ is the ground truth trajectory, $y_{m(\tau)}$ means the computed trajectory that best matches $\tau$; $y$ represents computed trajectory; $\text{len}(\tau)$ is ground truth trajectory that best matches $\gamma$; $t$ is the frame index; $\Delta$ means the IOU threshold that judges whether computed bounding box matches the ground truth bounding box (here we set $\Delta = 0.5$); $m(\cdot)$ is a mismatch function which is equal to 1 if there is a mismatch at $t$; otherwise, $m(\cdot)$ is 0.

### 4.2 MTMCT Results on CityFlow

In the CityFlow dataset, the spatio-temporal information is useful in improving the performance [7, 11, 16]. Especially, the camera links are applied to achieve the best performance [11]. Moreover, some systems utilize data association graph for MTMCT [9]. Most of the existing MTMCT methods are based on the tracking-by-detection scheme and adapted for multi-camera views. However, the proposed method also considers traffic scenarios as well as the camera links to enhance the MTMCT performance.

The qualitative results are shown in Fig. 6, which shows that our method is generalized well for different cameras and vehicles. Table 1 compares our methods with the state-of-the-art approaches on the CityFlow benchmark. Here, locality aware appearance metric (LAAM) [10] is the state-of-the-art approach on DukeMTMC dataset. LAAM improves the performance of DeepCC by training the model based on both intra-camera and inter-camera metrics. The ReID features in DeepCC are extracted by DenseNet-121 with softmax and triplet loss. In terms of LAAM, they use ResNet-50 pre-trained on ImageNet. The tracklet length is set to 10 frames, and the temporal window sizes for SCT and ICT are set to 500 frames and 2400 frames. According to the experimental results, our proposed method outperforms all the state-of-the-art methods, by incorporating the traffic-aware zones for isolated trajectories ReID and also taking advantage of the generated camera link model. Finally, we achieve IDF1 of 74.93% for MTMCT on CityFlow. Moreover, comparing to [11], our system can automatically generate camera links instead of human labeling.

### 4.3 Ablation Studies

The ablation studies of MTMCT is showed in Table 2, including the combinations of TNT, TSCT, TA, and CLM. The experimental results show that each proposed component helps enhance the robustness. First, we show all modules of the proposed method are necessary. When replacing TSCT with the TNT, IDF1 based on the TA feature drops by 4.3% on MTMCT. A similar but greater accuracy drop can be observed when the TA is replaced by the average pooling. The drop is consistent when using the same tracklet based

| Methods                | IDF1  |
|------------------------|-------|
| MOANA+BA [34]          | 0.3950|
| DeepSORT+BS [34]       | 0.4140|
| TC+BA [34]             | 0.4630|
| ZeroOne [28]           | 0.5987|
| DeepCC [25]            | 0.5660|
| LAAM [10]              | 0.6300|
| ANU [9]                | 0.6519|
| TrafficBrain [7]       | 0.6653|
| DDashcam [16]          | 0.6865|
| UWIPL[11]              | 0.7059|
| **Ours**               | **0.7493**|
ReID features. These results show that both the TSCT and TA are necessary components in our system. Second, from the ablation studies, the removal of the CLM causes a greater accuracy drop. The reason is the transition time constraint between cameras improves data association in tracking, so the global matching in ReID is much better in ICT. In the vehicle ReID, high inter-class similarity leads to small appearance variance of the targets. Consequently, after applying the CLM, the MTMCT performance can be largely improved. Then, we show that the TA and CLM are both important, IDF1 can be improved from 15.83% to over 50% by using TA or CLM. Therefore, CLM can improve MTMCT by the general ReID model.

In Table 3, we show the performance of SCT, which favorably compares with the state-of-the-art methods for SCT [32, 35, 40]. DeepSORT [40] is an online method incorporating Kalman-filter-based tracking and the Hungarian algorithm. TC [35] is an offline method using tracklet clustering which is the winner of the AI City Challenge at CVPR 2018 [21]. MOANA [32] is a state-of-the-art approach on the MOTChallenge 2015 3D benchmark. There are three sets of available detection results, i.e., SSDs [18], YOLOv3 [22] and Faster R-CNN [23], which are provided by the CityFlow dataset. According to [34], SSDs [18] performs the best and is used by us for comparison. The metrics of SCT include IDF1, Multiple Object Tracking Accuracy (MOTA), Multiple Object Tracking Precision (MOTP), Recall and the mostly tracked targets (MT). According to the experimental results, the proposed TSCT method achieves the best performance.

