City-Scale Multi-Camera Vehicle Tracking based on Space-Time-Appearance Features

Hui Yao, Zhizhao Duan, Zhen Xie, Jingbo Chen, Xi Wu, Duo Xu, Yutao Gao
Alibaba Group
Hangzhou, Zhejiang Province, China
{beiyu.yh, zhizhao.dzz, xiezhen.xz, jingbo.cjb, lingke.wx, manii.xd}@alibaba-inc.com
yutao.gao@alipay.com

Abstract

Multi-Camera Multi-Vehicle Tracking (MCMVT) is an essential task in the field of city-scale traffic management, which usually consists of three sub-tasks: object detection and re-identification (ReID), single-camera tracking, cross-camera trajectory association. Compared with existing methods, two challenges are considered and addressed in this paper: (1) low-confidence objects could be missed without extra data annotation, (2) precise association of trajectories from different cameras is affected by multiple factors. For the first challenge, a cascaded tracking method based on detection, appearance features and trajectory interpolation is proposed, exploiting potential real targets in low-confidence objects to improve detection and identification recall. For the second challenge, space, time and appearance features are proposed to be the most crucial factors for trajectory association, so a zone-gate and time-decay based matching mechanism is proposed to adjust original appearance matrix to link tracklets more precisely from different cameras. Extensive experimental results validate the effectiveness of the proposed innovative technologies.

1. Introduction

With the blossom of autonomous driving, the demand for the digital traffic management platform and the development of urban intelligence is becoming more and more prominent. Among them, Multi-Camera Multi-Vehicle Tracking (MCMVT) is one of the most important perception tasks. It aims at tracking the cross-camera trajectories of multiple vehicles. Fig. 1 shows the complete tracking chain of a certain vehicle under different cameras.

The MCMVT task usually consists of three sub-tasks: object detection and ReID, single-camera tracking, cross-camera trajectory association. The goal of the first two sub-tasks is to identify the trajectory of each target seen in single cameras with a tracking-by-detection manner. Outstanding detection and tracking results have been achieved in the computer vision community with accurate data annotation. However, city-scale annotation is heavy and expensive. Thus, fully utilizing pretrained detection models trained on public datasets is necessary. The challenge is that the performance of public models is unstable in unseen scenes. Many objects would be filtered by a common con-

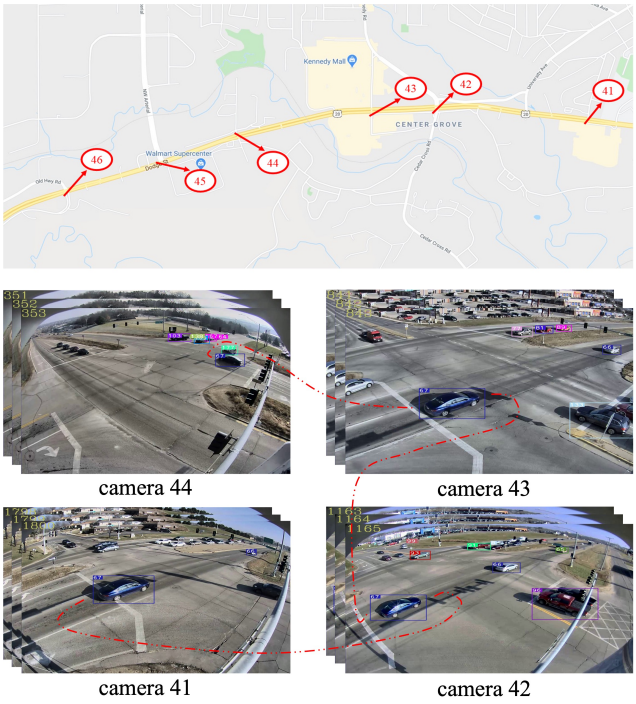

Figure 1. Illustration of MCMVT. The top is the camera position distribution map in the urban scene, arrows represent the directions that cameras are facing; the bottom is the tracking trajectory of vehicle 67 under different cameras.

fidence threshold, because the confidences of objects given by public models are relatively low and indistinguishable. When accurate targets in single cameras are obtained, the last sub-task aims to concatenate multiple trajectories of each single target. The challenge is that precise association of trajectories from different cameras is affected by multiple factors. To identify a unique target, we propose three constraints that need to be considered: space, time and appearance. How to design an efficient mechanism to take all of these constraints into consideration and improve the matching precision is to be addressed.

