Multi-Camera Tracking of Vehicles based on Deep Features Re-ID and Trajectory-Based Camera Link Models

Hung-Min Hsu[1,2], Tsung-Wei Huang[1], Gaoang Wang[1], Jiarui Cai[1],
Zhichao Lei[1], and Jenq-Neng Hwang[1]

[1]Department of Electrical and Computer Engineering, University of Washington
[2] Research Center for Information Technology Innovation, Academia Sinica, Taiwan
{hmhsu, twhuang, gaoang, jrcai, zl68, hwang}@uw.edu

Abstract

Due to the exponential growth of traffic camera networks, the need for multi-camera tracking (MCT) for intelligent transportation has received more and more attention. The challenges of MCT include similar vehicle models, significant feature variation in different orientations, color variation of the same car due to lighting conditions, small object sizes and frequent occlusion, as well as the varied resolutions of videos. In this work, we propose an MCT system, which combines single-camera tracking (SCT) and inter-camera tracking (ICT) which includes trajectory-based camera link model and deep feature re-identification.For SCT, we use a TrackletNet Tracker (TNT), which effectively generates the moving trajectories of all detected vehicles by exploiting temporal and appearance information of multiple tracklets that are created by associating bounding boxes of detected vehicles. The tracklets are generated based on CNN feature matching and intersection-over-union (IOU) in every single-camera view. In terms of deep feature re-identification, we exploit the temporal attention model to extract the most discriminant feature of each trajectory. In addition, we propose the trajectory-based camera link models with order constraint to efficiently leverage the spatial and temporal information for ICT. The proposed method is evaluated on CVPR AI City Challenge2019 City Flow dataset, achieving IDF1 70.59%, which outperforms competing methods.

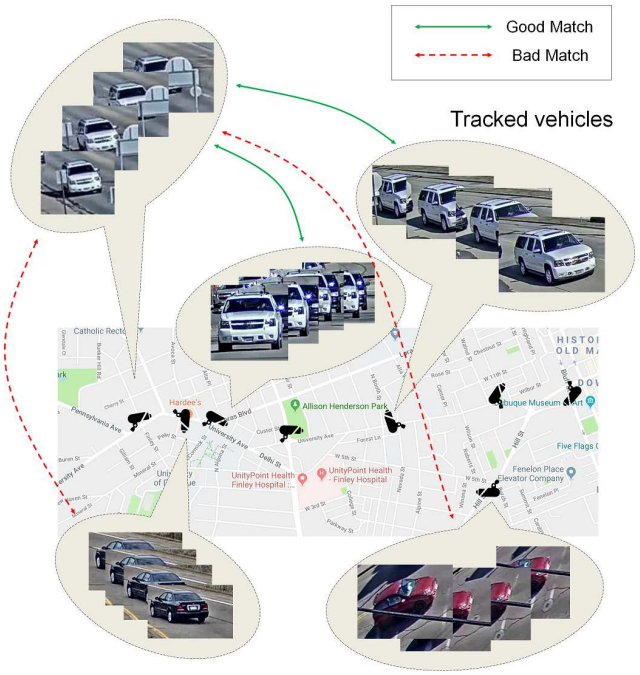

Figure 1. Illustration for vehicle MCT task. From a vehicle MCT dataset (this figure shows the map from [27] dataset), we have tracked vehicles from different cameras. Given a tracked vehicle in a camera, MCT task is aimed to search from other cameras with the same vehicle.

1. Introduction

For traffic flow prediction and analysis purpose, the demands of multi-camera tracking (MCT), which tracks multiple detected objects across multiple cameras of overlapping/non-overlapping views, rapidly increase in recent years. General speaking, MCT includes three parts, i.e., single-camera tracking (SCT), appearance feature re-identification (Re-ID) and the trajectory-based camera link model for spatial and temporal constraints. The goal of MCT is to generate tracks in every single camera and then associate the tracks that belong to the same vehicle in different cameras. However, the task is very challenging due

to several reasons. First, tracking vehicles in a long-time range is difficult because of heavy occlusion, the different appearance from different orientations of the same vehicle, similar appearances from different vehicles and varying lighting conditions, not to mention the frequent noisy object detections and occlusions. The tracks can be easily lost or switched in the SCT. Similarly, Re-ID is also a difficult task because the appearance features of vehicles may change dramatically on account of varied illuminations and viewing angles in two different cameras. Moreover, the trajectory-based camera link models, which are the major parts of inter-camera tracking (ICT), is also very critical in the cross-camera matching.

