Multi-camera Vehicle Tracking from End-to-end based on Spatial-Temporal Information and Visual Features

Wei Jin
Shenzhen University
Shenzhen, China
jinwei2017@email.szu.edu.cn

ABSTRACT

In large-scale traffic video analysis, continuous tracking of vehicles across cameras overcomes the time and space limitations of a single camera, and is conducive to transportation design and traffic flow optimization. In this work, we propose an end-to-end framework for multi-camera vehicle detection, tracking and re-identification in complex traffic environments with urban multi-junctions, which integrates visual features and temporal-spatial information of the trajectories for optimization. Based on detection and tracking of multi-vehicles in a single camera, our method distinguishes and marks the vehicle trajectories from different intersections where they enter and exit. Then, the visual features of the same vehicle keyframes are extracted to match between the cameras of the specific matching link, while taking into account the constraint of the trajectory time. In the end, our algorithm shortens vehicle trajectories' average matching time in two cameras to *2* seconds, and the accuracy is *81.59%* in the test scenarios, which greatly improves the efficiency and accuracy of vehicle re-identification.

CCS Concepts

• Computing methodologies→ Artificial intelligence→ Computer vision→Computer vision problems → Tracking

Keywords

Multi-Target Multi-Camera tracking; Vehicle ReID; Spatial-Temporal Information; Visual Features;

1. INTRODUCTION

In general, multi-target multi-camera tracking (MTMC) has to address three distinct but closely related problems [1], namely multi-target single-camera tracking (MTSC) , re-identification of targets across multiple cameras (ReID) and associated trajectories of the same target in different cameras. In recent years, a lot of attention has been paid to person ReID and MTMC problems [2,3,4,5,6], and other studies have proposed related datasets of vehicle ReID [1,7,8,9,10]. However, due to the change of shooting angle, occlusion, rapid vehicle movement, complicated road structure, etc., video-based solutions for multi-camera vehicle tracking from end-to-end has not been proposed yet.

CSAI2019, December 6–8, 2019, Beijing, China
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-7627-3/19/12…$15.00
DOI: https://doi.org/10.1145/3374587.3374629

In this paper, we propose a framework for continuous tracking of cross-camera vehicles, as shown in Figure 1. In order to obtain the

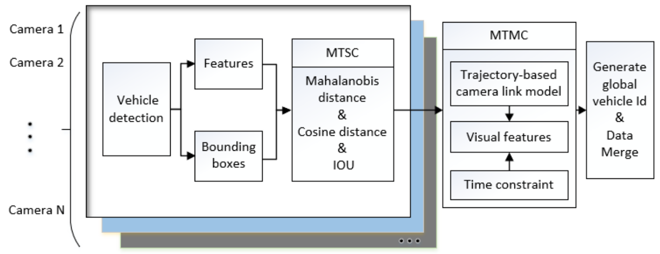
Camera 1
Camera 2
Camera N
Vehicle detection
Features
Bounding boxes
MTSC
Mahalanobis distance & Cosine distance & IOU
MTMC
Trajectory-based camera link model
Visual features
Time constraint
Generate global vehicle Id & Data Merge

Figure 1: The framework of our method, where each color represents one camera

trajectory of each vehicle in a single camera, we infer the vehicle trajectory through Kalman filtering, and carry out the Hungarian algorithm cascade matching[11] in Mahalanobis distance. Then, considering that vehicle may only partially appear in the field of view, we calculate the Intersection over Union (IOU) of the detection frame to match neighboring targets in adjacent frames. In the stage of matching vehicle trajectories of different cameras, we establish a cross-camera trajectory-based camera link model, clustering the vehicle trajectories into a limited number according to direction combination of the adjacent intersections.

On this basis, a unique global vehicle ID can be generated by matching the vehicle trajectories in a single camera based on visual features and time constraints.

