Efficient Online Multi-Camera Tracking with Memory-Efficient Accumulated Appearance Features and Trajectory Validation

Lap Quoc Tran* Huan Duc Vi*
Asilla
{laptq, huan}@asilla.net *

Abstract

Multi-camera tracking (MCT) plays a crucial role in various computer vision applications. However, accurate tracking of individuals across multiple cameras faces challenges, particularly with identity switches. In this paper, we present an efficient online MCT system that tackles these challenges through online processing. Our system leverages memory-efficient accumulated appearance features to provide stable representations of individuals across cameras and time. By incorporating trajectory validation using hierarchical agglomerative clustering (HAC) in overlapping regions, ID transfers are identified and rectified. Evaluation on the 2024 AI City Challenge Track 1 dataset [39] demonstrates the competitive performance of our system, achieving accurate tracking in both overlapping and non-overlapping camera networks. With a 40.3% HOTA score [29], our system ranked 9th in the challenge. The integration of trajectory validation enhances performance by 8% over the baseline, and the accumulated appearance features further contribute to a 17% improvement.

1. Introduction

Multi-camera tracking (MCT) focuses on detecting and tracking individuals across a multi-camera network, which can have overlapping or non-overlapping fields of view. MCT plays an important role in the research community because it facilitates real-world applications in monitoring public spaces, managing crowds, analyzing human behavior, and detecting anomalous activities.

A typical MCT system consists of two primary components: single-camera tracking (SCT) and cross-camera association (CCA). Most SCT approaches adopt the tracking-by-detection method. Initially, an object detector localizes individuals within the camera frames, and a person re-identification (Re-ID) module extracts distinctive appearance features for these detections. Consecutive frame detections are then associated to form single-camera tracklets if they are predicted to belong to the same person. Next, the CCA step links tracklets observed across different cameras to create multi-camera tracks that represent individuals' movement throughout the camera network. This association process typically utilizes motion and appearance cues.

MCT poses significant challenges from both an accuracy and implementation perspective. In terms of accuracy, MCT systems often suffer from identity (ID) switches where the IDs of tracked objects are incorrectly assigned or swapped. Two specific types of ID switches can occur: ID transfer and ID ascension. In an ID transfer, the ID of an object is transferred from one tracked object to another. By contrast, in an ID ascension, the ID of an object is incorrectly incremented to a higher ID value, leading the tracker to falsely predict the appearance of a new object. Typically, a dissimilarity threshold balances ID transfer and ID ascension, where a lower threshold reduces ID transfer but may promote ID ascension. This trade-off arises because factors like camera viewpoint, occlusion, brightness variations, and calibration errors can cause high dissimilarity between detections of the same person while exhibiting low dissimilarity between different individuals.

To address these challenges, many recent MCT algorithms adopt a batch-based approach, utilizing information from successive frames to predict the current frame's results. However, despite improved performance, batch-based algorithms require extensive computations, making them unsuitable for real-world systems that demand online and real-time processing. Consequently, online tracking algorithms that rely solely on past frames' information are actively being researched.

In this study, we propose an efficient online MCT system designed for both overlapping and non-overlapping camera networks. Our system can effectively detect and handle ID switch issues. At each time instance, our system first employs an online SCT module to detect individuals and associate detections within a single camera view. Then, for the same time instance, we link these

*These authors contributed equally to this work.

single-camera tracks across different cameras into a global track representing a unique identity by using a hierarchical logic that incorporates appearance and spatio-temporal cues. To address ID switches, our system introduces the use of memory-efficient, high-quality accumulated appearance features, and trajectory validation in overlapping areas. Specifically, to prevent ID transfers, we accumulate appearance features from all previous frames for each track and use this accumulated representation to build the dissimilarity matrix before association. Compared to exponential moving average features or averages from limited feature banks, our accumulated features are more generalized and stable for each person across cameras and time, allowing the use of smaller dissimilarity thresholds without a significant increase in ID ascension errors. Additionally, we propose a trajectory validation step using Hierarchical Agglomerative Clustering (HAC) in overlapping regions to identify and correct already occurred ID transfers.

We evaluated our online system on the 2024 AI City Challenge Track 1 dataset [39], achieving the 9th place with a 40.3% HOTA score on the test set, where trajectory validation contributed an improvement of 8% and accumulated appearance feature contributed an improvement of 17% over our baseline version regarding Association Accuracy. Our key contributions are:

1. An online MCT system performing well on overlapping and non-overlapping cameras.
2. Using memory-efficient accumulated appearance features with smaller dissimilarity thresholds to prevent ID transfers.
3. A trajectory validation method based on HAC to spot and fix occurred ID transfers.

2. Related Works

2.1. Object Detection

Object detection models play a crucial role in the tracking-by-detection paradigm by localizing humans in image frames. Several state-of-the-art models have been proposed, including YOLOv5 [20], YOLOv6 [26], YOLOv7 [37], and YOLOv8 [21]. In recent years, there has been growing interest in exploring the application of transformer-based architectures, which have achieved remarkable success in natural language processing, to object detection. This has led to the development of models like DETR [4], Deformable-DETR [58], DN-DETR [28], YOLOS [12], and DINO-DETR [49].

