Complementing Vehicle Trajectories Using Two Camera Viewpoints

Katsuaki Nakano
Department of Computer Engineering,
Kanazawa Institute of Technology
Kanazawa, Japan
b1814660@planet.kanazawa-it.ac.jp

Minoru Nakazawa
Department of Computer Engineering,
Kanazawa Institute of Technology
Kanazawa, Japan
nakazawa@infor.kanazawa-it.ac.jp

Michael Zuzak
Department of Computer Engineering,
Rochester Institute of Technology
Rochester, USA
mjzeec@rit.edu

Abstract—**Traffic volume surveying is a crucial activity to get traffic statistics for road management and traffic congestion control. In recent years, the target environment of traffic volume surveying has become more complex, such as the fully automated surveillance of many-way intersections. Further compounding this complexity, some local governments may not be able to install a camera at a high enough elevation to capture the entire intersection due to environmental, legal, safety, or cost restrictions. Therefore, bigger objects such as buses and trucks often occlude other vehicles in the captured image. This occlusion degrades the accuracy of counting and is one of the main problems that makes the automation of traffic counting at intersections difficult. In this work, we propose a Bird's Eye View (BEV) transformation method capable of 1) removing camera distortion created by wide-angle cameras installed at lower elevations (a common scenario in traffic volume surveys), and 2) utilizing multiple viewpoints to complement object trajectories to reduce accuracy loss caused by occlusion. Furthermore, we evaluate the effectiveness of the proposed method using real-world traffic survey data collected at two intersections in Japan. We find that the proposed method produces a 3% improvement in accuracy over automated counting using a single viewpoint.**

Index Terms—**Computer Vision, Traffic Survey, Multi-Camera Multi-Target Tracking**

I. Introduction

In Japan, traffic volume surveys for national roads and streets are conducted once every five years by the Ministry of Land, Infrastructure, Transport and Tourism (MLIT) to determine the traffic volume on roads throughout Japan and to provide the traffic statistics for road management planning [1]. Currently, these surveys on local roads are mainly conducted manually. Meanwhile, on roads under the direct control of the MLIT, AI analysis using CCTV camera images and image processing technologies uses constant observation to perform automatic traffic counting. The use of automated traffic counting technologies is expected to further increase because the MLIT discontinued manual observation of local road sections during the fall of 2021 [2]. As a result, the MLIT expects traffic volume surveys on local roads to be fully automated, creating an urgent need for automated survey techniques using sensors and AI. To automate traffic surveys on local roads, measurement by direction in complex environments such as many-way intersections will be required. Intersections present a particular challenge as some local governments may not be able to secure a camera position high enough to capture the entire intersection due to local restrictions. This results in bigger objects such as buses and trucks occluding other vehicles within the captured image. Prior work on automated driving has shown that occlusion makes the automation of object tracking tasks challenging [3].

In this study, our purpose is to develop a counting system that is robust to occlusion and camera distortions. The contributions of the work are summarized as follows:

- We develop a methodology that uses multiple low-angle camera viewpoints to complement broken trajectories without using appearance features. By doing so, traffic survey accuracy degradation caused by occlusion and camera distortion is reduced. While we apply the proposed method exclusively to a traffic survey scenario, it can be applied to any multi-view, multi-object tracking problem where occlusion limits accuracy.
- We evaluate the proposed methodology by applying it to real-world traffic survey data collected at two intersections in Japan. We find that our proposed method produces a 3% improvement in survey accuracy over prior automated methods that use a single viewpoint.

II. Preliminaries

A. Multi-Object Tracking

Multi-object tracking (MOT) is a crucial task in computer vision that involves tracking multiple objects moving in a video by assigning a unique ID to each of them in each frame. MOT methods are broken down into two categories: Tracking-by-Detection [4]–[6] and One-shot [7], [8]. Tracking-by-Detection methods divide the task into two phases: detection and tracking. Tracking-by-Detection has the advantage that we can select a suitable model for each of the detection and tracking tasks. YOLO [4] is a typical detection model for Tracking-by-Detection. For a tracking model, SORT [5] and its extension DeepSORT [6] are representative models. In general, a tracking model processes detection results inferred by a detection model to realize Tracking-by-Detection. Meanwhile, one-shot methods combine detection and tracking in a single model and are generally characterized by faster processing speeds than Tracking-by-Detection methods. FairMOT [7] and JDE [8] are representative one-shot models.

