Multi-Camera Vehicle Tracking Based on Occlusion-aware and Inter-vehicle Information

Yuming Liu, Xiaochun Zhang, Bingzhen Zhang, Xiaoyong Zhang, Sen Wang, Jianrong Xu
Shenzhen Urban Transport Planning Center Co., Ltd., Shenzhen, China
{liuyuming, zxc, zhangbingzhen, zhangxy, wangsen, xujianrong}@sutpc.com

Abstract

With the demands of analyzing and predicting traffic flow for applications in smart cities, Multi-Target Multi-Camera vehicle Tracking(MTMCT) at the city scale has become a fundamental problem. The MTMCT is challenging due to the view variations, frequent occlusions, and similar vehicle models in the same camera. This work proposes an MTMCT framework based on occlusion-aware and inter-vehicle information that can effectively match vehicle tracklets. The occlusion-aware module segments the tracklets of an occluded and occluding vehicle pair. It recalculates the similarity of the complete tracklets, which can handle the occlusions and suppress false detections. This work proposes an inter-vehicle information module to improve the matching accuracy. The module can enhance the ability to distinguish similar vehicles under the same camera at different times. The proposed whole framework consists of four modules: (1) vehicle detection and feature extraction by re-identification models, (2) single-camera tracking (SCT) to produce initial tracklets with an occlusion-aware module, (3) tracklets similarity by inter-vehicle association, (4) clustering in adjacent cameras for multi-camera tracklets matching. The proposed method obtains IDF_1 score of 0.8285 on the Track-1 multi-camera vehicle tracking task of the 2022 AI City Challenge.

1. Introduction

Multi-Target Multi-Camera vehicle Tracking (MTMCT) is an essential component in many tasks related to transportation. It can provide rich information for traffic signal time planning, automatic traffic monitoring, traffic flow prediction, and simulation. As shown in Figure 1, the MTMCT task is to track multiple vehicle targets from the cameras at different locations. Typically, the main components of an MTMCT system include vehicle detection, single-camera tracking (SCT), vehicle re-identification (ReID), and multi-camera tracklets matching (MCTM). Unlike

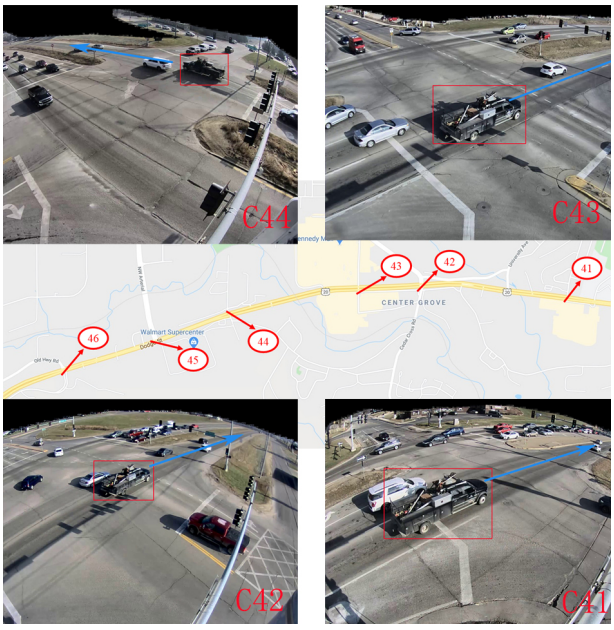

Figure 1. Multi-target multi-camera tracking task is to track multiple vehicle targets from the cameras at different locations

classical single-camera multiple object tracking (MOT), MTMCT needs to generate one complete global tracklets for an identical vehicle on a long road by different cameras without overlapping areas. For the SCT module, many tracking-by-detection methods have been proposed recently due to the improvement of object detection techniques [2, 26, 28]. This paradigm will generate a set of detections for each video frame independently that the detections link to tracks based on a similarity measure. That similarity often considers visual features extracted by a re-identification model for the MCTM module, Hsu *et al*. [11] propose strategies of distinguishing tracklets with several pre-defined zones for every camera and then apply the greedy algorithm to generate the final matching result. Chone Liu *et al*. [13] apply the hierarchical clustering method to match adjacent camera tracklets. Different lev-

els of clustering focus on the spatio-temporal relationship of different tracklets, which will help identify vehicles whose links suffer from significant appearance changes.

However, there are still several significant challenges for the MTMCT task. The representational ability of the extracted features under occlusion since the re-ID model can get confused by overlapping nearby targets limited by camera resolution, distortion, deformation of lighting conditions, and vehicle occlusion conditions. In real traffic scenarios, vehicles have similar appearances. So it is unreliable to use the re-identification model alone to distinguish these vehicles in the SCT stage. In addition, the identical vehicle instance may have a significant appearance change under different cameras because its position relative to the camera has changed. It is also tricky for the re-identification model to distinguish these tracklets directly. It leads to the fact that the clustering algorithm cannot directly distinguish the tracklets of similar vehicles under a single camera.

