MVTrajecter: Multi-View Pedestrian Tracking with Trajectory Motion Cost and Trajectory Appearance Cost

Taiga Yamane Ryo Masumura Satoshi Suzuki Shota Orihashi
NTT Human Informatics Laboratries, NTT Corporation
{taiga.yamane, ryo.masumura, satoshixv.suzuki, shota.orihashi}@ntt.com

Abstract

Multi-View Pedestrian Tracking (MVPT) aims to track pedestrians in the form of a bird's eye view occupancy map from multi-view videos. End-to-end methods that detect and associate pedestrians within one model have shown great progress in MVPT. The motion and appearance information of pedestrians is important for the association, but previous end-to-end MVPT methods rely only on the current and its single adjacent past timestamp, discarding the past trajectories before that. This paper proposes a novel end-to-end MVPT method called Multi-View Trajectory Tracker (MVTrajecter) that utilizes information from multiple timestamps in past trajectories for robust association. MVTrajecter introduces trajectory motion cost and trajectory appearance cost to effectively incorporate motion and appearance information, respectively. These costs calculate which pedestrians at the current and each past timestamp are likely identical based on the information between those timestamps. Even if a current pedestrian could be associated with a false pedestrian at some past timestamp, these costs enable the model to associate that current pedestrian with the correct past trajectory based on other past timestamps. In addition, MVTrajecter effectively captures the relationships between multiple timestamps leveraging the attention mechanism. Extensive experiments demonstrate the effectiveness of each component in MVTrajecter and show that it outperforms the previous state-of-the-art methods.

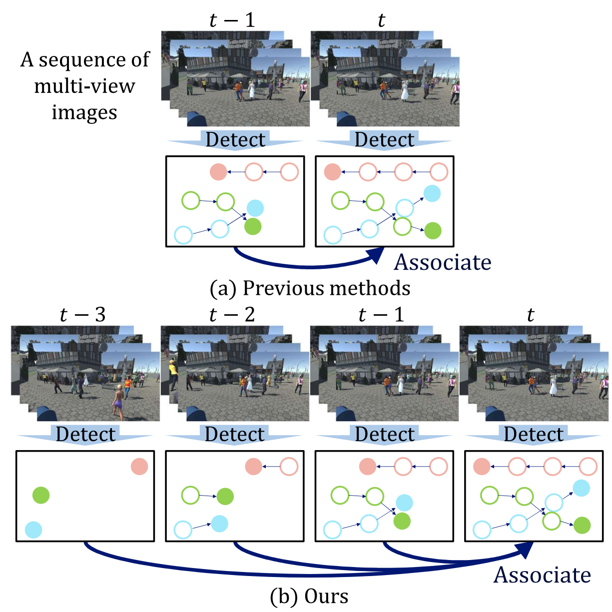

Figure 1. Comparison of (a) previous end-to-end MVPT methods and (b) MVTrajecter. This example shows a case where three pedestrians are on BEV occupancy maps and we want to associate pedestrians at t with past pedestrians. The same color indicates the same pedestrian. The filled circles indicate detected pedestrians at each timestamp, and the empty circles indicate past pedestrians.

1. Introduction

Pedestrian tracking in videos is a crucial component in various applications, such as surveillance systems [19], sports analysis [12], and human-computer interaction [68]. In pedestrian tracking, occlusion is a significant problem because it hampers tracking pedestrians hidden behind obstacles or other pedestrians. As a potential solution, many studies have focused on *Multi-View Pedestrian Tracking* (MVPT) [9, 20, 24, 25, 32, 69]. MVPT aims to track pedestrians in the form of a bird's eye view (BEV) pedestrian occupancy map from multi-view videos captured by multiple cameras. In contrast to monocular tracking [15, 45, 60], which uses a single non-calibrated camera, MVPT utilizes multiple calibrated cameras with partially overlapping fields of view. MVPT is, therefore, expected to be more robust to occlusion in the overlapping area than monocular tracking by combining complementary clues.

A variety of MVPT methods have been proposed. Early methods [4, 9–11, 22, 23, 27, 28, 30, 33, 46–48, 50, 53, 59, 66, 69, 72] relied on monocular object detection [54, 79], which uses the detection results of each view image by monocular detection to localize pedestrians in the BEV occupancy map and identify which pedestrians are identical at different timestamps. While these methods were important for early attempts in MVPT, occlusion in monocular detection hurt the final tracking performance. Furthermore, these methods optimized monocular detection and subsequent parts separately, resulting in suboptimal tracking performances. To overcome these problems, end-to-end trainable methods [14, 20, 62, 63, 71] that do not rely on monocular detection have become the dominant approach. These methods first generate a representation of the ground plane, called the BEV feature, for each timestamp and then track pedestrians through the detection and association processes using this feature. In the detection process, they directly predict a BEV pedestrian occupancy map from the BEV feature for each timestamp. In the association process, they identify which pedestrians are identical at the current and past timestamps by utilizing information on pedestrians extracted from the BEV feature. These methods are optimized in an end-to-end manner.

In end-to-end MVPT methods, improving the association process has been the main research topic to achieve better tracking performances. Previous methods utilize motion and/or appearance information of pedestrians between the current and an adjacent past timestamp. EarlyBird [62] and REMP [14] extract appearance information from the BEV feature at each timestamp and extract motion information using a Kalman filter [36]. TrackTacular [63], MVFlow [20], and MVTr [71] concatenate the BEV features at the current and an adjacent past timestamp and extract motion information between those two timestamps. Then, they associate pedestrians at the current and an adjacent past timestamp based on the appearance similarity and/or the distance between locations estimated by motions and actual detected locations. While these methods have outperformed non-end-to-end MVPT methods, they only utilize the information on pedestrians between the current and an adjacent past timestamp and discard the information on past trajectories[1] before that, as shown in Fig. 1(a). Therefore, they do not fully utilize past trajectories, leading to association errors when the information at an adjacent past timestamp is unreliable (*e.g.*, when an area is excessively crowded with pedestrians). Although many end-to-end monocular tracking methods [17, 18, 51, 67, 73, 80] have investigated leveraging information at multiple timestamps in past trajectories (see Sec. 2), it is not clear how to leverage this information in the end-to-end MVPT method.

In this paper, we propose a novel end-to-end MVPT method called **M**ulti-**V**iew **Traject**ory Track**er** (MVTrajecter), which performs the association based on motion and appearance information between the current and multiple timestamps in past trajectories, as shown in Fig. 1(b). To effectively utilize the information, we introduce trajectory motion cost (TMC) and trajectory appearance cost (TAC). These costs comprehensively identify which current pedestrian and which past trajectory are identical by calculating which current pedestrian is likely identical to which past pedestrian for each past timestamp. Specifically, TMC calculates the distance between current locations estimated by motion information from each past timestamp and actual detected current locations, and TAC calculates the appearance similarities between pedestrians at the current and each past timestamp. Even if a current pedestrian could be associated with a false pedestrian at some past timestamp, these costs enable MVTrajecter to associate that current pedestrian with the correct past trajectory based on the information at other past timestamps. Furthermore, in order for TMC and TAC to identify pedestrians accurately, it is necessary to appropriately capture the relationships of pedestrians between multiple timestamps and extract identifiable motion and appearance information. As shown in Fig. 2, MVTrajecter leverages the attention mechanism [64], which has been shown to have such ability. During training, MVTrajecter learns to capture relationships of pedestrians between multiple timestamps through end-to-end training with TMC and TAC as its training objectives. In experiments on three major MVPT datasets [9, 32, 65], we verify the effectiveness of MVTrajecter and demonstrate that utilizing information at multiple timestamps in past trajectories leads to a better tracking performance.

