Cluster Self-Refinement for Enhanced Online Multi-Camera People Tracking

Jeongho Kim Wooksu Shin Hancheol Park Donghyuk Choi
Nota Inc., Republic of Korea
{jeongho.kim,wooksu.shin,hancheol.park,donghyuk.choi}@nota.ai

Abstract

Recently, there has been a significant amount of research on Multi-Camera People Tracking (MCPT). MCPT presents more challenges compared to Multi-Object Single Camera Tracking, leading many existing studies to address them using offline methods. However, offline methods can only analyze pre-recorded videos, which presents less practical application in real industries compared to online methods. Therefore, we aimed to focus on resolving major problems that arise when using the online approach. Specifically, to address problems that could critically affect the performance of the online MCPT, such as storing inaccurate or low-quality appearance features and situations where a person is assigned multiple IDs, we proposed a Cluster Self-Refinement module. We achieved a third-place at the 2024 AI City Challenge Track 1 with a HOTA score of 60.9261%, and our code is available at https://github.com/nota-github/AIC2024_Track1_Nota.

1. Introduction

Multi-camera people tracking (MCPT) is an essential system for understanding and analyzing the pathways and behaviors of people. Recently, there has been extensive research into utilizing these systems as advanced surveillance systems, due to the advancements in deep learning-based models. MCPT is a system that detects and tracks people across multiple cameras. The process of MCPT typically proceeds in the following manner: 1) As illustrated in Fig. 1, the locations of people are detected by inputting footage from multiple cameras into a people detection model. These locations are indicated using bounding boxes and coordinates. 2) Detected people are assigned a local ID through a single camera tracking algorithm, and the appearance feature and location information for each ID is stored. 3) Information from each instance of single camera tracking is matched to assign a global ID. The majority of MCPT systems follow this process.

MCPT can be categorized into online MCPT and offline MCPT, depending on the timing of the video frames used for analysis. Online MCPT utilizes only past frames to predict tracking in the current frame. Therefore, it can be applied to all video sources, including real-time streaming and pre-recorded CCTV footage. On the other hand, offline MCPT uses not only past frames but also future frames to predict tracking in the current frame. Consequently, unlike online MCPT, it cannot be used with video sources like streaming, where future frames are unknown; however, by using future frames, it can improve tracking accuracy. We participated in the 2023 AI City Challenge Track 1 [16] and published a paper [11]. Among the seven papers presented, the top five utilized offline MCPT [7, 13, 17, 19, 25], while only the bottom two [8, 11], including our paper, used online MCPT. Nevertheless, online MCPT, with its broader applicability, is more advantageous for actual industrial use.

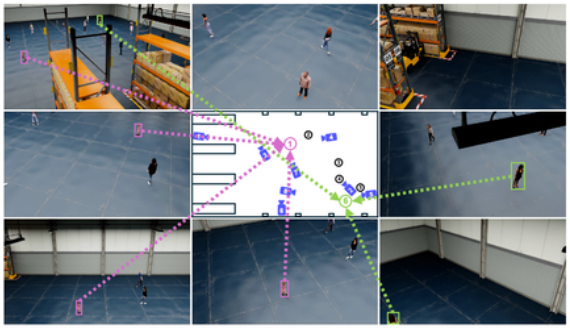


Figure 1. This is an example of Multi-Camera People Tracking. It involves tracking people across various cameras by mapping them to the same identities. The image in the center depicts a 2D map of the location, showing the estimated positions of people as captured by the cameras. The numbers provided represent their global IDs.