Many occlusions and missed results are generated when the vehicle waits for the traffic light in the SCT. We propose an occlusion re-ranking module to mark the occluded vehicle instances. It segments the tracklets of an occluded and occluding vehicle pair and recalculates the similarity of the sub-tracklets to re-map the id. The sub-tracklets will be stored for follow stage to improve the impermanence of MCTM.

Aiming at the challenge that it is difficult to distinguish similar vehicles in the MCTM, we propose an inter-vehicle information module. We employ the adjacent tracklets features to calculate the similarity and improve the matching accuracy of tracklets between multiple cameras. The motivation is a particular similarity between in-vehicle queues between cross-cameras. The surrounding vehicles of the target vehicle are also similar to other cameras. For the followed clustering stage, we use inter-vehicle information to calculate the similarity of the tracklets that can distinguish vehicles with similar appearances at different times of the same camera.

The main contributions of this paper are summarized as follows:

- Propose an occlusion-aware module to segment the tracklets of an occluded and occluding vehicle pair and recalculate the similarity of the sub-tracklets, which can reduce the occlusions and suppress false detections in the SCT stage.

- A inter-vehicle information module is proposed to improve the matching accuracy of tracklets between multiple cameras, which can significantly avoid feature mismatching between cross-cameras.

2. Related Work

2.1. Vehicle Detection

Object detection is the most popular task in computer vision, which locates and classifies the objects by a bounding box. Typically, this task can be organized into one-stage methods and two-stage methods. SSD [14] and YOLO [23] are representative one-stage methods that combine detection and recognition into one integrated model. The two-stage approach, Faster-RCNN [6], and Mask R-CNN [7], split it into region proposal network (RPN) and another classification model to improve the prediction of bounding boxes.

2.2. Single-Camera Vehicle Tracking

Single-Camera Tracking (SCT) could be divided into the tracking-by-detection paradigm, while the other is jointing object detection with Re-ID in a single network. Bewley *et al*. introduce a Simple Online and Realtime Tracking (SORT) algorithm [2], which tracks bounding boxes by using a Kalman filter and Hungarian algorithm correctly. Nicolai *et al*. [26] propose that the appearance features extracted from a deep network enhance the association cost algorithm on the base of SORT. The deep network provides a normalized vector with 128 features and uses the cosine distance between those vectors to compute similarity scores. On the other hand, some research joint the appearance embedding model into a single-shot detector so that the model can simultaneously output detections and the corresponding feature [1, 22].

2.3. Vehicle Re-identification

The aim of vehicle re-identification (Re-ID) is to retrieve vehicles that appear in different cameras. Thanks to the popularization of smart cities and smart transportation, vehicle ReID has received more attention and research. Chen *et al*. [3] and Rikiya *et al*. [21] mine informative samples in the vehicle Re-ID training phase. Shen *et al*. [19] apply the spatial-temporal constraints to reduce the sample search space. Zheng *et al*. [29] propose a joint learning framework, which combines ReID learning and data generation end-to-end. Zhou *et al*. [30] generate a multi-view feature by transforming a single-view feature against the orientation variation. In recent years, with the rapid development of transformer-based vision tasks, vehicle re-identification has been greatly improved as in [10, 15].

2.4. Multi-Camera Vehicle Tracking

Most MTMCT researches design involves the following steps, object detection, multi-target single-camera tracking, appearance feature extraction for ReID, and cross-camera tracklets matching. Methods [4, 5] establish a global graph for multiple tracklets in different cameras and optimize for

