Integrating Viewing Direction and Image Features for Robust Multi-View Multi-Object 3D Pedestrian Tracking

Rasho Ali *, Max Mehltretter, Christian Heipke

Institute of Photogrammetry and GeoInformation, Leibniz University Hannover, Germany
(ali, mehltretter, heipke)@ipi.uni-hannover.de

Keywords: Image Sequence Analysis, Multi-View Tracking, Detection and Localization in 3D

Abstract

Recently, there has been growing interest in the development of 3D multi-view, multi-object detection and tracking models (MV-MOD and MV-MOT), resulting in significant methodological advances. However, many of these developments do not address the critical challenge of generalization across different camera constellations, i.e., having camera constellations that differ between training and testing, limiting their effectiveness in real-world applications. A key factor often overlooked is the influence of the direction of the optical axis during image capture, which is not adequately propagated in the model. In this work, we propose a novel convolutional neural network-based method for 3D MV-MOD and MV-MOT that enhances generalization by incorporating the direction from which the images were captured as an additional input to this network. For each image, this directional information is combined with the 2D features extracted from that image, before 3D features are computed, using the 2D features from all images. We empirically evaluate the performance of the proposed method on the real-world Wildtrack dataset, demonstrating the effectiveness of the proposed approach.

1. INTRODUCTION

3D multi-view multi-object tracking is the task of using images from multiple cameras to detect and track objects in a scene. This task is typically performed in 3D, where an occupancy map of the objects represented in the bird’s eye view (BEV) is used to locate and track these objects. The input to a model addressing this task consists of images and the intrinsic and extrinsic parameters of the cameras for each view at each time stamp, while the output consists of the object occupancy map as a function of time, along with a unique ID for each tracked object. 3D MV-MOT plays a crucial role in a variety of real-world applications, including pedestrian safety, autonomous driving, and sports analysis. Many recent approaches follow a similar scheme, where feature maps from multiple views are projected onto a common plane or into a common volume in 3D object space. Subsequently, BEV features are computed and used for BEV-based detection and tracking. However, the camera parameters are only applied during the projection step, meaning the model does not further take into account the view-point from which an image was captured when combining feature maps from various images. Consequently, the model learns to handle features from different cameras only in the specific configuration it was trained on. For example, in a setup with two cameras pointing in similar directions and capturing an object from the front, and a third camera capturing the same object from the back, the model should treat the rear view features differently from those captured from the front. This distinction is important because the features can be significantly different. However, if the model learns to always handle features from cameras 1 and 2 differently from those from camera 3, changing the camera configuration will reduce the model’s ability to generalize.

To address this limitation, we propose a novel method that incorporates the direction from which an image was captured directly into the associated projected features. Thus, our method can distinguish between image features that are captured from similar directions and, therefore, should have similar representations in 3D object space, and image features that are captured from different directions and thus may be different although referring to the same object. While the proposed method is generally applicable to objects of various types, this paper focuses on pedestrian tracking. The contribution of this paper is a novel method which combines parameters of the extrinsic and intrinsic orientation with image feature maps, allowing the model to infer the direction from which each image was captured.

* Corresponding author

2. RELATED WORK

MV-MOT describes a scenario in which a number of cameras in a scene carry out the task of multi-object tracking. The cameras can have overlapping or non-overlapping fields of view, and they can be stationary or mounted on a moving platform. Generally, MV-MOT can be divided into three main steps:

1. Object Detection: Detect objects of interest in each image from each camera view.
2. Spatial Association: Match detections across views by associating those captured from the different cameras
3. Temporal Association: Link detections across consecutive time stamps to track the objects over time

In many approaches (You and Jiang, 2020; Ong et al., 2020; Cheng et al., 2023), each step of the MV-MOT task is performed separately. Typically, the first step involves using a state-of-the-art detection model, such as (Ren et al., 2016; Ge et al., 2021). The use of advanced detection models in the first step allows for better generalization across multiple scenes, as

single-view object detection models have been extensively researched in recent years (Ren et al., 2016; Carion et al., 2020; Zhu et al., 2020). Recently, with the advancement of end-to-end MV-MOD models (Hou et al., 2020; Hou and Zheng, 2021; Song et al., 2021), new end-to-end tracking models have been developed, as MV-MOD deals with the first two steps of MV-MOT, namely, object detection and spatial association. This enables the joint optimization of all three steps in MV-MOT during training. Both, end-to-end trainable methods for MV-MOD and MV-MOT are discussed in the flowing sections.