For the first challenge, inspired by ByteTrack [39], which uses low-confidence objects to assist in tracking, we propose a cascaded association strategy including ReID and box to match high-confidence objects, and mine the real targets from low-confidence objects using a strict IOU based matching method. Thus, low-confidence objects that may be filtered could be reserved, contributing to high target recall. Furthermore, trajectory interpolation is used to connect fragmented tracklets.

To address the second challenge, on the basis of the original matching matrix based on ReID feature similarity, we introduce the spatial-temporal constraint information to optimize the clustering matrix by establishing the zone-gate mechanism and time-decay filtering strategy. In this way, high precision could be achieved.

The contributions of this paper could be summarized as follows:

1). Propose a cascaded tracking method based on detection-appearance features and trajectory interpolation to improve target recall, without extra data annotation.

2). Propose a zone-gate and time-decay based matching mechanism to fully utilize space-time-appearance features, contributing to precise trajectory association

3). Achieve an IDF1 score of **0.8371**, which ranked the **3rd place** on the public leaderboard of Track 1, AI City Chalenge 2022.

2. Related Work

2.1. Object Detection and Re-identification

Object Detection. Object detection, as the foundational task for computer vision, is usually classified into two categories. One is the two-stage detectors and the representative works are R-CNN series [4, 5, 18, 26]. The other is the one-stage methods which become well-known owing to the YOLO series [2, 23–25]. Besides, it is also possible to categorize the detection methods into the anchor-based [15, 27] and the anchor-free [32, 44]. Recently, the transformer based detectors such as DETR [46] and Swin Transformer [17] are booming, pointing out a new development direction of object detection.

Re-identification. As an important component of MCMVT, Vehicle ReID can not only assist in the process of intra-camera tracking, but also plays an irreplaceable role in inter-camera tracking. Within the research field of ReID, many works focus on how to design efficient loss functions, such as triplet loss [8], circle loss [28], etc. Also, weakly supervised detection [45] and synthetic data [41] prove to be beneficial by expanding the training data and reducing the receptive field. He *et al.* [7] generate the pseudo labels of test samples using Identity Mining Method, then fine-tune the model on the test domain to improve the performance. Moreover, Some post-processing techniques are proposed to further optimize the identification resutls, such as model ensemble [19], re-ranking [42], image-to-track retrieval [16], etc. Luo *et al.* [19] who won the 1st place in Track 2 of AI City Challenge 2021 [22], prove that the tricks of the person ReID strong baseline [20, 21] also have a significant performance on Vehicle ReID. Due to the impressive effect, we consider retraining the model based on this method and use it as our ReID feature extractor.

2.2. Multiple Object Tracking

Multiple object tracking (MOT) is one of the dominant topics for autonomous driving. Its goal is to associate the same targets in the video sequence, and connect each of them into a tracking chain with a unique identity. The classical tracking methods are mainly based on the probability theory, especiallythe Kalman [34] and particle filters [6] lay a good mathematical foundation for tracking problems. Based on the Kalman filter, SORT [1] is widely used for tracking problems. In recent years, CNN-based tracking methods have become popular. DeepSort [35], as the representative method of separated detection and tracking (SDT), uses a stand-alone ReID model to reduce ID switch. JDE [12,33,36,40] adopts a weight-shared CNN for object detection and ReID feature extraction, which is considered a means of joint detection and tracking (JDT). CenterTrack [43] detects targets with the point-based heatmap and then tracks them with a simple greedy match. In addition, transformer-based tracking methods are also applied successfully recently. TransMOT [3] applies a graph transformer for the spatial-temporal modeling. TransCenter [37] is the first MOT architecture to predict the target heatmap based on the transformer.

2.3. Multi-Camera Multiple Object Tracking

Based on the results of aforementioned tasks, the mission of multi-camera multi-target tracking is to connect the tracking chains under different cameras in series. For this purpose, there has been a lot of excellent work [9–11, 14, 30, 31, 38]. For example, [14] proposes Direction Based Temporal Mask (DBTM) to reduce search space of matching. [38] pre-defines enter/exit areas and a time window to

constrain the clustering domain. However, the filtering conditions of the above methods are relatively rough, resulting in some limitations for further performance improvement.

3. Method

3.1. Vehicle Detection and Re-identification

As separate computer vision tasks, detection and re-identification are the basic work of tracking, and there are many mature solutions. So this paper will follow existing methods to generate detection boxes and ReID features.