2. RELATED WORKS

2.1 Vehicle Detection

Recently, the implementation of anchor-free object detectors based on robust key point estimation has made new progress. For example, CornerNet [12] takes a pair of vertices in the upper left and lower right corners of the prediction target bounding box as the key points. ExtremeNet [13] detect four extreme points (top-most, left-most, bottom-most, right-most) and one center point of objects using a standard keypoint estimation network. However, they all need a keypoint grouping stage to get the target bounding box estimate. Based on previous studies, CenterNet [14] uses the local peak point on the keypoint feature map as the center point of the prediction target, and directly regresses the position of the target without grouping the key points or post-processing such as Non-Maximum Suppression (NMS), which achieves the balance between detection accuracy and speed.

2.2 Multi-Target Single-Camera tracking (MTSC)

Tracking-by-detection scheme has been widely used in the field of multi-object tracking (MOT) [15,16], and it focuses on the relationship between trajectories. For example, based on the linear programming [17] or graph optimization method [18,19], the tracking problem can be solved by minimizing the total cost. When the detection target inevitably occludes, or there are problems such as objects with highly similar appearance in the field of view, the existing multi-object tracking method tends to add similarity of appearance features to obtain more reliable tracking accuracy. [20] combines Kalman Filter and Hungarian algorithm for motion estimation and data association of targets based on the position and size of the detection bounding box. It still achieves good performance at high frame rates. However, it is accurate if the target state estimation uncertainty is low, and once the target detection is lost, it cannot be re-associated with the previous trajectory. Based on this, [11]combines the measurement of target state estimation and appearance similarity, and apply deep cosine metric learning to pedestrian tracking [21], which improves the robustness of target occlusion or miss detection. Nevertheless, when the motion of the occluded target becomes irregular, a better strategy needs to be considered.

2.3 Vehicle Re-identification

The MTMC task is usually to get a picture-based query sequence, so it relies heavily on the appearance features extracted by ReID. Then the different targets are distinguished by supervising the classification and joint metric learning. At present, some studies in the field of pedestrian recognition have achieved good performance. [22] proposes an algorithm using adaptive weighted triplet loss and hard-identity mining for the training of CNN-based features. [23] adopts a re-ranking method in calculating the feature similarity. [2] compares the performance of the loss function of softmax, triplet and its combination, and summarizes the impact of common training tricks on the performance of the model. It achieves the stat-of-the-art performance of pedestrian recognition in Market1501 dataset [24]. In the field of vehicle re-identification, [25] proposes a two-stage framework for vehicle re-identification, which first generates a series of candidate visual-spatial-temporal paths with the query image as the starting and ending states, and then a Siamese-CNN and Path-LSTM network is utilized to determine whether each query pair has the same vehicle identity as the spatial-temporal regularization from the candidate path.

2.4 Multi-Target Multi-Camera tracking (MTMC)

Since the video capture is continuous and relatively fixed in position, the spatial-temporal information between the vehicle trajectories can be fully utilized in vehicle ReID, which is very effective to distinguish vehicles with similar appearances and different trajectories within a certain time. [26] uses a custom area list to describe the trajectory of the vehicle, and combines constraints of time window to establish a camera link model, which reduces the search range for vehicle re-identification. [27] proposes to use the cluster loss and trajectory consistent loss for the training of the vehicle ReID model, and then generate the matching result based on the ranking of the weighted feature and the trajectory information. [28] extracts the appearance features from the global image, the regions and areas around key points, and applies the vehicle tracking, driving direction, vehicle type and time constraints to the re-identification of vehicles. The method achieves the best performance in NVIDIA AI City Challenge 2019 [29].

3. METHODOLOGY

In order to continuously track the vehicles across the camera, we obtain the direction, trajectory and time stamp of each vehicle based on the detection and tracking of the vehicles in the single camera, and then associate the vehicle trajectory with the intersection information to establish a camera link model based on road topology. That is to say, for the key frames extracted by the vehicle in the current camera sequence, they only match the appearance characteristics with key frame sequence inferred by the camera link model and the time attention model. Then we calculate the optimal match sequence and update the globally unique vehicle ID.