In this paper, we propose an online MCPT methodology for the 2024 AI City Challenge Track 1 [24]. This track, held for the second time last year, aims to develop MCPT in indoor spaces such as warehouses and hospitals. Unlike last year, this year's challenge will exclusively use an expanded synthetic data set to track people across multiple cameras. For this track, we utilized our MCPT system presented last year as a baseline. However, this system had several areas for improvement, highlighting persistent issues inherent to

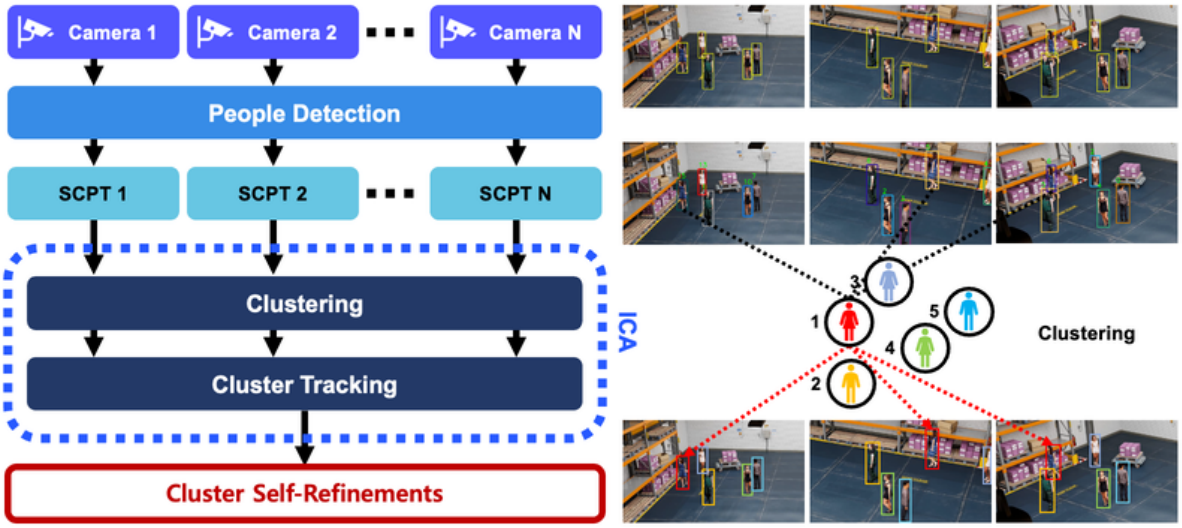


Figure 2. Overview of our system's architecture.

online MCPT.

Online MCPT, unlike offline MCPT, must predict the tracking results of the current frame solely based on accumulated information from past frames. Therefore, if incorrect information is stored or tracking is done wrongly at any point, this misinformation can accumulate, leading to repeated errors in the results. First, if inaccurate or low-quality appearance features are stored, it confuses predicting the results of subsequent frames and makes accurate tracking difficult. The second issue arises when more than one global ID is assigned to the same person. In situations where a new person enters a given location or an existing person disappears, a new global ID can be assigned to the same person. Traditional online MCPT systems cannot address these issues, leading to the continuous assignment of different global IDs and, consequently, inaccurate tracking.

In this study, to overcome these limitations of previous MCPT implementations, we propose Cluster Self-Refinement (CSR), a method that periodically cleanses and corrects the stored appearance information and assigned Global IDs. Moreover, we have enhanced the utilization of the pose estimation model to enable more accurate location estimation and the storage of a wider variety of higher-quality appearance information. Our experimental results show that we achieved significant performance improvements through CSR and further enhancements via pose estimation usage, achieving a HOTA score of 60.9261% on the official test dataset, which allowed us to secure third place in the 2024 AI City Challenge Track 1.

2. Related work

Multi-camera multi-object tracking (MCMT) has been researched across various domains in recent years, including vehicle tracking [26] and people tracking [7, 8, 11, 13, 17, 19, 25]. Regardless of the domain, most MCMT systems consist of processes for object detection, single-camera tracking, and inter-camera association.

2.1. Object Detection

The performance of the object detection model is crucial in MCMT. This is because accurate bounding box predictions are essential for extracting precise appearance and location information. Various models have been researched to enable real-time usage, including those based on the You Only Look Once (YOLO)[10, 12, 22] methodology, two-stage models based on the Region Proposal Network (RPN) such as the R-CNN family[4–6, 18], and more recently, models based on the Transformer block, representing the state-of-the-art, such as Detection with Transformer (DETR)[3, 26, 28].