2.2. Image-based Person Re-Identification

Image-based person Re-ID aims to retrieve people with the same person across different images. It involves constructing a gallery that stores samples of identities observed previously, which serves as a reference for comparison with future queries. Deep feature learning has been a prominent focus in person Re-ID research. Various approaches [2, 3, 9, 14, 30, 36, 56] have been proposed to learn more discriminative features for person Re-ID. Additionally, researchers have explored different loss functions to guide feature learning, with works by [10, 17, 31], among others, contributing to this area. To optimize the ranking order of results, ranking optimization techniques have been employed. These techniques aim to improve the ordering of ranked lists through approaches such as re-ranking [38, 54, 55] and rank fusion [47, 53] methods.

2.3. Single-camera Tracking

Single-camera tracking (SCT) can be classified into two main types: tracking-by-detection and joint-detection tracking. The tracking-by-detection paradigm involves detecting objects in each frame and then associating them across frames to form tracks. Several effective tracking-by-detection methods have been developed, including BoT-SORT [1], DeepSORT [43], Bytetrack [52], and StrongSORT [11]. These methods utilize object detection outputs to track objects over time, employing techniques such as Kalman filters and data association algorithms.

Joint-detection tracking integrates object detection and person Re-ID into a unified framework. By jointly learning both tasks, these methods aim to improve tracking performance. SiamMOT [33], JDE [41], FairMOT [51], and CenterTrack [57] are examples of joint-detection tracking methods. They leverage the complementary information from object detection and Re-ID to enhance the accuracy and robustness of tracking.

2.4. Multi-camera Tracking

Multi-camera tracking (MCT) approaches can be broadly classified into online and offline methods. Online approaches [13, 34, 50] handle the task frame-by-frame, while offline trackers perform the task as a post-processing step using the outputs of single-camera tracking [15, 23, 35, 46]. In MCT, appearance similarity plays a significant role in matching tracklets. Many approaches [18, 19, 25, 32, 48] leverage embedding feature vectors to compute appearance similarity. However, relying solely on appearance features may lead to identity switches. To enhance performance, recent works [5, 8, 15, 27, 44, 45] have incorporated additional constraints, such as camera topology, temporal information, and motion rules. Additionally, graph-based approaches have been employed, where associations across frames and cameras are determined using graph structures [6, 7, 16, 42].

3. Method

Figure 1 provides an overview of our proposed online system.

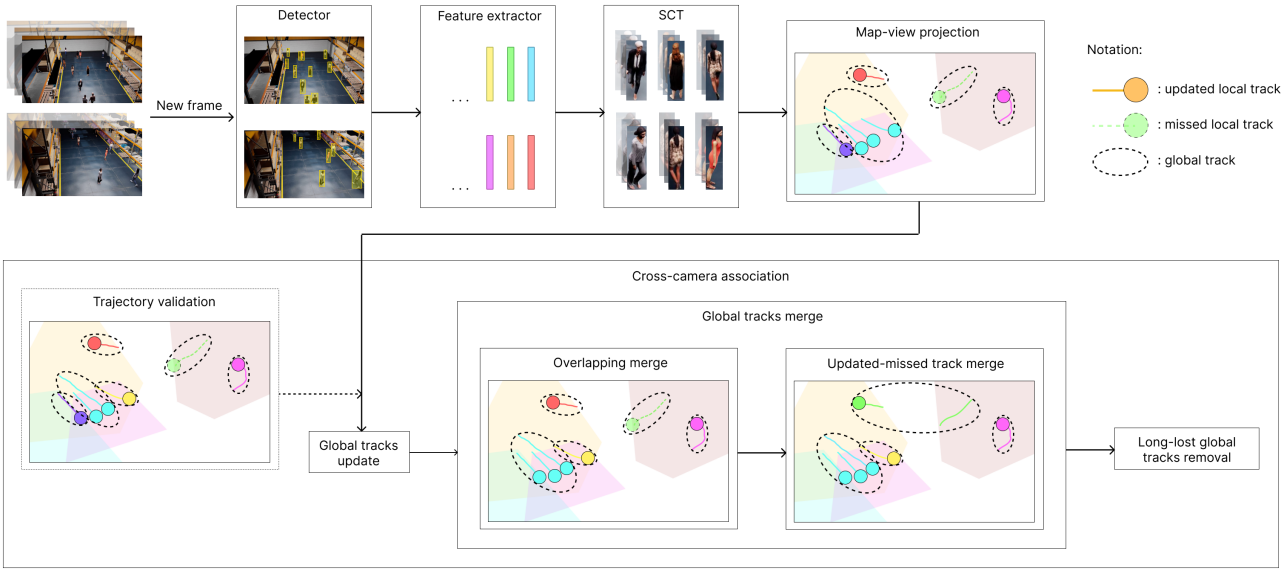

Figure 1. An overview of our proposed online MCT system. Our system employs a state-of-the-art object detector, Re-ID model, and online SCT method while developing a CCA module that works "online". Regarding the CCA module, at each time instance and after the SCT step, our system predicts which tracklets belong to the same person in order to merge them into a single track. In the *overlapping merge* step, our system considers tracklets that are moving in close proximity to each other for the merge. In the *updated-missed track merge* step, our system considers a pair of tracklets from which one is present (illustrated by a solid line) and the other is missed (illustrated by a dashed line).

Incoming frames go through a sequence of steps in our system: person detection, feature extraction, single-camera tracking, map-view projection, global track update, global track merge, and removal of long-lost global tracks. First, we employ a person detector to obtain person bounding boxes from each camera view. The location of the person on the image frame is represented by a point, which can be either the midpoint of the bottom edge for rectangle bounding boxes or a foot point for human pose. For each detected bounding box, we extract discriminative appearance features using a ReID model and normalize them to have a norm of 1. This visual feature is used in both the single-camera tracking (SCT) and the cross-camera association (CCA) steps. For each time instance, we employ a tracking-by-detection SCT module to update existing (or create new) local tracks with new detections for each camera. Using homography matrices, we project the current location of every local track from all cameras onto a shared map-view perspective.