Our contributions are summarized as follows:

1. We propose a novel end-to-end MVPT method called MVTrajecter. This comprehensively identifies which current pedestrian and which past trajectory is identical utilizing information between the current and multiple timestamps in past trajectories through TMC and TAC.
2. Given the use of TMC and TAC, we design the architecture of MVTrajecter leveraging the attention mechanism and effectively capture the relationships between the current pedestrians and past trajectories, extracting identifiable motion and appearance information.
3. We demonstrate that utilizing information at multiple timestamps in past trajectories leads to a better tracking performance. MVTrajecter outperforms previous methods and achieves a new state-of-the-art performance on GMVD, Wildtrack, and MultiviewX.

2. Related Work

Multi-View Pedestrian Detection. Multi-view pedestrian detection (MVPD) [9, 32, 65, 76] is a task to predict a BEV pedestrian occupancy map from a multi-view image. Unlike MVPT, MVPD aims to detect pedestrians rather than track pedestrians and processes the multi-view image rather than the multi-view video. MVPD is an important component of MVPT methods because they solve MVPT by detecting pedestrians at each timestamp and associating them across

[1]Trajectories are parts of the final track and typically consist of more than two timestamps.

different timestamps (as described in Sec. 1). In MVPD, multi-view aggregation is essential because each view has an overlapping but different field of view. Early methods [3, 8, 25, 56] detected pedestrians for each view using monocular detection [54, 79] and performed multi-view aggregation based on the detection results in each view. These methods optimized monocular detection and multi-view aggregation separately, resulting in suboptimal detection performances. In recent years, end-to-end trainable methods [2, 21, 31, 32, 52, 58, 77] that do not rely on monocular detection have become mainstream in MVPD. End-to-end MVPD methods generate a BEV feature by projecting image features extracted from each view image onto a ground plane, predicting the BEV occupancy map. The information from each view is aggregated through this BEV feature. MVDet [32] is a pioneering end-to-end method that generates the BEV feature by projecting image features onto the ground plane using a perspective transformation. To further improve detection performance, recent studies have proposed more effective projections [2, 31, 58], data augmentations [21, 52, 61], and network architectures [2, 40]. Our MVPT method extends MVDet to perform the association of pedestrians utilizing information on past trajectories.

Pedestrian Tracking. Many monocular pedestrian tracking methods, including both non-deep learning [34, 35, 41, 49, 75] and deep learning [1, 5, 7, 13, 17, 18, 51, 67, 73, 80] ones, leverage information at multiple timestamps in past trajectories for robust association. Fusion-based methods [1, 17, 18, 67, 73] fuse the appearance features at multiple timestamps using the exponential moving average (EMA) frame-by-frame and generate the representative appearance feature for each trajectory. Graph-based methods [5, 7, 13] construct a graph that represents the association based on appearance features at multiple timestamps and generate trajectories by optimizing the graph. GTR [80] extends DETR [6] to tracking and directly generates trajectories based on interactions between queries and pedestrian features across multiple timestamps. MotionTrack [51] extracts motion information on the whole trajectory utilizing the spatial distribution pattern and velocity-time correlation based on locations over multiple past timestamps. While these methods have improved their tracking performances on benchmark datasets [15, 45, 60], they cannot be applied for MVPT directly because they are designed for monocular tracking and are unable to consider the relationships between multiple views and the BEV map. Encouraged by the success of the above monocular tracking methods, many non-end-to-end MVPT methods [4, 10, 23, 47, 53, 66] also leverage information at multiple timestamps in past trajectories using techniques such as the dynamic programming, graph, and transformer [64]. However, previous end-to-end MVPT methods [14, 20, 62, 63, 71] have not incorporated this information, and it is not clear how to leverage it.

3. Methodology

In this section, we first formulate the problem setting of our MVTrajecter. We then provide a detailed explanation of MVTrajecter, including its overall detection and association procedure, model architecture, and training method.

3.1. Problem Setting of MVTrajecter

MVTrajecter processes a multi-view video in a sequential frame-by-frame manner, where a frame consists of images from multiple views at one timestamp. Let $\boldsymbol{I}^t = \{I_1^t, \cdots, I_S^t\}$ be a multi-view image at the current timestamp t in a multi-view video from synchronized S cameras, where $I_s^t \in \mathbb{R}^{3\times H\times W}$ is an image from the s-th camera at t. Here, H and W are the height and width of the image, respectively. MVTrajecter aims to detect the current pedestrians from $\boldsymbol{I}^t$ and generate the current trajectories $\{\tau_1^{t-K:\,t}, \cdots, \tau_N^{t-K:\,t}\}$ of all N pedestrians by associating the current pedestrian locations $\{p_1^t, \cdots, p_N^t\}$ with past trajectories $\{\tau_1^{t-K:\,t-1}, \cdots, \tau_N^{t-K:\,t-1}\}$, where pedestrian locations and trajectories represent those on the BEV map. Here, K is the length of past trajectories that MVTrajecter utilizes for the association. $p_n^{t-k} \in \mathbb{R}^2 \cup \{\emptyset\}$ represents the location at timestamp $t-k$ of the pedestrian n, where $k = 0, \cdots, K$. $p_n^{t-k} = \emptyset$ indicates that the pedestrian n cannot be located at $t-k$. Trajectory $\tau_n^{t-K:\,t-k} = \{p_n^{t-K}, \cdots, p_n^{t-k}\}$ represents a sequence of pedestrian n's locations from $t-K$ to $t-k$.

We can obtain the final tracks by solving the above problem in order from the first to last timestamps in the video and linking generated trajectories pedestrian-by-pedestrian.

Note that previous end-to-end methods [14, 20, 62, 63, 71] set $K = 1$ and use information on pedestrians between t and $t-1$ for the association. In contrast, MVTrajecter utilizes information on pedestrians from $t-K$ to t.

3.2. Procedure of MVTrajecter

MVTrajecter consists of two parts: detecting pedestrians at the current timestamp t and associating the current pedestrians with past trajectories. In this subsection, we explain the procedures of these parts in order.

In the detection part, we predict the current BEV occupancy map $M^t \in \mathbb{R}^{1\times X\times Y}$ from the current multi-view image $\boldsymbol{I}^t$, where X and Y are the width and height of the BEV occupancy map, respectively. We can obtain the current pedestrian locations $\{p_1^t, \cdots, p_N^t\}$ from M^t.