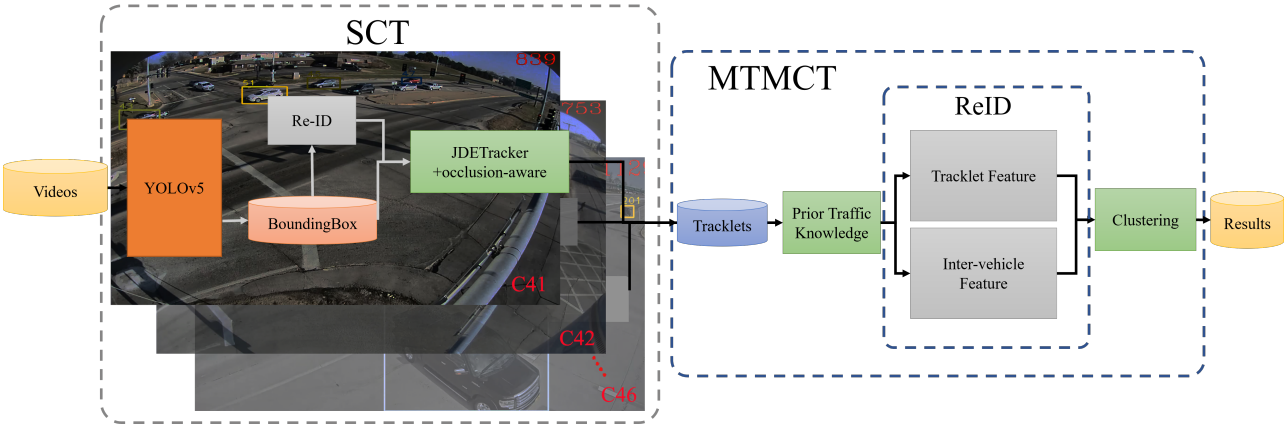

Figure 2. The proposed MTMCT framework is shown in Figure 2. The steps of MTMCT are as follows: (1) Vehicle detection and feature extraction by re-identification models, (2) single-camera tracking to produce raw tracklets with occlusion-aware module, (3) tracklets similarity by prior traffic knowledge and inter-vehicle information, (4) clustering in adjacent cameras for multi-camera tracklets matching. The detailed process will be described below.

an MTMCT solution. In the cross-camera tracklets matching stage, most methods regard matching as a tracklets clustering problem that needs to calculate the similarity between tracklets. Many researchers focus on reducing searching space by the spatial-temporal constraints and traffic rules. Chong Liu *et al*. [13] propose Direction Based Temporal Mask, which helps reduce matching space for visual re-identification. It also proposes sub-clustering in adjacent cameras to merge adjacent tracklets and then uses these matched local tracklets for query expansion. The method helps link the vehicles suffering from great appearance changes. Jin Ye *et al*. [27] also establishes the distance matrix to associate all candidate trajectories between two consecutive cameras. It reduces the search space by constraints of travel time, road structure, and traffic rules, where Re-ID features compute all distance matrices.

3. Proposed Method

3.1. Overview

The proposed MTMCT framework is shown in Figure 2. The steps of MTMCT are as follows: (1) vehicle detection and feature extraction by re-identification models, (2) single-camera tracking to produce raw tracklets with an occlusion-aware module, (3) tracklets similarity by prior traffic knowledge and inter-vehicle information, (4) clustering in adjacent cameras for multi-camera tracklets matching. The detailed process will be described below.

3.2. Vehicle Detection

Vehicle detection is the basis of the MTMCT task, and the performance directly affects the effectiveness of the whole framework. The AI City Challenge organization provides vehicle detection baselines from popular deep models, including YOLOv3, SSD [14], and Mask R-CNN [7]. Surprised by the performance and simplicity of the latest version of YOLO, this paper applies the YOLOv5x model to detect vehicles. The model pre-trained on the COCO dataset provided by the YOLO organization has proved its performance in cars detection. The official model pre-trained on COCO datasets contains 80 categories, but only cars, trucks, and buses are used in this task. Therefore, the detection model in this paper combines these three types of targets and uses non-maximum suppression to avoid the same target being detected multiple times.

3.3. Vehicle Re-identification

Following existing vehicle re-identification works, we apply FastReID Toolbox [9] to build a re-identification model. We use the BOT-R50-IBN backbone, which has a strong generalization ability to extract appearance features for the presence of occlusions, different illumination conditions, and viewpoint changes. The BOT-R50-IBN consists of ResNet50 [8], an attention-like non-local module, and an instance batch normalization (IBN) module [24] which can learn more robust features. The model uses generalized mean pooling (GeM) to aggregate feature maps generated by the backbone into a global feature. The loss function used is Cross-Entropy loss and Triplet loss like:

$$L_r = L_c + \alpha L_t \tag{1}$$

while the L_c and L_t stands cross-entropy loss and triplet loss respectively. The cross-entropy loss is formulated as follows:

$$L_c = \sum_{C}^{i=1} y_i log(\hat{y_i}) + (1 - y_i) log(1 - \hat{y_i}) \tag{2}$$