For the detection task, we use the YOLOv5x[1] model which is pre-trained on the COCO dataset [13]. As for ReID task, we retrain the models as our ReID feature extractor following the work proposed by Luo *et al.* [19], and configures are set to the same. The loss function we use can be formulated as:

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

where L_c and L_t denote softmax cross-entropy loss [29] and triplet loss respectively. α is a balance weight, set to 1 by default.

3.2. Single Camera Tracking

Considering the performance and robustness, we devide the single-camera tracking problem into three parts, which are box detection, feature embedding and target association. We use the methods described above for the first two parts. ByteTrack [39] is a state-of-art method which mine the real target from the low confidence box sufficiently to improve the tracking performance. Therefore, we take the ByteTrack's tracker management parts and association strategy as reference. Beyond that, we find that ReID features are of great importance, which will affect the performance of MCMVT. The way only using IOU will cause false matching, which will lead to a confusion in the appearance representation of the tracking chain. So we use a cascaded matching strategy.

Specifically, we first associate the high confidence box with ReID features, then the unmatched trackers are associated with boxes by IOU. Lastly, we just match the low confidence boxes with IOU to enhance the stability of tracking. The Kalman filter is used for track updating.

3.3. Multi-Camera Vehicle Tracking

Different from the purpose of MOT, MCMVT needs to match the inter-camera tracklets to obtain the complete tracking chain of each target. A general solution is to use ReID features to associate vehicle candidates under different cameras. However, due to the similar appearance, the ambiguity of the cropped image and the numerous candidates in the gallery, directly using appearance features for ID clustering faces many challenges. Hence, based on the Sub-Clustering in Adjacent Cameras (SCAC) proposed by Liu *et al.* [14], we cluster each pair of adjacent cameras separately, and then extend the clustering results to the entire scene chain.

First of all, to improve the robustness and representation ability of features, for tracklet i under camera N, we calculate the average features of all frames and use it as the trajectory feature $f_{t_i^N}$ for cross-camera matching. Then a cosine similarity matching matrix M between camera N and $N+1$ can be established based on the above ReID features, i.e.,

$$M = \begin{bmatrix} m(t_1^N, t_1^{N+1}) & \cdots & m(t_1^N, t_j^{N+1}) & \cdots & m(t_1^N, t_J^{N+1}) \\ \vdots & & \vdots & & \vdots \\ m(t_i^N, t_1^{N+1}) & \cdots & m(t_i^N, t_j^{N+1}) & \cdots & m(t_i^N, t_J^{N+1}) \\ \vdots & & \vdots & & \vdots \\ m(t_I^N, t_1^{N+1}) & \cdots & m(t_I^N, t_j^{N+1}) & \cdots & m(t_I^N, t_J^{N+1}) \end{bmatrix} \tag{2}$$

where I denotes the total number of tracklets in camera N, J denotes the total number of tracklets in camera $N+1$. N is the camera ID, ranging from 41 to 45, and

$$m(t_i^N, t_j^{N+1}) = \cos(f_{t_i^N}, f_{t_j^{N+1}}), \tag{3}$$

means cosine similarity between tracklet t_i^N and t_j^{N+1}.

Further, we propose a more advanced spatial-temporal constraint method to dynamically regulate the matching matrix, which can shrink the matching space and reduce the association difficulty. This module includes two parts, namely the zone-gate mechanism based on bi-directional main road partition, and matching probability adjustment based on the temporal decay curve.

3.3.1 Zone-Gate Mechanism

From Fig. 2, it can be seen that for intersection (camera) N, there are a total of 12 driving routes for vehicles. For all of these driving routes, if the tracklet under this camera needs to associate with the next intersection $N+1$, it must pass through zone 3 and 4; similarly, if it needs to associate with the previous intersection $N-1$, it must go through zone 7 and 8. With the simplified definition, we only focus on the four bi-directional zones (3, 4, 7 and 8) on the main road, regardless of whether it comes from or goes to the bypasses. Following the above-mentioned principles, we divided the zones of different intersections as shown in Fig. 3.

For tracklet t_i^N and t_j^{N+1}, we construct the following gate function:

$$g(t_i^N, t_j^{N+1}) = \begin{cases} 1 & \text{if } 3 \in Z_{t_i^N}, 8 \in Z_{t_j^{N+1}} \\ 1 & \text{if } 4 \in Z_{t_i^N}, 7 \in Z_{t_j^{N+1}} \\ 0 & \text{else} \end{cases} \tag{4}$$