2.1 End-To-End MV-MOD

End-to-end multi-view multi-object detection models detect pedestrians in each view and associate the detections between the views of a single epoch. Generally, these approaches follow a similar paradigm, which can be divided into three main steps. First, image feature maps extracted from the different views are projected onto a common ground plane (Hou et al., 2020; Hou and Zheng, 2021; Vora et al., 2023) or into a common 3D feature volume represented by a voxel grid (Song et al., 2021; Harley et al., 2023; Aung et al., 2024). Second, the projected features are aggregated to create common BEV-features. Third, the aggregated BEV-features are used as input to a detection head to predict the BEV positions of the pedestrians. Some methods add a 2D detection head to detect pedestrians in each of the input images (Hou et al., 2020; Hou and Zheng, 2021), which helps to achieve higher activations at pedestrian locations and thus to extract more meaningful feature maps. Other methods use a non-parameterized layer for the BEV aggregation step (e.g., an average pooling layer) to handle a variable number of images and thus to augment the number of images used during training (Vora et al., 2023); and some weight the image features before projection (Aung et al., 2024), emphasizing regions of interest, i.e., pedestrian locations.

Despite advancements in the field, a key limitation of current approaches is their inability to generalize across different scenes or even the same scene captured from different views, as shown in (Vora et al., 2023; Teepe et al., 2024b). One reason for this limitation is that the projected features do not contain information about the direction from which the images were captured. Consequently, such models learn to handle features from different images only in the specific configuration they were trained on. Additionally, some of the methods use a homography to project image feature maps onto a common plane in 3D object space, commonly assumed to be a ground plane. However, this approach is suboptimal for objects above the ground plane as such objects violate the assumption inherent in the homography, i.e., that every 3D object point is located on a 2D plane. In such cases, only the position of the feet is accurately projected, while other body parts are misaligned and projected into different directions depending on the position of the camera, resulting in distorted patterns and shadow-like artifacts. This distortion becomes larger the more parallel the optical axis of a camera is to the ground, resulting in very elongated projections. These distortions result in the fact that features belonging to the same body part of the same person lie at different positions in the assumed plane, meaning that such features cannot be aggregated in a straightforward manner, i.e, by averaging or by convolution.

In so-called 3D feature pulling, or simply 3D-pulling, as presented by Harley et al. (2023), a 3D voxel grid is generated in object space and the voxels are projected into the image space of each camera, creating a mapping between the 3D voxels and their corresponding pixel coordinates. However, the actual 3D position of an object is only determined by combining features from multiple images. This contrasts with 3D feature lifting, where 3D positions are initially estimated independently per image. 3D feature lifting, which was introduced in (Philion and Fidler, 2020), estimates the depth for each pixel along the corresponding camera ray using a trainable layer to simulate a point cloud from the camera image. The estimated depth is used to project the image features along the corresponding camera ray, generating a point cloud within each camera frustum. These point clouds are then mapped into a shared voxel grid. Harley et al. (2023) and Teepe et al. (2024b) showed that the 3D lifting approach achieves results similar to 3D feature pulling. However, as shown in (Teepe et al., 2024b), this type of projection method has proven to be less robust to scene changes where the inference scene differs from the training scene, compared to the non-parametric 3D-pulling. This finding makes 3D-pulling an attractive option for robust multi-view detection. In addition, its non-parametric nature makes it immune against any overfitting issues in the projection step.

In the context of object geotagging, Nassar et al. (2020) proposed a method that leverages soft geometry constraints based on the geo-location of camera poses to identify the same object across two views. Their approach employs a Siamese network architecture which processes pairs of images by concatenating camera pose information (e.g., image geolocation and the heading angle of the object inside the image) with image crops. These combined features are then decoded using a CNN to re-identify the same object in the second view. In contrast to Nassar et al. (2020), which focuses on object geotagging using two images, our model addresses multi-view multi-object detection and tracking. While their approach is limited to pairwise comparisons via Siamese networks, our method is designed to handle multiple camera views simultaneously.