The CCA module maintains a list of global tracks, each containing one or several local tracks on the map. Local tracks belonging to the same global track are expected to represent the same person but from different camera views. At each time instance, the CCA module starts by observing the state of local tracks after the SCT step to update the representative state for each global track. Importantly, global tracks are then merged with each other if they are expected to represent the same person. Further details about the CCA module are presented in Section 3.1.

3.1. Cross-camera Association

At each time instance t, this module maintains a list of M_t global tracks $G_t = \{g_{t,i} | i = \overline{1 \dots M_t}\}$ and N_t local tracks $L_t = \{l_{t,j}^{(k)} | j = \overline{1 \dots N_t}\}$ where $k = \overline{1 \dots M_t}$ is the index of a global track. M_t and N_t change over time. A local track $l_{t,j}^{(k)}$ can be assigned to one and only one global track, while a global track $g_{t,i}$ can contain one or many local tracks, which means $g_{t,i} = \{l_{t,j}^{(k)} | j = \overline{1 \dots N_t}$ and $k = i\}$. At each time instance, there are three sequential steps involved: global track update, global track merge, and removal of long-lost tracks.

3.1.1 Global Track Update

At each time instance t, a local track returned by SCT can fall into one of three categories:

- Newly created local track: If a local track is newly created by the SCT, our system will create a new global track, mark it as *updated*, and assign the local track to this global track. The newly created global track becomes a candidate for merging with other global tracks in the global track merge steps.
- Updated local track: If an existing local track is updated with detection at the current time instance, the corre-

sponding global track that it's assigned to is also marked as *updated*.

- Missed local track: If an existing local track misses detection at the current time instance, and every local track assigned to the corresponding global track belongs to this category, the global track is marked as *missed*.

The states of global tracks defined in this step are used to determine candidate tracks to be merged in the next step. Additionally, for each global track $g_{t,i}$, a representative location $p_{g_{t,i}}$ on the map is extrapolated by averaging the locations $p_{l_{t,j}}$ of its assigned local tracks:

$$p_{g_{t,i}} = \frac{\sum_j p_{l_{t,j}^{(i)}}}{|g_{t,i}|}$$

3.1.2 Overlapping Merge

In a camera system with overlapping fields of view, it is possible for a person to be present in multiple cameras simultaneously, resulting in multiple tracks on the shared map. The overlapping merge step aims to group these tracks into a single track.

At each time instance, our online system retrieves a list of global tracks that are not marked as *missed* within a short time window (e.g., a few seconds). A pairwise dissimilarity matrix is then constructed among these global tracks. The dissimilarity between two global tracks is calculated based on both the appearance distance and the trajectory distance between the tracks as follows.

The appearance distance between two global tracks is computed by aggregating the appearance distances among their respective local tracks. Cosine dissimilarity is used for this purpose. Assume the appearance feature of a local track is $F_{l_{t,j}}$, then the appearance distance $\Delta\text{AP}_{g_{t,x},g_{t,y}}$ between two global track $g_{t,x}$ and $g_{t,y}$ is computed:

$$\Delta\text{AP}_{g_{t,x},g_{t,y}} = \frac{\sum_j \sum_q 1 - \text{cosine\_similarity}(\frac{F_{l_{t,j}^{(x)}}}{\|F_{l_{t,j}^{(x)}}\|}, \frac{F_{l_{t,q}^{(y)}}}{\|F_{l_{t,q}^{(y)}}\|})}{|g_{t,x}| \times |g_{t,y}|}$$

The trajectory distance between two global tracks $g_{t,x}$ and $g_{t,y}$ is determined based on the representative locations $p_{g_{t,x}}$ and $p_{g_{t,y}}$ obtained in the global track update step, as described in Section 3.1.1. The discrete Fréchet distance is employed in our online system to measure the trajectory distance, considering the locations from the 10 most recent detections of each local track.

$$\Delta\text{TR}_{g_{t,x},g_{t,y}} = \text{discrete\_Frechet}(\{p_{g_{t-i,x}} | i = \overline{0 \dots 9}\}, \{p_{g_{t-i,y}} | i = \overline{0 \dots 9}\})$$

Algorithm 1 Overlapping merge

Input: Global track list G_t.
Output: Some $g_{t,i}$ in G_t are merged together.

construct cost matrix $C_{M_t \times M_t} = \infty$
for each $g_{t,x}$ in G_t **do**
  for each $g_{t,y}$ in G_t **do**
    if $g_{t,x}$ is $g_{t,y}$ **then**
      continue
    end if
    if is_missed($g_{t,x}$) or is_missed($g_{t,y}$) **then**
      continue
    end if
    $C[x, y] \leftarrow \Delta\text{OL}_{g_{t,x},g_{t,y}}$
  end for
end for
clusters $\leftarrow$ AgglomerativeClustering(C, θ_{OL})
for each *cluster* in *clusters* **do**
  oldest $\leftarrow$ get oldest global track from *cluster*
  for each $g_{t,x}$ in *cluster* **do**
    merge $g_{t,x}$ to *oldest*
    remove $g_{t,x}$ from G_t
  end for
end for

Subsequently, a dissimilarity matrix is constructed using a fusion of the appearance distance and the trajectory distance.

$$\Delta\text{OL}_{g_{t,x},g_{t,y}} = \alpha_{\text{OL}} \cdot \Delta\text{AP}_{g_{t,x},g_{t,y}} + (1 - \alpha_{\text{OL}}) \cdot \sigma_1(\Delta\text{TR}_{g_{t,x},g_{t,y}})$$

where $\sigma_1(x) = \frac{1}{1 + e^{-\frac{x}{15}}} - 0.5$ is a compression function.