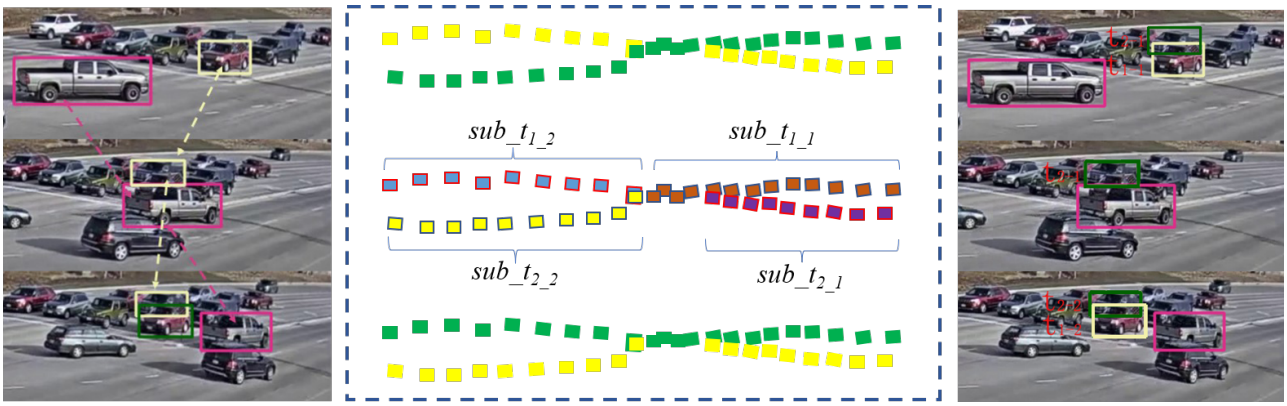

Figure 3. The illustrations of Occlusion-aware Module. Split the raw tracklets into $sub\_t_{1-1}, sub\_t_{1-2}, sub\_t_{2-1}, sub\_t_{2-2}$ due to Intersection-over-Union. Then calculate the similarity of re-id features between the sub-tracklets respectively, according to the similarity, sub_t_{1-1} and sub_t_{2-2} are merged, and sub_t_{1-2} and sub_t_{2-1} are merged.

where C is the number of vehicle categories in the dataset, $\hat{y}_i$ is the i ground-truth label, and y_i is the i predicted probability.

Triplet loss focuses on optimizing the distance between the training samples, to ensure that the embedding $g(x^a)$ of the anchor vehicle is closer to its positive $g(x^p)$ than the negative vehicle, the triplet loss with N samples can be formulated as:

$$L_t = max(D, 0) \tag{3}$$

where the D is,

$$D = \sum_{i=1}^{N} \left[\|g(x_i^a) - g(x_i^p)\|^2 - \|g(x_i^a) - g(x_i^n)\|^2 + \beta \right] \tag{4}$$

3.4. Single-Camera Tracking

3.4.1 Basic Algorithm

In single-camera tracking, the goal of the task is to associate detections in video frames with the corresponding tracklets. Following the tracking-by-detection paradigm, we crop the detected target from the detection and use the ReID model to extract the appearance features of the target. Then we refer to the JDE [25] to build a tracker management module, which uses the extracted appearance features. It performs Cascade Matching and Kalman filter correction with the position information of the vehicle in the image to generate the tracklets. Finally, the tracklets of each vehicle under a single camera are obtained. The tracklets will contain the following vectors :

$$Traj^{id} = \{Traj_n^{id}, Traj_{n+1}^{id} ... Traj_m^{id}\} \tag{5}$$

while $Traj_t^{id}$ represents the feature of each track at time t. It consists of appearance feature f_t and position feature b_t.

$$Traj_t^{id} = f_t, b_t \tag{6}$$

3.4.2 Occlusion-Aware Module

Single-camera multi-target detection algorithms usually encounter occlusions and missed detections. The literature [20] introduces the concept of occlusion and occlusion tracklets and determines whether it is an active or inactive tracklets through non-maximum suppression. The minimum overlap criterion adds additional constraints such as static tracklets to identify and remove false detections. However, this constraint ignores the appearance features of historical tracklets, so we propose an occlusion-aware module and re-identify occlusion tracklets through tracklets segmentation. Similarly, the IOU of the two tracklets is calculated and set at a threshold ζ_{iou}. An occluded and occluding tracklets pair will be marked while the IOU is greater than the ζ_{iou} threshold. We divide the original two tracklets into sub-tracklets and calculate the similarity of appearance features between the sub-tracklets, respectively. Finally, we join the most similar sub-tracklets according to the time sequence and re-assign the id. As shown in Figure 3, the original SCT module switches the id between black car and red car due to occlusion. The proposed module will split the raw tracklets into $sub\_t_{1-1}, sub\_t_{1-2}, sub\_t_{2-1}, sub\_t_{2-2}$. Then calculate the similarity of re-id features mentioned above between the sub-tracklets, respectively. According to the similarity, sub_t_{1-1} and sub_t_{2-2} are merged, and sub_t_{1-2} and sub_t_{2-1} are merged.

