CROSS-CAMERA MULTI-OBJECT TRACKING SYSTEM

Luu Van Tin, Ngo Duc Thang, Nguyen Quang Sang，林垣志
Department of Computer Science, National Yang Ming Chiao Tung University, Taiwan

ABSTRACT

Multi-camera multi-vehicle tracking is a fundamental task for intelligent traffic systems. The goal of our participation in the AICUP 2024 competition is to detect and identify the same vehicles across different cameras. In this paper, we propose a new MCMT system composed of four parts: (1) adapting data to varying conditions based on image quality detectors, (2) improving state-of-the-art UCMCTrack models by adding appearance features, with (3) automatically extracting the homography matrix help to convert vehicle motion from the image plane to the ground plane, and (4) separating query groups based on specific conditions to refine the tracking process. The proposed system achieves promising results, ranking 6th in the Cross-Camera Multi-Target Vehicle Tracking Competition with a final score of 1.3439, which is the sum of the IDF1 score and MOTA. The source code is available at https://github.com/tinery/AICUP-2024-competition.

Keywords *Multi-Target Vehicle Tracking, Re-identification, Tracking-by-detection, AICUP.*

1. INTRODUCTION

In recent years, surveillance camera systems have been widely used on roads, due to security needs, crime prevention and in case of accidents. Manual detection to identify an object when it moves out of the view of a single camera is facing many difficulties, takes a lot of time and consumes human resources. Therefore, the Cross-Camera Multi-object Tracking system using AI is gradually being developed by researchers to solve the mentioned problems. It is also called Multi-Camera Multi Vehicle Tracking (MCMT) and is mentioned at the AI city challenge [1], [2], [3]. The purpose of this system is to track the Vehicle target on multiple cross cameras as shown in Figure 1(a), the vehicles are tracked, we can easily see the yellow vehicle being tracked. The MCMT task usually consists of three sub-tasks: Vehicle detection and ReID, Multiple-Target Single-Camera Tracking (MSCT), Cross-camera trajectory association. The overall process can be succinctly outlined as follows: Initially, the Vehicle detection module identifies vehicle positions and categories within frames, extracting features using ReID. Subsequently, leveraging the vehicle positions and acquired features, the MSCT module generates potential trajectories for each camera individually.

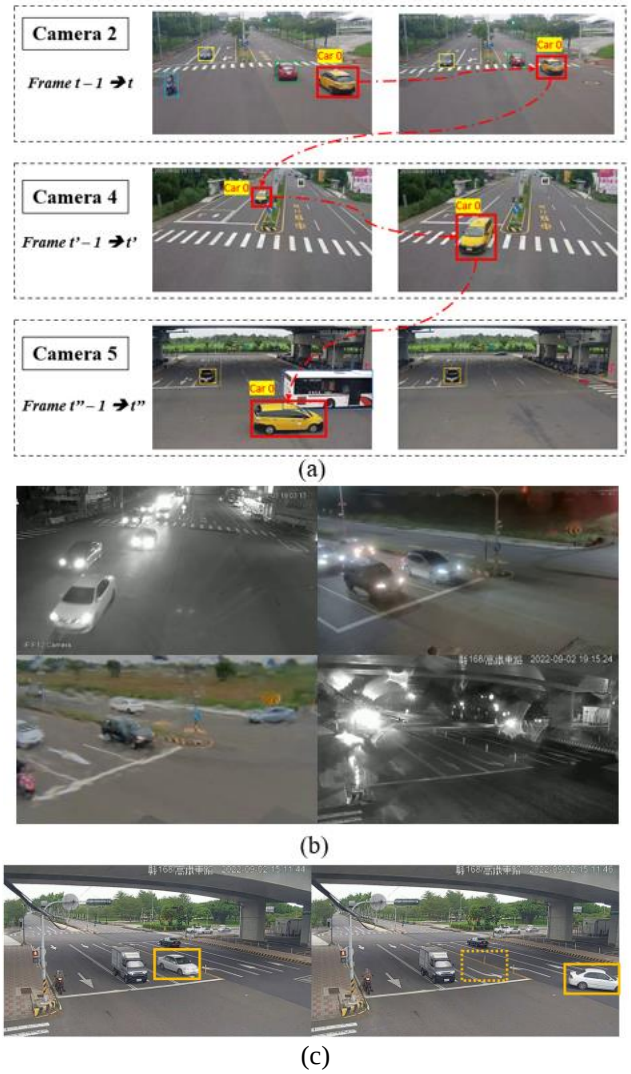


Figure 1. (a) Illustration of Cross-camera Multi-Object Vehicle Tracking. (b) Top-left and top right: Color and colorless images are extracted at the same time of day from 2 different cameras. Bottom: Images are affected by different conditions. (c) IoU = 0 of an object in two adjacent frames

Finally, the Cross-camera trajectory association module cross-references these trajectory candidates across various cameras to establish connections between targets and global identities.