2.2 MV-MOT

Research in MV-MOT has so far been relatively limited compared to single-view multi-object tracking (SV-MOT). MV-MOT models can be broadly classified into three main categories. The first category uses SV-MOT models, such as (Nguyen and Heipke, 2020; Henschel, 2021; Zhang et al., 2022), to address pedestrian detection and temporal association tasks. This is followed by spatial association, which utilizes epipolar geometry to find correspondences based on locations on the ground plane (Hu et al., 2006; Xu et al., 2016). In the second category, the models first solve the spatial association task, followed by the temporal association as done in (Nguyen et al., 2022; Cheng et al., 2023). ReST (Cheng et al., 2023) uses a Graph Neural Network for the tracking task. In this model, the nodes of the graph, which represent individual tracked pedestrians, are determined based on the appearance, position, and speed of the pedestrians. The position is defined by the feet of the pedestrians, represented by the midpoint of the bottom of their bounding boxes, using a homography, while the speed is represented as the difference in position between two frames. The edges of the graph encode the relationships between pedestrians across frames, representing temporal and spatial associations. A major drawback of such models is their dependence on the accuracy of the bounding box predictions. If the position of the feet is incorrectly predicted, e.g., due to occlusion, a common problem in tracking tasks, this can lead to incorrect projections and consequently incorrect spatial associations.

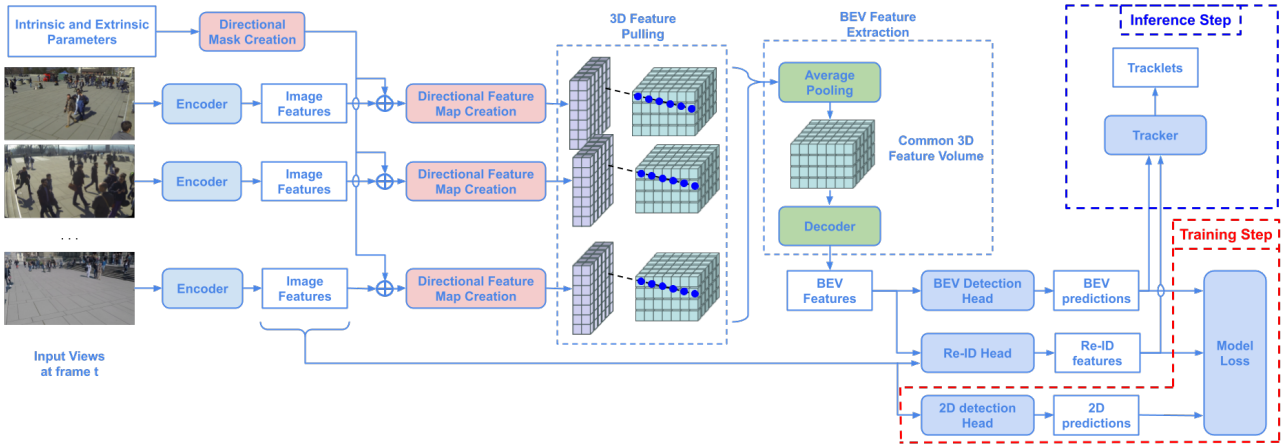

Figure 1. The figure illustrates the model’s training and inference setups. The areas used only during training are enclosed within a dashed red box, while the areas used only during inference are enclosed within a dashed blue box. The input views are first encoded, and the resulting image feature map, combined with the intrinsic and extrinsic parameters, are used to compute directional feature maps, which are projected into a 3D voxel grid for each image. The 3D voxel grids from the different images are then aggregated into a common 3D voxel grid, from which BEV features are extracted using a 2D convolutional layer. These BEV features are subsequently used to predict the BEV positions of pedestrians. Finally, these detections, along with their corresponding re-identification features, are utilized to associate detections into tracklets.

The last category is end-to-end MV-MOT. Many papers have recently discussed end-to-end SV-MOT (Zhou et al., 2020; Zhang et al., 2021; Ali et al., 2023), which is strongly driven by the significant improvements in single-view object detection approaches. However, the same cannot be said regarding end-to-end MV-MOT as only a few works have addressed this task. End-to-end MV-MOT is similar to end-to-end MV-MOD in that it follows the same first three steps: feature projection, aggregation, and detection. However, MV-MOT may include an additional re-identification head, as in (Teepe et al., 2024a). Alternatively, in (Engilberge et al., 2023; Teepe et al., 2024b), the aggregated features from frame t are concatenated with the aggregated features from frame $t-1$ and are used as input to a BEV tracking head that predicts the motion of each detection in BEV space.