[1] https://github.com/ultralytics/yolov5

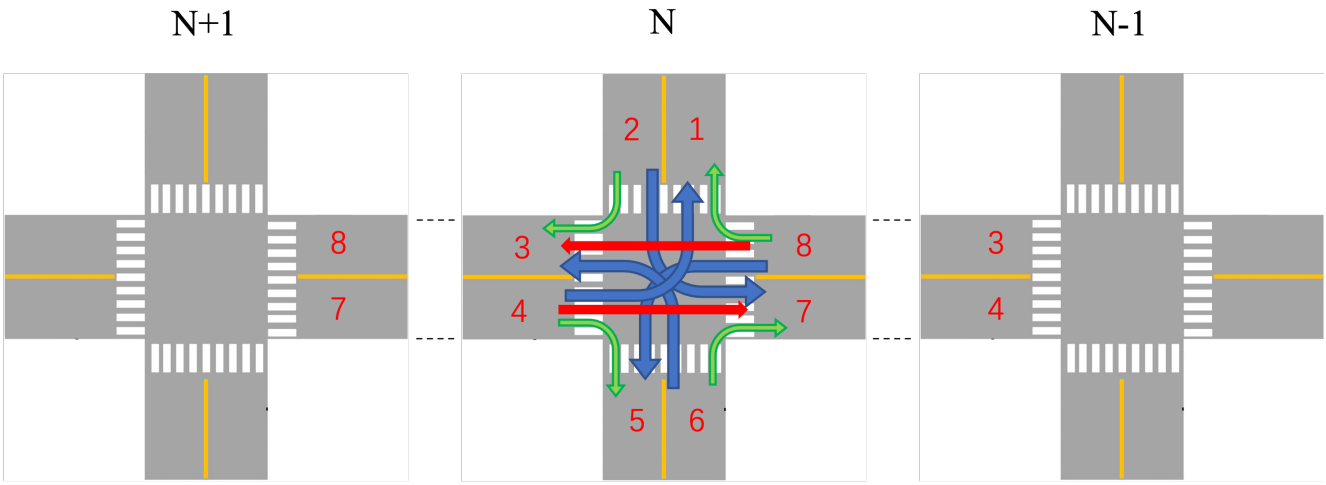

Figure 2. The analysis diagram of zones matching of intersections. The numbers represent zone IDs, arrows represent all possible driving routes at intersection N.

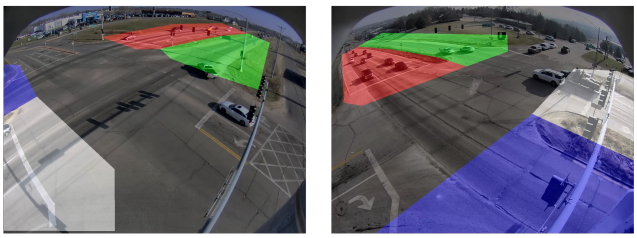

Figure 3. Illustration of split zones in bi-directional main road. Red, green, blue and white correspond to zones 8, 7, 3 and 4 respectively in Fig. 2.

where Z_t denotes the list of zones that trajectory t traverses. As shown in Eq. (4), the gate will be switched **ON** only when t_i^N passes through zone 3 and t_j^{N+1} passes through zone 8, or t_i^N passes through zone 4 and t_j^{N+1} passes through zone 7.

With the gate function $g(\cdot)$, we obtain the space-aware mask matrix G and filtered similarity matrix $M^{'}$:

$$M^{'} = G \cdot M \tag{5}$$

which greatly reduces the matching space.

3.3.2 Time-Decay Strategy

Empirically, humans often consider the elapsed time as an important factor when they try to identify a target across different cameras. Inspired by this, this paper introduces the time variable, to adjust the matching probability between different vehicles. Through this method, the performance of MCMVT is further improved.

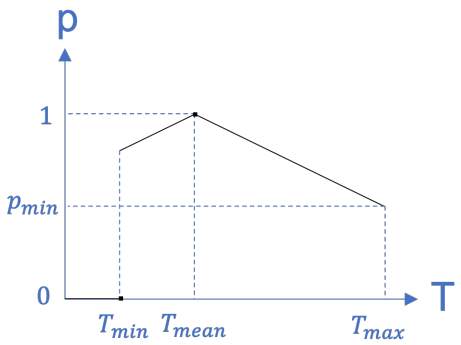

Figure 4. The time decay curve, the horizontal axis represents elapsed time, and the vertical axis is the matching probability.