The Hierarchical Agglomerative Clustering algorithm (HAC) is applied to this dissimilarity matrix, with a predefined dissimilarity threshold θ_{OL}, to identify clusters of related global tracks. HAC is a bottom-up hierarchical clustering approach that begins by considering each data point as a separate cluster. The algorithm then iteratively merges the most similar clusters based on the pair-wise dissimilarity between clusters, continuing this process until all data points or clusters are merged into a single cluster or until the dissimilarity between any two clusters exceeds the predefined threshold. The use of HAC with a predefined threshold is suitable for online multi-camera tracking systems, as the number of clusters is typically not known in advance. We demonstrate our algorithm in Algorithm 1.

Finally, the system merges the global tracks within each cluster. A new global track is created, incorporating all the local tracks associated with the original global tracks in the cluster. The presence of the original global tracks is replaced by the presence of the newly created global track.

3.1.3 Updated-Missed Tracks Merge

The updated-missed tracks merge step addresses cases where a person is missed in tracking but then reappears with a new global ID. This situation often occurs when a person moves between disjoint cameras in a network or when a person temporarily disappears and reappears in the same camera.

Based on the global track status determined in the global track update step (Section 3.1.1), the system retrieves two lists: the *updated* global tracks and the *missed* global tracks. For each pair of an *updated* global track and a *missed* global track, the system calculates four values to build dissimilarity matrices: appearance distance, spatial distance, speed distance, and fusion distance. The spatial distance ΔST is computed as the Euclidean distance between the location of the first detection of the *updated* global track $g_{t,x}$ and the location on the map estimated by Kalman filter [22] of the *missed* global track $g_{t,y}$:

$$\Delta\text{ST}_{g_{t,x},g_{t,y}} = \text{euclid}(p_{g_{t_{x,\text{first}},x}}, \text{Kalman}(p_{g_{t_{y,\text{last}},y}}))$$

where $t_{x,\text{first}}$ and $t_{x,\text{last}}$ are the time instances when global track x is seen for the first and last time.

The speed distance ΔSP represents the ratio between the speed required to travel from the last detection of the *missed* global track $g_{t,y}$ to the first detection of the *updated* global track $g_{t,x}$ and the average speed of the *missed* global track $g_{t,y}$ during its tracking period:

$$\Delta\text{SP}_{g_{t,x},g_{t,y}} = \frac{\text{euclid}(p_{g_{t_{x,\text{first}},x}}, p_{g_{t_{y,\text{last}},y}})}{(t_{x,\text{first}} - t_{y,\text{last}}) \times \text{average\_speed}(g_{t,y})}$$

The fusion distance ΔFU is the fusion of the appearance feature distance and the spatial distance:

$$\Delta\text{FU}_{g_{t,x},g_{t,y}} = \alpha_{\text{UM}} \cdot \Delta\text{AP}_{g_{t,x},g_{t,y}} + (1-\alpha_{\text{UM}}) \cdot \sigma_2(\Delta\text{ST}_{g_{t,x},g_{t,y}})$$

where $\sigma_2(x) = \dfrac{1}{1+e^{-\frac{x}{30}}} - 0.5$ is a compression function.

As indicated in Algorithm 2, in our online system, we employ a 2-round matching scheme to handle different scenarios and improve the accuracy of the tracking process. Each round serves a specific purpose and utilizes different criteria for matching. In the first round, the Hungarian algorithm [24] is applied using the fusion distance ΔFU to merge pairs of matched global tracks, while filtering out pairs with an appearance distance ΔAP exceeding a predefined appearance dissimilarity threshold θ_{AP}. This round is particularly effective for tracking individuals who have been lost for a relatively short period, where a short-term Kalman prediction [22] is still reliable. This idea is similar to the SORT-like approaches, which also adopt Kalman filter [22] for the short-term association.

Algorithm 2 Updated-missed tracks merge

Input: Global track list G_t.
Output: Some $g_{t,i}$ in G_t are merged together.

construct cost matrix $C_{\text{AP}} = \infty$, $C_{\text{ST}} = \infty$, $C_{\text{SP}} = \infty$
for each $g_{t,x}$ in G_t **do**
  for each $g_{t,y}$ in G_t **do**
    if !is_missed($g_{t,x}$) and is_missed($g_{t,y}$) **then**
      $C_{\text{AP}}[x,y] \leftarrow \Delta\text{AP}_{g_{t,x},g_{t,y}}$
      $C_{\text{ST}}[x,y] \leftarrow \Delta\text{ST}_{g_{t,x},g_{t,y}}$
      $C_{\text{SP}}[x,y] \leftarrow \Delta\text{SP}_{g_{t,x},g_{t,y}}$
    end if
  end for
end for
$C_{\text{FU}} \leftarrow \alpha_{\text{UM}} \cdot \text{C}_{\text{AP}} + (1 - \alpha_{\text{UM}}) \cdot \sigma_2(\text{C}_{\text{ST}})$
matches $\leftarrow$ Hungarian(C_{FU})
matches_1 $\leftarrow$ filter(*matches*, C_{AP}, θ_{AP})
matches $\leftarrow$ Hungarian(C_{AP})
matches_2 $\leftarrow$ filter(*matches*, C_{SP}, θ_{SP})
matches $\leftarrow$ *matches_1* $\cup$ *matches_2*
for each (updated $g_{t,x}$, missed $g_{t,y}$) in *matches* **do**
  merge $g_{t,x}$ to $g_{t,y}$
  remove $g_{t,x}$ from G_t
end for

In the second round, the Hungarian algorithm [24] is applied again, but this time using the appearance distance ΔAP as the matching criterion, while filtering out pairs with speed distance ΔSP exceeding a predefined speed dissimilarity threshold θ_{SP}. This round aims to track individuals who have been lost for a long time by relying primarily on appearance information. During extended periods of loss, appearance information becomes crucial for re-establishing track associations.