As object tracking methods are usually built on top of existing object detection methods, they typically inherit the drawbacks of the respective detection model. For example, (Engilberge et al., 2023; Teepe et al., 2024a) use a homography to project image features onto a common plane, which leads to the aforementioned shadow-like distorted features. On the other hand, (Teepe et al., 2024b) experimented with 3D-lifting for the projection step and report robustness problems. This could be due to overfitting of the model to the trained views, preventing it from generalizing effectively.

3. A New Method for Robust Pedestrian Detection and Tracking including Viewing Direction

Given a set of N images $I_{i,t}, i = 1, 2, ..., N$ captured at a time step t by N synchronized cameras with known extrinsic and intrinsic parameters, our method predicts the 2D positions of the observed pedestrians on a ground plane in the bird’s eye view as a function of time. The method builds on the main ideas used in MV-MOD, namely, MvDet (Hou et al., 2020) and EarlyBird (Teepe et al., 2024a), with the use of 3D feature pulling instead of using a homography to project the image features onto a common plane. Additionally, we introduce a novel layer that generates directional feature maps, incorporating camera intrinsic and extrinsic parameters. This enables our method to consider the directions from which the images have been captured when fusing features from different images after 3D feature pulling. Our method can be broken down into six main parts, as shown in Figure 1:

1. For each image $I_{i,t}$, the feature map is extracted using a ResNet18 backbone CNN (He et al., 2016), chosen for its compact size, which reduces the computational resources needed to process multiple images as input.
2. For each image, the method generates a directional feature map; these are described in more detail in the next section.
3. The directional feature map of each image is projected into a 3D voxel grid, generating a 3D feature volume for each image.
4. All the 3D feature volumes from the different images are aggregated to create a common 3D feature volume. The aggregation is done using average pooling.
5. A Resnet-18 based decoder is used to extract BEV features from the aggregated 3D feature volume.
6. Both, image and BEV features, are used to compute pedestrian re-identification (re-ID) features in the re-ID head, while only the BEV features are used as input to the BEV detection head to predict the BEV positions of the pedestrians.
7. Tracklets are generated based on the BEV detections of the current frame and the re-identification features of both the current and the previous time step.

3.1 Directional Feature Map

Image feature maps which are the output of the image encoder, in our case ResNet18, and so-called directional masks are used to calculate directional feature maps, one for each image. The directional mask is a 2D matrix with size $(w \times h \times 3)$, where w and h are the width and height of the corresponding image

feature map. Each pixel of the matrix contains a unit vector representing the direction from the projection center of the camera to that pixel. To compute the mask, we follow these steps:
Given an undistorted image, the rotation matrix R describing the rotation between image and object coordinate system, and the intrinsic parameters, i.e, the coordinates of the principal point u_0, v_0 and the camera constant c, the directional mask can be calculated using the following equations:

$$d = \frac{1}{\lambda} R \left(\begin{pmatrix} u \\ v \\ 0 \end{pmatrix} - \begin{pmatrix} u_0 \\ v_0 \\ c \end{pmatrix} \right) \tag{1}$$

$$\hat{d} = \frac{d}{|d|} \tag{2}$$

where u, v are the pixel coordinates in the image coordinate system, λ is a scaling factor which can be ignored as we normalize the calculated vectors, d is a vector from the projection center to the object point while $\hat{d}$ is the unit vector in the same direction. The directional mask is obtained by computing $\hat{d}$ for each pixel in an image. Afterwards, the directional mask is concatenated with the respective image feature map, and fed into a convolutional block. The convolutional block consists of a 3×3 2D convolution layer, followed by a ReLU activation layer, and a 1×1 2D convolution, producing an output feature map to which we refer to as directional feature map.