In the association part, we calculate association costs between the current pedestrian locations $\{p_1^t, \cdots, p_N^t\}$ and past trajectories $\{\tau_1^{t-K:\,t-1}, \cdots, \tau_N^{t-K:\,t-1}\}$, indicating which current pedestrian is likely identical to which past trajectory. To this end, we utilize motion and appearance information of pedestrians between the current and past K timestamps. First, for each pedestrian at each past timestamp, we predict the motion to the current timestamp. The motion

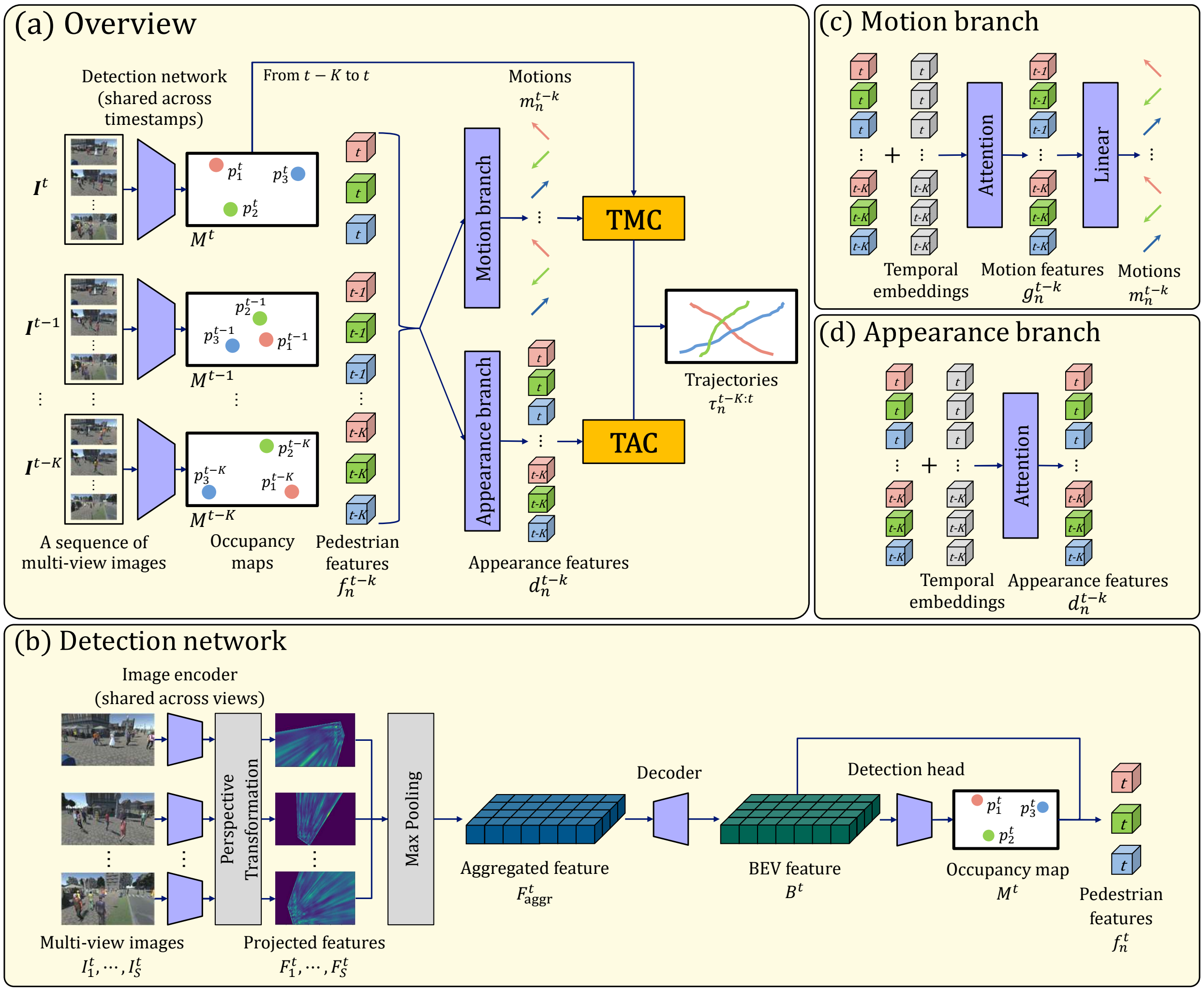

Figure 2. (a) Overview of MVTrajecter. It consists of a (b) detection network, (c) motion branch, (d) appearance branch, TMC, and TAC (see Sec. 3.3). The purple blocks have trainable parameters. This is an example of the association for pedestrians at t.

to be predicted is defined as the offset between the current and past locations. For pedestrian n at $t-k$, this offset is $p_n^t - p_n^{t-k}$. Then, based on the predicted motions, we calculate the motion-based association cost named trajectory motion cost (TMC). Next, for each pedestrian at the current timestamp and past K timestamps, we extract the appearance feature that represents the appearance of the pedestrian. Then, based on the predicted appearance features, we calculate the appearance-based association cost named trajectory appearance cost (TAC). In Sec. 3.3, we will formulate TMC and TAC in detail. After obtaining TMC and TAC, we calculate the overall association cost by fusing them to account for motion and appearance information simultaneously. Finally, we use Hungarian Matching [39] for the overall association cost to perform association. As a result of these processes, we generate the current trajectories $\{\tau_1^{t-K:t}, \cdots, \tau_N^{t-k:t}\}$. We will explain the model architecture to implement this procedure in Sec. 3.4.

3.3. Association Cost

MVTrajecter calculates TMC and TAC and performs the association by combining them. Let $i \in \{1, \cdots, N\}$ be the pedestrian whose location is p_i^t at the current timestamp t, and $j \in \{1, \cdots, N\}$ be the pedestrian whose past trajectory is $\tau_j^{t-K:t-1}$. We define the overall association cost between i and j by $C(i,j) = (1-\alpha)C_{\text{TMC}}(i,j) + \alpha C_{\text{TAC}}(i,j)$, where α is a predetermined weighting parameter. Here, $C_{\text{TMC}}(i,j)$ and $C_{\text{TAC}}(i,j)$ are TMC and TAC between i and j, respectively (we will explain how to calculate them later). During inference, we calculate $C(i,j)$ for all pairs of i and j and use Hungarian Matching with a certain threshold to decide which pair of i and j represents identical pedestrians. We initialized unmatched current pedestrians as new trajectories.

Trajectory Motion Cost (TMC). Let $m_j^{t-k} \in \mathbb{R}^2$ be the predicted motion of j from timestamp $t-k$ to t. When the distance $||p_i^t - (p_j^{t-k} + m_j^{t-k})||$ between the current detected location ($i.e.$, p_i^t) and estimated current location by motion ($i.e.$, $p_j^{t-k} + m_j^{t-k}$) is small, i is likely identical to j among pedestrians at $t-k$. Furthermore, when this distance is large for multiple k, i and j are less likely to be identical than when the distance is large for only one k. On the basis of this, we define TMC between i and j by summing the Euclidean distance for each past timestamp, as:

$$C_{\mathrm{TMC}}(i,j) = \sum_{k=1}^{K} ||p_i^t - (p_j^{t-k} + m_j^{t-k})||. \tag{1}$$

When $C_{\mathrm{TMC}}(i,j)$ is small, i and j are likely to be identical. If $p_j^{t-k} = \emptyset$, we calculate $||p_i^t - (p_j^{t-k} + m_j^{t-k})||$ using the Kalman filter [36]. TMC utilizes motion information not only from an adjacent past timestamp but also from multiple timestamps in past trajectories.