5 CONCLUSION

In this paper, we propose a novel approach for multi-target multi-camera tracking (MTMCT) of vehicles, which includes traffic-aware single camera tracking (TSCT), trajectory-based camera link model (CLM), and vehicle re-identification (ReID). Finally, hierarchical clustering is utilized to merge the vehicle trajectories from different cameras and generates the MTMCT results. From our experiments, the proposed method is shown to be effective and robust. It also achieves a new state-of-the-art performance on the CityFlow dataset.

ACKNOWLEDGEMENT

This work was partially supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government. [20ZD1100, Development of ICT Convergence Technology for Daegu-GyeongBuk Regional Industry]
REFERENCES

[1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. 2006. Surf: Speeded up robust features. In European conference on computer vision. Springer, 404–417.

[2] Jiarui Cai, Yizhou Wang, Haojie Zhang, Heng-Min Huo, Chengqian Ma, and Jeng-Neng Hwang. 2020. IA-MOT: Instance-Aware Multi-Object Tracking with Motion Consistency. BMIT Challenge Workshop, IEEE Conference on Computer Vision and Pattern Recognition (2020).

[3] Wongoon Choi. 2015. Near-online multi-target tracking with aggregated local flow descriptor. In Proceedings of the IEEE international conference on computer vision. 3029–3037.

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

[5] Jiyang Gao and Ram Nevatia. 2018. Revisiting Temporal Modeling for Video-based Person ReID. arXiv preprint arXiv:1805.02104 (2018).

[6] Peilun Li, Guozhen Li, Zhangxi Yan, Youzeng Li, Meiqi Lu, Pengfei Xu, Yang Gu, Yihang Lou, Yan Bai, Jun Liu, Shiqi Wang, and Lingyu Duan. 2019. Veri-wild: A large dataset and a new method for vehicle re-identification based on visual-spatio-temporal features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 275–284.

[7] Jiarui Cai, Yizhou Wang, Haotian Zhang, Hung-Min Hsu, Chengqian Ma, and Jenq-Neng Hwang. 2019. Multi-Camera Vehicle Tracking System. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 167–174.

[8] Yunzhong Hou, Heming Du, and Liang Zheng. 2019. A Locality Aware City-Scale Multi-Camera Vehicle Tracking System. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 167–174.

[9] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. 2017. Re-ranking person appearance model for robust multiple object tracking in 3D. IEEE Access 7 (2019), 31934–31945.

[10] Hongye Liu, Yonghong Tian, Yaowei Yang, Lu Pang, and Tiejun Huang. 2016. Multi-Camera Vehicle Tracking with powerful visual features and spatial-temporal cue. In Proc. CVPR Workshops. 203–212.

[11] Yunzhong Hou, Liang Zheng, and Shaozi Li. 2019. A Locality Aware Appearance Metric for Multi-Target Multi-Camera Tracking. arXiv preprint arXiv:1911.12037 (2019).

[12] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. 2018. Features for Multi-Target Multi-Camera Vehicle Tracking. arXiv preprint arXiv:1703.07737 (2017).

[13] Zheng Tang, Gaoang Wang, Haojie Zhang, Aotian Zheng, and Jeng-Neng Hwang. 2019. 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. 8797–8806.

[14] Yi Zhou and Ling Shao. 2018. Aware attentive multi-view inference for vehicle re-identification with k-reciprocal encoding. In Proceedings of the 27th ACM International Conference on Multimedia. 1318–1327.

[15] Haotian Zhang, Yizhou Wang, Jiarui Cai, Hung-Min Hsu, Haorui Ji, and Jenq-Neng Hwang. 2017. 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. 108–115.

[16] Dorin Comaniciu and Peter Meer. 2002. Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 5 (2002), 603–619.