Specifically, we obtain the actual distance between adjacent intersections according to the GPS position of each camera, and then set the average elapsed time of each road based on experience. Our temporal probability curve is shown as Eq. (6) and Fig. 4.

$$p = \begin{cases} 0, & \text{if } T < T_{min} \\ 1 + (1 - p_{min})\frac{T - T_{mean}}{T_{max} - T_{mean}}, & \text{if } T_{min} \le T < T_{mean} \\ 1 - (1 - p_{min})\frac{T - T_{mean}}{T_{max} - T_{mean}}, & \text{if } T \ge T_{mean} \end{cases} \tag{6}$$

The matching probability decays linearly from point $(T_{mean}, 1)$ to both ends, whose decay slope is determined by $(T_{mean}, 1)$ and (T_{max}, p_{min}). Theoretically, a vehicle can stay between two intersections for a long time, but cannot appear at the next intersection in a very short time. Therefore, when the time reaches the maximum threshold T_{max}, the matching probability drops to p_{min}; when the

time is less than T_{min}, the matching probability is truncated to 0.

It is worth noting that, the elapsed time T is related to the driving direction of the main road, i.e.:

$$T = \begin{cases} T_{j,s}^{N+1} - T_{i,e}^{N}, & \text{if } 3 \in Z_{t_i^N}, 8 \in Z_{t_j^{N+1}} \\ T_{i,s}^{N} - T_{j,e}^{N+1}, & \text{if } 4 \in Z_{t_i^N}, 7 \in Z_{t_j^{N+1}} \end{cases} \tag{7}$$

where T_s means the start time, T_e means the end time.

Then we generate a temporal decay matrix P and apply it to the following equation,

$$M^{''} = G \cdot M^{'} \tag{8}$$

to get the final spatial-temporal constrained matching matrix $M^{''}$.

3.4. Trajectory Post-Processing

Despite using the cascaded tracking framework, the final results still have fragmented trajectories, which inevitably leads to true targets missing. To further improve the recall rate, we design an interpolated post-processing module for interrupted trajectories.

Assume that the time series in which trajectory t exists are $[1, 2, ..., T_1]$, $[T_2, T_2 + 1, ..., T_n]$, where $T_2 > T_1$. It is reasonable to believe that in the interval $[T_1, T_2]$, the trajectory is temporarily lost due to occlusion or other reasons, so we use linear interpolation to complete the tracking box B at time T, which belongs to $[T_1, T_2]$. The interpolation formula is as follows:

$$B_T = B_{T_1} + (B_{T_2} - B_{T_1}) \frac{T - T_1}{T_2 - T_1} \tag{9}$$

Fig. 5 shows the visual tracking results before and after post-processing. It can be seen that this module plays an important role in maintaining the coherence of the tracking trajectory and improving the detection recall and identification recall.

4. Experiments

4.1. Datasets

We participated in Track 1, which takes CityFlowV2[2] as the dataset (name it AIC22-T1). The dataset contains 3.58 hours (215.03 minutes) of videos collected from 46 cameras spanning 16 intersections in a mid-sized U.S. city. The dataset is divided into 6 scenarios. Scenario S06 is used for testing.

For the ReID module, we used the benchmark of AI City Challenge 2021 Track 2 (name it AIC21-T2), which consists of a real-world dataset and an additional synthetic dataset[3].

[2]https://www.aicitychallenge.org/2022-track1-download/

[3]After consulting the official by email, it is confirmed that the dataset can be used.

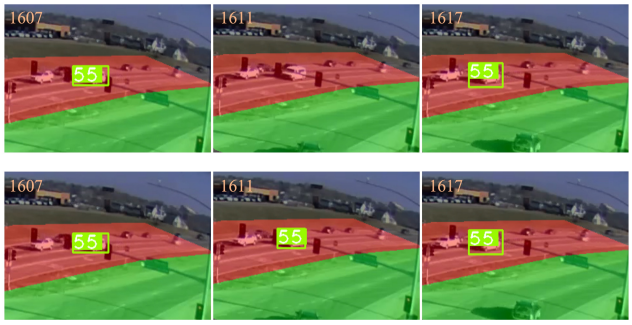

Figure 5. Illustration of tracking visualization before and after trajectory interpolation under the C042 camera. Top: before interpolation; Bottom: after interpolation.

4.2. Evaluation Metric

For track 1: City-Scale Multi-Camera Vehicle Tracking, the evaluation metric is $IDF1$, which measures the ratio of correctly identified detections over the ground-truth and the average number of calculated detections. Its specific calculation formula is:

$$IDF1 = \frac{2 \cdot IDTP}{2 \cdot IDTP + IDFP + IDFN} \tag{10}$$

where $IDTP$ is the number of true positive identities, $IDFP$ is the number of false positive identities and $IDFN$ represents the number of false negative identities.