3.1 Vehicle Detection

We use the stat-of-the-art detector named CenterNet [14] to build our detection framework, which is an anchor-free algorithm based on the center point estimation of the target bounding box. In addition, the DLA-34 network structure is used in our test. The detection training datasets consist of CityFlow[1] and COCO[30] datasets, and random scale and random flip are used for data enhancement. In order to reduce the interference of the target detection algorithm on the subsequent work, we suppress the detection frame with resolution less than $40\ pixels\ *\ 40\ pixels$, and obtain more reliable detection results.

3.2 Multi-Target Single-Camera tracking (MTSC)

After obtaining the vehicle detection bounding box, we follow the pipeline of DeepSORT [11] to correlate the detection of the same vehicle between frames, and some changes are made. In the multi-target detection framework of DeepSORT, the Kalman filter is used to estimate the motion state of the tracking target in the next frame, which uses an 8-dimensional space to describe. It is $(u, v, r, h, x^*, y^*, r^*, h^*)$, where u, v is the position of the target center point, and r is the aspect ratio, and h is high. When the Mahalanobis distance between the detection target and the predicted position exceeds a certain threshold, it is considered an impossible match. Similarly, the cosine distance is used to measure the difference between the appearance of the detection target and the tracking trajectory, which will filter the matching results with large differences in appearance. Then, based on the weighted distance matrix of the Mahalanobis distance and the cosine distance, the Hungarian algorithm is used to allocate the target detection frame and the tracking trajectory. Finally, in order to alleviate the large change caused by the sudden change of appearance or partial occlusion, the matching of the unmatched trajectories disappearing only one frame is based on the Intersection over Union.

Due to the serious occlusion between the vehicle targets or the loss of detection, the trajectory of target is often lost and the switching of target easily occurs during the tracking process. The practice in the DeepSORT is to use a cascading matching strategy to preferentially match the tracking with shorter occlusion time. When the occlusion time exceeds the preset maximum threshold number of frames A_{max}, the tracked trajectory is deleted from the to-be-tracked list. However, if the threshold is set too small, the occluded target will be regarded as a newly appearing vehicle when it reappears in the field of view. While the setting is too large, the vehicle newly appearing in the field of view is likely to be matched with the previous trajectory. Therefore, It is difficult to find a suitable balanced between the two. At the same time, due to the large traffic volume, the occlusion between vehicles is very common, and the accuracy of DeepSORT will drop sharply at this time.

Therefore, we model the state $State_i$ of the tracking trajectory $\mathrm{tracklet}_i$ to reduce the effect of occlusion on tracking. We assume that the motion state of the vehicle is marked as a deleted state, an occluded state, and a tracked state. And the state space of the tracking bounding box $track_i$ predicted by Kalman filtering is (x_i, y_i, a_i, h_i). When the $track_i$ is completely outside the field of view or the number of consecutive untracked times $Time\_since\_updated$ is greater than A_{max}, the track state is marked as deleted, and $track_i$ will be deleted directly. In addition, $S_i \cap S_j$ is represented as the area of the overlapping portion of $track_i$ and detection bounding box $detection_j$. When it exceeds the threshold S_0 of its own area, it can be judged that $track_i$ is occluded by other targets of the current frame. So we mark the track state as covered, and retain $tracklet_i$ until the next matching is completed. The others are normally tracked. We calculate $State_i$ by the following formula:

$$State_i = \begin{cases} deleted \;\; Time\_since\_updated > A_{max} \\ \quad or \;\; x_i + a_i * h_i < 0 \;\; or \;\; x_i > W \\ \quad or \;\; y_i > H \;\; or \;\; y_i + h_i < 0 \;, \\ covered \;\; (S_i \cap S_j)/S_i > S_0 \;, \\ tracked \;\; otherwise \end{cases} \quad (1)$$

Where W and H are the length and width of the video frame, respectively. When the $tracklet_i$ that has disappeared from the field of view will be deleted in time, and that of the occlusion state will be retained until the next match. A relatively small A_{max} can be set to handle only possible missed detections. At this point, the $tracklet_i$ that uses cascading matches will be a smaller number and more trustworthy.