3.1.4 Removal of Long-Lost Tracks

In an open-world scenario, individuals may exit the camera network and not revisit for a long time. The long-lost tracks removal step aims to remove global tracks that have been inactive for a long period, indicating that the corresponding person is no longer present in the scene. This step helps prevent the ID transfer and reduces computational overhead.

For each global track marked as *missed* in the global track update step (Section 3.1.1), the system checks the number of consecutive missed detections. If this number exceeds a `max_frame_skipped` period, the global track is removed from the system.

3.2. Accumulated Appearance Feature

In Section 3.1.2 we mentioned the appearance feature of a local track $F_{l_{t,j}}$, which is used to compute the appearance distance between two global tracks, without going into de-

tails how this appearance feature is computed. A local track is presented by a sequence of detections. There are several possible options to compute the appearance feature of a local track using the appearance features of its detections:

- Each local track maintains a single vector of exponential moving average (EMA) appearance feature, and this single vector is used to calculate the appearance distance. This is also the approach used in StrongSORT [11].
- Each local track maintains a bank of feature vectors, and the appearance distance is computed by obtaining the average of the feature bank. This is also the approach used in DeepSORT [43].

However, neither of these strategies is suitable for long-term tracking. The EMA appearance feature, while memory-efficient by requiring the maintenance of a single vector over time, is particularly sensitive to recent environmental changes. In practice, most failures in appearance matching occur due to appearance misrepresentation caused by factors like occlusion and illumination variations. For instance, the EMA appearance features of the same person may differ significantly when they move from a partially occluded or dark area in one camera to an unobstructed or brighter area in another camera. Conversely, two different people may have similar EMA appearance features if they both recently stayed in a dark area.

On the other hand, maintaining a bank of appearance features throughout time and averaging them results in a more generalized and robust feature that is less sensitive to noise. However, storing such a feature bank is impractical due to memory limitations. For example, a single feature bank for 1 hour of video at 30 frames per second, using a float32 feature vector of 512 dimensions, would occupy more than 200 megabytes.

To address these challenges, we propose a memory-efficient accumulated appearance feature for each local track $l_{t,j}$ at time instance t, as follows:

$$F_{l_{t,j}} = F_{l_{t-1,j}} + \frac{f_{l_{t,j}}}{\|f_{l_{t,j}}\|}$$

where $f_{l_{t,j}}$ is the appearance feature of the detection at time t of $l_{t,j}$. By normalizing the appearance feature vectors before adding them up, we can eliminate the concern about overflow error.

Figure 2 demonstrates the comparison of appearance distance in a pair of a positive pair and a negative pair extracted from the 2024 AI City Challenge Track 1 dataset, using the 3 mentioned methods.

Moreover, to improve the quality of the appearance feature, we suggest filtering the person detections to retain only high-quality detections that are minimally influenced by factors like lighting conditions or significant occlusion. Several techniques can be employed for this purpose, including utilizing a pose estimation model to ensure the visibility of discriminative body parts, evaluating the contrast of the cropped image, or simply selecting detections with high confidence scores from the detection model. In our online system, we adopt the approach of selecting detections with high confidence scores.

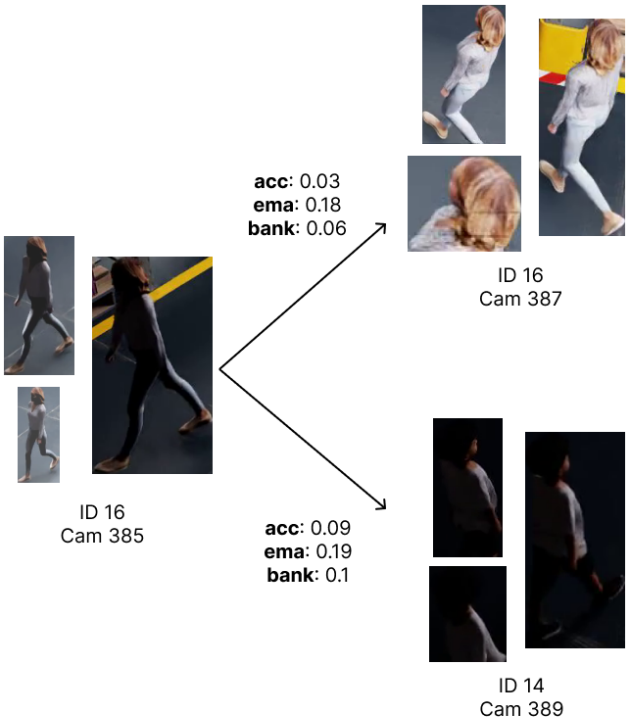

Figure 2. Comparison of appearance distances in a positive pair and a negative pair, extracted from the 2024 AI City Challenge Track 1 dataset. The appearance distance is measured using cosine dissimilarity. In the figure, **acc** represents our proposed method using accumulated features. **ema** represents the method using exponential moving average features with a smoothing factor of 0.1. **bank** represents the method using a feature bank with a size of 500. Our proposed method provides better separation and allows a stricter dissimilarity threshold.