Trajectory Appearance Cost (TAC). Let $d_n^{t-k} \in \mathbb{R}^E$ be the E-dimensional appearance feature of pedestrian n at $t-k$. When the similarity between d_i^t and d_j^{t-k} is high, i is likely identical to j among pedestrians at $t-k$. Based on this, we define the conditional probability that i is identical to j among pedestrians at $t-k$ using the dot production as:

$$\mathrm{Pr}^{t-k}(j|i) = \frac{\exp(d_i^t \cdot d_j^{t-k})}{\sum_{n=1}^{N} \exp(d_i^t \cdot d_n^{t-k})}. \tag{2}$$

If $p_j^{t-k} = \emptyset$, we set $\mathrm{Pr}^{t-k}(j|i) = 0$. When $\mathrm{Pr}^{t-k}(j|i)$ is high for multiple k, i and j are more likely to be identical than when the probability is high for only one k. Therefore, we define TAC between i and j by summing the probability for each past timestamp, as:

$$C_{\mathrm{TAC}}(i,j) = -\sum_{k=1}^{K} \mathrm{Pr}^{t-k}(j|i). \tag{3}$$

When $C_{\mathrm{TAC}}(i,j)$ is small, i and j are likely to be identical. Unlike TMC, since high $\mathrm{Pr}^{t-k}(j|i)$ indicates identical pedestrian, we multiply $\mathrm{Pr}^{t-k}(j|i)$ by -1. TAC utilizes appearance information not only from an adjacent past timestamp but also from multiple timestamps in past trajectories.

3.4. Model Architecture

The overview of MVTrajecter is shown in Fig. 2(a). MVTrajecter is an end-to-end trainable method that outputs everything required for our detection and association procedure. It consists of the detection part, which predicts the BEV occupancy map and extracts pedestrian locations and pedestrian features, and the association part, which predicts motions and appearance features from pedestrian features.

Detection part. Based on [32, 62, 65], the detection network consists of the image encoder, perspective transformation [32], max pooling, decoder, and detection head, as shown in Fig. 2(b). The input to the detection part is a multiview image $\boldsymbol{I}^t = \{I_1^t, \cdots, I_S^t\}$. The image encoder extracts E-channel feature maps from each view image, and then, the extracted feature maps are projected onto a ground plane by perspective transformation [32] (formulated in Supp. A). We denote these projected features as $\{F_1^t, \cdots, F_S^t\}$, where $F_s^t \in \mathbb{R}^{E\times X\times Y}$ is the projected feature of the s-th camera at t. The max pooling operation aggregates projected features along the camera direction as:

$$F_{\mathrm{aggr}}^t(e,x,y) = \max_{s\in\{1,\cdots,S\}} F_s^t(e,x,y), \tag{4}$$

where F_{aggr}^t is referred to as the aggregated feature, and $F_s^t(e,x,y)$ is the value at coordinates (e,x,y) on F_s^t. This aggregation allows the later modules to handle any number of cameras and camera layouts [65]. The decoder refines F_{aggr}^t and generates a BEV feature $B^t \in \mathbb{R}^{E\times X\times Y}$. The detection head predicts a BEV occupancy map $M^t \in \mathbb{R}^{1\times X\times Y}$ from B^t. We apply a 3×3 max pooling to M^t following [79] and extract pedestrians over a certain threshold. As a result, we obtain current pedestrian locations $\{p_1^t, \cdots, p_N^t\}$. We also extract pedestrian features by $f_n^t = B^t(p_n^t)$ for the subsequent association part, where f_n^t is a pedestrian feature of pedestrian n at t, and $B^t(x,y) \in \mathbb{R}^E$ represents a feature vector at coordinates (x,y) on B^t.

Association part. The association part predicts motions and appearance features from current and past pedestrian features $\{f_1^{t-k}, \cdots, f_N^{t-k}\}_{k=0}^{K}$. We keep past pedestrian features $\{f_1^{t-k}, \cdots, f_N^{t-k}\}_{k=1}^{K}$ and do not need to recalculate them. The association part has a motion branch to predict motions and an appearance branch to predict appearance features. We utilize the attention mechanism [64] for both branches. The attention mechanism has the ability to capture the multiple temporal changes of pedestrians [44, 70, 74] and match pedestrians with high similarity at different timestamps [26, 80, 81]. The former ability fits to predict motions over multiple timestamps and the latter ability fits to predict identifiable appearance features. Before feeding pedestrian features into the attention mechanism, we add sinusoidal temporal embeddings [64] to them in order to distinguish the timestamps (from $t-K$ to t) of pedestrian features.

The motion branch consists of an attention layer and a linear layer, as shown in Fig. 2(c). The attention layer captures the temporal changes of pedestrians from pedestrian features, generating motion features $\{g_1^{t-k}, \cdots, g_N^{t-k}\}_{k=1}^{K}$. Then, the linear layer predicts motions $\{m_1^{t-k}, \cdots, m_N^{t-k}\}_{k=1}^{K}$ from motion features:

$$g_n^{t-k} = \mathrm{Attention}(f_n^{t-k}, \{f_1^{t-k}, \cdots, f_N^{t-k}\}_{k=0}^{K}), \tag{5}$$
$$m_n^{t-k} = \mathrm{Linear}(g_n^{t-k}), \tag{6}$$

where $g_n^{t-k} \in \mathbb{R}^E$ is the motion feature of pedestrian n at $t-k$, and $m_n^{t-k} \in \mathbb{R}^2$ is the motion of pedestrian n from $t-k$ to t. For the Attention function, the first argument indicates query, and the second argument indicates key and value. These predicted motions are utilized for TMC.

The appearance branch consists of an attention layer, as shown in Fig. 2(d). This generates appearance features $\{d_1^{t-k}, \cdots, d_N^{t-k}\}_{k=0}^K$ by increasing the similarity between identical pedestrian's features at different timestamps, as:

$$d_n^{t-k} = \mathrm{Attention}(f_n^{t-k}, \{f_1^{t-k}, \cdots, f_N^{t-k}\}_{k=0}^K), \quad (7)$$

where $d_n^{t-k} \in \mathbb{R}^E$ is the appearance feature of pedestrian n at $t-k$. These appearance features are utilized for TAC.

3.5. Training

During training, we split multi-view videos into non-overlapping multi-view image sequences with window size $K+1$ and feed $\{\boldsymbol{I}^{t-K}, \cdots, \boldsymbol{I}^t\}$ into MVTrajecter. We train the entire MVTrajecter in an end-to-end manner. The overall training objective consists of detection loss, the loss related to TMC, and the loss related to TAC. First, we predict occupancy maps $\{M^{t-K}, \cdots, M^t\}$ from $\{\boldsymbol{I}^{t-K}, \cdots, \boldsymbol{I}^t\}$. The detection loss $\mathcal{L}_{\mathrm{det}}$ is calculated using a mean squared error (MSE) for all predicted occupancy maps $\{M^{t-K}, \cdots, M^t\}$, as:

$$\mathcal{L}_{\mathrm{det}} = \frac{1}{K+1} \sum_{k=0}^{K} \mathrm{MSE}(M^{t-k}, \bar{M}^{t-k}), \quad (8)$$

where $\bar{M}^{t-k}$ is a smoothed ground truth occupancy map at $t-k$ using a Gaussian kernel following [32].