To deal with the problems in SCT, a more efficient and reliable descriptor of appearance features is critically needed. With the significant advances of object detections, many SCT methods follow the tracking-by-detection framework [6, 32], which has been proven effective in many works of human and vehicle tracking. By taking advantage of well-embedded appearance and temporal relationship, the tracking-by-detection framework can not only use the similarity measure of features between objects but also use the locations of corresponding objects to determine if they are the same object.

To associate the vehicular tracks in different cameras for ICT, appearance feature based vehicle Re-ID is one of the most effective approaches. In terms of vehicle re-id, some works [31, 14, 34] focus on generating discriminant visual features by deep convolutional neural networks (DCNNs). Besides, trajectory-based camera links and transition time among neighboring cameras [13, 28, 27] are also important cues in ICT. Based on these spatial and temporal constraints, the searching and matching space can be greatly reduced.

In this paper, we propose an innovative framework for MCT system for vehicles. The flowchart of our proposed MCT system is shown in Figure 2. First, we use a TrackletNet Tracker (TNT) [29] in the single camera tracking (SCT). 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 tracklets, we estimate their similarity by a Siamese TrackletNet based on both appearance and temporal information. A graph model is 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. After SCT, a temporal attention model is adopted [5] to extract embedded features for each trajectory. Cross-entropy loss and triplet loss are jointly used in training. Finally, based on the feature similarity and the built trajectory-based camera link model, we can generate global IDs in MCT. To summarize, we claim the following contributions,

- An effective TNT tracker is used for the SCT task.
- A temporal attention model is exploited to extract the feature of each trajectory.
- Trajectory-based camera link models are constructed using spatial and temporal information.

The rest of this paper is organized as follows. We provide an overview of related works in Section 2 and our proposed MCT system is introduced in Section 3. The experiments and evaluations of our method on the CVPR AI City Challenge 2019 City Flow dataset [27] are shown in Section 4. Finally, the conclusion is drawn in Section 5.

2. Related Works

Single-Camera Tracking (SCT). Most of the recent multi-object tracking (MOT) approaches are based on tracking-by-detection schemes [6, 32], i.e., given detection results, we would like to associate detections across frames and estimate object locations when unreliable detections or occlusions occur. Many tracking methods are based on graph models [26, 15, 28, 25, 10, 11, 3, 24, 30] and solve the tracking problem by minimizing the total cost. In [26, 15, 25, 11], the detected objects are treated as the vertices in the graph models, while in [28, 3, 24, 30], the graph vertices are based on tracklets. For detection-based graph models, there are two major disadvantages. First, one of the critical assumptions in graph models is the conditional independence of the vertices. However, detections are not conditionally independent from frame to frame; therefore if we want to track an object in the long run, the temporal information can be more effectively utilized. Second, the detection-based graph usually comes with a very high-dimensional affinity matrix, which makes it very hard to find the global minimum solution in the optimization. However, a tracklet-based graph model can better utilize the information from a short trajectory to measure the relationship between vertices, if the mis-association can be carefully handled in the tracklet generation step.

Features are essential in the tracking-by-detection framework. There are two types of features that are used in common, *i.e.*, appearance features and temporal features. For appearance features, many works adopt CNN-based features for Re-ID tasks [18, 33, 26, 35]. For example, [18] proposes an adaptive weighted triplet loss for training and a new technique for hard-identity mining. [33] adopts a re-ranking technique [35] in calculating the feature similarity. However, histogram-based features, like color histograms, HOG, and LBP, are still powerful if no labeled training data are provided [28]. As for temporal features, the location, size, and motion of bounding boxes are commonly used. Given the appearance features and temporal