In addition, reference mcetrics such as IDP, IDR, $Precision$, $Recall$ are also provided, but are not used for ranking.

4.3. Implementation Details

Our tracking system runs on a PC with four Tesla V100 (32 GB) GPUs, the deep learning framework is PyTorch 1.8. Following [14], in the vehicle detection part, we set the image size to 1280 and the confidence threshold to 0.1 to predict each frame of all test videos. Then the cropped images are resized to $(384, 384)$ to extract ReID features, whose dimension is 2048 dimensions. In the single-camera tracking stage, the high confidence is set to 0.4, while the low confidence is set to 0.1. Besides, the IOU cost confidence for the high confidence box association is set to 0.8 and the IOU cost confidence for the low confidence box association is 0.5. In the inter-camera matching stage, the temporal decay probability p_{min} is set to 0.7, the average elapsed time T_{mean} between adjacent cameras is shown in Tab. 1.

4.4. Quantitative and Qualitative Evaluation

4.4.1 Quantitative Results

Based on the method described in Sec. 3, our team (TAG) tested scenario 6 in the CityFlowV2 dataset, the generated results can get an IDF1 score of **0.8371**, which ranked the

Camera Pairs	(041, 042)	(042, 043)	(043, 044)	(044, 045)	(045, 046)
Average Elapsed Time (s)	82	32	57	33	50