Then, we predict motions $\{m_1^{t-k}, \cdots, m_N^{t-k}\}_{k=1}^K$ and appearance features $\{d_1^{t-k}, \cdots, d_N^{t-k}\}_{k=0}^K$. Let $\mathcal{L}_{\mathrm{TMC}}$ be the loss related to TMC. For pedestrian n and $k = 1, \cdots, K$, we minimize $||p_n^t - (p_n^{t-k} + m_n^{t-k})||$. We optimize $\{m_n^{t-K}, \cdots, m_n^{t-1}\}$ because $\{p_n^{t-K}, \cdots, p_n^t\}$ are fixed by the detection part. Therefore, we define $\mathcal{L}_{\mathrm{TMC}}$ as:

$$\mathcal{L}_{\mathrm{TMC}} = \frac{1}{N}\frac{1}{K} \sum_{n=1}^{N}\sum_{k=1}^{K} ||m_n^{t-k} - (\bar{p}_n^t - \bar{p}_n^{t-k})||, \quad (9)$$

where $\bar{p}_n^{t-k}$ represents the ground truth location of pedestrian n at $t-k$. Let $\mathcal{L}_{\mathrm{TAC}}$ be the loss related to TAC. For pedestrian n and $k = 1, \cdots, K$, we maximize $\mathrm{Pr}^{t-k}(j = n|i = n)$. Therefore, we define $\mathcal{L}_{\mathrm{TAC}}$ using negative log-likelihood, as:

$$\mathcal{L}_{\mathrm{TAC}} = -\frac{1}{N}\frac{1}{K} \sum_{n=1}^{N}\sum_{k=1}^{K} \log \mathrm{Pr}^{t-k}(j = n|i = n). \quad (10)$$

We set the overall loss $\mathcal{L}_{\mathrm{all}}$ using the uncertainty loss [38, 78] to balance $\mathcal{L}_{\mathrm{det}}$, $\mathcal{L}_{\mathrm{TMC}}$, and $\mathcal{L}_{\mathrm{TAC}}$ as follows:

$$\mathcal{L}_{\mathrm{all}} = \frac{1}{e^{w_1}}\mathcal{L}_{\mathrm{det}} + \frac{1}{e^{w_2}}\mathcal{L}_{\mathrm{TMC}} + \frac{1}{e^{w_3}}\mathcal{L}_{\mathrm{TAC}} + w_1 + w_2 + w_3, \quad (11)$$

where w_1, w_2, and w_3 are learnable parameters that automatically balance the three losses.

4. Experiments

4.1. Datasets

In this subsection, we explain the three MVPT datasets used in our experiments. More details can be found in Supp. B.

Wildtrack [9]. This is a real-world dataset consisting of videos captured with 7 cameras. The videos comprise 400 frames at a frame rate of 2 fps. This dataset covers a $12\,\mathrm{m} \times 36\,\mathrm{m}$ region quantized into a 480×1440 grid using square grid cells of $2.5\,\mathrm{cm}^2$. Each frame includes 20 pedestrians on average. Wildtrack splits them into the first 360 frames for training and the remaining 40 frames for testing. We use the last 40 frames in the training split as the validation data.

MultiviewX [32]. This is a synthetic dataset using the Unity engine and closely follows the style of Wildtrack. This dataset consists of videos captured by 6 cameras and covers a $16\,\mathrm{m} \times 25\,\mathrm{m}$ region. The region is quantized into a 640×1000 grid. Each frame includes 40 pedestrians on average. MultiviewX also splits them into the first 360 frames for training and the remaining 40 frames for testing. We use the last 40 frames in the training split as the validation data.

GMVD [65]. This is a large-scale synthetic dataset and includes 7 scenes with varying numbers of cameras and camera layouts. Furthermore, each scene contains multiple sequences with various environmental conditions including time and weather. This makes GMVD more challenging than Wildtrack or MultiviewX. Except for the camera setup and the size of the covered region, all parameters follow those of MultiviewX. Each frame includes 20–40 pedestrians on average. GMVD splits them into 6 scenes (with 43 sequences and 4983 frames) for training and 1 scene (with 10 sequences and 1012 frames) for testing. Following Vora et al. [65], we use MultiviewX as the validation data in the experiments on GMVD. Since GMVD is the largest dataset, we set it as the main dataset for our experiments.

4.2. Evaluation Metrics

Following previous studies [10, 62, 63], we used five standard metrics provided by Kasturi et al. [37] and Ristan et al.[55]: IDF1, Multiple Object Tracking Accuracy (MOTA), Multiple Object Tracking Precision (MOTP), Mostly Tracked (MT), and Mostly Lost (ML). A predicted pedestrian was classified as a true positive if its distance from the ground truth was within 1.0 meters. We used IDF1 and MOTA as the primary performance indicators following previous studies [62, 63, 71].

4.3. Implementation Details

Following previous methods [32, 62, 63], we employed ResNet18 [29], ResNet18 U-Net [57], and a 2-layer convolutional neural network for the image encoder, decoder, and detection head in the detection network, respectively. ResNet18 was pretrained on ImageNet [16]. We employed

Method	IDF1↑	MOTA↑	MOTP↑	MT↑	ML↓
EarlyBird [62]	69.7	70.8	86.1	41.6	12.4
MVFlow [20]	70.5	71.1	85.7	41.0	12.7
TrackTacular [63]	74.4	71.2	85.2	52.4	12.5
MVTr [71]	74.3	75.6	84.2	63.1	10.3
Ours	**77.2**	**78.1**	**87.7**	**66.0**	**8.7**

Table 1. Comparison with previous methods on GMVD when all models used their own detection results for tracking.

Method	IDF1↑	MOTA↑	MOTP↑	MT↑	ML↓
EarlyBird [62]	70.6	72.6	85.9	43.9	11.6
MVFlow [20]	73.2	73.9	83.7	58.1	9.0
TrackTacular [63]	75.6	76.2	84.9	64.3	8.8
MVTr [71]	75.3	76.0	85.0	64.1	9.0
Ours	**77.2**	**78.1**	**87.7**	**66.0**	**8.7**

Table 2. Comparison with previous methods on GMVD when all models used the same detection results as ours for tracking.

2 transformer attention blocks [64] with 8 attention heads for each attention layer.

The input images were resized to 720×1280. We set the channel size E to 1024. Unless otherwise mentioned, we set the temporal length K of past trajectories to 7, which is the maximum size we can set due to the limitations of GPU memory. We set the weighting parameter α between TMC and TAC to 0.98, the threshold of detection to 0.4, and the threshold of Hungarian Matching to 0.1. These hyperparameters were tuned on the validation data.

We optimized MVTrajecter using the Adam optimizer [42]. We set the batch size to 1 and accumulated the gradient over 16 batches. The learning rate was initialized to 1.0×10^{-3} and was decayed to 1.0×10^{-6} following a cosine schedule. We set the maximum training epochs to 20 for GMVD and 50 for Wildtrack and MultiviewX and stopped the training if the overall loss for the validation data did not decrease for 3 epochs in a row. More detailed implementations are provided in Supp. C. All experiments were conducted on a Nvidia 80GB A100 GPU.

4.4. Comparison with State-of-the-Art Methods

To verify the effectiveness of MVTrajecter, we compared it with previous state-of-the-art methods. Table 1 shows the comparison results on GMVD. Each method used its own detection results for tracking. Since the previous methods had not been evaluated on GMVD, we re-implemented EarlyBird [62], MVFlow [20], TrackTacular [63], and MVTr [71], which are the most recent state-of-the-art methods. Their network architectures and training settings followed our implementation details as much as possible. MVTrajecter outperformed the previous methods on all metrics. These results demonstrate that MVTrajecter is superior to the other methods in terms of end-to-end tracking performance, including detection and association.