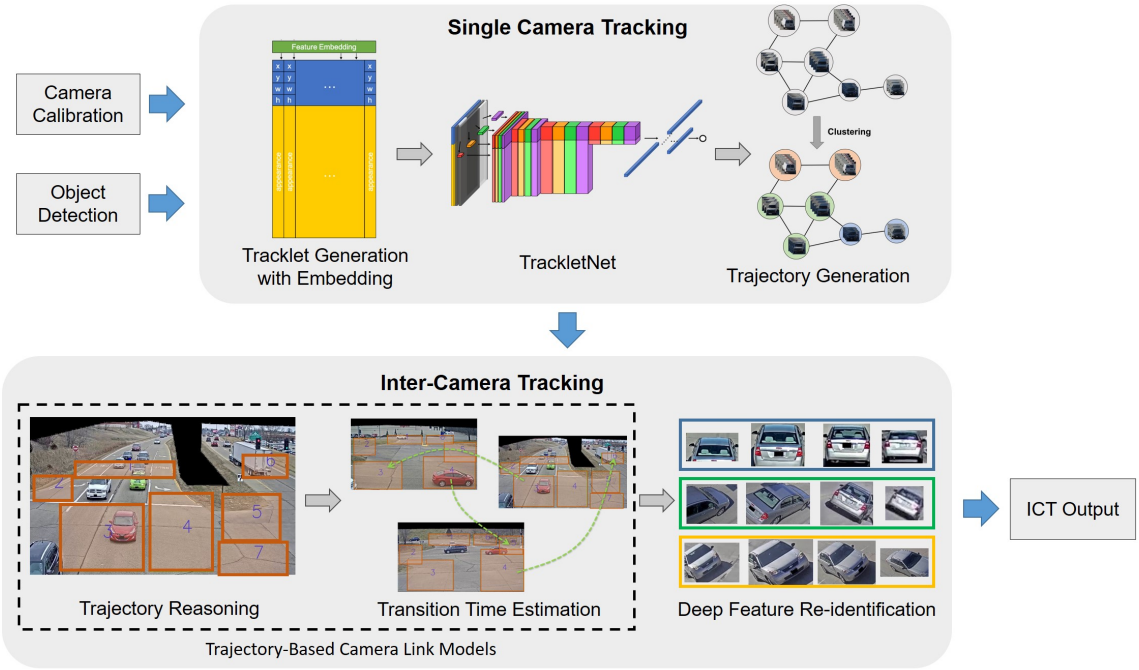

Figure 2. The flowchart of MCT in Track 1.

features, the tracker can fuse them together to achieve better performance [33, 15, 28]. However, it is still empirical and difficult to determine the weighting of each feature.

Appearance Feature based Re-ID. The feature extraction methods generally fall into two primary categories, one is the traditional keypoint-descriptor methods, like SURF [2], ORB [19], and the other are the deep learning based feature extractors, like CNN [31, 14, 34]. Comparing with handcrafted features, CNN feature extractors usually perform better because they can extract the features robustly by supervised learning which jointly extracts the discriminant feature and estimate the classification/regression models. However, most of the functional CNN feature extractors are trained on different types of objects which may have prominent discriminating features. However, for this task, we need to distinguish the differences within one single class – vehicle. To deal with this problem, we need to retrain the CNN model, and more importantly, create more discriminant features for different types of vehicles.

Camera Link Models. To reduce the searching and matching space in the ICT, some works [13, 28, 27] also consider spatial-temporal constraints with camera link models. For example, in [13], bidirectional transition time distribution is exploited with the camera link models in the process of estimation with an unsupervised scheme. In [27, 28], the transition time distribution is built for each connected pair of cameras by using the estimated vehicle speed. With a reliable camera link model, the candidate set for matching becomes much smaller. As a result, the accuracy of across camera association can be significantly improved.

3. Proposed Method

3.1. Single-Camera Tracking (SCT)

We adopt the TrackletNet Tracker (TNT) [29] for SCT in the Track-1 challenge. The tracking system is based on a tracklet graph-based model, as shown in Figure 3, which has three key components, 1) tracklet generation, 2) connectivity measurement, and 3) graph-based clustering. Given the detection results in each frame, the tracklets are generated based on the intersection-over-union (IOU) compensated by the epipolar geometry constraint due to camera motion and the appearance similarity between two consecutive frames. Each generated tracklet is treated as one node in the graph. Between every two tracklets, the connectivity is measured as the edge weight in the graph model, where the connectivity represents the likelihood of the two tracklets being from the same object. To calculate the connectivity, a multi-scale TrackletNet is built as a classifier, which can combine both temporal and spatial features in the likelihood estimation. Clustering [28] is then conducted to minimize the total cost