3.5. Multi-Camera Tracklets Matching

After the SCT stage, the next step is to match single-camera tracklets. The multi-camera tracklets matching mainly includes three parts. The road prior knowledge is used to reduce the matching space. The inter-vehicle information is used to establish a matching similarity matrix. The clustering algorithm completes the final matching of the multi-camera tracklets.

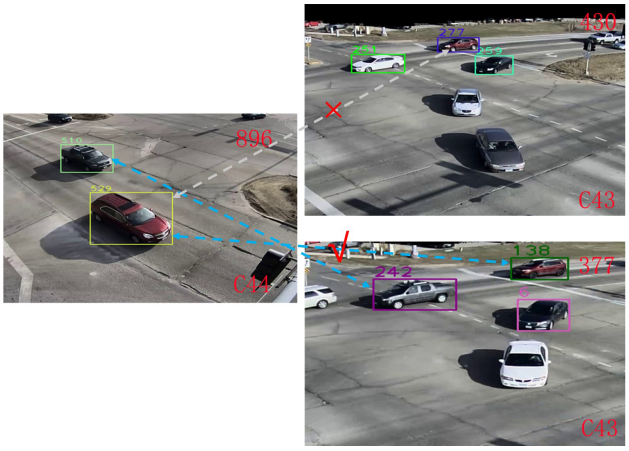

Figure 4. The illustrations of Inter-vehicle Information Module. For the id_{529} vehicle in the C44 camera, the red vehicle id_{277} and id_{138} are both similar to it. There is a id_{242} vehicle around id_{138} in C43 that is similar to id_{520} in C44, so the id_{138} vehicle in C43 should be considered a matching vehicle.

3.5.1 Prior Traffic Knowledge

In the MTMCT task, prior knowledge of the road is the essential part that can greatly reduce the search space of the tracklets. It also filters out the matching tracklets pairs unreasonably in real road scenes. Following previous work [11, 13, 17, 27], The proposed method uses the road camera distribution to filter out useless matching pairs, including both spatial and temporal dimensions. We divide the crossroads into four zones first. Each tracklet under a single camera must contain the start−end zone and time so that it will be marked as,

$$traj = \{c, zs, ze, ts, te\} \tag{7}$$

Where the c represents camera id, zs, ze, ts, te stands for the start zone, end zone, start time, and end time. If the tracklet is only across one zone, the tracklets are considered noise. According to the road distribution information, it can be determined that two adjacent camera zones are connected. For example, the C44 camera zone 1 will be connected to the C43 camera zone 2, which is marked as $C44_1 - C43_2 - Dis$. Dis is the distance between the two cameras. In the tracklets matching stage, any tracklets under SCT will be considered whether they satisfy the constraint mentioned above and estimate time conflicts of the tracklets according to Dis. In addition, the entire region topology can be constructed as a binary tree structure, and the connectivity of non-adjacent regions can be judged by Depth-First-Search (DFS).

3.5.2 Inter-Vehicle Information Module

The similarity calculation first considers the appearance features of the ReID model. However, it is not easy to distinguish similar vehicles with the same camera. Under different cameras, the vehicles around the same target are similar. For example, as shown in Figure 4, for the id_{529} vehicle in the C44 camera, the red vehicle id_{277} and id_{138} are both similar to it. The time lag between the two vehicles being tracked is only about 1 second. Thus, it is not easy to distinguish only by the vehicle's appearance. But there is a id_{242} vehicle around id_{138} in C43 that is similar to id_{520} in C44, so the id_{138} vehicle in C43 should be considered a matching vehicle. Inspired by this information, we propose an inter−vehicle information module to enhance the capability of distinguishing. For each SCT tracklets, the k neatest vehicles for the target during the tracking time will be counted and sorted. The features of the top n neatest surrounding vehicles will be taken as the inter-vehicle features. Therefore each SCT tracklets will be represented by the following:

$$Traj^{id} = f, b, \hat{f} \tag{8}$$

$\hat{f}$ present the inter−vehicle features. For a pair of tracklets, the proposed method identifies valid inter-vehicle by Hungarian matching, which also uses the appearance feature of inter−vehicle from the same ReID model. The final similarity calculation is formulated as,

$$S(Traj_i, Traj_j) = \lambda S(f_i, f_j) + \mu \frac{\sum_{e=1}^{\leq n} MS(\hat{f}_{ie}, \hat{f}_{je})}{n} \tag{9}$$

S refers to cosine similarity, while the MS refers to cosine similarity of matched inter-vehicle. λ, μ stands for the scale coefficient. It can seem that if there are more inter-vehicle matching pairs between the two tracklets, the similarity score will be higher.