Since tracking results are affected by detection results, we also compared the tracking performance using the same detection results to validate the effectiveness of MVTrajecter's association only. Table 2 shows the comparison results on GMVD using the same detection results as MVTrajecter. MVTrajecter outperformed all the comparison methods. This demonstrates that MVTrajecter's association, which uses motion and appearance information over multiple past timestamps, is superior to the association of the other methods. Specifically, its improvement over EarlyBird, which mainly depends on appearance information for the association, is due to utilizing information from multiple past timestamps and incorporating motion information. Improvements over MVFlow, TrackTacular, and MVTr, which only use motion information for the association, are due to utilizing information from multiple past timestamps and incorporating appearance information. Furthermore, all comparison methods improved their tracking performances when using MVTrajecter's detection results. This indicates that MVTrajecter's detection is superior to those of the other methods. A detailed evaluation and analysis of MVTrajecter's detection are provided in Supp. D.

Method	IDF1↑	MOTA↑	MOTP↑	MT↑	ML↓
ReST† [10]	85.7	81.6	81.8	79.4	4.9
BEV-SUSHI† [66]	93.4	87.5	**94.3**	90.2	**2.4**
REMP† [14]	–	88.5	86.8	–	–
EarlyBird [62]	92.3	89.5	86.6	78.0	4.9
MVFlow [20]	93.5	91.3	–	–	–
TrackTacular [63]	95.3	91.8	85.4	87.8	4.9
MVTr [71]	93.1	92.3	92.7	**95.1**	4.9
Ours	**96.5**	**94.3**	93.0	90.2	4.9

Table 3. Comparison with previous methods on Wildtrack. Methods with † used external data or external models.

Method	IDF1↑	MOTA↑	MOTP↑	MT↑	ML↓
REMP† [14]	–	81.0	85.8	–	–
EarlyBird [62]	82.4	88.4	86.2	82.9	1.3
TrackTacular [63]	85.6	92.4	80.1	92.1	2.6
MVTr [71]	82.9	91.4	**95.0**	96.1	**0.0**
Ours	**85.8**	**92.8**	**95.0**	**97.4**	**0.0**

Table 4. Comparison with previous methods on MultiviewX. The method with † used external data or external models.

To validate the versatility of MVTrajecter, which is not limited to GMVD, we also experimented using MultiviewX and Wildtrack. Tables 3 and 4 show the comparison results on Wildtrack and MultiviewX, respectively, where the results of the previous methods are directly copied from the original papers. MVTrajecter outperformed the previous methods on both Wildtrack and MultiviewX, achieving new state-of-the-art tracking performances of 94.3 MOTA on Wildtrack and 92.8 MOTA on MultiviewX. These results demonstrate that MVTrajecter is effective for various datasets. Qualitative results are provided in Supp. G.

4.5. Ablation Study

We conducted ablation studies to investigate the effect of past trajectory length, TMC, and TAC, as well as the advantages of employing the attention mechanism to predict mo-

K	IDF1↑	MOTA↑	MOTP↑	MT↑	ML↓
1	71.0	74.2	87.0	60.7	9.0
3	73.6	76.4	87.3	63.5	8.8
5	76.0	77.2	**87.8**	64.8	**8.7**
7	**77.2**	**78.1**	87.7	**66.0**	**8.7**

Table 5. Effect of past trajectory length K.

	IDF1↑	MOTA↑	MOTP↑	MT↑	ML↓
Motion baseline	67.8	73.7	86.2	58.1	9.0
+ TMC	71.8	74.2	86.6	61.9	**8.7**
Appearance baseline	64.7	72.3	87.0	58.4	9.0
+ TAC	72.3	73.8	87.3	62.3	8.8
+ TMC and TAC	**77.2**	**78.1**	**87.7**	**66.0**	**8.7**

Table 6. Effect of TMC and TAC.

tions and appearance features. All experiments were conducted on GMVD. We also conducted more detailed ablation studies in Supp. E and analyzed the inference speed of our proposed method in Supp. F.

Effect of Past Trajectory Length K. Table 5 shows the comparison results using different lengths K of past trajectories. When $K = 1$, our method is similar to the comparison methods in Table 1 and discards the information on past trajectories before an adjacent past timestamp, yielding a low tracking performance. As we increased K, we consistently observed improvements in MOTA and IDF1, and our method outperformed the comparison methods in Table 1 even when $K = 5$. The largest $K = 7$ achieved 3.9 points higher MOTA and 6.2 points higher IDF1 than the smallest $K = 1$. These performance improvements demonstrate the advantage of using longer trajectories for the association.

Effect of TMC and TAC. To investigate the effect of TMC and TAC on the tracking performance, we implemented two baselines of MVTrajecter, an appearance baseline and a motion baseline, and added TMC and TAC to them. The motion and appearance baselines only utilize the motion and appearance information between the current and an adjacent past timestamp, respectively. Table 6 shows the impacts of TMC and TAC on tracking performances. When TMC was added to the motion baseline or TAC was added to the appearance baseline, their tracking performances were greatly improved. These results demonstrate the effectiveness of utilizing the information for multiple past timestamps via TMC and TAC. In addition, when both TMC and TAC were added to the baselines, tracking performances were improved further. This indicates that both motion and appearance information from multiple past timestamps are necessary for a better tracking performance.

Effectiveness of Employing the Attention Mechanism to Predict Motions. While we employed the attention mechanism to predict motions, we could also employ the Kalman filter [36] which is often used in monocular tracking to predict motions. We compared the attention mechanism with the Kalman filter to verify the advantage of employing the attention mechanism. Table 7 shows the comparison results.

	IDF1↑	MOTA↑	MOTP↑	MT↑	ML↓
Ours w/ KF	76.0	75.3	**87.8**	64.1	8.8
Ours	**77.2**	**78.1**	87.7	**66.0**	**8.7**

Table 7. Comparison of motion prediction. ”w/ KF” predicts motions using the Kalman filter [36].

	IDF1↑	MOTA↑	MOTP↑	MT↑	ML↓
Ours w/ EB	75.6	76.2	87.4	62.9	8.8
Ours	**77.2**	**78.1**	**87.7**	**66.0**	**8.7**

Table 8. Comparison of appearance feature extraction. ”w/ EB” extracts appearance features in the same way as EarlyBird [62].

Employing the attention mechanism achieved a better tracking performance on almost all metrics than employing the Kalman filter. This result demonstrates that the motions predicted by the attention mechanism are more accurate than those predicted by the Kalman filter. We presume this is because the attention mechanism captures the relationships between multiple timestamps more accurately.

Effectiveness of Employing the Attention Mechanism to Extract Appearance Features. We re-extracted appearance features for all timestamps within the time window (*i.e.*, from $t - K$ to t) as t changed by employing the attention mechanism. In contrast, EarlyBird [62] and REMP [14] extract appearance features only for t. To investigate the advantage of our appearance feature extraction, we compared it with EarlyBird's appearance feature extraction. Table 8 shows the comparison results. Our appearance feature extraction achieved a higher tracking performance than EarlyBird's. We presume this is because as t changes, important appearance information also changes, and the attention mechanism has the ability to adaptively select that information based on the relationships among multiple timestamps.