The MOT challenge [9], a person tracking benchmark, uses 20 indicators to evaluate the performance of object tracking models, of which Multiple Object Tracking Accuracy (MOTA) [10] is a measured task effectiveness indicator. However, one-shot methods are often optimized with parameters specific to human tracking in the MOT challenge. In traffic volume surveys, the classes are not limited to people, but also include a wide range of objects such as cars, buses, cyclists, etc. Thus, our proposed system employs Tracking-by-Detection models as tracking models.

B. Multi-Camera Multi-Target Tracking

Multi-camera multi-target tracking (MTMCT) is another important task in computer vision that aims to detect and track multiple targets in a multi-camera environment. The general pipeline of MTMCT methods is composed of the following two phases: 1) the local trajectory generation phase that tracks and detects target objects to generate a local trajectory for each perspective, and 2) the cross-camera trajectory matching phase that matches a local trajectory across all cameras to generate a complete trajectory within the entire multi-camera network. An important unsolved problem in MTMCT is object occlusion, which leads to incomplete local trajectories within specific cameras that degrade the accuracy of cross-camera trajectory matching [11]. We note that this problem is present in any MOT task, hence, improvements in MOT models contribute to addressing the challenges created by occlusion.

The main challenge addressed by MTMCT is the considerable variations in visual appearance between different perspectives from different cameras. Several works have explored methods to overcome these differences in appearance between camera perspectives [12], [13]. Hu et al. uses epipolar geometry to calculate the correspondence between all cameras [12]. To do so, the authors project lines and detected object coordinates into a reference plane to find the correspondence. Xu et al. uses a Hierarchical Composition of Tracklet (HCT) to match local trajectories by considering multiple cues of objects such as their visual appearance and coordinates on the ground plane [13].

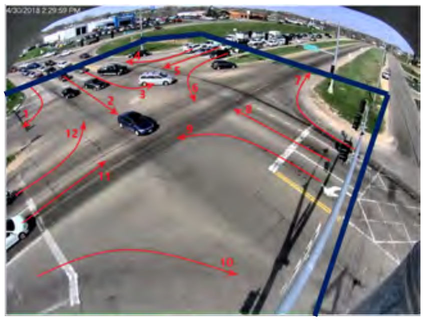


Fig. 1. One viewpoint in the AI City Challenge [14]

Typical datasets for evaluating MTMCT algorithms include the AI City Challenge [14] and the Duke-MTMC dataset [15]. In such datasets, the Region of Interest (ROI), surrounded by blue lines in Figure 1, and the Motion of Interest (MOI), represented by red arrows in Figure 1, are set. Vehicles are often counted according to these identifiers. In some cases, such as the AI City Challenge camera viewpoint shown in Figure 1, the camera is positioned higher than the traffic signal, making occlusion unlikely. As a result, existing methods are likely not designed to handle occlusion, limiting their applicability to intersections with height restrictions, where occlusion is common. In addition, methods that use appearance features have the problem of producing different matching results when the differences in appearance features between different cameras are too large, or when distinct objects are visually similar. Therefore, in this study, we propose a method to complement broken trajectories without appearance features by using camera calibration and homography matrices for BEV transformation that is applicable to wide angle cameras located in low elevation positions.

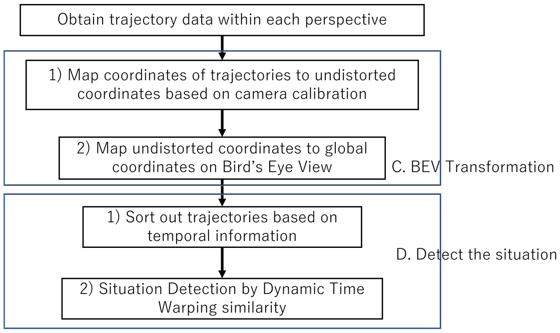

Obtain trajectory data within each perspective
1) Map coordinates of trajectories to undistorted coordinates based on camera calibration
2) Map undistorted coordinates to global coordinates on Bird's Eye View
C. BEV Transformation
1) Sort out trajectories based on temporal information
2) Situation Detection by Dynamic Time Warping similarity
D. Detect the situation