3.3 Trajectory-based camera link model

In real-world scenarios, it’s easier to obtain the relative position and orientation of the camera in the map, but accurate geographic coordinates and camera parameters are often unknown. Fig.2.1 shows the camera position and shooting direction information in one of the scenes in CityFlow[1] dataset. Since the vehicle travels on the road, it usually follows a fixed road structure and traffic rules, and its travel is directional. When we associate the trajectory of each vehicle with the intersection, each vehicle link can simply be clustered into a directional combination of a limited number of adjacent intersections without the need to obtain accurate camera parameters and geographic coordinates.

In order to effectively distinguish the different vehicle trajectories in the camera, we obtain image coordinates of the identification points of the intersections on the image as the clustering center of the vehicle trajectory, as is shown in Fig.2.2, 2.3, 2.5, 2.6 . Then, the Euclidean distance between the bottom-center point of the starting and ending vehicle tracking bounding box and the image coordinates of the Identification point of each intersection is calculated. The intersection corresponding to the shortest distance can be considered as the intersection where the vehicle enters or exits. At this time, each vehicle trajectory corresponds to a pair of links from intersection m to intersection n in camera i , denoted as $T_i^{mn} = \{[start_i^m, end_i^n]\}$, through which the vehicle trajectory can be distinguished. We only consider the different vehicle trajectories of m and n, which will filter out vehicles parked on the side of the road or unnecessary tracking. Compared with the region of interest (ROI) filtering these unnecessary targets, we do not need human intervention, but also reduce the computation of ReID and improve the accuracy.

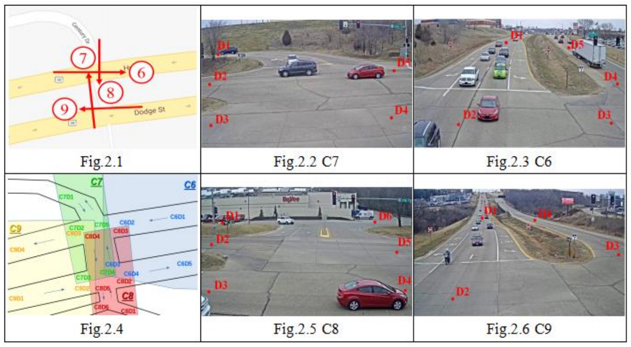

Figure 2(2.1-2.6): Fig.2.1 shows the camera distribution and shooting directions in one of the scenes in the CityFlow dataset. The red arrows denote the locations and directions of cameras. Fig.2.2 shows examples of camera views in camera 7, which is abbreviated as C7. And so on, for views in Fig.2.3, Fig.2.5, Fig.2.6. We use the red dot in the figures to identify each intersection, its position is approximately selected, and the number starting with D indicates. Fig.2.4 shows each camera's relative shooting range and intersections, based on which we reconstruct the trajectory-based camera link.

The relative geographical location of the camera group and the shooting direction provide a sufficient basis for the reconstruction of the topological relationship between the roads, as is shown in Fig.2.4. In order to link the vehicle trajectories in each camera together, we define the camera link as the following formula:

$$L_{ij}^{mn} = \{[out_i^m, in_j^n]\} \quad (2)$$

It indicates that the vehicle can be driven out of the m intersection in camera i and from the intersection n in camera j. For a two-way lane, the same intersection may have a vehicle exiting or a vehicle entering. Therefore, the link between the camera i and the camera j can be defined as $L = (L_{ij}, L_{ji})$, expressed as a set of all two-way lane camera links.

After defining the correspondence between the vehicle trajectory, the camera link and the intersection, for the camera link $L_{ij}^{mn} = \{[out_i^m, in_j^n]\}$, we extract the vehicle trajectory $T_i^{\alpha m} = \{[start_i^\alpha, end_i^m]\}$ that exits from the m intersection in the camera i according to out_i^m to join the gallery sequence g_i^m, where α indicates the intersection that is arbitrarily entered. According to the in_j^n, the vehicle trajectory $T_j^{n\beta} = \{[start_j^n, end_j^\beta]\}$ that is drawn from the n intersection in the camera j is extracted and added to the query sequence q_j^n, where β represents an arbitrary exit intersection. Thereby, the vehicle matching sequence (q_j^n, g_i^m) corresponding to the camera link L_{ij}^{mn} can be generated according to the traveling trajectory T_i^{mn} of the vehicle, and for the link L between the camera i and the camera j, the vehicle matching sequence $M = \{(q_i, g_j), (q_j, g_i)\}$ can be generated.