2.2. Multi-Object Single Camera Tracking

Single-camera tracking is a system that utilizes detection results to assign IDs to bounding boxes, thereby identifying trajectories. Research in this area continues to be actively conducted, with recent developments focusing on methods to prevent ID-switches, particularly those based on the Re-ID model proposed in DeepSort[21]. Models such as ByteTrack[27], OC-Sort[2], and BoT-Sort[1] have been introduced, building on this foundation.

2.3. Multi Camera People Tracking

Multi-camera people tracking (MCPT) is a task within MCMT focused on the people domain. This task is presented in the 2024 AI City Challenge Track 1 [24] , in which we participated this year, continuing the same task

from last year. In the previous competition, a total of 7 papers were published, with the top 5 teams using offline MCPT [7, 13, 17, 19, 25] and the bottom 2 teams using online MCPT [8, 11]. This indicates that the ability to utilize future frame predictions in offline MCPT significantly impacts performance enhancement. Notably, last year's winning paper leveraged offline advantages by storing the most relevant appearance information from the entire time frame for ID assignment and correcting any incorrectly assigned IDs by reviewing the entire trajectory. However, such offline MCPT systems have the limitation of being inapplicable in streaming contexts, posing challenges for real-world industrial applications.

Therefore, we focused on overcoming the limitations of online MCPT, which include the inability to select and store good appearance information and refine trajectories, unlike offline MCPT. To address these issues, we propose enhanced methods of utilizing pose estimation and introduce Cluster Self-Refinement.

3. Methods

In this section, we will describe our proposed methods. We are building on the online MCPT system that we proposed last year and focusing on addressing the limitations of existing online MCPT systems. In Sec. 3.1, we will summarize the method we used as a baseline, which was proposed last year, in Sec. 3.2, we will explain the enhanced method of utilizing pose estimation, and in Sec. 3.3, we will describe Cluster Self-Refinement (CSR).

3.1. Baseline Summary

Last year, we participated in a challenge where we proposed an online Multi-Camera People Tracking (MCPT) system [11]. The system, shown in Fig. 2, had three main stages: people detection, single-camera people tracking (SCPT), and inter-camera association (ICA). At every time, the process goes through the above stages sequentially, and the synchronization across all cameras must be aligned.

At a specific time t from n cameras, we input each frame into a people detection model to detect individuals in each camera's view. Subsequently, the detection results are sent to SCPT to proceed with tracking within each camera. Tracking is conducted by comparing the bounding boxes obtained from a detection model and appearance features vectorized through a re-identification model with the information of previously tracked subjects. It assigns local IDs to results that share similar locations and appearances. Finally, the tracking results from each camera are combined, and in the ICA phase, global IDs are assigned to each tracking result. A global ID is assigned to individuals appearing across all cameras, with the same person having the same global ID.

ICA is divided into two stages: Clustering and cluster tracking. The Clustering stage involves calculating the distance between each tracklet based on the appearance features and detection results received from SCPT, and then creating clusters using the Hungarian algorithm. Here, the distance is the sum of the Euclidean distance and the appearance distance. The Euclidean distance is calculated after mapping the location from a single camera onto a virtual 2D map using a homography matrix, while the appearance distance is calculated through the cosine distance between appearance features. Then, during the Cluster tracking stage, the process involves matching based on the distances between clusters that are already being tracked and those that have been newly created. The distance between two clusters is determined by calculating the average mapping location of tracklets within the cluster and the average distance of appearance features. This process is conducted online for every frame.

Our MCPT proposed last year made an important contribution by using a pose estimation model to solve the occlusion problem. Occlusion can make it difficult to accurately estimate a person's exact position on a 2D map because it can hide parts of the body, including the feet. To address this issue, we used the pose estimation model to calculate the ratio between the coordinates of body parts and the feet in the training data. When the feet were not visible, we used this ratio to estimate their coordinates. Another issue was storing appearance features when occlusion occurred. To resolve this problem, we only stored appearance features when the confidence scores of the keypoints produced by the pose estimation model were all above a certain threshold value. This ensured that only the appearance feature where the entire body was visible was stored.