Fig. 2. Chart of the Proposed Method

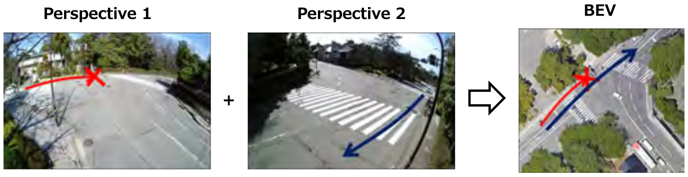


Fig. 3. Overview of the Proposed Method

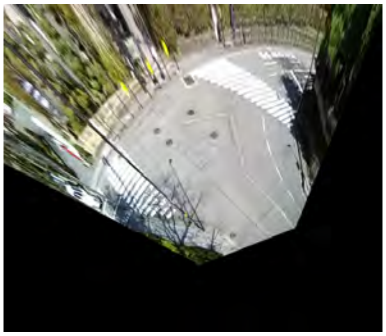


Fig. 4. Mapped BEV without Camera Calibration

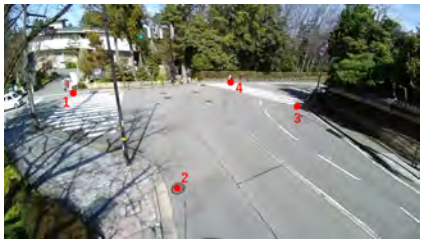


Fig. 5. Corresponding Points in Perspective 1

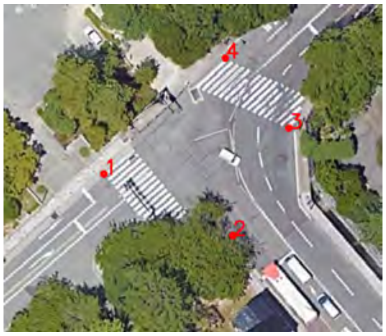


Fig. 6. Corresponding Points in BEV

III. Proposed Occlusion-Resistant Multi-Target Multi-Camera Tracking Algorithm

A. *Overview of Proposed Method*

Our proposed method converts vehicle trajectories obtained from two camera perspectives into BEV images. Then, it detects a situation where the vehicle trajectory is broken in one perspective but is correctly tracked for a longer period in the other perspective. This scenario indicates that occlusion has occurred in the intersection, causing an object detection to drop in the occluded perspective, while remaining present in the other non-occluded perspective. By doing so, trajectories can be maintained as long as at least one camera retains it's detection of the target. This allows existing MCMT systems to overcome accuracy loss caused by occlusion events.

Figure 3 shows an example of an occluded trajectory being recovered with the proposed method. In this example, the red trajectory in perspective 1 and the blue trajectory in perspective 2 show the same object passing through at the same time. However, notice that in perspective 1, the object IDs are switched in the middle of the trajectory. Throughout the process shown in Figure 2, the trajectories of these two different viewpoints are mapped into BEV, and the Dynamic Time Warping (DTW) similarity of the trajectories is used to detect situations where one trajectory is interrupted while the other trajectory is being tracked correctly (i.e., occlusion). Similarly, an interruption in perspective 2 can be recovered using perspective 1 in the same way, thereby reducing the overall number of broken trajectories and improving MCMT tasks, such as vehicle counting.

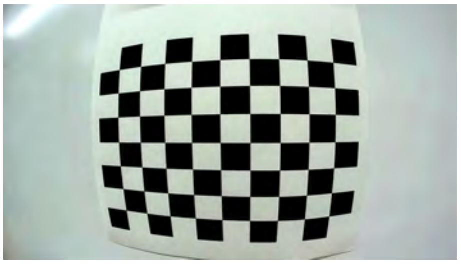


Fig. 7. An Input Image for Calibration

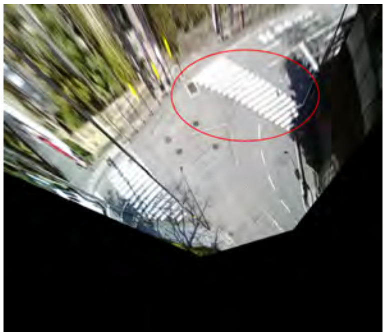


Fig. 8. Transformation Result

B. *Trajectory Mapping to BEV*

1) Camera Calibration: If a wide-angle camera image is input as-is for overhead view conversion, a distorted overhead view is obtained. This will result in a distorted image when converted to BEV, where objects near the edges of the image will appear to be curved. To resolve this, a more accurate bird's-eye view conversion requires distortion correction by camera calibration.

Camera calibration is a technique for correcting images by determining internal camera parameters such as lens focal length, camera position, and camera orientation. Calibration was performed using OpenCV functions. The input data for obtaining the internal parameters is the chessboard shown in Figure 7. These images are taken with a wide-angle camera from various angles and distances. A total of 40 of these boards were input for parameter estimation. The findChessBoardCorners function in OpenCV is input-sensitive, and the corner detection may not work correctly depending on the light conditions, so the contrast value of the input images was set high.