In the AI city challenge [1-3] competition in recent years, many researchers have also solved the MCMT

problem [4-8]. There have been a number of methods to deal with occlusion problems such as [7], [8] and most competition challenges such as environmental conditions, parallel vehicles, etc. Despite achieving the top performance of the current MCMT model, there are still some challenges in this competition that these methods have not been able to solve. Among the challenges are the camera recording at night, switching to IR mode without color information, rainy weather conditions along with the glare effect from vehicle headlights also cause very bad data. Therefore, matching between cameras with color information and without color information becomes difficult, all shown in Figure 1b. In addition, in this problem the data is sampled at 1 frame per second, so the IOUs of the object between two consecutive frames are very much zero, shown in Figure 1c. This also greatly hinders IOU-based MSCT methods such as [9-11], so in this problem we propose to use IOU-free methods [12].

To solve the above problems, we propose a new MCMT system that adapts data in different good and bad conditions based on image quality detectors. In bad conditions where the appearance feature becomes noisy, we add weights so that the model focuses more on the motion feature. Besides, we use the IOU-free UCMCTrack method [12], one of the powerful methods reaching the top state of the art of the MSCT task. The original UCMCTrack method does not have an appearance feature and finding the Homography matrix requires manual operations, so we propose an automatic Homography predictor and add a weighted appearance feature to improve it under good data conditions, all are shown in figure 2.

The contributions of this article are summarized below:

- Proposal to improve the UCMCTrack method by adding appearance features.
- Proposing a method to automatically extract the Homography matrix.
- Proposing a system that adapts to different environmental conditions based on image quality and the addition of condition-based query clustering methods.

2. RELATED WORK

2.1. Object detection

Object detection is one of the fundamental tasks of computer vision and image processing, involving identifying specific objects in digital images and videos, employing AI techniques to achieve accurate results. There are two main categories in this task: one-stage and two-stage object detection methods. This task can be also further classified into 2 kinds of design: CNN-based and transformer-based object detectors, distinguished by the technique and architecture of backbones.

SSD [13] and YOLO [14] represent one-stage detectors, prioritizing real-time processing by compromising some accuracy. On the other hand, Faster R-CNN [15] and Cascade R-CNN [16], belonging to the two-stage detector category, offer high accuracy and flexibility at the expense of time efficiency. Another branch comprises transformer-based designs, such as DETR [17] and Swim Transformer [18], which process the input images as sequences of patches and leverage self-attention mechanisms to capture long-range correlation. One of the current SOTA models that responds in real time is the YOLOv8 version [36].

2.2. Multi-Target Single-Camera Tracking

Multi-Target Single-Camera (MTSC) trackers are divided by tracking-by-detection methods [19-21] and joint-detection-tracking methods [22-24].

Tracking-by-detection techniques locate detection boxes and use motion and visual cues to connect them. These techniques have been widely used in MTSC activities. Simple Online and Realtime Tracking (SORT) [10] represents an implementation of the Multiple Object Tracking (MOT) framework that employs the Kalman filter algorithm to track multiple targets based on motion, utilizing observations from deep detection models. DeepSORT [9] extends this approach by integrating deep visual features for object association within the SORT framework. It decreases ID switches while maintaining computational complexity.

Motion prediction or appearance embedding are included into detection frameworks [25, 26] in a number of joint-detection-tracking techniques. These joint trackers achieve comparable performance while keeping computational costs low. However, they face the challenge of competition among different components, which limits the upper bound of tracking performance.

2.3. Re-identification

Re-identification (ReID) is a crucial element within multi-camera traffic flow systems, tasked with identifying the same vehicle across various camera captures [27]. Most ReID research leverage CNN-based techniques to develop discriminative vehicle representations from raw pixel data. Siamese networks [26] and triplet loss functions [28] are commonly utilized for learning embeddings that can accurately encode vehicle identities while remaining robust to irrelevant factors. However, these approaches face performance constraints stemming from the scarcity of labeled data, which incurs significant costs for data annotation.

To solve these problems, there are several works focusing on designing ReID-specific architectures. Zhou et al. [29] proposed a method for extracting view-invariant features by transforming single-view features into multi-view features and leveraging GAN model to synthesize