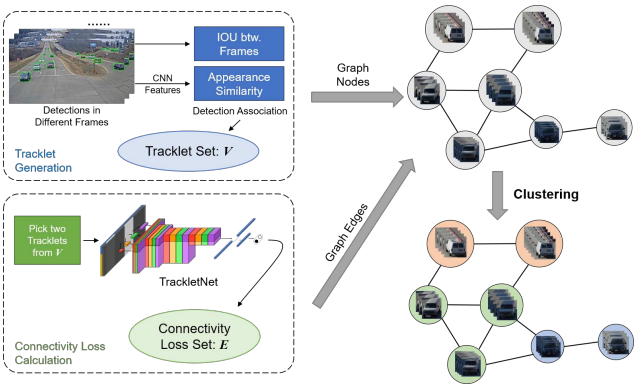

Figure 3. The TNT framework for the Single-Camera tracking. Given the detections in different frames, detection association is computed to generate Tracklets for the Vertex Set V. After that, every two tracklets are put into the TrackletNet to measure the connectivity, which forms the similarity on the Edge Set E. A graph model G can be derived from V and E. Finally, the tracklets with the same ID are grouped into one cluster using the graph partition approach.

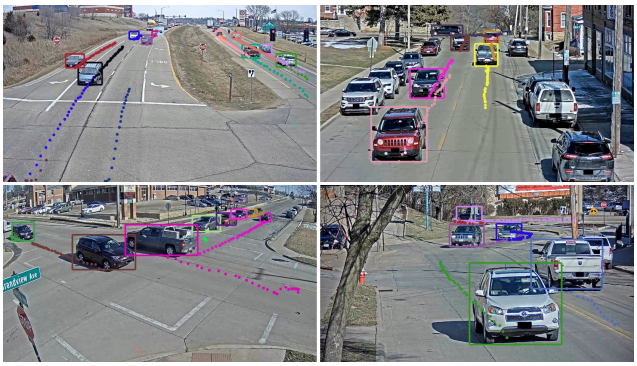

Figure 4. Examples of single-camera tracking (SCT) in different scenarios. Each color represents a unique ID of tracked vehicle. The parked vehicles are excluded from the tracking.

on the graph. After clustering, the tracklets from the same ID can be merged into one group.

The reason we use TNT as our tracking method is due to its robustness in dealing with occlusions and false detections. More specifically, 1) a TrackletNet focuses on the continuity of the embedded features along the time. In other words, the convolution kernels only capture the dependency along time. 2) The network integrates object ReID, temporal ,and spatial dependency as one unified framework. Based on the tracking results from TNT, the continuous trajectory of each object ID across frames can thus be obtained. Some examples of SCT in different scenarios are shown in Figure 4.

3.2. Deep Feature Re-ID

Frame-Level Feature Extraction. To reduce noise, all the images are fed into a Mask-RCNN [7] to remove the background. The unmasked area is zero-filled if the detected object is classified as a vehicle and its confidence score is above a certain threshold. The pre-processed frame features are then extracted from a ResNet50 [8] network that pre-trained on ImageNet. The 2048-dim fully-connected layer before the classification layer is used to represent the appearance of the vehicle.

Temporal Attention Model. After we extract the frame-level features, we combine them into clip-level features using temporal attention modeling (TA) [5]. The structure of the temporal attention modeling is shown in Figure 5. The spatial convolutional network is a 2D convolution operation and the temporal convolutional network is a 1D convolution operation. We train these two networks to get more reliable attention scores for the frames in video clips. After the weighted average, we can get the clip-level features f_c.

Loss Function. Triplet loss is firstly proposed by the FaceNet [4] to address face verification problem, which is similar to a vehicle ReID task. In an end-to-end metric learning paradigm, the input image is projected to an embedding vector space, then the distance of the embedded features are directly compared and optimized. Given an anchor feature a, the projection of a positive feature x_p belongs to the same vehicle y_a is closer to the anchor's projection than that of a negative feature x_n belonging to another class y_b, by at least a margin m. To train the model more efficiently, we adopt batch sample (BS) [12] instead of batch hard (BH) [4] in the triplet generation.