The Figure 8 shows the result of the distortion correction using the camera's internal parameters obtained from the camera calibration.

2) BEV Mapping by Homography Matrix: The homography matrix is a 3 × 3 matrix of projections for the planes in 3D space. The format of this matrix is in Equation (1). The homography matrix can be obtained using OpenCV's findHomography function.

$$H = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \tag{1}$$

$$x' = \frac{M_{11}x + M_{12}y + M_{13}}{M_{31}x + M_{32}y + M_{33}} \tag{2}$$

$$y' = \frac{M_{21}x + M_{22}y + M_{23}}{M_{31}x + M_{32}y + M_{33}} \tag{3}$$

To obtain the homography matrix, it is necessary to input four or more corresponding coordinates of the image to be converted to overhead coordinates and the BEV image. Then, the coordinates x and y of each pixel are transformed to x' and y' in equations (2) and (3) based on the homography matrix parameters obtained. The four red dots in the Figure 5 and 6 depict the corresponding points overlaid onto each image. The Figure 8 is the result of the transformation based on the corresponding points. Compared to Figure 4 without distortion correction, the distortion of the crosswalk in the upper right corner of the image, which is curved toward the back, has been corrected, resulting in a more accurate transformation result.

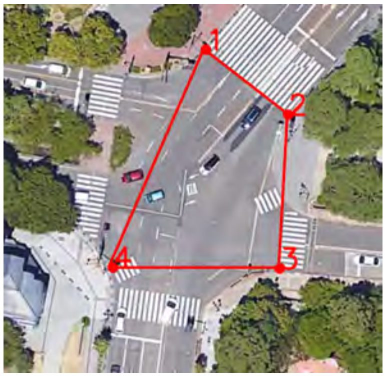


Fig. 9. Region of Interest

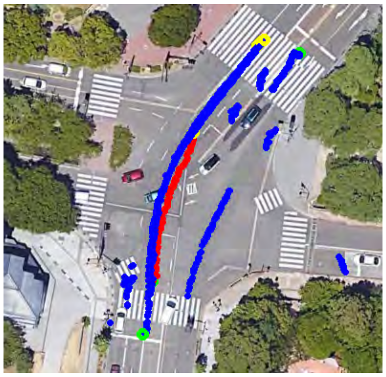


Fig. 10. Trajectories Selected by Temporal Information

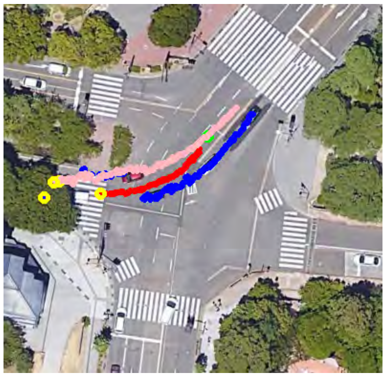


Fig. 11. A Trajectory Decided by DTW Similarity

C. *Occlusion Scenario Detection*

1) Target Trajectories: To perform traffic counting, each image has a defined region of interest based on the four corresponding points specified in the overhead view transformation. An example of a region of interest is shown in Figure 9. Traffic counting is performed by counting trajectories that move through the boundaries of this region. Trajectories with a start or end point within the region of interest indicate an interrupted trajectory (e.g., an occlusion event). In prior counting methods, these trajectories are not counted properly because they do not cross the region of interest boundary, degrading counting accuracy. Therefore, to improve accuracy, trajectories that have a start or end point within the region of interest must be complemented. Figure 10 shows an example trajectory that is interrupted in the region of interest. The green points are the start of the trajectory and the yellow points are the end of the trajectory.

2) Extracting Similar Trajectories by Temporal Information: For each vehicle trajectory from one viewpoint that exists within the range, candidate similar trajectories from another viewpoint that existed at the same time are selected. Since the manual frame synchronization resulted in a time gap of up to 5 frames, we extract all trajectories from the other viewpoints that existed in the 5 frames before and after the frame of interest. Figure 10 shows the extracted trajectory derived from temporal information. The red trajectory is obtained from the perspective 1. The blue trajectories are obtained from the perspective 2 existing in the 5 frames before or after when the red trajectory existed.