The online MCPT proposed in this paper, like last year, uses YOLOv8 [10] (detector), BoT-Sort [1] (tracker), and ResNet50-IBN [23] (ReID), with the pose estimation model switched from HRNet [20] to RTMPose [9] for faster inference speed. The specific application methods will be discussed in the experiment detail section Sec. 4.1.

3.2. Enhanced Utilizing Pose Estimation

3.2.1 Angle Aware Position Estimation

When parts of the human body are obscured, the previous method [11] estimated the coordinates of the feet using predetermined ratios, as mentioned in section Sec. 3.1. However, as shown in the Fig. 3, the proportions of the human body can vary depending on the camera angle. In such cases, estimating the position of the feet in each angle using a fixed ratio can result in some discrepancies from the actual location. We attempted to solve this issue by adjusting the interpolation ratio to suit the camera angle. The interpolation ratio for each camera is updated for every frame using

Figure 3. Example image illustrating the differences in body proportions according to camera angle.

the exponential moving average (EMA) method as follows:

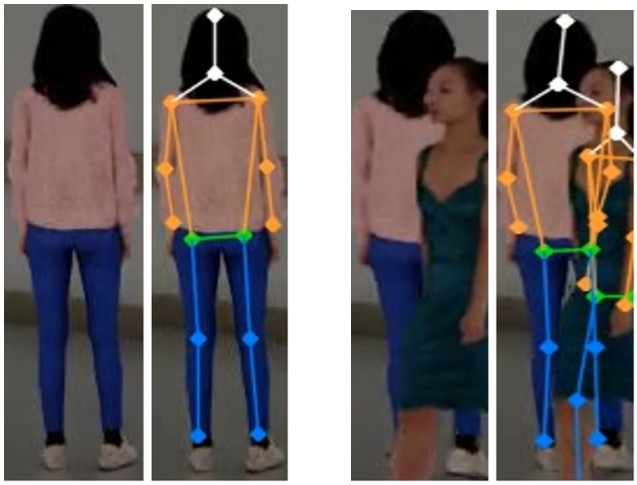
$$r_i^t = \alpha r_i^{t-1} + (1-\alpha)c_i^t$$

c_i^t represents the average interpolation ratio calculated using the keypoints of people captured by the $i-th$ camera at time t, and for accurate calculation, only keypoints with a confidence score above 0.5 are considered. r_i^t is the smoothed interpolation ratio for the $i-th$ camera at time t, calculated using the EMA method, and is used for estimating the feet coordinates of people with partially obscured bodies at that time. α is set as the momentum term at 0.9, and for all i, r_i^0 is set as the average interpolation ratio calculated based on the training and validation sets.

3.2.2 Appearance Feature Filtering

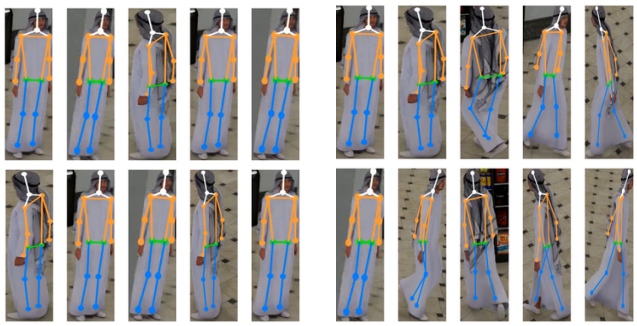
In the Inter-Camera Association (ICA) phase of previous methods, cluster tracking of the current frame was carried out using appearance features (up to 10) stored in the cluster tracklet previously. Thus, which appearance features are stored at this time significantly affects the subsequent performance of MCPT.