Table 1. The average elapsed time of different adjacent camera pairs

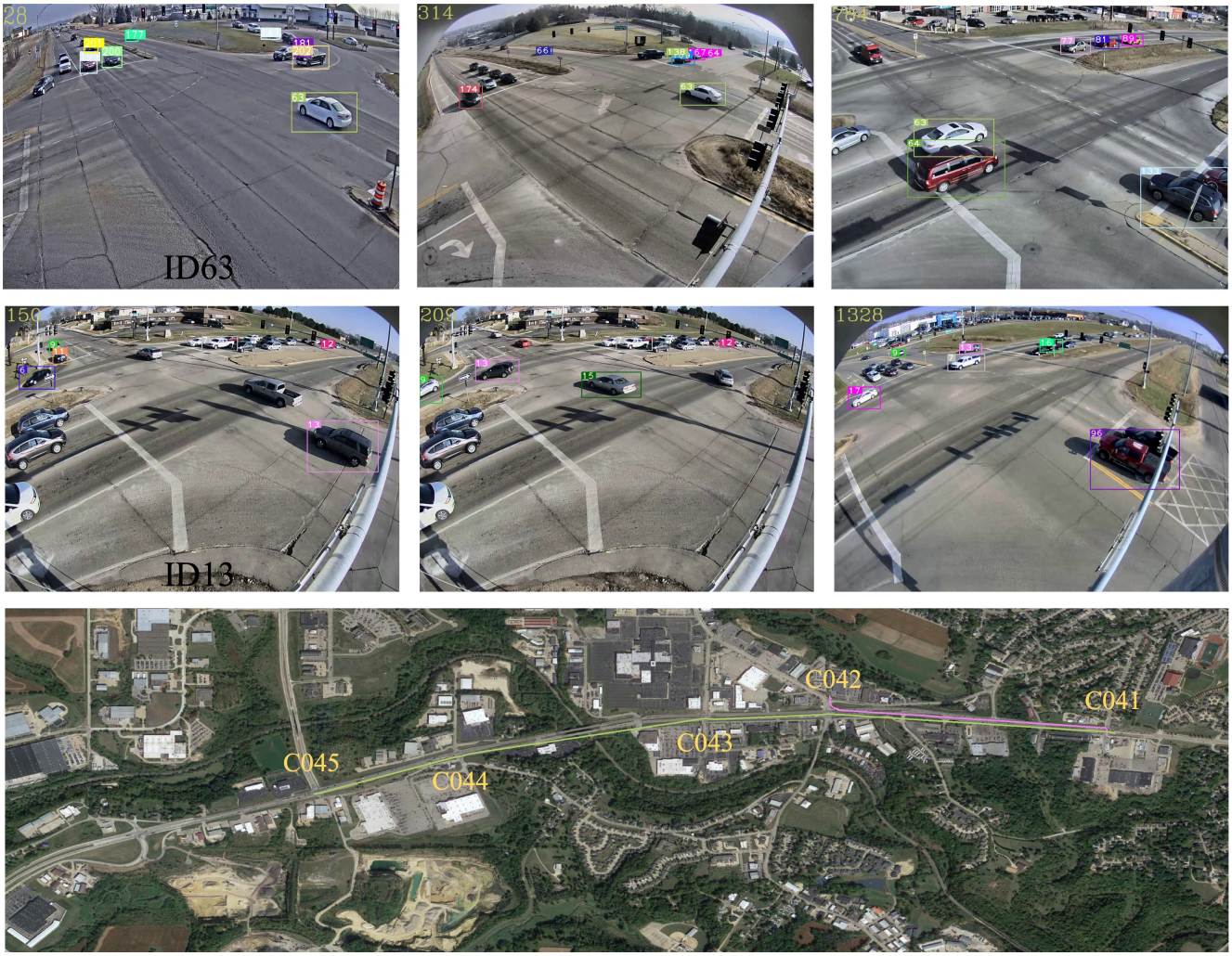

Figure 6. Top: visualization of multi-vehicle tracking across cameras, the first row is the results of ID 63 and the second is ID 13; Bottom: projection of the above two vehicle tracking results on GIS map.

3rd place on the public leaderboard. The overall results of the leaderboard are shown in Tab. 2.

4.4.2 Qualitative Results

In order to illustrate the effect of vehicle tracking more intuitively, we show the tracking positions of vehicles 13 and 63 at different times in Fig. 6. It can be seen that even when the vehicle 13 appears in the field of view with a tiny target, it can be well matched and tracked. Meanwhile, it gives the results of projecting the trajectories of both vehicles onto the GIS map using matrix transformation. With such tools, traffic flow analysis and road labeling can be more easily performed.

4.5. Ablation Study

To analyze the influence of different datasets on the ReID model, comparative experiments are conducted on the dataset of AIC21-T2 and AIC22-T1. As shown in Tab. 3, the performance of the model trained on AIC21-T2 is better. Through visualization, we found that the reason why AIC22-T1 performs not well is the lack of data diversity.

Rank	Team ID	Team Name	IDF1 Score
1	28	matcher	0.8486
2	59	BOE	0.8437
3	**37**	**TAG**	**0.8371**
4	50	FraunhoferlOSB	0.8348
5	70	appolo	0.8251
6	36	Li-Chen-Yi	0.8218
7	10	Terminus-AI	0.8171
8	118	FourBeauties	0.8166
9	110	Orange Peel	0.8140
10	94	SKKU Automation Lab	0.8129

Table 2. The public leaderboard of track 1, our team takes the third place.

backbone	IDF1	IDP	IDR
ResNeXt101-IBN-a*	0.7813	0.8498	0.7230
ResNeXt101-IBN-a	0.7851	0.8481	0.7309

Table 3. ResNeXt101-IBN-a* is trained on AIC22-T1 and ResNeXt101-IBN-a is trained on AIC21-T2.

Module	IDF1	IDP	IDR	Precision	Recall
baseline	0.8057	0.8480	0.7675	0.8903	0.8059
+Zone-Gate	0.8095	0.8710	0.7561	0.8979	0.7794
+Time Based Decay	0.8151	0.8893	0.7523	0.9179	0.7765
+ByteTrack	0.8235	0.8827	0.7718	0.9024	0.7848
+ReID-ByteTrack	0.8344	**0.9008**	0.7771	**0.9193**	0.7898
+Post-Processing	**0.8371**	0.8878	**0.7918**	0.9046	**0.8069**

Table 4. Ablation experiments on each incremental module.

One vehicle instance in AIC22-T1 has fewer views than that in AIC21-T2. Therefore, we adopt the model trained on AIC21-T2 as the ReID feature extractor.

Tab. 4 lists the ablation experiment results about adding different modules. Among them, the baseline represents solutions proposed by Liu *et al.* [14]. It can be seen that the zone-gate mechanism based on the bi-directional main road division and the time-decay matching probability adjustment further improve the IDP score. It is mainly due to the constraints of the spatial-temporal information on the original appearance matrix, which reduce the search space and the matching difficulty. Meanwhile, the introduction of the optimized ByteTrack framework can mine the useful information of the low-confidence boxes and promote the tracking performance; the last post-processing module greatly improves IDR score by linking interrupted trajectories, although it causes a decrease in IDP.

5. Conclusion

In this paper, we realize a complete MCMVT tracking scheme, including detection, ReID, single-camera tracking and multi-camera matching. In particular, to further enhance the tracking performance, we propose more advanced solutions and strategies. First, we design a cascaded tracking method to relieve the problem of true targets detection and tracking missing and improve the identification recall. Secondly, the zone-gate and time-decay based mechanism is proposed to optimize the matching space, ensuring a high identification precision. Finally, we use trajectory interpolation as the post-processing unit to keep tracked trajectories coherent, and achieve an IDF1 score of **0.8371** in the track 1, which ranked the **3rd place** on the public leaderboard.