The objective of triplet loss is to maximize the distance between features of different identity pairs while minimize that of the same identity [9]. The BS triplet loss in a mini-batch $\mathcal{X}$ is defined as,

$$\mathcal{L}_{BStri}(\theta; \mathcal{X}) = \sum_{\text{all batches}} \sum_{a \in B} l_{triplet}(a), \tag{1}$$

where

$$l_{triplet}(a) = [m + \sum_{p \in P(a)} w_p D_{ap} - \sum_{n \in N(a)} w_n D_{an}]_+, \tag{2}$$

with w_p and w_n are the weighting of positive and negative samples, respectively, D_{ap} and D_{an} are the distances between the anchor sample to the positive samples and negative sample, respectively, and m is the defined margin.

According to BS strategy, the weighting of positive and negative samples are defined as follows,

$$\begin{aligned} w_p &= P(x_p == \text{multinomial}_{x \in P(a)}\{D_{ax}\}), \\ w_n &= P(x_n == \text{multinomial}_{x \in N(a)}\{D_{ax}\}), \end{aligned} \tag{3}$$

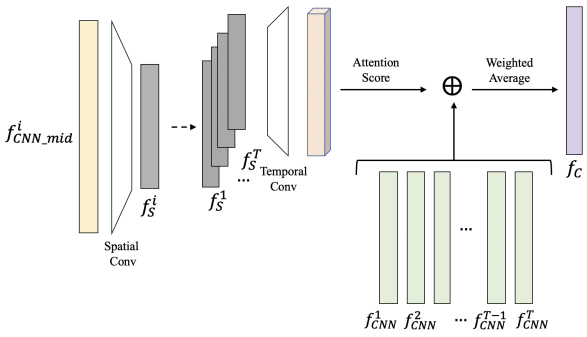

Figure 5. The structure of temporal attention model [5]. The frame-level features are passed through the spatial and temporal convolutional networks to obtain the attention score for each frame, and then calculate the weighted average using the attention scores to get clip-level features.

where x_p and x_n are positive and negative samples, respectively.

In addition to BS triplet loss, we also include cross-entropy ($Xent$) loss [22] in the training as follows,

$$\mathcal{L}_{Xent} = -\sum_{i=1}^{P} \log\left(\text{prob}(i)\right) q(i), \tag{4}$$

where $q(i)$ is the one-hot ground truth label, $\text{prob}(i)$ is the probability of the probe vehicle belongs to vehicle i.

The overall loss function is a weighted combination of BS triplet loss and cross-entropy loss,

$$\mathcal{L}_{Total} = \lambda_1 \mathcal{L}_{BStri} + \lambda_2 \mathcal{L}_{Xent}. \tag{5}$$

3.3. Trajectory-Based Camera Link Models

Because the movement of a vehicle on the road usually follows certain driving patterns based on road structures and traffic rules, we can group them into limited numbers of trajectories. By exploiting the spatial-temporal relationships between the trajectories in different cameras, we proposed the trajectory-based camera link models for multi-camera tracking of vehicles.

Distinguishing Trajectories. To efficiently distinguish different trajectories within a camera, we define several zones on the image (Figure 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, and our goal is to use a zone list to describe a trajectory uniquely. For example, in Figure 6, the straight and the right-turn trajectories can be described using different zone lists they go through.

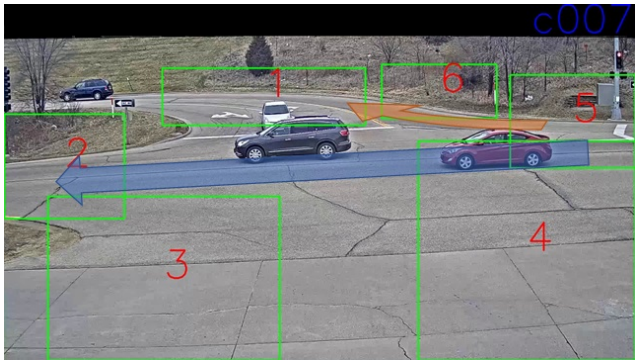

Figure 6. Examples of using zone lists to describe trajectories. The straight trajectory (blue) can be described by zone list [5, 2], and the right-turn trajectory (orange) can be described by zone list [5, 6, 1].