3.4 Multi-Target Multi-Camera tracking (MTMC)

In the pedestrian ReID task, the depth feature extraction and matching model in [2] has proved to be very effective, and its performance on the Market1501 dataset [24] can reach *94.5%* of Rank1 and *85.9%* of mAP. In this paper, we use it to train the appearance of the vehicle to improve the performance of the vehicle's ReID. In single-camera multi-vehicle tracking, we obtain a sequence of consecutive vehicle tracking pictures of each vehicle in a single video to form their respective vehicle trajectories. The trajectory-based camera link model helps to correlate individual vehicle trajectories to generate a limited vehicle matching sequence. However, it is not sensible to use a single camera to capture the entire sequence of vehicle tracking pictures in the video, which can greatly increase the consumption of computing resources. Moreover, the random use of a single vehicle picture may produce unstable results.

The solution in [27] is to calculate the average feature of the trajectory of the gallery images. However, we consider that the sequence of pictures for each trajectory may contain features of different angles and resolutions of the vehicle, which may be lost in the calculation of the average. Since the appearance of the vehicle in adjacent frames is less variable, it may be more efficient to take a limited number of vehicle pictures in the vehicle trajectories to form a new one. Then the vehicle matching sequence based on the camera link model can be generated. In addition, taking into account the time information of the traffic in the video, we record the time t_s when each vehicle exits from the m intersection in the camera i and the time t_e enters from the n intersection in the camera *j*. For the defined vehicle matching sequence (q_j^n, g_i^m), We estimate the maximum interval time of the vehicle traveling on the invisible road sections and consider the matching sequence that satisfies $t_e < t_s$ as an impossible match, which can also greatly reduce the matching search space and improve the matching precision.

4. EXPERIMENTS

4.1 Dataset for Vehicle ReID

We conducted experiments on the city-scale traffic camera dataset CityFlow [1], which has wider coverage and provides information such as the original video and shooting time than the Veri-wild [7], VeRi-776 [8], VehicleID [9] and PKUVD [10] datasets. The CityFlow dataset includes *10* intersections in a mid-sized U.S. city, *3.25* hours (*195.03* minutes) of synchronized video taken by 40 cameras, and the longest distance between two simultaneously recorded cameras is *2.5* km. Of these, *58.43* minutes of videos are used for training and another *136.60* minutes of videos are used for testing.

4.2 Implementation Details

Referring to the organization of the Market1501 dataset, we organize the vehicle trajectories tracked in each video into respective vehicle matching sequences M based on the camera link model. In the vehicle ReID, we used the ResNet50[31] network as the Backbone and added the proposed tricks to train the vehicle ReID model, including Warmup, Random erasing augmentation, Label Smoothing, Last Stride, BNNeck and Center loss. One of the mini-batch contains each K picture of the P car. After the picture passes the Backbone network, the global feature is obtained. Then, according to this feature, a classification loss (ID loss) and a triplet loss (batch hard mining) are calculated. In the test step, the cosine distance is used to perform the similarity measure to obtain the matching result of each vehicle matching sequence.