By obtaining high-quality and memory-efficient appearance features for each local track, we reduce noise in the appearance feature over time, resulting in more distinct visual clusters for different identities. Consequently, we can choose a smaller appearance dissimilarity threshold to effectively eliminate ID transfers without increasing the occurrence of ID ascension. Experimental results supporting this assumption are provided in Section 4.4.

3.3. Trajectory validation

To detect and address the issue of ID transfer, our system proposes an approach that leverages the overlapping areas between cameras. As shown in Figure 1, prior to updating global tracks with new results from the SCT, our system

Algorithm 3 Trajectory validation

Input: global track $g_{t,i} = \{l_{t,j}^{(k)} | j = \overline{1 \dots N_t}, k = i\}$
Output: some $l_{t,j}$ are detached from $g_{t,i}$

construct cost matrix $C_{|g_{t,i}| \times |g_{t,i}|}$
for each $l_{t,j}$ in $g_{t,i}$ **do**
  for each $l_{t,q}$ in $g_{t,i}$ **do**
    if $l_{t,j}$!= $l_{t,q}$ **then**
      $C[j, q] \leftarrow \Delta\text{OL}_{l_{t,j}, l_{t,q}}$
    end if
  end for
end for
clusters $\leftarrow$ AgglomerativeClustering(C, θ_{OL})
if $|clusters| > 1$ **then**
  main_cluster $\leftarrow$ cluster of $\text{argmin}_j\ (\Delta\text{OL}_{g_{t,i}, l_{t,j}})$
  for $l_{t,j}$ in $g_{t,i}$ **do**
    if (cluster of $l_{t,j}$) != *main_cluster* **then**
      $g_{t,i} \leftarrow g_{t,i} \setminus l_{t,j}$
      create new $g_{t,M_t+1} = \{l_{t,j}\}$
    end if
  end for
end if

performs a trajectory validation step.

At each time instance, for each global track, we employ HAC with dissimilarity threshold θ_{OL} (as described in Section 3.1.2) on its local tracks. If any local track is separated from the main cluster, our system creates a new global track for that local track. Subsequently, the new global track joins the global track merge block, similar to any other global track, and is merged with its true identity as indicated in Algorithm 3.

To determine the main cluster, especially in cases where HAC clusters the local tracks into multiple clusters of equal cardinality, our system calculates the average dissimilarity scores between the global track and the local tracks within each cluster based on appearance and trajectory distance. The cluster with the smallest dissimilarity score is selected as the main cluster.

In Section 4.4, we provide experimental results that demonstrate the effectiveness of this strategy.

4. Experiments

4.1. Dataset and setting

We evaluated our online MCT system using the 2024 AI City Challenge Track 1 dataset [39]. The dataset consists of approximately 1000 cameras divided into 90 scenes. All videos in the dataset have a resolution of 1080p, 30 FPS, and a duration of 13 minutes. Camera matrices are provided. The training set includes 40 scenes, the validation set includes 20 scenes, and the test set includes 30 scenes. The test set is more challenging than the validation set due to factors such as a higher number of cameras per scene, the presence of private rooms and corridors, and a narrower camera network coverage.

4.2. Evaluation Metrics

We used the mean Average Precision (mAP) to evaluate the performance of both the detection model and the Re-ID model. For multi-camera tracking, we employed the HOTA metrics [29], which provide a unified metric that balances detection accuracy, association accuracy, and localization accuracy.

4.3. Implementation Details

For the detection model, we utilized YOLOv8 [21] and trained it from scratch on the training set. The input size was set to 640×360, and the model achieved a validation mAP@50 of 0.96. During testing, we employed a high confidence threshold of 0.4 to ensure high-quality detection (as discussed in Section 3.2). The midpoint of the bottom edge of the bounding box was used to project the person's location onto the map. For the Re-ID model, we employed TransReID base [14] and trained it from scratch on the training set. The input size was 128×256, and the feature dimension was set to 768. The model achieved a validation mAP of 0.6. The balanced threshold for the Re-ID task, determined by GOM [40], was determined to be 0.2. We utilized StrongSORT [11] for SCT. The IoU dissimilarity threshold was set to 0.54. A probationary period of 0.3 seconds was applied to create a local track, and a track was deleted if it was missed for 5 seconds.

In Section 4.4, we demonstrated the effectiveness of the accumulated appearance feature. Initially, we set the appearance dissimilarity threshold $\theta_{\text{AP}} = 0.12$ with a feature bank size of 15. Subsequently, we reduced the threshold to 0.1 and evaluated the performance change. This appearance dissimilarity threshold was applied in both StrongSORT [11] and the global track merge steps.

In the overlapping merge step, we used a fusion weight $\alpha_{\text{OL}} = 0.8$. In the updated-missed tracks merge, the fusion weight was set $\alpha_{\text{UM}} = 0.9$, and the speed dissimilarity threshold was set $\theta_{\text{SP}} = 2$. This choice was motivated by the high-quality detection and appearance features' effectiveness in handling spatial location noise, particularly when people move in close proximity to each other.

Specifically for the 2024 AI City Challenge Track 1 [39], we set `max_frame_skipped` equal to the length of the videos because of the in-house nature of the dataset.