3.5.3 Hierarchical Clustering

Since the works [11–13] show promising competitiveness in the MTMCT task by using hierarchical clustering methods, we apply similar clustering methods to match tracklets. Based on the constraints motioned above and the proposed inter-vehicle information, hierarchical classification algorithms are employed to complete the final tracklets matching. Encouraged, we only consider tracklets from adjacent regions to ensure high-confidence clustering at the first clustering. The remaining trajectories are clustered again to explore matching pairs of tracklets spanning multiple cameras. Finally, we can merge matched tracklets successfully to form completed tracklets.

Method	IDF_1	IDP	IDR	Precision	Recall
baseline	79.58	82.62	76.76	86.25	80.13
+occ	81.15	85.95	76.86	**89.49**	80.03
+occ+inter	**82.85**	**86.54**	**79.46**	88.81	**81.54**

Table 1. The performance of each module proposed.

4. Experiments

4.1. Dataset

The CityFlowV2 [16] consisted of 3.58 hours (215.03 minutes) of video captured by 46 cameras spanning 16 intersections in a mid-sized U.S. city. The dataset is divided into six simultaneous scenarios. Three are used for training, two for validation, and the other for testing. The dataset contains 313931 bounding boxes for 880 distinct annotated vehicle identities. The vehicles that passed through more than one camera are labeled. Each video's time offset and geographic location are provided in each scenario to utilize Spatio-temporal knowledge. The resolution of each video is about 960p, and the videos have a frame rate of 10 FPS. CityFlowV2 covers various road traffic types, including intersections, road extensions, and highways. The subset for vehicle ReID, namely CityFlowV2-ReID, is split into a training set with 52 717 images from 440 identities and a test set including 31238 images from another 440 identities. An additional 1103 images are sampled as queries.

4.2. Evaluation Metrics

For the MTMCT task, the IDF_1 score [18] is used to rank the performance on the leaderboard. IDF_1 calculates the ratio of the number of correctly identified detections to the ground truth and the average number of calculated detections. Denote $IDTP$ as the count of true positive IDs, $IDTN$ as the count of true negative IDs, $IDFP$ as the count of false-positive IDs, and $IDFN$ as the count of false-negative IDs. The IDF_1 scores could be calculated as:

$$IDF_1 = \frac{2IDTP}{2IDTP + IDFP + IDFN} \tag{10}$$

4.3. Implementation Details

The proposed method is implemented in PyTorch 1.7.1 and is performed on two NVIDIA V100 GPUs. In the detection stage, we use the YOLOv5x model pre-trained on COCO to perform vehicle detection with a confidence threshold of 0.1. We use FastReID Toolbox to train our model in the vehicle Re-ID training process. The BOT-R50-IBN is the backbone for training and inference. The model is trained using Adam with the initial learning rate of 0.00035, batch size of 4, and the weight decay of 0.0005. In the SCT stage, we use a modified JDETracker to perform single-camera vehicle tracking with a confidence threshold of 0.1 and an area threshold of 750 pixels. In the occlusion-aware module, the IOU threshold is 0.75. To build an inter-vehicle module, we set the k to 3 while the n is set to 3.

Team ID	Name	IDF_1
28	matcher	0.8486
59	BOE	0.8437
37	TAG	0.8371
50	FraunhoferIOSB	0.8348
70	appolo	0.8251
36	Li-Chen-Yi	0.8218
10	Terminus-AI	0.8171
118	FourBeauties	0.8166
110	Orange Peel	0.8140
94	SKKU Automation Lab	0.8129
107	**SUTPC(Ours)**	0.8285

Table 2. Comparison of other team.

4.4. Experiments Results

Table 1 shows the effect of using each module separately on the results that verified the effects of the proposed module. The occlusion-aware module effectively improves IDF_1 scores from 79.58% to 81.15%. Furthermore, the inter-vehicle information module made significant progress both in the IDF_1, IDP, and IDR scores. Finally, the whole proposed method achieves 82.85% IDF_1 scores. Table 2 shows the comparison of other teams, which indicates that the proposed method has good competitiveness with other teams.

4.5. Visualization

Figure 5 shows the final matching results of the proposed method on CityFlowV2. From left to right are the C41, C42, and C43 cameras. Furthermore, the identical vehicles are marked with the same ID across the different cameras. In the first row, it can seem that a black truck marked as id_{56} and green box has been tracked correctly from C41 to C43 as same as the id_{18} green car in the second row and the id_{71} blue car in the third row. The proposed method generates correct matching pairs in different cameras even if the tracklets have different angles or occlusion.