Taking into account the performance of the calculation, we finally choose $K = 5$ to generate a sequence of pictures for each vehicle trajectories. The vehicle id is finally unified into the id of the query vehicle, and the result is the maximum value of rank1 in the K-time matching result of the vehicle, which can filter out some bad matches. Figure 3 shows the matching results between the query vehicle and the gallery vehicle generated by the vehicle ReID model. The average time to complete the matching of the two intersections in the test is *2* seconds. Based on the following formula, we evaluate the continuous tracking performance of video-based multi-camera vehicles.

$$A = (TP+TN) / (TP+TN+FP+FN) \qquad (3)$$

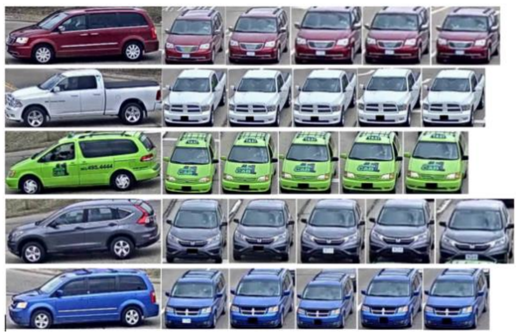
Figure 3: The matching results between the query vehicle (leftmost column) and the gallery vehicle (else) generated by the vehicle ReID model.

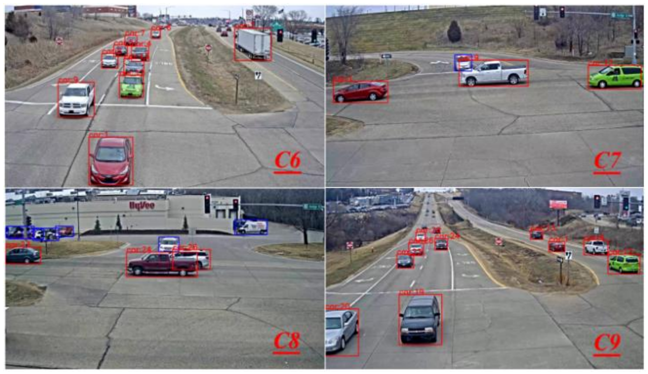

Figure 4: An example of the performance of multi-camera vehicle tracking for one scene of the test video. The red bounding boxes represent the vehicle that has been tracked, and the blue bounding boxes represent the vehicle to be tracked.

Where A is the accuracy, TP stands for true positives, indicating the correct match; TN stands for true negatives; FP stands for false positives; FN stands for false negatives. In the scenario tested in Figure 2, the accuracy of the vehicle ReID is *81.59%*. Figure 4 shows an example of the performance of multi-vehicle tracking for one scene of the test video. The trajectory-based camera link model, time constraint and vehicle ReID model fusion are used to solve the problem of multi-camera vehicle tracking, and the performance of vehicle matching is improved while reducing the amount of calculation.

5. CONCLUSION

In this paper, we propose a camera link model based on trajectory and time constraints to deal with the correlation problem of multi-camera vehicle trajectories. During vehicle tracking, additional handling of vehicle occlusion problems in complex road conditions is performed, which reduces frequent switching of target IDs in a single camera. In addition, the ReID model of the vehicle is retrained, which has better robustness for different vehicles in a single camera with similar appearance and different appearance of the same vehicle. The vehicle visual feature matching is also completed by selecting key frames of the vehicle, which reduces the calculation amount, reduces the excessive dependence on the detection, tracking and re-identification, and further improves the accuracy of continuous tracking of the vehicle. In the experiment, the average time to complete the matching of the two intersections was reduced to *2* seconds, and the *81.59%* accuracy was achieved, indicating that the proposed method is very effective.

6. REFERENCES

[1] Tang Z, Naphade M, Liu M Y, et al. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 8797-8806.

[2] Luo H, Gu Y, Liao X, et al. Bag of tricks and a strong baseline for deep person re-identification[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2019: 0-0.

[3] Wei L, Zhang S, Gao W, et al. Person transfer gan to bridge domain gap for person re-identification[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 79-88.

[4] Zheng Z, Zheng L, Yang Y. Unlabeled samples generated by gan improve the person re-identification baseline in vitro[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 3754-3762.

[5] Ristani E, Solera F, Zou R, et al. Performance measures and a data set for multi-target, multi-camera tracking[C]//European Conference on Computer Vision. Springer, Cham, 2016: 17-35.

[6] Zheng L, Shen L, Tian L, et al. Scalable person re-identification: A benchmark[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1116-1124.