4.4. Results

We present the experimental results on the 100% test set of the 2024 AI City Challenge Track 1 [39] in Table 1. The inclusion of trajectory validation significantly improved the

Method	HOTA (%)	DetA (%)	AssA (%)	LocA (%)
Baseline	16.4	46.7	6.0	89.2
+ Trajectory validation ($\theta_{\text{AP}} = 0.12$, feature bank)	27.7	54.8	14.7	89.5
+ Trajectory validation, $\theta_{\text{AP}} = 0.1$, accumulated feature	40.3	53.8	32.5	89.6

Table 1. Experimental results on the 100% test set of the 2024 AI City Challenge Track 1 [39]. Note that in our baseline version, scene 071 was excluded from our submission due to corruption issues caused by camera 649, which may result in a slight shift in the baseline version's score due to detection loss. In the two improved versions, this scene is included back.

Ranking	Team ID	HOTA
	...	
7	5	45.1575
8	124	40.3361
9	**162**	**40.3361**
10	21	33.4879
11	90	31.5208
	...	

Table 2. Leaderboard of the 2024 AI City Challenge Track 1 [39]. Our proposed online system achieved a rank of 9th with a HOTA score [29] of 40.3%.

Association Accuracy by 8%, leading to an improved overall HOTA score [29]. Additionally, incorporating the accumulated appearance feature with a smaller appearance dissimilarity threshold further improved the Association Accuracy by 17%.

With our best-improved version achieving a HOTA score [29] of 40.3% on the test set, our online system ranked 9th out of 17 teams on the AI City Challenge 2024 Track 1 [39] leaderboard, as shown in Table 2.

5. Conclusion

In this study, we have presented an efficient online multi-camera tracking (MCT) system that overcomes the challenges of identity switches and achieves accurate tracking in both overlapping and non-overlapping camera networks. Our system incorporates appearance and spatio-temporal cues, along with memory-efficient accumulated appearance features and trajectory validation using hierarchical agglomerative clustering (HAC) in overlapping regions. Experimental evaluations on the 2024 AI City Challenge Track 1 dataset [39] have demonstrated the effectiveness of our system, achieving competitive performance. The proposed system provides a valuable contribution to the field of MCT, offering a real-time and online solution for tracking individuals across multiple cameras in various real-world scenarios. Future work can explore further enhancements and optimizations to improve the system's performance and extend its applicability to other tracking domains.

6. Acknowledgement

We would like to express our sincere gratitude for the invaluable support provided by Asilla, Inc. throughout the course of this work.

References

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

[2] Ejaz Ahmed, Michael Jones, and Tim K Marks. An improved deep learning architecture for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3908–3916, 2015. 2

[3] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II 14*, pages 850–865. Springer, 2016. 2

[4] 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

[5] Long Chen, Haizhou Ai, Rui Chen, Zijie Zhuang, and Shuang Liu. Cross-view tracking for multi-human 3d pose estimation at over 100 fps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3279–3288, 2020. 2

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

[7] Cheng-Che Cheng, Min-Xuan Qiu, Chen-Kuo Chiang, and Shang-Hong Lai. Rest: A reconfigurable spatial-temporal graph model for multi-camera multi-object tracking, 2023. 2

[8] Nhat Minh Chung, Huy Dinh-Anh Le, Vuong Ai Nguyen, Quang Qui-Vinh Nguyen, Thong Duy-Minh Nguyen, Tin-Trung Thai, and Synh Viet-Uyen Ha. Multi-camera multi-vehicle tracking with domain generalization and contextual constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3327–3337, 2022. 2

[9] Zuozhuo Dai, Mingqiang Chen, Xiaodong Gu, Siyu Zhu, and Ping Tan. Batch dropblock network for person re-identification and beyond. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3691–3701, 2019. 2

[10] Xingping Dong and Jianbing Shen. Triplet loss in siamese network for object tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 459–474, 2018. 2

[11] Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. Strongsort: Make deepsort great again, 2023. 2, 6, 7

[12] Yuxin Fang, Bencheng Liao, Xinggang Wang, Jiemin Fang, Jiyang Qi, Rui Wu, Jianwei Niu, and Wenyu Liu. You only look at one sequence: Rethinking transformer in vision through object detection. *Advances in Neural Information Processing Systems*, 34:26183–26197, 2021. 2

[13] Bipin Gaikwad and Abhijit Karmakar. Smart surveillance system for real-time multi-person multi-camera tracking at the edge. *Journal of real-time image processing*, 18(6): 1993–2007, 2021. 2

[14] Shuting He, Hao Luo, Pichao Wang, Fan Wang, Hao Li, and Wei Jiang. Transreid: Transformer-based object re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15013–15022, 2021. 2, 7

[15] Yuhang He, Jie Han, Wentao Yu, Xiaopeng Hong, Xing Wei, and Yihong Gong. City-scale multi-camera vehicle tracking by semantic attribute parsing and cross-camera tracklet matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 576–577, 2020. 2

[16] Yuhang He, Xing Wei, Xiaopeng Hong, Weiwei Shi, and Yihong Gong. Multi-target multi-camera tracking by tracklet-to-target assignment. *IEEE Transactions on Image Processing*, 29:5191–5205, 2020. 2

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

[18] Yunzhong Hou, Liang Zheng, Zhongdao Wang, and Shengjin Wang. Locality aware appearance metric for multi-target multi-camera tracking. *arXiv preprint arXiv:1911.12037*, 2019. 2