3.2 3D Feature Pulling

In 3D feature pulling, a predefined discrete voxel grid $V \in \mathbb{N}^{X \times Y \times Z}$ in the object coordinate system is projected into the image plane of each camera similarly to (Harley et al., 2023). Here, X and Y represent the number of discretization steps along the tow horizontal directions of the ground plane, while Z denotes the number of discretization steps along the vertical dimension. For each voxel, the directional feature map is then sampled in each image using bilinear interpolation, creating N 3D feature volumes, one for each camera. Since not all voxels are guaranteed to fall within the frustum of a given camera, some voxels will end up outside of some of the images, these voxels are filled with zeros. The projection of the voxel grid into image space of a camera is calculated using the collinearity equations:

$$\begin{pmatrix} u \\ v \\ 0 \end{pmatrix} - \begin{pmatrix} u_0 \\ v_0 \\ c \end{pmatrix} = \lambda R^T \left(\begin{pmatrix} x \\ y \\ z \end{pmatrix} - \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} \right) \tag{3}$$

Where x_0, y_0, z_0 are the coordinates of the projection center in object coordinate system and x, y, z describe the position of an object point in the object coordinate system.

3.3 BEV Feature Extraction

Similar to (Vora et al., 2023), we employ a non-parametric pooling layer to aggregate the N 3D feature volumes into a common 3D feature volume. The choice of a pooling layer, as in (Vora et al., 2023), is motivated by its permutation invariance, ensuring that the operation is unaffected by the order in which images from different views are processed. Additionally, since the pooling layer is non-parametric, it reduces the risk of overfitting introduced by this aggregation step. Similar to (Vora et al., 2023), we choose to use an average-pooling layer. After the pooling step, features along the Z-axis are concatenated and used as input to the Resnet-18 based decoder block to extract BEV features.

3.4 Prediction Heads and Losses

Prediction Heads The architectures of the BEV detection, 2D detection and the re-identification heads follow (Teepe et al., 2024a). The BEV detection head predicts an occupancy map on the ground plane and an offset to the center point of these predicted cells to mitigate quantization errors that may arise from the grid-based representation. To help the model focus on pedestrian-related features, the 2D detection head predicts the centers of the 2D bounding boxes, which contribute to higher activations at pedestrian positions. The re-identification head is used to predict distinctive features for each pedestrian, allowing them to be differentiated from one another. These features are extracted from both, BEV features and the image feature maps.

Losses BEV detections are optimized using a focal loss while the offsets are optimized with an L1 loss. Following Kendall et al. (2018), an uncertainty term is used to automatically balance the single-task losses before combining them. The idea is that instead of using fixed weights for different loss terms, the model learns to balance them automatically by predicting uncertainty parameters. Similar to the re-identification method used in FairMOT (Zhang et al., 2021) re-identification is achieved by learning re-ID features through classification and metric learning, from both, the BEV and the image feature maps. The classification branch treats each identity as a distinct class and uses cross-entropy loss to learn discriminative features. The identities are the unique IDs given to each pedestrian in the training dataset. The metric learning component ensures that embeddings of the same identity are pulled closer together in feature space while pushing different identities apart; we use the SupCon Loss (Khosla et al., 2020) for this step. The model loss is the sum of all the mentioned losses.

3.5 Tracking

Tracking is achieved in two steps, following the approach outlined in (Chen et al., 2018) and using the same thresholds as (Teepe et al., 2024a). In the first step, tracklets are initialized based on the predictions in the BEV. In subsequent timestamps, the BEV predictions are linked to existing tracklets using the following steps: The re-ID features and the motion prediction based on a Kalman Filter (Kalman, 1960) with a constant velocity assumption, computed from the frames $t-2$ and $t-1$, are used to generate initial tracking results for frame t. The process employed in the following is similar to DeepSORT (Wojke et al., 2017), where the Mahalanobis distance D_m is calculated between the predicted and the detected position. Additionally, the cosine similarity D_c between ID-features of detections in the previous and current time step are calculated, and both distances are combined into $D = \gamma D_c + (1-\gamma) D_m$ where γ is a weighting factor set to 0.98. At this step, the Mahalanobis distance is set to infinity if it is larger than $0.4m$ to prevent unrealistic motion trajectories. At the end of this step, detections are matched to existing tracks using the Hungarian method (Kuhn, 1955). Note that typically not all detections are matched, e.g., due to occlusions or inaccuracies in BEV detection, which can lead to errors in motion prediction. In the second step, the model attempts to match previously unmatched detections by increasing the Mahalanobis distance threshold from $0.4m$ to