[7] Lou Y, Bai Y, Liu J, et al. Veri-wild: A large dataset and a new method for vehicle re-identification in the wild[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 3235-3243.

[8] Liu X, Liu W, Mei T, et al. A deep learning-based approach to progressive vehicle re-identification for urban surveillance[C]//European Conference on Computer Vision. Springer, Cham, 2016: 869-884.

[9] Liu H, Tian Y, Yang Y, et al. Deep relative distance learning: Tell the difference between similar vehicles[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 2167-2175.

[10] Yan K, Tian Y, Wang Y, et al. Exploiting multi-grain ranking constraints for precisely searching visually-similar vehicles[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 562-570.

Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric[C]//2017 IEEE International Conference on Image Processing (ICIP). IEEE, 2017: 3645-3649.

[12] Law H, Deng J. Cornernet: Detecting objects as paired keypoints[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 734-750.

[13] Zhou X, Zhuo J, Krahenbuhl P. Bottom-up object detection by grouping extreme and center points[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 850-859.

[14] Zhou X, Wang D, Kr ähenb ühl P. Objects as Points[J]. arXiv preprint arXiv:1904.07850, 2019.

[15] Geiger A, Lauer M, Wojek C, et al. 3d traffic scene understanding from movable platforms[J]. IEEE transactions on pattern analysis and machine intelligence, 2013, 36(5): 1012-1025.

[16] Zhang H, Geiger A, Urtasun R. Understanding high-level semantics by modeling traffic patterns[C]//Proceedings of the IEEE international conference on computer vision. 2013: 3056-3063.

[17] Schulter S, Vernaza P, Choi W, et al. Deep network flow for multi-object tracking[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 6951-6960.

[18] Tang S, Andriluka M, Andres B, et al. Multiple people tracking by lifted multicut and person re-identification[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 3539-3548.

[19] Tang Z, Wang G, Xiao H, et al. Single-camera and inter-camera vehicle tracking and 3D speed estimation based on fusion of visual and semantic features[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2018: 108-115.

[20] Bewley A, Ge Z, Ott L, et al. Simple online and realtime tracking[C]//2016 IEEE International Conference on Image Processing (ICIP). IEEE, 2016: 3464-3468.

[21] Yi D, Lei Z, Liao S, et al. Deep metric learning for person re-identification[C]//2014 22nd International Conference on Pattern Recognition. IEEE, 2014: 34-39.

[22] Ristani E, Tomasi C. Features for multi-target multi-camera tracking and re-identification[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 6036-6046.

[23] Zhang Z, Wu J, Zhang X, et al. Multi-target, multi-camera tracking by hierarchical clustering: Recent progress on dukemtmc project[J]. arXiv preprint arXiv:1712.09531, 2017.

[24] Zheng L, Shen L, Tian L, et al. Scalable person re-identification: A benchmark[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1116-1124.

[25] Shen Y, Xiao T, Li H, et al. Learning deep neural networks for vehicle re-id with visual-spatio-temporal path proposals[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 1900-1909.

[26] Hsu H M, Huang T W, Wang G, et al. Multi-Camera Tracking of Vehicles based on Deep Features Re-ID and Trajectory-Based Camera Link Models[C]//AI City Challenge Workshop, IEEE/CVF Computer Vision and Pattern Recognition (CVPR) Conference, Long Beach, California. 2019.

[27] He Z, Lei Y, Wu S B W. Multi-Camera Vehicle Tracking with Powerful Visual Features and Spatial-Temporal Cue[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2019: 203-212.

[28] Tan X, Wang Z, Jiang M, et al. Multi-camera vehicle tracking and re-identification based on visual and spatial-temporal features[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2019: 275-284.

[29] Naphade M, Tang Z, Chang M C, et al. The 2019 AI City Challenge[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2019: 452-460.

[30] Lin T Y, Maire M, Belongie S, et al. Microsoft coco: Common objects in context[C]//European conference on computer vision. Springer, Cham, 2014: 740-755.

[31] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.