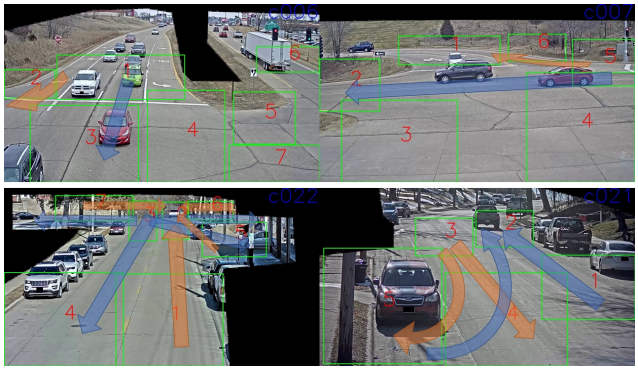

Figure 7. Examples of transitions for camera pairs with overlapping views (top) and non-overlapping views (bottom). The colors of the trajectories correspond to different possible transitions between cameras. For the top camera pair, from source camera (left) to destination camera (right), transition blue is $(\{[1,3]\},\{[5,2]\})$, and transition orange is $(\{[1,2]\},\{[5,6,1]\})$. For the bottom camera pair, from source camera (left) to destination camera (right), transition blue is $(\{[3,4],[3,7],[3,5],[3,2,6]\},\{[1,2],[5,4,2]\})$, and transition orange is $(\{[1,2],[5,2],[6,2],[7,3,2]\},\{[3,4],[3,4,5]\})$.

However, due to the viewing angle of the camera, the bounding boxes of a tracked vehicle of a certain trajectory may not go through the corresponding zone list perfectly without touching other zones. Therefore, we defined the distance between a tracked vehicle and a trajectory as,

$$dist(tr, \hat{tr}) = \sum_{z \in tr \cup \hat{tr}} |\mathbf{1}(z \in tr) - a_z|, \tag{6}$$

where tr is the zone list of the trajectory, $\hat{tr}$ is the actual zones gone through by the tracked vehicle and a_z is the overlapping ratio of the vehicle to zone z, *i.e.*, the overlapping area divided by the vehicle bounding box area. Besides, the orders of the zones in the zone list and the tracked

vehicle are also considered. If the order in the tracked vehicle conflicts with the zone list, the distance is set to infinity. Finally, we can assign the tracked vehicle to the closest trajectory within the camera. An example of computing the distance is shown in Figure 8.

Transition Between Cameras. To link two cameras together, we define the transition between the two cameras as $L = (T_{src}, T_{dst})$, where $T_{src} = \{tr_{src_1}, tr_{src_2}, ..., tr_{src_m}\}$ is the trajectories in the source camera and $T_{dst} = \{tr_{dst_1}, tr_{dst_2}, ..., tr_{dst_n}\}$ is the trajectories in the destination camera. Usually, a camera pair can have more than one transition due to the bidirectional traffic. Examples of transition are shown in Figure 7. Note that for the camera pair with overlapping view, T_{src} and T_{dst} usually consists of a single trajectory, and for the camera pair with non-overlapping view, T_{src} and T_{dst} can consist of multiple trajectories.

To apply the temporal constraint on the transition, for both T_{src} and T_{dst}, we first define the transition zones z_{src} and z_{dst} such that $z_{src} \in tr_{src_i} \forall tr_{src_i} \in T_{src}$ and $z_{dst} \in tr_{dst_i} \forall tr_{dst_i} \in T_{dst}$. Then, given a pair of tracked vehicles, tr_{src} and tr_{dst}, in source camera and destination camera, we can define the transition time as,

$$\Delta t = t_{src} - t_{dst}, \tag{7}$$

where t_{src} and t_{dst} are the times the vehicles passing z_{src} and z_{dst} respectively. For each transition L, we define a time window $(\Delta t_{min}, \Delta t_{max})$ so that only the tracked vehicle pair whose transition time is inside the window are considered as valid. With appropriate time window, the search space of the re-identification can be greatly reduced.

Note that the source and destination defined here is relative, and can be different from the vehicle's driving starting point and destination. Therefore, we can have $\Delta t < 0$, depending on the definition of source and destination cameras.