5. Conclusion

This paper proposes an effective framework for the MTMCT task guide by occlusion-aware and inter-vehicle information. Unlike other general MTMCT frameworks, we propose an occlusion-aware module to segment the tracklets of an occluded and occluding vehicle pair. The

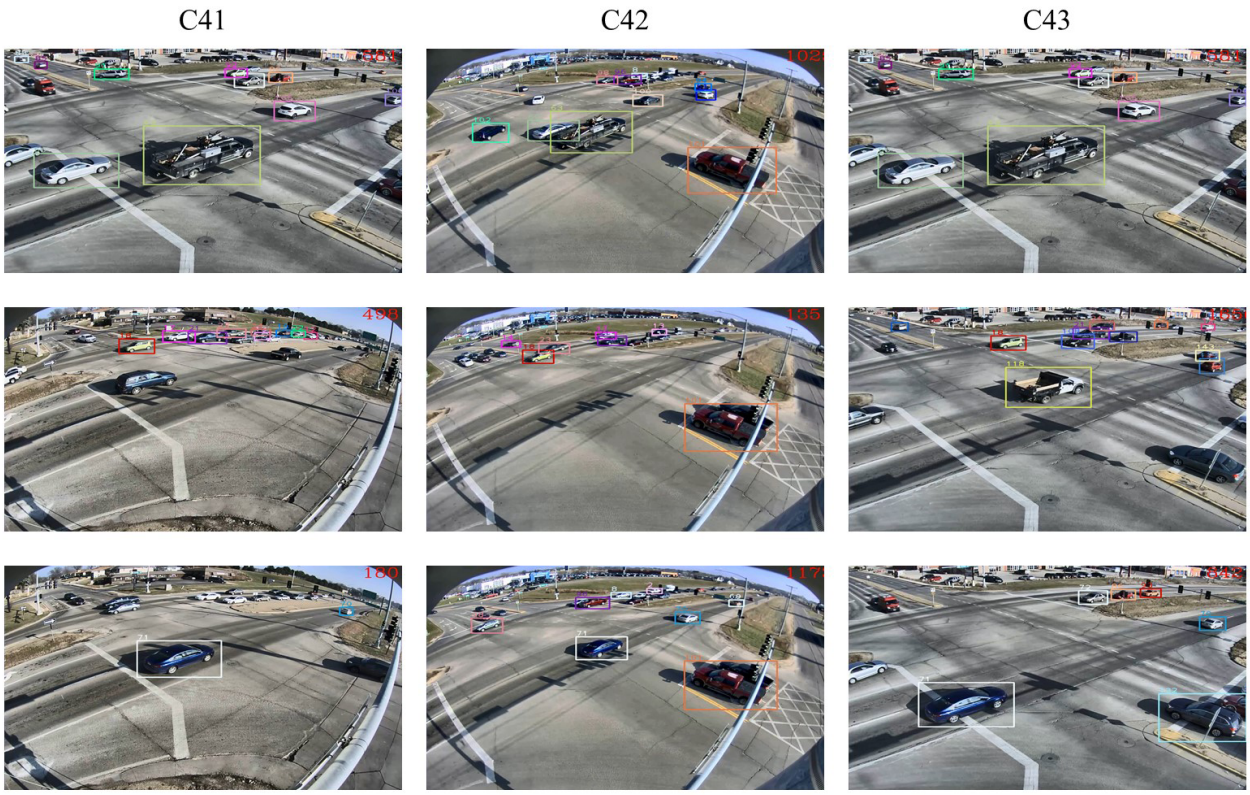

Figure 5. The final matching results of the proposed method on CityFlowV2. From left to right are the C41, C42, and C43 cameras. Furthermore, the same vehicles are marked as the same ID across the different cameras. In the first row, it can seem that a black truck marked as id_{56} and green box has been tracked correctly from C41 to C43 as same as the id_{18} green car in the second row and the id_{71} blue car in the third row.

proposed method recalculates the similarity of the complete tracklets, which can improve the occlusions and lose challenge in the SCT stage. In the tracklets matching stage, the proposed method employs an inter-vehicle information module to improve the matching accuracy of tracklets between multiple cameras. It can significantly avoid feature mismatching between cross-cameras and distinguish vehicles with similar appearances at different times of the same camera. The result shows the effectiveness of the system, which achieves 0.8285 IDF_1 scores on the Track-1 of the 2022 AI City Challenge.