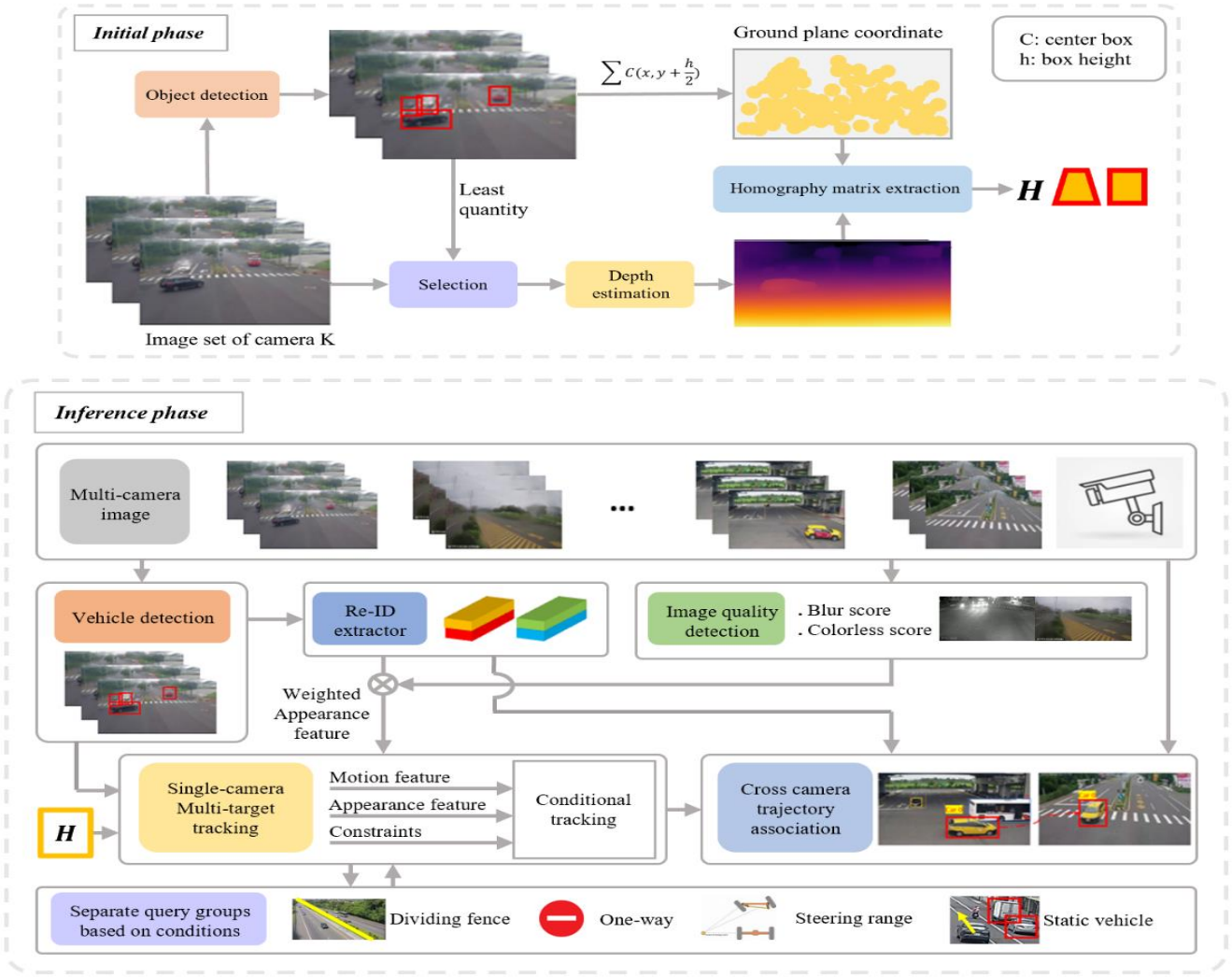


Figure 2. Pipeline for our Cross-camera Multi-target tracking. The system initially runs the first step to synthesize the homography matrix of each camera, then executes it as a block diagram in the inference phase.

additional vehicle images during the training process. Gu Jianyang et al. [30] focusing on cross-domain vehicle ReiD and developed a neural architecture search (NAS) approach that incorporates a twin's contrastive mechanism, a multi-scale interaction LossBranch-Split strategies to design a multi-branch architecture aimed at learning diverse global features and enhancing feature discriminability. And compared to the state-of-the-art (SOTA) vehicle ReID methods, their experiments have demonstrated superior performance.

2.4. Cross-camera trajectory association

There are two ways to approach inter-camera association including matching trajectories or obtaining tracklets. Many previous works attempt to tackle this problem from different aspects. Certain methods [32, 33] rely on a global graph for multiple tracklets in different cameras and optimize for a multi-camera multi-target tracking solution. Other approaches [34] utilized the Hungarian matching algorithm to attain global optimization outcomes using the distance matrix encompassing all tracklet candidates between consecutive cameras, while additional method [35] presents a hierarchical clustering method to collect possible trajectory pairs between two cameras. This study uses the trajectories generated by the former three modules. That associates all tracklets with the same identities by appearance features and spatial-temporal information by using two consecutive cameras to match tracklets according to the entry and exit of the road.

3. METHOD

3.1. Overview

The proposed MCMT system is shown in Fig. 2, which includes Vehicle detection, Re-ID extraction, Image quality detection, Single-camera Multi-target tracking, Automatic

Homography matrix extraction, Separate query groups based on conditions, Cross camera trajectory association.

3.2. Vehicle detection and Re-ID

Vehicle detection is the fundamental and crucial step for multi-camera vehicle tracking. Similar to most MCMT tracking methods, we adopt the tracking-by-detection paradigm, utilizing the state-of-the-art network YOLOv8x, is an advanced version of a real-time object detection system, renowned for its high accuracy and speed in detecting objects in images and videos. It performs detection by applying a single neural network to the entire image, dividing it into regions, and predicting bounding boxes and probabilities for each region. The resolution of the image is full HD, so we only let the model learn with image sizes smaller than 720x1280, specifically 640x640, excluding yolo methods with 4 level feature levels and requiring high resolution like 1280x1280. We also tried finetuning and comparison between YOLOv8 and YOLOv9, and found that YOLOv8 is more stable in both Recall and Precision.