Method	MODA%	MODP%	Prec.%	Recall%	IDF1%	MOTA%
MVDet (Hou et al., 2020)	88.2	75.7	94.7	93.6	-	-
GMVD (Vora et al., 2023)	86.7	76.2	95.1	91.4	-	-
SHOT (Song et al., 2021)	90.2	76.5	96.1	94.0	-	-
MVDeTr (Hou and Zheng, 2021)	91.5	**82.1**	**97.4**	94.0	-	-
(Aung et al., 2024)	**94.1**	78.8	96.4	**97.7**	-	-
ReST (Cheng et al., 2023)	-	-	-	-	86.7	84.9
EarlyBird (Teepe et al., 2024a)	91.2	81.8	94.9	96.3	92.3	89.5
TrackTacular (3D-Pulling) (Teepe et al., 2024b)	92.1	76.2	97.0	95.1	**95.3**	**91.8**
Our (w.o. Directional Information)	88.7	81.7	94.1	94.6	90.0	86.5
Our (Directional Features)	89.3	81.7	93.7	95.7	92.0	87.4
Our (Simplified Directional Features)	89.0	81.5	93.6	95.5	91.6	87.8

Table 1. Comparison of our method to various state-of-the-art detection and tracking methods with training and inference on all 7 views.

$2.5m$. If a detection still does not find a match, it is classified as a new track. Finally, tracks without matches for 10 consecutive frames are ended.

4. Experiments

4.1 Datasets and Metrics

WildTrack is a real-world multi-view dataset consisting of image sequences captured by seven cameras at a size of 1920×1080 pixels, each comprising of 400 frames per camera, annotated at two frames per second. The dataset covers an area of $12m \times 36m \times 2m$, which is quantized into a $480 \times 1440 \times 4$ grid with a cell size of 2.5 cm² in the X and Y directions and 50 cm in the Z direction. The extension in Z has been chosen to be $2m$ as it approximately represents the maximum height of an adult human. Following Chavdarova et al. (2018), Vora et al. (2023) and Teepe et al. (2024a), we use the first 90% of the frames for training and the last 10% for testing. Figure 2 shows the distribution of the cameras used in the WildTrack dataset, along with the frustums of their fields of view, illustrating the coverage from each camera.

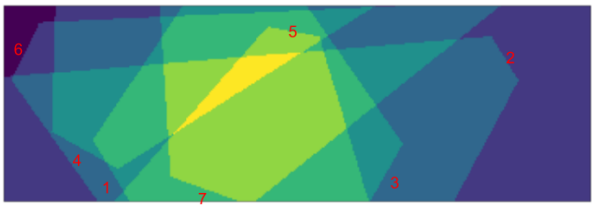

Figure 2. The distributions of the cameras used in WildTrack dataset, along with the frustums of their fields of view and the overlap between the cameras fields of view in BEV.

We use the WildTrack dataset because it offers a real-world multi-view setting with pedestrian annotations, making it suitable for evaluating our model's performance in multi-camera pedestrian detection and tracking. Additionally, it allows comparison with state-of-the-art models for generalization to new camera configurations, where camera positions differ between the training and test sets.

Detection Metrics We utilize the standard evaluation metrics proposed in (Chavdarova et al., 2018) to assess the performance of our multi-view detection model. Comparisons with the ground truth are based on the Euclidean distance between the predicted and the ground truth BEV position (GT) of a pedestrian. A detection is classified as a true positive, if it falls within a radius of $r = 0.5m$ around the GT, which approximately corresponds to the average radius of a human body. The primary performance indicator is Multiple Object Detection Accuracy (MODA), which accounts for normalized missed detections and false positives, considering both, false negatives and false positives. MODA is computed as: $MODA = 1 - \frac{FP+FN}{GT}$ where n is the number of GT pedestrians, and FP and FN are the false positives and false negatives, respectively. We also use Multiple Object Detection Precision (MODP) to assess localization precision:
$MODP = \frac{\sum 1-p[p<s]/s}{TP}$ where p is the distance from a detection to its GT, s is a threshold set to $0.5m$ to ascertain true positives, and TP is the count of true positives. Additionally, precision and Recall are computed.