Ordered Transition. To further reduce the search space of the re-identification, we consider the relationship between different tracked vehicles. For some of the roads, the order of a series of vehicles does not change much due to the traffic rules or the road condition. Therefore, we define an ordered transition which has the following constraint: given two tracked vehicles tr_{src_1} and tr_{src_2} in source camera and the corresponding vehicles tr_{dst_1} and tr_{dst_2} in destination camera,

$$sign(t_{src_1} - t_{src_2}) = sign(t_{dst_1} - t_{dst_2}), \tag{8}$$

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

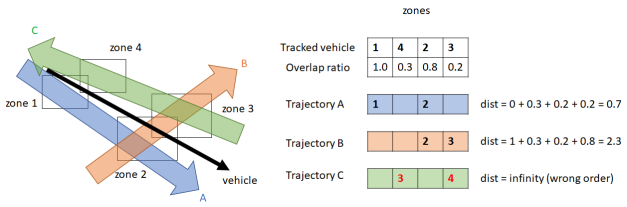

Figure 8. Example of distances between tracked vehicle (black) and trajectories A, B and C (blue, orange, green), given 4 zones in a camera. Although the tracked vehicle touches zones 4 and 3, but its distance to trajectory A is still much smaller than that to trajectory B. Besides, its distance to trajectory C is infinity because of the different zone order.

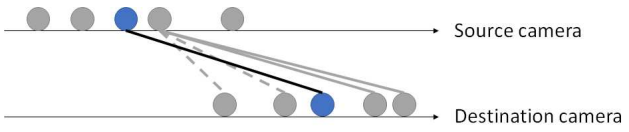

Figure 9. Example of ordered transition. When performing re-identification of a sequence of vehicles passing through the source and destination cameras, after the vehicle pair with smaller distance (blue) is matched in greedy algorithm, the search space of its neighbors (gray) is roughly reduced to half due to the order constraint.

Optimization. To apply the trajectory-based camera link model on the MCT, we use the greedy algorithm because it can be seen that after applying the transition time constraint, the search space of re-identification becomes minimal, and the rank-1 accuracy will be close to 1 for those high-confidence matches. First, we calculate the pairwise distance of the features of all the pair of tracked vehicles whose transition time is valid. Then, we greedily select the smallest pair-wise distance to match the tracked vehicles. For each ordered transition, we further remove the pairs which conflict with previously matched pairs (Figure 9). We repeat the process until there is no valid transition pair or the minimum distance is larger than a threshold.

4. Experiments and Results

Datasets. As given in [1], the benchmark dataset contains 3.25 hours (195.03 minutes) of videos collected from 40 cameras spanning 10 intersections in a mid-sized U.S. city, in which 58.43 minutes of videos are for training and the other 136.60 minutes are for testing. The resolution of each video is at least 960p and the majority of the videos have a frame rate of 10 FPS. It covers a diverse set of location types, including intersections, stretches of roadways, and highways. In total, the dataset contains 229,680 bounding boxes for 666 distinct annotated vehicle identities that pass through at least 2 cameras.

Figure 10. MCT result of cameras with highly overlapping views. Top) car id: 253, bottom) car id: 73.

Figure 11. MCT result of cameras with non-overlapping and partially overlapping views. Car id: 336.

Implementation Details. For training the TNT in SCT, we use the dataset of AI City Challenge 2018 [16], since with over 3.3k vehicles, which contains much richer information than the training set in the benchmark dataset. To extract deep embedded features for Re-ID, we use ResNet50 as the backbone network, training with the combination of Htri loss and Xent loss. For the camera transition time window, we set the lower bound and upper bound based on the road channelization information and specific road section situation [27, 20, 21, 23].

Evaluation and Results. The IDF1 score [17] is used to rank the performance of each team. IDF1 measures the ratio of correctly identified detections over the average number of ground-truth and computed detections. The final ranking on the testing set is shown in Table 1. We outperforms other

Rank	Team ID	Team Name	IDF1 Score
1	**21**	**UWIPL**	**0.7059**
2	49	DDashcam	0.6865
3	12	Traffic Brain	0.6653
4	53	Desire	0.6644
5	97	ANU	0.6519
6	59	Zero_One	0.5987
7	36	DGRC	0.4924
8	107	IIAI-VOS	0.4504
9	104	Owlish	0.3369
10	52	CUNY-NPU	0.2850

Table 1. The IDF1 score on Track 1. Our team is shown in bold type.

IDF1	IDP	IDR	Precision	Recall
0.7059	0.6912	0.7211	0.7470	0.7793

Table 2. The evaluation results of our proposed method.

teams in terms of IDF1 score of 0.7059, which shows the effectiveness of our proposed method. Besides, precision and recall performance is shown in Table 2. Qualitative results are shown in Figure 10 (highly overlapping view) and 11 (non-overlapping and partially overlapping views). It shows that our method is generalized for different scenarios.