5. Conclusion

We proposed a new end-to-end multi-view pedestrian tracking (MVPT) method called MVTrajecter. It comprehensively identifies which current pedestrian is identical to which past trajectory based on the motion and appearance information between the current and multiple timestamps in past trajectories. In addition, it effectively captures the relationships between multiple timestamps employing the attention mechanism. We demonstrated the effectiveness of each component in MVTrajecter, and MVTrajecter outperformed previous methods on three major MVPT datasets.

Even though we found that utilizing information from multiple timestamps in past trajectories leads to a better tracking performance, we only utilized information at the past 7 timestamps due to the limitations of GPU memory. Therefore, MVTrajecter cannot track someone whose detection continuously fails in over 7 timestamps. In future work, we plan to explore efficient ways to utilize information over the longer term.

References

[1] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. BoT-SORT: Robust associations multi-pedestrian tracking. *arXiv:2206.14651*, 2022. 3, 4

[2] Sithu Aung, Haesol Park, Hyungjoo Jung, and Junghyun Cho. Enhancing multi-view pedestrian detection through generalized 3D feature pulling. In *WACV*, 2024. 3, 2

[3] Pierre Baqué, François Fleuret, and Pascal Fua. Deep occlusion reasoning for multi-camera multi-target detection. In *ICCV*, 2017. 3

[4] J. Berclaz, F. Fleuret, and P. Fua. Robust people tracking with global trajectory optimization. In *CVPR*, 2006. 1, 3

[5] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *CVPR*, 2020. 3

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 3

[7] Orcun Cetintas, Guillem Brasó, and Laura Leal-Taixé. Unifying short and long-term tracking with graph hierarchies. In *CVPR*, 2023. 3

[8] Tatjana Chavdarova and François Fleuret. Deep multi-camera people detection. In *ICMLA*, 2017. 3

[9] Tatjana Chavdarova, Pierre Baqué, Stéphane Bouquet, Andrii Maksai, Cijo Jose, Timur Bagautdinov, Louis Lettry, Pascal Fua, Luc Van Gool, and François Fleuret. Wildtrack: A multi-camera hd dataset for dense unscripted pedestrian detection. In *CVPR*, 2018. 1, 2, 6, 3

[10] 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. In *ICCV*, 2023. 3, 6, 7

[11] Riu Cherdchusakulchai, Sasin Phimsiri, Visarut Trairattanapa, Suchat Tungjitnob, Wasu Kudisthalert, Pornprom Kiawjak, Ek Thamwiwatthana, Phawat Borisuitsawat, Teepakorn Tosawadi, Pakcheera Choppradi, et al. Online multi-camera people tracking with spatial-temporal mechanism and anchor-feature hierarchical clustering. In *CVPRW*, 2024. 1

[12] Yutao Cui, Chenkai Zeng, Xiaoyu Zhao, Yichun Yang, Gangshan Wu, and Limin Wang. SportsMOT: A large multi-object tracking dataset in multiple sports scenes. In *ICCV*, 2023. 1

[13] Peng Dai, Renliang Weng, Wongun Choi, Changshui Zhang, Zhangping He, and Wei Ding. Learning a proposal classifier for multiple object tracking. In *CVPR*, 2021. 3

[14] Kosta Dakic, Kanchana Thilakarathna, Rodrigo N Calheiros, and Teng Joon Lim. Resource-efficient multiview perception: Integrating semantic masking with masked autoencoders. *arXiv:2410.04817*, 2024. 2, 3, 7, 8

[15] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. MOT20: A benchmark for multi object tracking in crowded scenes. *arXiv:2003.09003*, 2020. 1, 3

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[17] Yunhao Du, Junfeng Wan, Yanyun Zhao, Binyu Zhang, Zhihang Tong, and Junhao Dong. GIAOTracker: A comprehensive framework for mcmot with global information and optimizing strategies in visdrone 2021. In *ICCV*, 2021. 2, 3, 4

[18] Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. StrongSORT: Make deepsort great again. *IEEE Transactions on Multimedia*, 2023. 2, 3, 4

[19] Mohamed Elhoseny. Multi-object detection and tracking (MODT) machine learning model for real-time video surveillance systems. *CSSP*, 2020. 1

[20] Martin Engilberge, Weizhe Liu, and Pascal Fua. Multi-view tracking using weakly supervised human motion prediction. In *WACV*, 2023. 1, 2, 3, 7, 4, 5

[21] Martin Engilberge, Haixin Shi, Zhiye Wang, and Pascal Fua. Two-level data augmentation for calibrated multi-view detection. In *WACV*, 2023. 3

[22] Ran Eshel and Yael Moses. Homography based multiple camera detection and tracking of people in a dense crowd. In *CVPR*, 2008. 1

[23] Wei Feng, Feifan Wang, Ruize Han, Yiyang Gan, Zekun Qian, Junhui Hou, and Song Wang. Unveiling the power of self-supervision for multi-view multi-human association and tracking. *TPAMI*, 2024. 1, 3

[24] James Ferryman and Ali Shahrokni. PETS2009: Dataset and challenge. In *PETS*, 2009. 1

[25] Francois Fleuret, Jerome Berclaz, Richard Lengagne, and Pascal Fua. Multicamera people tracking with a probabilistic occupancy map. *TPAMI*, 2007. 1, 3

[26] Hiroshi Fukui, Taiki Miyagawa, and Yusuke Morishita. Multi-object tracking as attention mechanism. In *ICIP*, 2023. 5

[27] Ruize Han, Wei Feng, Yujun Zhang, Jiewen Zhao, and Song Wang. Multiple human association and tracking from egocentric and complementary top views. *TPAMI*, 2021. 1

[28] Ruize Han, Wei Feng, Feifan Wang, Zekun Qian, Haomin Yan, and Song Wang. Benchmarking the complementary-view multi-human association and tracking. *IJCV*, 2024. 1

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[30] Martin Hofmann, Daniel Wolf, and Gerhard Rigoll. Hypergraphs for joint multi-view reconstruction and multi-object tracking. In *CVPR*, 2013. 1

[31] Yunzhong Hou and Liang Zheng. Multiview detection with shadow transformer (and view-coherent data augmentation). In *ACMMM*, 2021. 3, 2

[32] Yunzhong Hou, Liang Zheng, and Stephen Gould. Multiview detection with feature perspective transformation. In *ECCV*, 2020. 1, 2, 3, 5, 6

[33] Weiming Hu, Min Hu, Xue Zhou, Tieniu Tan, Jianguang Lou, and Steve Maybank. Principal axis-based correspondence between multiple cameras for people tracking. *TPAMI*, 2006. 1

[34] Chang Huang, Yuan Li, and Ramakant Nevatia. Multiple target tracking by learning-based hierarchical association of detection responses. *TPAMI*, 2012. 3

[35] Hao Jiang, Sidney Fels, and James J. Little. A linear programming approach for multiple object tracking. In *CVPR*, 2007. 3

[36] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *J Fluids Eng*, 1960. 2, 5, 8

[37] Rangachar Kasturi, Dmitry Goldgof, Padmanabhan Soundararajan, Vasant Manohar, John Garofolo, Rachel Bowers, Matthew Boonstra, Valentina Korzhova, and Jing Zhang. Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *TPAMI*, 2008. 6, 2