References

[1] 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. IEEE, 2016. 2

[2] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 2

[3] Peng Chu, Jiang Wang, Quanzeng You, Haibin Ling, and Zicheng Liu. Transmot: Spatial-temporal graph transformer for multiple object tracking. *arXiv preprint arXiv:2104.00194*, 2021. 2

[4] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2

[5] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2

[6] Fredrik Gustafsson, Fredrik Gunnarsson, Niclas Bergman, Urban Forssell, Jonas Jansson, Rickard Karlsson, and P-J Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Transactions on signal processing*, 50(2):425–437, 2002. 2

[7] Shuting He, Hao Luo, Weihua Chen, Miao Zhang, Yuqi Zhang, Fan Wang, Hao Li, and Wei Jiang. Multi-domain learning and identity mining for vehicle re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 582–583, 2020. 2

[8] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 2

[9] 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. 2

[10] Hung-Min Hsu, Yizhou Wang, and Jenq-Neng Hwang. Traffic-aware multi-camera tracking of vehicles based on reid and camera link model. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 964–972, 2020. 2

[11] 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. 2

[12] Wei Li, Yuanjun Xiong, Shuo Yang, Mingze Xu, Yongxin Wang, and Wei Xia. Semi-tcl: Semi-supervised track contrastive representation learning. *arXiv preprint arXiv:2107.02396*, 2021. 2

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 3

[14] 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. 2, 3, 5, 7

[15] 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

[16] Xinchen Liu, Wu Liu, Tao Mei, and Huadong Ma. A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In *European conference on computer vision*, pages 869–884. Springer, 2016. 2

[17] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 2

[18] Xin Lu, Buyu Li, Yuxin Yue, Quanquan Li, and Junjie Yan. Grid r-cnn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7363–7372, 2019. 2

[19] Hao Luo, Weihua Chen, Xianzhe Xu, Jianyang Gu, Yuqi Zhang, Chong Liu, Yiqi Jiang, Shuting He, Fan Wang, and Hao Li. An empirical study of vehicle re-identification on the ai city challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4095–4102, 2021. 2, 3

[20] Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. Bag of tricks and a strong baseline for deep person re-identification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019. 2

[21] Hao Luo, Wei Jiang, Youzhi Gu, Fuxu Liu, Xingyu Liao, Shenqi Lai, and Jianyang Gu. A strong baseline and batch normalization neck for deep person re-identification. *IEEE Transactions on Multimedia*, 22(10):2597–2609, 2019. 2

[22] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2021. 2

[23] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2

[24] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 2

[25] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 2

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 2

[27] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 2

[28] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6398–6407, 2020. 2

[29] 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. 3

[30] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and Jenq-Neng Hwang. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8797–8806, 2019. 2

[31] Zheng Tang, Gaoang Wang, Hao Xiao, Aotian Zheng, and Jenq-Neng Hwang. Single-camera and inter-camera vehicle tracking and 3d speed estimation based on fusion of visual and semantic features. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 108–115, 2018. 2

[32] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019. 2

[33] 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. 2

[34] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995. 2

[35] 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. 2

[36] Mingze Xu, Chenyou Fan, Yuchen Wang, Michael S Ryoo, and David J Crandall. Joint person segmentation and identification in synchronized first-and third-person videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 637–652, 2018. 2

[37] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. Transcenter: Transformers with dense queries for multiple-object tracking. *arXiv preprint arXiv:2103.15145*, 2021. 2

[38] 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. 2

[39] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. *arXiv preprint arXiv:2110.06864*, 2021. 2, 3

[40] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129(11):3069–3087, 2021. 2

[41] Zhedong Zheng, Minyue Jiang, Zhigang Wang, Jian Wang, Zechen Bai, Xuanmeng Zhang, Xin Yu, Xiao Tan, Yi Yang, Shilei Wen, et al. Going beyond real data: A robust visual representation for vehicle re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 598–599, 2020. 2

[42] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1318–1327, 2017. 2

[43] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision*, pages 474–490. Springer, 2020. 2

[44] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2

[45] Xiangyu Zhu, Zhenbo Luo, Pei Fu, and Xiang Ji. Voc-reid: Vehicle re-identification based on vehicle-orientation-camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 602–603, 2020. 2

[46] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 2