3) Extracting Similar Trajectories by Dynamic Time Warping: Dynamic Time Warping (DTW) [16] is a method to measure the distance and similarity between time-series data. DTW can be applied to time-series data of different lengths because it finds the shortest path using a brute-force comparison of distances at each point. To extract similar trajectories, we employed DTW for vehicle trajectories using Euclidean distance. The trajectory shown in pink in the Figure 11 is the similar trajectory determined by DTW. Because extremely short trajectories may be selected when evaluating DTW alone, the Euclidean distance between the start and end points of each trajectory is also compared. If the similar trajectory determined by DTW is longer than the target trajectory in red, the pair of trajectories between perspectives is detected as a candidate target to be complemented.

IV. Evaluation

A. Target Data

To evaluate our proposed methodology, we applied it to a multi-viewpoint recording of an intersection in Hirosaka, Japan. A bird's-eye view of the evaluation intersection, obtained from Google Earth, is shown in Figure 12. The viewpoint of the two cameras over the evaluation intersection is shown in Table I. The video used for evaluation was recorded at a height of 4.5m from the ground using a Panasonic HX-A1 with the wide-angle setting enabled. The recording setup is depicted in Figure 13. The videos recorded from the two cameras were manually synchronized to ensure that frames from each viewpoint corresponding to the same time instant were processed together.

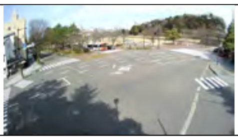
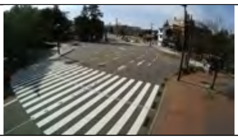
B. Experimental Methodology

For evaluation, a 10-minute video was recorded by both viewpoints at the Hirosaka intersection in Kanazawa City shown in Table I. ID pairs that were interrupted in Perspective 1 and could be followed in Perspective 2 were manually annotated to provide ground truth. We evaluated the proposed methodology based on the agreement rate between the annotations and the ID pairs selected by the system.

C. Experimental Results

Table II and III show the overall traffic volume, annotation results, and system detection results. The annotation results show that out of the 294 vehicles in the video, 14 IDs are tracked from perspective 1 and 8 IDs are tracked from perspective 2. The proposed system was able to detect 9 IDs out of 14 and 6 IDs out of 8, respectively. This corresponds to 68% of the total annotated trajectories that can be complemented in the video. This results in a vehicle measurement accuracy of roughly 81% at the Hirosaka intersection. According to the results in perspective 1, our proposed method detected 9 IDs out of the entire 294 IDs. Therefore, the vehicle measurement accuracy can be improved by 3%, from 81% to 84%, by complementing the trajectories detected by the proposed system.

TABLE I
Target Intersection

Perspective 1	Perspective 2
	

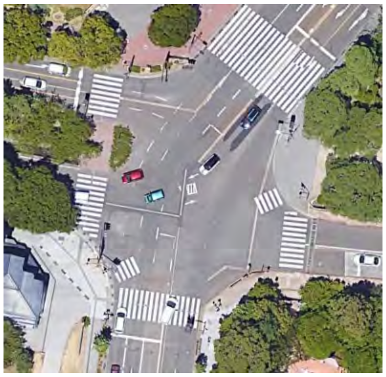


Fig. 12. BEV (Bird's Eye View) Obtained from Google Earth

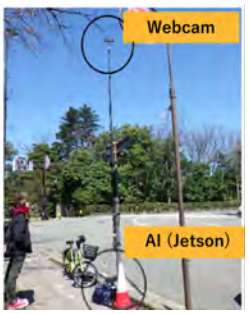


Fig. 13. Configuration for Intersection Data Collection

D. Discussion

The proposed method improved vehicle measurement accuracy by utilizing a complementary viewpoint to reduce the impact of occlusion. While vehicle measurement accuracy improved, there remained several cases that were not corrected. These cases occurred when an incorrect ID was assigned to a complementary trajectory. Figure 14 and 15 show an example of a failure case.

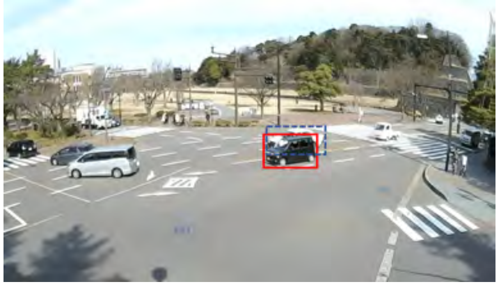


Fig. 14. A Failure Case from Perspective 1

In this case, the white taxi (dotted blue bounding box) behind the black car (red bounding box) in perspective 1 is occluded. As a result, the track on the white taxi is lost. To be correct, the proposed method must select the white taxi. However, the black car (blue bounding box) behind the white taxi was selected in perspective 2 instead. These cases occur