As illustrated in the Fig. 4, if the appearance feature of a bounding box (bbox) containing multiple people is stored, the corresponding cluster tracklet might later engage in incorrect tracking due to ID switching with another person's cluster tracklet. To prevent this, we pre-filtered the appearance features to be stored in the tracklet based on the number of keypoints contained within the bbox. The pose estimation model we utilized predicts 14 keypoints. Therefore, by not storing the appearance feature of a bounding box if it contains more than 15 keypoints, we have addressed this issue.



(a) single person in bounding box (b) two people in bounding box

Figure 4. Example image of Appearance Feature Filtering. The result from using a pose estimation model shows that (a) has 14 keypoints marked, while (b) has more than 15 keypoints marked. Therefore, (b) cannot be stored in the tracklet.



(a) w/o Procrustes analysis (b) w Procrustes analysis

Figure 5. Example image showing diverse appearance features stored using Procrustes analysis.

Moreover, as shown in Fig. 5a, if all appearance features stored within a cluster tracklet are similar, it may fail to match when the same person appears later in a different appearance. Therefore, for more accurate matching, a cluster tracklet needs to contain a variety of appearance features. To achieve this, we intend to use Procrustes analysis, a measurement that can indicate the similarity between poses. Procrustes distance is used to determine the degree of similarity between two shapes. For instance, given two matrices (keypoints) A and B, it involves finding a transformation of A that best matches B, and then determining the similarity through the difference between the transformed A and B. The transformation is based on an orthogonal rotation matrix and found using Singular Value Decomposition (SVD). The formulas are as follows:

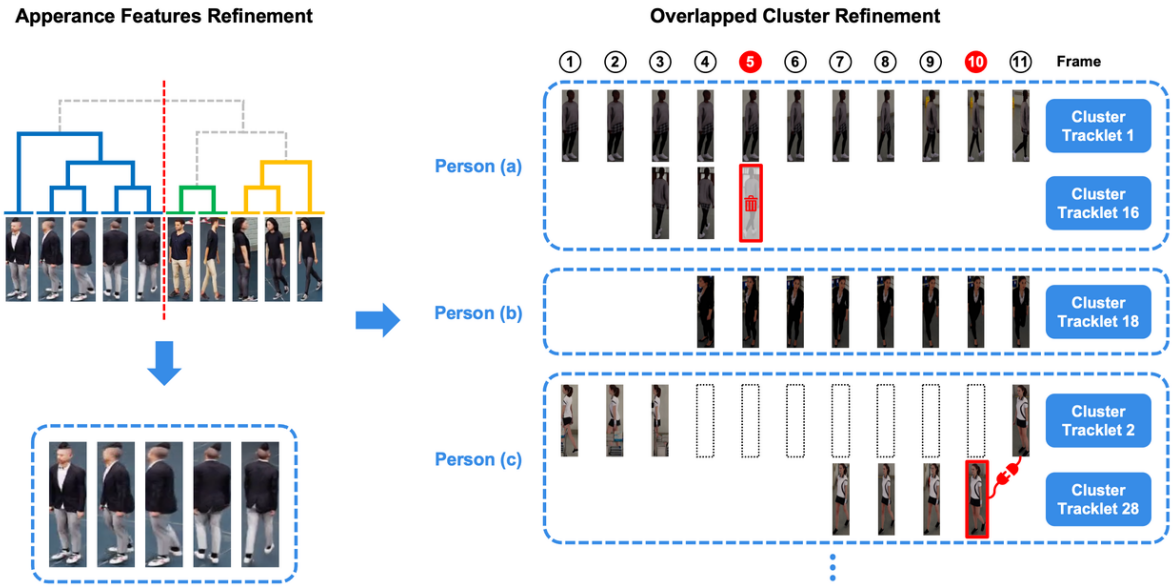


Figure 6. Overview of Cluster Self-Refinement. The left side depicts the refinement of appearance features, utilizing agglomerative clustering to check if different people are stored and, if correct, refine the appearance features in the cluster tracklet. The right side illustrates overlapped cluster refinement, addressing situations where one person has more than one global ID. The CSR procedure is carried out at regular intervals, as denoted by the red circle shown above.