5. Conclusion

In this paper, we propose a novel approach for multi-camera tracking (MCT), which includes single-camera tracking (SCT), deep appearance feature re-identification (Re-ID) and also spatial-temporal constraints for the trajectory-based camera link models. From our experiments, the proposed method is efficient, effective ,and robust, with achieved IDF1 70.59%, which outperforms other competing methods in the challenge.

Acknowledgement The authors would like to thank many people who helped in the improvement of the performance of the proposed system: Kelvin Lin, Charles Tung Fang, Yizhou Wang, Chengqian Ma, Shawn Hsiao, Nansong Yi, Yao-Chung Liang, Shih-Hao Yeh, Xinyu Zhao, Huihao Chen and Zexin Li.

References

[1] Ai city challenge 2019 official website. `https://www.aicitychallenge.org`. Accessed: 2019-02-08.

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

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

[4] Schroff Florian, Kalenichenko Dmitry, and Philbin James. Facenet: A unified embedding for face recognition and clustering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[5] Jiyang Gao and Ram Nevatia. Revisiting temporal modeling for video-based person reid. *arXiv preprint arXiv:1805.02104*, 2018.

[6] Andreas Geiger, Martin Lauer, Christian Wojek, Christoph Stiller, and Raquel Urtasun. 3d traffic scene understanding from movable platforms. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):1012–1025, 2014.

[7] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[8] 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.

[9] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

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

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

[12] Ratnesh Kumar, Edwin Weill, Farzin Aghdasi, and Parthsarathy Sriram. Vehicle re-identification: an efficient baseline using triplet embedding. *arXiv preprint arXiv:1901.01015*, 2019.

[13] Young-Gun Lee, Jenq-Neng Hwang, and Zhijun Fang. Combined estimation of camera link models for human tracking across nonoverlapping cameras. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2254–2258. IEEE, 2015.

[14] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International Conference on Artificial Neural Networks*, pages 52–59. Springer, 2011.

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

[16] Milind Naphade, Ming-Ching Chang, Anuj Sharma, David C Anastasiu, Vamsi Jagarlamudi, Pranamesh Chakraborty, Tingting Huang, Shuo Wang, Ming-Yu Liu, Rama Chellappa, et al. The 2018 nvidia ai city challenge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 53–60, 2018.

[17] 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.

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

[19] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *International Conference on Computer Vision (ICCV)*, pages 2564–2571. IEEE, 2011.

[20] David Schrank, Bill Eisele, Tim Lomax, and Jim Bak. 2015 urban mobility scorecard. 2015.

[21] David Schrank, Tim Lomax, and Bill Eisele. 2012 urban mobility report. *Texas Transportation Institute*, pages 1–57, 2011.

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

[23] Jinjun Tang, Shen Zhang, Wenhui Zhang, Fang Liu, Weibin Zhang, and Yinhai Wang. Statistical properties of urban mobility from location-based travel networks. *Physica A: Statistical Mechanics and its Applications*, 461:694–707, 2016.

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

[25] Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Multi-person tracking by multicut and deep matching. In *European Conference on Computer Vision*, pages 100–111. Springer, 2016.

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

[27] 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. *arXiv preprint arXiv:1903.09254*, 2019.

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

[29] Gaoang Wang, Yizhou Wang, Haotian Zhang, Renshu Gu, and Jenq-Neng Hwang. Exploit the connectivity: Multi-object tracking with trackletnet. *arXiv preprint arXiv:1811.07258*, 2018.

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

[31] Xiang Wu, Ran He, Zhenan Sun, and Tieniu Tan. A light cnn for deep face representation with noisy labels. *IEEE Transactions on Information Forensics and Security*, 13(11):2884–2896, 2018.

[32] Hongyi Zhang, Andreas Geiger, and Raquel Urtasun. Understanding high-level semantics by modeling traffic patterns. In *Proceedings of the IEEE international conference on computer vision*, pages 3056–3063, 2013.

[33] 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.

[34] Wenzhi Zhao and Shihong Du. Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach. *IEEE Transactions on Geoscience and Remote Sensing*, 54(8):4544–4554, 2016.

[35] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Reranking person re-identification with k-reciprocal encoding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1318–1327, 2017.