For vehicle re-identification, we utilized Fast-ReID model with backbone architecture Resnet50. This method is one of the stable methods that reached the state of the art in recent years and is now used by some MOT methods to extract appearance features such as BoT-SORT. However, this method is still ineffective in bad weather conditions and images lose color. Therefore, we have limited the weight of the appearance feature to cases of color loss, which is why the image quality detector was created.

3.3. Image quality detection

3.3.1 Colorless

We observed that the RGB histograms of grayscale images exhibit nearly identical values across the red, green, and blue channels. This uniformity indicates the absence of color differentiation, characteristic of grayscale images. The consistent histogram pattern remains even with translated color values, resulting in a uniform grayscale representation. This observation can be useful in image quality detection processes to distinguish between colored and colorless images, as shown in Fig. 3.

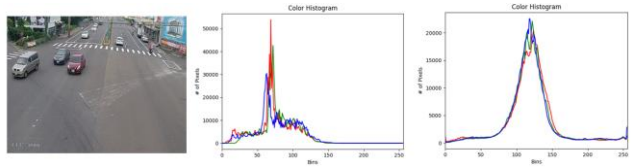
Based on this concept, we employ a mathematical approach to quantify the absence of color. The formula could be written as following:

$$t = arg\,max(H) - arg\,max(H_j) \tag{1}$$

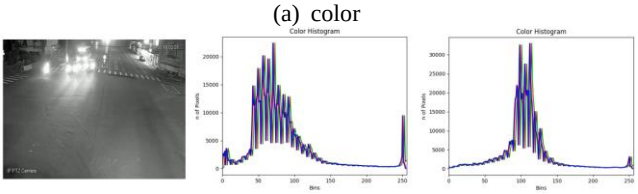
where t represents the threshold value, which signifies the point of maximum histogram value, H represents the histogram of the RGB values.

$$e = H_j(x+t) - H_j(x) \tag{2}$$

where x is the position along the x-axis. A colorless image is identified when e is close to zero, indicating minimal deviation in the histogram values after translation along the x-axis. This mathematical characterization allows us to systematically evaluate the image quality and effectively detect colorless images in a consistent and objective manner, as shown in Fig. 4.



(a) color



(b) colorless

Figure 3. Visualization of color images and colorless images.

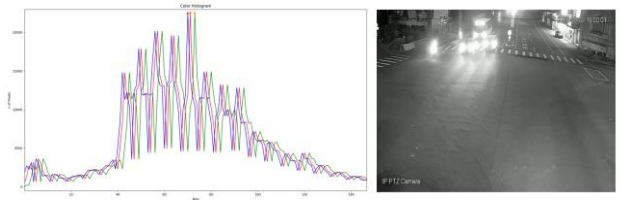


Figure 4. Visualization of colorless image after apply mathematical approach.

3.3.2 Blur image

In our approach to image quality detection, we establish a reference frame to compare against current frames, assessing the degree of blur present in the images, as shown in Fig. 5.

$$FM = \frac{\sum_0^m (b_m)}{h * w - (objectarea)}, \begin{cases} b = 1 & if\ f > M/1000, \\ b = 0 & else \end{cases} \tag{3}$$
$$M = \max(abs(f_{shift\_center}))$$

where f is converted to the frequency domain using FFT from the ignored object image. The formula we previously discussed is applied with a key modification: regions containing objects are ignored. By focusing on the background, we obtain a more accurate representation of the image's sharpness. The formula could be written as following:

$$blur\ score = \frac{FM_{cur} - FM_{ref}}{FM_{cur}} \tag{4}$$

This calculates the blur score by comparing the frequency magnitude of the current and reference frames, excluding object areas. A negative score indicates that the current frame is blurrier than the reference frame, providing a quantitative measure of image quality.

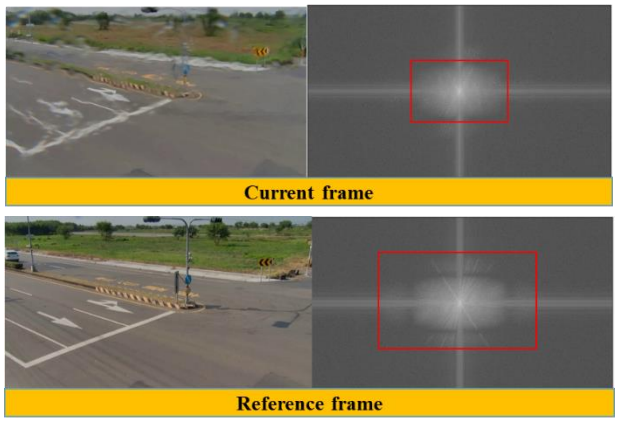


Figure 5. Establish a reference frame to compare against current frames.

3.4. Automatic Homography matrix extraction

The UCMCTrack method involves manual calibration to find the homography matrix, which is a critical step in camera calibration. However, this manual approach can be challenging and may not yield high accuracy. The calibration matrix is essential for transforming the perspective of a 2D image into a 3D world coordinate system. Achieving precision in this transformation is crucial, as even small errors can lead to significant inaccuracies in the final application. Therefore, while UCMCTrack provides a foundational approach to calibration, alternative methods that automate the process or incorporate more sophisticated algorithms may be necessary to enhance accuracy and reliability.