1. $\text{SVD}(B^T A) = U\Sigma V^T$,
2. $R = UV^T$ (from step 1),
3. $D_p(A, B) = \sqrt{\sum (RA - B)^2}$ (using R from step 2).

where A, B are 2-dimensional matrices representing the coordinates of keypoints, D_p denotes the Procrustes distance between the two matrices, and R is the orthogonal rotation matrix that rotates A to be most similar to B. We measure the distance between a keypoint we wish to additionally store and the already stored keypoints using this distance function. If the distance to all stored keypoints exceeds a threshold value, it indicates a different pose and thus is stored; if the distance to any stored keypoint is less than the threshold value, it indicates a similar pose is already stored, and thus it is not stored. By using Procrustes analysis for appearance feature filtering, we can obtain a tracklet composed of people in various appearances, as shown in the Fig. 5b.

3.3. Cluster Self-Refinement

In the Online MCPT, the allocation of multiple global IDs to the same person or storing inaccurate information (such as appearance features) into a cluster tracklet can significantly impact the performance of MCPT. Since it can not correct duplicated (incorrect) global IDs or erroneous information within tracklets using future scenes like Offline Tracking. Hence, performing refinements using only the information gathered thus far can enhance online MCPT. As depicted in Fig. 6, we have implemented an CSR procedure for the currently monitored cluster tracklets, periodically executing two specific steps in sequence.

3.3.1 Appearance Features Refinement

A cluster tracklet should only store the appearance features of a single person. If the appearance features of several people be stored within a tracklet, as illustrated in Fig. 6, there's a risk of ID Switching occurring with another cluster tracklet, which could compromise the quality of future tracking. Therefore, we first use agglomerative clustering with cosine distance as the metric to divide the stored appearance features in the tracklet into two feature clusters. Then, by measuring the cosine distance between the two feature clusters and finding it exceeds a certain threshold, we infer that the cluster tracklet consists of different people, leading to the deletion of appearance features from the feature cluster stored later. Even if more than three different people are present in the cluster tracklet, using agglomerative clustering reduces the likelihood, as shown in the Fig. 6, of a single person being split into both feature clusters. After deleting one feature cluster, even if two people remain in the remaining cluster, the periodic execution of CSR ensures that eventually, only the appearance features of a single person remain in a cluster tracklet.

(a) Train set (b) Test set(under 80) (c) Test set (above 81)

Figure 7. This compares the camera views of the train data and the test data. It can be observed that different views appear for scene numbers above 81.

3.3.2 Overlapped Cluster Refinement

In this phase, we verify whether the person corresponding to the newly added cluster tracklet is the same as the one being tracked in an existing tracklet. In other words, we review whether multiple cluster tracklets are tracking the same person and then proceed with a refinement process, where we retain the original tracklet and delete the rest. We aimed to address two scenarios that result in the creation of overlapped cluster tracklets: cases where the person in the newly added cluster tracklet is already being tracked and cases where the person was being tracked but then lost. We will explain how we handled these two scenarios in turn. A lost cluster tracklet refers to a tracklet that was being tracked but then disappeared, and it has been stored for reference.

Firstly, we take a pair of currently tracked cluster tracklets and conduct a comparison. We measure the average cosine distance between the appearance features stored in each tracklet and the Euclidean distance between the 2D coordinates of the two tracklets. If both distances are below a certain threshold, we consider the two tracklets to be tracking the same person and delete one cluster as shown person (a) in the Fig. 6. In this case, we delete the cluster tracklet with the larger global ID, meaning the one that was added later. This process is repeated for all pairs of cluster tracklets.

Secondly, after the above process is completed, we take one tracked cluster tracklet and one lost cluster tracklet and compare them. Here, unlike the first method, we only measure the average cosine distance between the appearance features stored within each tracklet. This is because the 2D coordinates of the lost cluster tracklet are values stored in the past and therefore do not match the current timeline. Thus, if the measured cosine distance is below a certain threshold, we consider the two tracklets to belong to the same person's cluster tracklet. At this point, we also delete the later-added tracked cluster tracklet and update its location information in the lost cluster tracklet, restarting the tracking process as shown person (c) in the Fig. 6.