[38] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 2018. 6

[39] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 1955. 4

[40] Wei-Yu Lee, Ljubomir Jovanov, and Wilfried Philips. Multi-view target transformation for pedestrian detection. In *WACVW*, 2023. 3

[41] Bastian Leibe, Konrad Schindler, and Luc Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *ICCV*, 2007. 3

[42] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *ICLR*, 2018. 7, 2

[43] 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. *IJCV*, 2021. 4

[44] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. TrackFormer: Multi-object tracking with transformers. In *CVPR*, 2022. 5

[45] Anton Milan, Laura Leal-Taixe, Ian Reid, Stefan Roth, and Konrad Schindler. MOT16: A benchmark for multi-object tracking. *arXiv:1603.00831*, 2016. 1, 3

[46] Duy MH Nguyen, Roberto Henschel, Bodo Rosenhahn, Daniel Sonntag, and Paul Swoboda. LMGP: Lifted multicut meets geometry projections for multi-camera multi-object tracking. In *CVPR*, 2022. 1

[47] Alexandru Niculescu-Mizil, Deep Patel, and Iain Melvin. MCTR: Multi camera tracking transformer. *arXiv:2408.13243*, 2024. 3

[48] Jonah Ong, Ba-Tuong Vo, Ba-Ngu Vo, Du Yong Kim, and Sven Nordholm. A bayesian filter for multi-view 3D multi-object tracking with occlusion handling. *TPAMI*, 2020. 1

[49] AG Amitha Perera, Chukka Srinivas, Anthony Hoogs, Glen Brooksby, and Wensheng Hu. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *CVPR*, 2006. 3

[50] Zekun Qian, Ruize Han, Wei Feng, and Song Wang. From a bird's eye view to see: Joint camera and subject registration without the camera calibration. In *CVPR*, 2024. 1

[51] Zheng Qin, Sanping Zhou, Le Wang, Jinghai Duan, Gang Hua, and Wei Tang. MotionTrack: Learning robust short-term and long-term motions for multi-object tracking. In *CVPR*, 2023. 2, 3

[52] Rui Qiu, Ming Xu, Yuyao Yan, Jeremy S Smith, and Xi Yang. 3D random occlusion and multi-layer projection for deep multi-camera pedestrian localization. In *ECCV*, 2022. 3

[53] Kha Gia Quach, Pha Nguyen, Huu Le, Thanh-Dat Truong, Chi Nhan Duong, Minh-Triet Tran, and Khoa Luu. DyGLIP: A dynamic graph model with link prediction for accurate multi-camera multiple object tracking. In *CVPR*, 2021. 1, 3

[54] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1, 3

[55] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV*, 2016. 6

[56] Gemma Roig, Xavier Boix, Horesh Ben Shitrit, and Pascal Fua. Conditional random fields for multi-camera object detection. In *ICCV*, 2011. 3

[57] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 6

[58] Liangchen Song, Jialian Wu, Ming Yang, Qian Zhang, Yuan Li, and Junsong Yuan. Stacked homography transformations for multi-view pedestrian detection. In *ICCV*, 2021. 3

[59] Andreas Specker. OCMCTrack: Online multi-target multi-camera tracking with corrective matching cascade. In *CVPRW*, 2024. 1

[60] Peize Sun, Jinkun Cao, Yi Jiang, Zehuan Yuan, Song Bai, Kris Kitani, and Ping Luo. DanceTrack: Multi-object tracking in uniform appearance and diverse motion. In *CVPR*, 2022. 1, 3

[61] Satoshi Suzuki, Shotaro Tora, and Ryo Masumura. Scene generalized multi-view pedestrian detection with rotation-based augmentation and regularization. In *ICIP*, 2024. 3

[62] Torben Teepe, Philipp Wolters, Johannes Gilg, Fabian Herzog, and Gerhard Rigoll. EarlyBird: Early-fusion for multi-view tracking in the bird's eye view. In *WACVW*, 2024. 2, 3, 5, 6, 7, 8, 4

[63] Torben Teepe, Philipp Wolters, Johannes Gilg, Fabian Herzog, and Gerhard Rigoll. Lifting multi-view detection and tracking to the bird's eye view. In *CVPRW*, 2024. 2, 3, 6, 7, 4, 5, 8, 9

[64] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, 2017. 2, 3, 5, 7

[65] Jeet Vora, Swetanjal Dutta, Kanishk Jain, Shyamgopal Karthik, and Vineet Gandhi. Bringing generalization to deep multi-view pedestrian detection. In *WACV*, 2023. 2, 5, 6, 1, 3

[66] Yizhou Wang, Tim Meinhardt, Orcun Cetintas, Cheng-Yen Yang, Sameer Satish Pusegaonkar, Benjamin Missaoui, Sujit Biswas, Zheng Tang, and Laura Leal-Taixé. BEV-SUSHI: Multi-target multi-camera 3d detection and tracking in bird's-eye view. *arXiv:2412.00692*, 2024. 1, 3, 7

[67] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *ECCV*, 2020. 2, 3, 4

[68] Tim Wengefeld, Steffen Müller, Benjamin Lewandowski, and Horst-Michael Gross. A multi modal people tracker for real time human robot interaction. In *RO-MAN*, 2019. 1

[69] Yuanlu Xu, Xiaobai Liu, Yang Liu, and Song-Chun Zhu. Multi-view people tracking via hierarchical trajectory composition. In *CVPR*, 2016. 1

[70] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. TransCenter: Transformers with dense representations for multiple-object tracking. *TPAMI*, 2022. 5

[71] Yihan Yang, Ming Xu, Jason F Ralph, Yuchen Ling, and Xiaonan Pan. An end-to-end tracking framework via multi-view and temporal feature aggregation. *CVIU*, 2024. 2, 3, 6, 7, 4, 5

[72] Quanzeng You and Hao Jiang. Real-time 3D deep multi-camera tracking. *arxiv:2003.11753*, 2020. 1

[73] En Yu, Zhuoling Li, and Shoudong Han. Towards discriminative representation: Multi-view trajectory contrastive learning for online multi-object tracking. In *CVPR*, 2022. 2, 3, 4

[74] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. MOTR: End-to-end multiple-object tracking with transformer. In *ECCV*, 2022. 5

[75] Li Zhang, Yuan Li, and Ramakant Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008. 3

[76] Qi Zhang, Wei Lin, and Antoni B Chan. Cross-view cross-scene multi-view crowd counting. In *CVPR*, 2021. 2

[77] Qi Zhang, Kaiyi Zhang, Antoni B Chan, and Hui Huang. Mahalanobis distance-based multi-view optimal transport for multi-view crowd localization. In *ECCV*, 2024. 3

[78] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. FairMOT: On the fairness of detection and re-identification in multiple object tracking. *IJCV*, 2021. 6

[79] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv:1904.07850*, 2019. 1, 3, 5

[80] Xingyi Zhou, Tianwei Yin, Vladlen Koltun, and Philipp Krähenbühl. Global tracking transformers. In *CVPR*, 2022. 2, 3, 5

[81] Ji Zhu, Hua Yang, Nian Liu, Minyoung Kim, Wenjun Zhang, and Ming-Hsuan Yang. Online multi-object tracking with dual matching attention networks. In *ECCV*, 2018. 5