Acknowledgements

This work was supported by "Innovation Chain + Industry Chain" Project of Shenzhen under Grant 20190830020003

References

[1] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 941–951, 2019. 2

[2] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016. 1, 2

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2

[4] Weihua Chen, Lijun Cao, Xiaotang Chen, and Kaiqi Huang. A novel solution for multi-camera object tracking. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 2329–2333. IEEE, 2014. 2

[5] Weihua Chen, Lijun Cao, Xiaotang Chen, and Kaiqi Huang. An equalized global graph model-based approach for multi-camera object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(11):2367–2381, 2016. 2

[6] Xinlei Chen and Abhinav Gupta. An implementation of faster rcnn with study for region sampling. *arXiv preprint arXiv:1702.02138*, 2017. 2

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international*

conference on computer vision, pages 2961–2969, 2017. 2, 3

[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. 3

[9] Lingxiao He, Xingyu Liao, Wu Liu, Xinchen Liu, Peng Cheng, and Tao Mei. Fastreid: A pytorch toolbox for general instance re-identification. *arXiv preprint arXiv:2006.02631*, 2020. 3

[10] Shuting He, Hao Luo, Pichao Wang, Fan Wang, Hao Li, and Wei Jiang. Transreid: Transformer-based object re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15013–15022, 2021. 2

[11] Hung-Min Hsu, Tsung-Wei Huang, Gaoang Wang, Jiarui Cai, Zhichao Lei, and Jenq-Neng Hwang. Multi-camera tracking of vehicles based on deep features re-id and trajectory-based camera link models. In *CVPR Workshops*, pages 416–424, 2019. 1, 5

[12] Philipp Kohl, Andreas Specker, Arne Schumann, and Jurgen Beyerer. The mta dataset for multi-target multi-camera pedestrian tracking by weighted distance aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1042–1043, 2020. 5

[13] Chong Liu, Yuqi Zhang, Hao Luo, Jiasheng Tang, Weihua Chen, Xianzhe Xu, Fan Wang, Hao Li, and Yi-Dong Shen. City-scale multi-camera vehicle tracking guided by crossroad zones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4129–4137, 2021. 1, 3, 5

[14] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 2, 3

[15] Xingan Ma, Kuan Zhu, Haiyun Guo, Jinqiao Wang, Min Huang, and Qinghai Miao. Vehicle re-identification with refined part model. In *2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 603–606. IEEE, 2019. 2

[16] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Xiaodong Yang, Yue Yao, Liang Zheng, Pranamesh Chakraborty, Christian E. Lopez, Anuj Sharma, Qi Feng, Vitaly Ablavsky, and Stan Sclaroff. The 5th AI City Challenge. In *Proc. CVPR Workshops*, pages 4263–4273, Virtual, 2021. 6

[17] Jinlong Peng, Tao Wang, Weiyao Lin, Jian Wang, John See, Shilei Wen, and Erui Ding. Tpm: Multiple object tracking with tracklet-plane matching. *Pattern Recognition*, 107:107480, 2020. 5

[18] 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. 6

[19] Yantao Shen, Tong Xiao, Hongsheng Li, Shuai Yi, and Xiaogang Wang. Learning deep neural networks for vehicle re-id with visual-spatio-temporal path proposals. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1900–1909, 2017. 2

[20] Andreas Specker, Daniel Stadler, Lucas Florin, and Jurgen Beyerer. An occlusion-aware multi-target multi-camera tracking system. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4173–4182, 2021. 4

[21] Rikiya Suzuki, Sumio Fujita, and Tetsuya Sakai. Arc loss: Softmax with additive angular margin for answer retrieval. In *Asia Information Retrieval Symposium*, pages 34–40. Springer, 2019. 2

[22] Zheng Tang, Milind Naphade, Stan Birchfield, Jonathan Tremblay, William Hodge, Ratnesh Kumar, Shuo Wang, and Xiaodong Yang. Pamtri: Pose-aware multi-task learning for vehicle re-identification using highly randomized synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 211–220, 2019. 2

[23] Ultralytics. Yolov5. https://github.com/ultralytics/YOLOv5. 2

[24] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 3

[25] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *European Conference on Computer Vision*, pages 107–122. Springer, 2020. 4

[26] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017. 1, 2

[27] Jin Ye, Xipeng Yang, Shuai Kang, Yue He, Weiming Zhang, Leping Huang, Minyue Jiang, Wei Zhang, Yifeng Shi, Meng Xia, et al. A robust mtmc tracking system for ai-city challenge 2021. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4044–4053, 2021. 3, 5

[28] Fengwei Yu, Wenbo Li, Quanquan Li, Yu Liu, Xiaohua Shi, and Junjie Yan. Poi: Multiple object tracking with high performance detection and appearance feature. In *European Conference on Computer Vision*, pages 36–42. Springer, 2016. 1

[29] Zhedong Zheng, Xiaodong Yang, Zhiding Yu, Liang Zheng, Yi Yang, and Jan Kautz. Joint discriminative and generative learning for person re-identification. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2138–2147, 2019. 2

[30] Yi Zhou and Ling Shao. Aware attentive multi-view inference for vehicle re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6489–6498, 2018. 2