	# Scenes	# Cameras	# People	Frames
Train	40	360	1,045	23,994
Val	20	174	601	23,994
Test	30	418	-	23,994

Table 1. The statistics of the datasets for Challenge Track 1

4. Experiment

4.1. Experiment Details

We conducted experiments using only the dataset from Challenge Track 1 [24]. Unlike last year [16], where real data was included in the test set, this year's training, validation, and test datasets were all composed entirely of synthetic data in multiple indoor settings generated using the NVIDIA Omniverse Platform. As shown in Tab. 1, it consisted of more cameras and people than last year.

In this MCMT track, team rankings on the leaderboard are determined by the HOTA score [15], which addresses limitations of previous metrics like MOTA and IDF1 by integrating accurate detection, association, and localization into a unified measure. HOTA is calculated as a combination of detection accuracy (DetA) and ID association accuracy (AssA). Additionally, in this challenge, global 2D coordinates on a 2D map were used as the location of the object, not the coordinates of the bounding box on a camera.

For people detection, single-camera people tracking, ReID, and pose estimation models, we used Yolov8x [10], BoT-SORT [1], ResNet50-IBN [23], and RTMPose-m [9], respectively. Among these, in people detection, as seen in Fig. 6, some views of the training and test data were very different, making detection with models fine-tuned on the training data ineffective. Therefore, for scenes after scene 81, we used a model pretrained on COCO [14]. Furthermore, RTMPose was pre-trained using the CrowdPose dataset.

Method	HOTA	DetA	AssA	LocA
Baseline	57.86	67.43	50.16	90.28
+ CSR	60.02	67.58	53.71	90.30
+ CSR + EUP	60.93	68.37	54.96	90.62

Table 2. The results of ablation study on using a CSR and EUP. CSR and EUP stand for Cluster Self-Refinement and Enhanced Utilizing Pose estimation respectively.

Rank	Team ID	Team Name	HOTA
1	221	RIIPS	71.94
2	79	SJTU-Lenovo	67.22
3	**40 (ours)**	**NetsPresso**	**60.93**
...	...	...	...

Table 3. Public leaderboard for the Challenge Track 1

4.2. Experiment Results

As can be seen in Tab. 2, the application of Cluster Self-Refinement significantly improved performance from the baseline. Notably, as intended, the reduction in incorrect ID matching increased AssA. Furthermore, performance was also enhanced through the Enhanced Utilizing Pose Estimation.

In Tab. 3, We submitted our proposed system for public evaluation in the AI City Challenge Track 1 and secured 3rd place out of 17 participating teams with a HOTA score of 60.93

5. Conclusion

In this paper, we propose a tracking self-diagnosis method called Cluster Self-Refinement applied to online MCPT. This method enables the application of the advantage of reviewing and modifying the entire tracklet of offline MCPT to online MCPT, by periodically reviewing and refining the information of past stored tracklets, enabling more accurate tracking. Moreover, it has expanded the use of pose estimation beyond the existing baseline models, further improving tracking accuracy. Our proposed method was ranked third in the 2024 AI City Challenge Track 1 [24].

References

[1] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking. *arXiv preprint arXiv:2206.14651*, 2022. 2, 3, 6

[2] 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*, pages 9686–9696, 2023. 2

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 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.