To enhance the accuracy of homography matrix extraction, we propose an automatic extraction method based on the depth estimation method, combining object detection to predict the ground surface. shown in Fig. 2.

From the given dataset of each camera, we use object detection to extract the midpoints at the bottom of the bounding box, which are considered the areas where the vehicle can move and are also the ground. At the same time, we also select the panels with the fewest objects, or the ground that is most clearly shown. We use the DepthAnything [39] method to predict relative depth, combined with ground information we can predict the approximate homography matrix. Finally, we extract the homography matrix by sampling nine arbitrary points near the edges of the image on the newly identified ground plane, indicated by a yellow contour. The depth data informs us of the actual coordinates, simplifying the extraction of the homography matrix (H matrix).

3.5. Single-camera Multi-target tracking

In our tracking system overview, we present a schematic that delineates our enhancements to single-camera multi-target tracking. The green boxes highlight our additions, which include similar distance motion and appearance (Dm, Da) and a constraint (Dc). With the primary features being motion and appearance, we have incorporated quality conditions and constraints into our system. The homography matrix (H) is obtained automatically. All are shown in figure 6

After the features are extracted and the quality weights and constraints are applied, the next step is to calculate the cost matrix. The cost matrix is a two-dimensional matrix where each element represents the cost of assigning a track to a new detection. In other words, each row of the cost matrix represents a track, while each column represents a new detection.

Once the cost matrix is calculated. The Hungarian algorithm is used to find maximum-weight matchings, which is sometimes called the assignment problem. If the cost of assigning a track to a new detection is too high (above a certain threshold), the track and detection are considered as not matched.

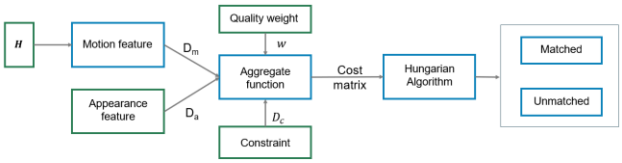


Figure 6. Pipeline of our single-camera multi-target tracking system.

Next, we go deeper into the two modules, Constraint and Aggregate function, respectively, with the Aggregate function to synthesize motion feature appearance feature quality weight and constraint information.

3.5.1. Separate query groups based on conditions

a. Steering range

Our task involves a four-wheeled vehicle, which inherently limits the maximum real-world steering angle that can be achieved, as illustrated in the image. By calculating the cosine between the previous and current vector information, we can determine the deviation. We aim for a matching distribution as shown in the bottom right image. Steering angles exceeding the limit are directly set to zero, and the distribution is designed to draw more attention to the center, acknowledging the existence of an error margin (e) in the image domain. The areas between the blue and red lines gradually decrease, accounting for potential noise cases.

In the image domain, steering angle limits vary depending on the vehicle's direction of movement. The double red arrows represent an incorrect steering angle limit

perceived in the image, which is actually smaller in reality with green arrows. This discrepancy is considered noise, and therefore, its significance is reduced in our analysis. By acknowledging and adjusting for these errors, we can refine our vehicle tracking and prediction models to better reflect real-world dynamics.

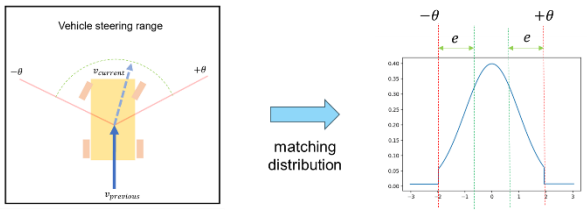


Figure 7. Illustrate the match distribution of the limitation of the maximum real-world steering angle.

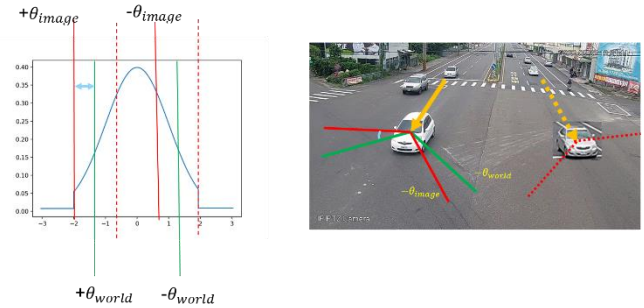


Figure 8. Illustrate the errors in the image and reality domains.

b. Static vehicle

In addition to steering angles, the image also considers static vehicles, which are identified by high Intersection over Union (IoU) values and have been consistently tracked across multiple frames. This is indicative of parked or stationary vehicles within the traffic scene. The formula depicted in the image defines a static object as follows:

$$D_{i,j}^{static} = \begin{cases} D_{i,j}^{IoU} & if \ track_{age} > N, \\ 0 & else \end{cases} \tag{5}$$

where D^{static} represents the static condition, and $track_{age}$ denotes the duration (in frames) that an object has been tracked. If the $track_{age}$ exceeds a certain threshold N, the object is considered static. This approach allows for the separation of moving vehicles from static ones, enhancing the accuracy of traffic analysis and monitoring systems.