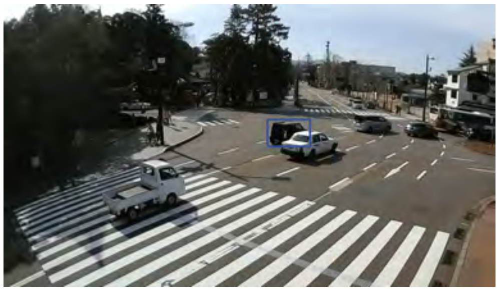


Fig. 15. A Failure Case from Perspective 2

because the trajectory similarity between the two vehicles on the BEV is too similar to distinguish. Such a scenario could be resolved by calculating similarity using not only the DTW, but also the vector information of the trajectory.

TABLE II
EXPERIMENT RESULT 1

Total	Annotated	Detected
294	14	9

TABLE III
EXPERIMENT RESULT 2

Total	Annotated	Detected
294	8	6

V. CONCLUSION

In this paper, we propose a bird's-eye view transformation method that addresses situations in which a camera with a low position and wide-angle distortion is used. Additionally, we propose a methodology to complementing vehicle trajectories on the transformed bird's-eye view to handle occlusion events. To evaluate the effectiveness of the proposed method, we evaluated it on real-world traffic data from two intersections in Japan. The proposed method resulted in a 3% improvement in accuracy over automated counting using a single viewpoint.

In future work, we will extend the proposed method to consider interrupted trajectories originating in either camera viewpoint to further improve the accuracy of traffic surveying. Furthermore, we will apply the ICP (Iterative Closest Point) algorithm to detected trajectory pairs in separate viewpoints to automatically integrate these trajectories.

REFERENCES

[1] "National Road Traffic Survey," Ministry of Land, Infrastructure, Transport and Tourism, https://www.mlit.go.jp/road/road_fr4_000071.html (accessed Apr. 2, 2023).

[2] "National Road Traffic Survey in 2015, Summary of Genaral Traffic Survey Results," Ministry of Land, Infrastructure, Transport and Tourism, https://www.mlit.go.jp/common/001187536.pdf (accessed Apr. 2, 2023).

[3] G. Tian, X. Zhang, S. Guo, Y. Liu, X. Liu and K. Wang, "Occlusion Handling Based on Motion Estimation for Multi-Object Tracking," 2021 IEEE International Conference on Unmanned Systems (ICUS), Beijing, China, 2021, pp. 1031-1036, doi: 10.1109/ICUS52573.2021.9641411.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi,"You only look once: Unified, real-time object detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[5] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," 2016 IEEE International Conference on Image Processing, 2016.

[6] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," 2017 IEEE International Conference on Image Processing, 2017.

[7] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "FairMOT: On the fairness of detection and re-identification in multiple object tracking," International Journal of Computer Vision, vol. 129, no. 11, pp. 3069–3087, 2021.

[8] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," Computer Vision – ECCV 2020, Cham: Springer International Publishing, 2020, pp. 107–122.

[9] "Mot challenge," MOT Challenge, https://motchallenge.net/ (accessed Jun. 10, 2023).

[10] P. Dendorfer et al., "MOTChallenge: A benchmark for single-camera multiple target tracking," International Journal of Computer Vision, vol. 129, no. 4, pp. 845–881, 2020. doi:10.1007/s11263-020-01393-0

[11] Y. He, X. Wei, X. Hong, W. Shi, Y. Gong, "Multi-Target Multi-Camera Tracking by Tracklet-to-Target Assignment", IEEE Transactions on Image Processing. March. 2020.

[12] W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou and S. Maybank, "Principal axis-based correspondence between multiple cameras for people tracking", IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 4, pp. 663-671, Apr. 2006.

[13] Y. Xu, X. Liu, L. Qin and S.-C. Zhu, "Cross-view people tracking by scene-centered spatio-temporal parsing", Proc. AAAI Conf. Artif. Intell., pp. 4299-4305, 2017.

[14] "2020 AI City Challenge," AI City Challenge, https://www.aicitychallenge.org/2020-ai-city-challenge/ (accessed July 31, 2023)

[15] "DUKE MTMC," DUKE MTMC, https://exposing.ai/duke_mtmc/ (accessed July 31, 2023)

[16] S. Salvador and P. Chan, "FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space", Intelligent Data Analysis 11.5 pp. 561-580, 2017.