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2

[7] Hsiang-Wei Huang, Cheng-Yen Yang, Zhongyu Jiang, Pyong-Kun Kim, Kyoungoh Lee, Kwangju Kim, Samartha Ramkumar, Chaitanya Mullapudi, In-Su Jang, Chung-I Huang, et al. Enhancing multi-camera people tracking with anchor-guided clustering and spatio-temporal consistency id re-assignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5238–5248, 2023. 1, 2, 3

[8] Yuntae Jeon, Dai Quoc Tran, Minsoo Park, and Seunghee Park. Leveraging future trajectory prediction for multi-camera people tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5398–5407, 2023. 1, 2, 3

[9] Tao Jiang, Peng Lu, Li Zhang, Ningsheng Ma, Rui Han, Chengqi Lyu, Yining Li, and Kai Chen. Rtmpose: Real-time multi-person pose estimation based on mmpose. *arXiv preprint arXiv:2303.07399*, 2023. 3, 6

[10] Glenn Jocher. Yolov8, 2023. https://github.com/ultralytics/ultralytics. 2, 3, 6

[11] Jeongho Kim, Wooksu Shin, Hancheol Park, and Jongwon Baek. Addressing the occlusion problem in multi-camera people tracking with human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5462–5468, 2023. 1, 2, 3

[12] Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, Yiduo Li, Bo Zhang, Yufei Liang, Linyuan Zhou, Xiaoming Xu, Xiangxiang Chu, Xiaoming Wei, and Xiaolin Wei. Yolov6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*, 2022. 2

[13] Zongyi Li, Runsheng Wang, He Li, Bohao Wei, Yuxuan Shi, Hefei Ling, Jiazhong Chen, Boyuan Liu, Zhongyang Li, and Hanqing Zheng. Hierarchical clustering and refinement for generalized multi-camera person tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5519–5528, 2023. 1, 2, 3

[14] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 6

[15] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International journal of computer vision*, 129:548–578, 2021. 6

[16] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Yue Yao, Liang Zheng, Mohammed Shaiqur Rahman, Meenakshi S. Arya, Anuj Sharma, Qi Feng, Vitaly Ablavsky, Stan Sclaroff, Pranamesh Chakraborty, Sanjita Prajapati, Alice Li, Shangru Li, Krishna Kunadharaju, Shenxin Jiang, and Rama Chellappa. The 7th ai city challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 5538–5548, 2023. 1, 6

[17] Quang Qui-Vinh Nguyen, Huy Dinh-Anh Le, Truc Thi-Thanh Chau, Duc Trung Luu, Nhat Minh Chung, and Synh Viet-Uyen Ha. Multi-camera people tracking with mixture of realistic and synthetic knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5495–5505, 2023. 1, 2, 3

[18] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015. 2

[19] Andreas Specker and Jürgen Beyerer. Reidtrack: Reid-only multi-target multi-camera tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5441–5451, 2023. 1, 2, 3

[20] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[21] Balaji Veeramani, John W Raymond, and Pritam Chanda. Deepsort: deep convolutional networks for sorting haploid maize seeds. *BMC bioinformatics*, 19:1–9, 2018. 2

[22] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*, 2022. 2

[23] Guanshuo Wang, Yufeng Yuan, Xiong Chen, Jiwei Li, and Xi Zhou. Learning discriminative features with multiple granularities for person re-identification. In *Proceedings of the 26th ACM international conference on Multimedia (MM)*, 2018. 3, 6

[24] Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Yue Yao, Liang Zheng, Mohammed Shaiqur Rahman, Meenakshi S. Arya, Anuj Sharma, Pranamesh Chakraborty, Sanjita Prajapati, Quan Kong, Norimasa Kobori, Munkhjargal Gochoo, Munkh-Erdene Otgonbold, Ganzorig Batnasan, Fady Alnajjar, Ping-Yang Chen, Jun-Wei Hsieh, Xunlei Wu, Sameer Satish Pusegaonkar, Yizhou Wang, Sujit Biswas, and Rama Chellappa. The 8th AI City Challenge. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024. 1, 2, 6, 7

[25] Wenjie Yang, Zhenyu Xie, Yaoming Wang, Yang Zhang, Xiao Ma, and Bing Hao. Integrating appearance and spatial-temporal information for multi-camera people tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5259–5268, 2023. 1, 2, 3

[26] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 2

[27] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *European conference on computer vision*, pages 1–21. Springer, 2022. 2

[28] 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