Figure 9. The static vehicles in the image.

c. One way & dividing fence

The image illustrates traffic management conditions for one-way streets and dividing fences. The conditions specified for one-way streets include vehicles remaining within a given polygon region and avoiding reverse situations. Additionally, vehicles must not cross a labeled road when a dividing fence is present. These rules are crucial for maintaining orderly traffic flow and ensuring road safety. The formula in the image defines the one-way condition as follows:

$$D_{i,j}^{one\_way} = \begin{cases} D_{i,j}^{cosine}(v_{ref}, v_{vehicle}) & if \ (v \ inside \ polygon) \ and \ not \ \text{int}erest, \\ 2 & elif \ \text{int}erset, \\ 0 & else \end{cases} \tag{6}$$

where D^{one_way} represents the one-way condition based on the vehicle's position relative to the polygon and the dividing fence. This mathematical model aids in the automated enforcement of traffic regulations.

In essence, when the current vector of a vehicle is within the polygon and does not intersect with the fence, the distance is calculated using the cosine distance between the reference segment vector, and the vehicle's current vector. If the vehicle's path intersects with the fence, indicating a violation of the one-way condition, the distance is assigned the maximum value of 2, which is the largest value for cosine distance. This assignment acts as a penalty for the intersection violation, emphasizing the importance of adhering to one-way traffic rules and road divisions**.**

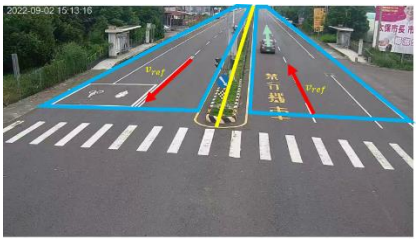


Figure 10. Illustrates traffic management conditions for one-way streets and dividing fences.

3.5.2. Aggregate function

From the above conditions, we have a general definition of the distance function of the constraint. Equation (7) separates the static object from the rest when it is given a smaller weight, with the threshold T in practice chosen to be 0.8. Formula (8)(9) as mentioned in section 3.5.1.a, when applied to the cost matrix, it is the opposite of the desired distribution and is exponential. Formula (10) represents the last condition for a one-way road.

$$\widehat{D}_{i,j}^{static} = \begin{cases} 1 + \epsilon - D_{i,j}^{static} & if \ D_{i,j}^{static} > T, \\ 1 & else \end{cases}, \quad \epsilon \text{ is a very small number} \tag{7}$$

$$\widehat{D}_{i,j}^{steer} = \begin{cases} 1 + e^{s\times(t\times\widehat{D}_{i,j}^{static}\times D_{i,j}^{steer}-1)} & if\ D_{i,j}^{steer} \le \theta_1, \\ \delta & else \end{cases}, \quad (8)$$

$$D_{i,j}^{steer} = D_{i,j}^{cosine}\left(v_{ref}, v_{vehicle}\right) \quad (9)$$

$$D_{i,j}^{c} = \begin{cases} \widehat{D}_{i,j}^{steer} & if\ D_{i,j}^{one-way} < \theta_2, \\ \delta & else \end{cases} \quad (10)$$

With δ is a very large number, going toward infinity. θ_1 θ_2, s and t are the thresholds of steering angle range, opposite direction, slope and translate of the exponential function, respectively.

Finally, the cost matrix function is defined by the formula below

$$Cost_{i,j} = \min\{w \times D_{i,j}^{a}, D_{i,j}^{m}\} \times D_{i,j}^{c} \quad (11)$$

For w we currently use δ and 1, corresponding to bad and good image quality.

In other words, we can divide the query group based on the constraints on the conditions that we have set, for example when there are constraints on driving direction in the vehicle re-identification (ReID) system. Typically, ReID queries consider all cases without distinction. However, by combining the steering angle conditions, we can classify vehicles into separate query groups, as shown in the figure with two separate groups. Each group contains less media, potentially increasing the accuracy of the ReID process. This approach offers several advantages, such as improved accuracy due to more precise matching of vehicles with similar trajectories, reduced search space which speeds up the process, and contextual filtering takes advantage of steering direction as a valuable criterion in complex traffic situations.

3.6. Cross camera trajectory association

Cross camera trajectory association is the final step for MCMT system. This process links the trajectories of tracked objects across multiple cameras, ensuring correct identification as they move through different areas. By integrating data, the system accurately tracks objects, enhancing surveillance and behavior analysis in complex areas, which is crucial for security, traffic management, and smart city systems.

In this case, we reuse the FastReID feature in section 3.2 and query for vehicles that have cross-camera correlations. Note that our results were skipped in this section because the team did not have time. to extract the embedding feature and query

4. EXPERIMENTS

4.1. Dataset and Evaluation Setting

4.1.1 Datasets

We participated in the AI Cup competition organized by the Ministry of Education: Cross-Camera Multi-Target Vehicle Tracking Competition – Model Development Session.

The dataset we train on both YOLOv8x and FastReID is provided by AI Cup competition without any additional data. As for the FastReID method, we only train with color images, because the appearance feature is removed in that case.