[19] Omar Javed, Khurram Shafique, and Mubarak Shah. Appearance modeling for tracking in multiple non-overlapping cameras. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, pages 26–33. IEEE, 2005. 2

[20] Glenn Jocher, Alex Stoken, Ayush Chaurasia, Jirka Borovec, Yonghye Kwon, Kalen Michael, Liu Changyu, Jiacong Fang, Piotr Skalski, Adam Hogan, et al. ultralytics/yolov5: v6. 0-yolov5n'nano'models, roboflow integration, tensorflow export, opencv dnn support. *Zenodo*, 2021. 2

[21] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023. 2, 7

[22] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960. 5

[23] Philipp Kohl, Andreas Specker, Arne Schumann, and Jurgen Beyerer. The mta dataset for multi-target multi-camera pedestrian tracking by weighted distance aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1042–1043, 2020. 2

[24] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 5

[25] Quoc Cuong Le and Moncef Hidane. Appearance features for online multiple camera multiple target tracking. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, pages 2089–2096, 2020. 2

[26] Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, et al. Yolov6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*, 2022. 2

[27] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3265–3273, 2022. 2

[28] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13619–13627, 2022. 2

[29] 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(2): 548–578, 2020. 1, 7, 8

[30] Hao Luo, Pichao Wang, Yi Xu, Feng Ding, Yanxin Zhou, Fan Wang, Hao Li, and Rong Jin. Self-supervised pre-training for transformer-based person re-identification. *arXiv preprint arXiv:2111.12084*, 2021. 2

[31] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4004–4012, 2016. 2

[32] Ergys Ristani and Carlo Tomasi. Features for multi-target multi-camera tracking and re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6036–6046, 2018. 2

[33] Bing Shuai, Andrew Berneshawi, Xinyu Li, Davide Modolo, and Joseph Tighe. Siammot: Siamese multi-object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12372–12382, 2021. 2

[34] Andreas Specker and Jürgen Beyerer. Toward accurate online multi-target multi-camera tracking in real-time. In *2022 30th European Signal Processing Conference (EUSIPCO)*, pages 533–537. IEEE, 2022. 2

[35] Andreas Specker, Daniel Stadler, Lucas Florin, and Jurgen Beyerer. An occlusion-aware multi-target multi-camera tracking system. In *Proceedings of the IEEE/CVF con-*

ference on computer vision and pattern recognition, pages 4173–4182, 2021. 2

[36] Ran Tao, Efstratios Gavves, and Arnold WM Smeulders. Siamese instance search for tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1420–1429, 2016. 2

[37] 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. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7464–7475, 2023. 2

[38] Hanxiao Wang, Shaogang Gong, Xiatian Zhu, and Tao Xiang. Human-in-the-loop person re-identification. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 405–422. Springer, 2016. 2

[39] 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, 7, 8

[40] Zheng Wang, Xin Yuan, Toshihiko Yamasaki, Yutian Lin, Xin Xu, and Wenjun Zeng. Re-identification = retrieval + verification: Back to essence and forward with a new metric, 2020. 7

[41] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *European conference on computer vision*, pages 107–122. Springer, 2020. 2

[42] Longyin Wen, Zhen Lei, Ming-Ching Chang, Honggang Qi, and Siwei Lyu. Multi-camera multi-target tracking with space-time-view hyper-graph. *International Journal of Computer Vision*, 122:313–333, 2017. 2

[43] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric, 2017. 2, 6

[44] Xipeng Yang, Jin Ye, Jincheng Lu, Chenting Gong, Minyue Jiang, Xiangru Lin, Wei Zhang, Xiao Tan, Yingying Li, Xiaoqing Ye, et al. Box-grained reranking matching for multi-camera multi-target tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3096–3106, 2022. 2

[45] Hui Yao, Zhizhao Duan, Zhen Xie, Jingbo Chen, Xi Wu, Duo Xu, and Yutao Gao. City-scale multi-camera vehicle tracking based on space-time-appearance features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3310–3318, 2022. 2

[46] Jin Ye, Xipeng Yang, Shuai Kang, Yue He, Weiming Zhang, Leping Huang, Minyue Jiang, Wei Zhang, Yifeng Shi, Meng Xia, et al. A robust mtmc tracking system for ai-city challenge 2021. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4044–4053, 2021. 2

[47] Mang Ye, Chao Liang, Yi Yu, Zheng Wang, Qingming Leng, Chunxia Xiao, Jun Chen, and Ruimin Hu. Person reidentification via ranking aggregation of similarity pulling and dissimilarity pushing. *IEEE Transactions on Multimedia*, 18 (12):2553–2566, 2016. 2

[48] Sisi You, Hantao Yao, and Changsheng Xu. Multi-target multi-camera tracking with optical-based pose association. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(8):3105–3117, 2020. 2

[49] 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

[50] Xindi Zhang and Ebroul Izquierdo. Real-time multi-target multi-camera tracking with spatial-temporal information. In *2019 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4. IEEE, 2019. 2

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

[52] 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

[53] Liang Zheng, Shengjin Wang, Lu Tian, Fei He, Ziqiong Liu, and Qi Tian. Query-adaptive late fusion for image search and person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1741–1750, 2015. 2

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

[55] Jiahuan Zhou, Pei Yu, Wei Tang, and Ying Wu. Efficient online local metric adaptation via negative samples for person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2420–2428, 2017. 2

[56] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3702–3712, 2019. 2

[57] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European conference on computer vision*, pages 474–490. Springer, 2020. 2

[58] 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