4.1.2 Evaluation Metrics.

For MTMC tracking, we utilize the CLEAR metrics [37], which include MOTA, FP, FN, and others, along with the IDF1 score to evaluate tracking performance. The evaluation system for the AI CUP competition sums the IDF_1 score [38] and MOTA to determine the final result, which is displayed on the Public/Private Leaderboard. MOTA emphasizes the performance of the detector, while IDF1 is the ratio of correctly identified detections over the average number of ground truth and computed detections, measuring the accuracy of the tracked ID.

The total score could be calculated as:

$$Total\ Score = \frac{2IDTP}{2IDTP + IDFP + IDFN} - \frac{FN + FP + IDS}{GT} + 1 \quad (x)$$

4.2. Implementation details

All the experiments were implemented under the Pytorch framework and NVIDIA 2080Ti GPU platform. We fine-tune both object detection and ReID based on the large COCO and VERI-Wild datasets. With object detection, use basic data augmentation methods such as mixup, mosaic, hsv, translate, scale.

One-way zones, lane barriers and reference vectors in the Constraint section are labeled with each camera. θ_1, θ_2, s and t in formulas (8) and (9) are in practice used as 0.3, 1.1, 8 and 0.85 respectively.

4.3. Results

We utilize the evaluation system to verify the effectiveness of our algorithm and optimize it based on the IDF_1 score and MOTA results. Table 1 shows the effect of using several optimization modules separately. Our baseline is based on the UCMCTrack [11], while the AICUP baseline is based on the BoT-SORT [12]. The static constraint corresponds to static vehicles, and the image quality detection module assesses the clarity of the images, providing a blur score and a colorless score. Combining this module with the Re-ID extractor helps to obtain a weighted appearance feature. The direction constraint accounts for the model considering steering range conditions, while the map constraint

addresses other conditions such as dividing fences and one-way roads. By acknowledging and considering these specific criteria, we can improve vehicle tracking and prediction models to robustly reflect real-world dynamics.

In the final ranking of the AICUP competition 2024, we placed sixth among the top-10 participating teams. The comparison of our method with other teams on the public leaderboard is shown in Table 2.

Method	Score	IDF_1	MOTA
AICUP's Baseline	0.8053	0.4675	0.3410
Our baseline (UCMCTrack)	0.8305	0.4577	0.3727
+Static constraint & image quality	0.8980	0.4933	0.4046
+Direction constraint	0.9665	0.5198	0.4467
+Map constraint	**1.0012**	0.5414	0.4598

Table 1. Total score changes after several optimizations.

Rank	Team Name	General Final Score	Public Score	Private Score
1	Team_5077	1.2369	1.1748	1.2636
2	Team_5045	1.2231	1.1397	1.2588
3	Team_4978	1.1987	1.1258	1.2300
4	Team_5084	1.1956	1.1238	1.2264
5	Team_5141	1.1933	1.1237	1.2232
6	**Team_5093**	**1.1343**	**1.0012**	**1.1914**
7	Team_5080	1.1013	0.9676	1.1586
8	Team_4944	1.0948	0.9968	1.1369
9	Team_5149	1.0941	0.9879	1.1396
10	Team_5076	1.0350	1.0205	1.0411

Table 2. The General Final Score of competition.

5. CONCLUSION

In this paper, we proposed a new cross-camera multi-object tracking system. We enhance the UCMCTrack method by incorporating appearance features, proposed a method to automatically extracting the Homography matrix, and adapted data for varying conditions using our developed image quality detectors. Our results show improvements in both IDF_1 score and MOTA, achieving a total score of 1.0012 and ranking seventh place on the public leaderboard.

6. REFERENCES

[1] Milind Naphade, Zheng Tang, Ming-Ching Chang, David C. Anastasiu, Anuj Sharma, Rama Chellappa, Shuo Wang, Pranamesh Chakraborty, Tingting Huang, Jenq-Neng Hwang, and Siwei Lyu, "The 2019 ai city challenge," In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 452-460, 2019.

[2] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 4263–4273, 2021.

[3] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Xiaodong Yang, Liang Zheng, Anuj Sharma, Rama Chellappa, and Pranamesh Chakraborty, "The 4th ai city challenge," In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 2665-2674, 2020.

[4] Zhiqun He, Yu Lei, Shuai Bai, and Wei Wu, "Multi-camera vehicle tracking with powerful visual features and spatial-temporal cue," In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 203-212, 2019

[5] Yunzhong Hou, Heming Du, and Liang Zheng. "A locality aware city-scale multi-camera vehicle tracking system," In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 167-174, 2019.

[6] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 416-424, 2019.

[7] Xipeng Yang, Jin Ye, Jincheng Lu, Chenting Gong, Minyue Jiang, Xiangru Lin, Wei Zhang, Xiao Tan, Yingying Li, Xiaoqing Ye, and Errui Ding. "Box-Grained Reranking Matching for Multi-Camera Multi-Target Tracking," In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 3096-3106, 2022.

[8] Fei Li, Zhen Wang, Ding Nie, Shiyi Zhang, Xingqun Jiang, Xingxing Zhao, and Peng Hu, "Multi-Camera Vehicle Tracking System for AI City Challenge 2022," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3264-3272, 2022.

[9] 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)*, pp. 3645-3649, 2017.

[10] 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)*, pp. 3464-3468, 2016.

[11] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky, "BoT-SORT: Robust associations multi-pedestrian tracking," *arXiv preprint arXiv:2206.14651*, 2022.

[12] Kefu Yi, Kai Luo, Xiaolei Luo, Jiangui Huang, Hao Wu, Rongdong Hu, and Wei Hao, "UCMCTrack: Multi-Object Tracking with Uniform Camera Motion Compensation," In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38, no. 7, pp. 6702-6710, 2024.

[13] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg, "SSD: Single Shot MultiBox Detector," *European Conference on Computer Vision (ECCV)*, 2015.

[14] Joseph Redmon and Ali Farhadi, "YOLO9000: better, faster, stronger," In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 6517-6525, 2016.

[15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence 39*, pp. 1137-1149, 2015.

[16] Zhaowei Cai and Nuno Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6154–6162, 2017.

[17] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko, "End-to-end object detection with transformers," *European Conference on Computer Vision (ECCV)*, pp. 213–229, 2020.

[18] 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 (ICCV)*, pp. 9992-10002, 2021.

[19] Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng, "StrongSORT: Make DeepSORT Great Again," *IEEE Transactions on Multimedia*, pp. 8725-8737, 2022.

[20] Jinkun Cao, Jiangmiao Pang, Xinshuo Weng, Rawal Khirodkar, and Kris Kitani, "Observation-centric sort: Rethinking sort for robust multi-object tracking," In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 9686-9696, 2023.

[21] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu, "Quasi-dense similarity learning for multiple object tracking," In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 164-173, 2020.

[22] Zhongdao Wang, Liang Zheng, Yixuan Liu, and Shengjin Wang, "Towards real-time multi-object tracking," *European Conference on Computer Vision (ECCV)*, pp. 107-122, 2020.

[23] 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*, vol. 129, pp. 3069-3087, 2021.

[24] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," *European Conference on Computer Vision (ECCV)*, pp. 474-490, 2020.

[25] C. Liang, Z. Zhang, X. Zhou, B. Li, S. Zhu, and W. Hu, "Rethinking the Competition Between Detection and ReID in Multiobject Tracking," *IEEE Transactions on Image Processing*, vol. 31, pp. 3182-3196, 2022.

[26] B. Shuai, A. Berneshawi, X. Li, D. Modolo, and J. Tighe, "Siammot: Siamese multi-object tracking," In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12372-12382, 2021.

[27] Zhedong Zheng, Tao Ruan, Yunchao Wei, Yi Yang, and Tao Mei, "Vehiclenet: Learning robust visual representation for vehicle re-identification," *IEEE Transactions on Multimedia*, pp. 2683–2693, 2020.

[28] Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven CH Hoi, "Deep learning for person reidentification: A survey and outlook," *IEEE Transactions on Pattern Analysis and Machine Intelligenc*e, 2021.

[29] Yi Zhou and Ling Shao, "Viewpoint-Aware attentive multi-view inference for vehicle re-identification." In *Proceedings of the IEEE conference on computer vision and pattern recognitio*n, pp. 6489–6498, 2018

[30] Jianyang Gu, Kai Wang, Hao Luo, Chen Chen, Wei Jiang, Yuqiang Fang, Shanghang Zhang, Yang You, and Jian Zhao, "MSINet: Twins Contrastive Search of Multi-Scale Interaction for Object ReID," In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[31] E. Almeida, B. Silva, and J. Batista, "Strength in Diversity: Multi-Branch Representation Learning for Vehicle Re-Identification," *2023 IEEE 26th International Conference on Intelligent Transportation Systems* (*ITSC*), pp. 4690-4696, 2023.

[32] W. Chen, L. Cao, X. Chen, and K. Huang, "A novel solution for multi-camera object tracking," In *2016 IEEE international conference on image processing (ICIP)*, pp. 2329-2333, 2014.

[33] W. Chen, L. Cao, X. Chen, and K. Huang, "An equalized global graph model-based approach for multicamera object tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 11, pp. 2367-2381, 2016.

[34] Jin Ye, Xipeng Yang, Shuai Kang, Yue He, Weiming Zhang, Leping Huang, Minyue Jiang, Wei Zhang, Yifeng Shi, Meng Xia, and Xiao Tan, "A Robust MTMC Tracking System for AI-City Challenge 2021," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 4029-4048, 2021.

[35] 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 (CVPR) Workshops*, pp. 4129-4137, 2021.

[36] Dillon Reis, Jacqueline Hong, Jordan Kupec, and Ahmad Daoudi, "Real-time flying object detection with YOLOv8," *arXiv preprint arXiv:2305.09972*, 2023.

[37] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, pp. 1–10, 2008.

[38] 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 (ECCV)*, pp. 17–35, 201

[39] Yang, Lihe, et al. "Depth anything: Unleashing the power of large-scale unlabeled data." *arXiv preprint arXiv:2401.10891* (2024).