Tracking Metrics To evaluate the tracking performance quantitatively, we utilize Multiple Object Tracking Accuracy (MOTA) (Bernardin and Stiefelhagen, 2008) and the Identity-F1 score (IDF1) (Ristani et al., 2016). These metrics provide a thorough evaluation of tracking quality. MOTA combines three different error metrics, including identity (ID) switches, false positives, and false negatives, to calculate a single score. By summing up these metrics over all epochs and dividing the sum by the total number of pedestrians in all epochs, we obtain the total error rate, with MOTA being: $\text{MOTA} = 1 - \frac{FN+FP+IDSW}{GT}$.
The IDF1 score specifically assesses the accuracy and consistency of pedestrian identifiers and their trajectories by combining ID precision (IDP) and ID Recall (IDR) using their harmonic mean: $\text{IDF1} = \frac{2 \times IDP \times IDR}{IDP+IDR}$.
IDP quantifies the proportion of true positives relative to the total of true positives and false positives, while IDR measures the proportion of true positives against the total of true positives and false negatives.

4.2 Implementation Details

To reduce memory usage, the images are first resized to an input resolution of 1280×720 pixels, similar to (Hou et al., 2020). For augmentation during training, we follow (Hou and Zheng, 2021; Harley et al., 2023): we apply random resizing and cropping on the RGB input, in a scale range of [0.8,1.2] and adapt the camera intrinsic matrix K accordingly. We train the detector using an Adam optimizer with a one-cycle learning rate scheduler with a maximum learning rate of 10^{-3}. Training with a batch size of 1, while accumulating gradients over several batches, we reach an effective batch size of 8. To initialize

	Method	Inference on $\{2,4,5,6\}$				Inference on $\{1,3,5,7\}$			
		MODA	MODP	Prec	Recall	MODA	MODP	Prec	Recall
Trained on {2,4,5,6}	MVDet (Hou et al., 2020)	85.2	72.2	92.6	92.5	43.2	68.2	94.6	45.8
	MVDeTr (Hou and Zheng, 2021)	75.4	79.5	**96.9**	77.9	41.7	73.7	92.0	45.7
	SHOT (Song et al., 2021)	81.9	74.1	94.1	87.4	51.4	72.5	94.4	54.6
	GMVD (Vora et al., 2023)	84.0	72.9	92.4	91.6	75.1	71.1	94.3	79.9
	EarlyBird Teepe et al. (2024a)	**91.0**	**81.5**	96.8	**94.1**	78.1	79.9	94.9	**82.5**
	Our (w.o. Directional Information)	86.4	80.0	95.8	90.4	76.5	**80.1**	95.4	80.4
	Our (Directional Features)	86.9	79.8	96.1	90.6	**78.8**	79.9	**96.4**	81.8
	Our (Simplified Directional Features)	86.2	79.6	95.6	90.4	77.9	79.3	95.9	81.4
Trained on{1,3,5,7}	MVDet (Hou et al., 2020)	27.8	68.7	90.8	31.0	78.2	73.6	89.5	88.6
	MVDeTr (Hou and Zheng, 2021)	5.6	65.5	62.4	14.0	72.5	78.9	95.0	76.5
	SHOT (Song et al., 2021)	15.3	62.9	89.2	17.4	79.7	76.4	95.7	83.5
	GMVD (Vora et al., 2023)	62.6	67.4	86.7	73.9	80.8	74.0	94.2	86.0
	EarlyBird (Teepe et al., 2024a)	**85.9**	**78.5**	**96.9**	**88.4**	**85.9**	78.5	**96.1**	89.5
	Our (w.o. Directional Information)	75.8	**78.5**	91.9	83.1	83.8	**80.9**	94.9	88.6
	Our (Directional Features)	77.5	77.7	92.0	84.8	85.3	79.7	95.5	**89.6**
	Our (Simplified Directional Features)	76.1	78.4	92.0	83.3	85.3	80.7	95.9	89.1

Table 2. Comparison of methods trained on different camera sets, including results for MVDet, MVDeTR, SHOT, and GMVD as reported by Vora et al. (2023), and EarlyBird trained as per Teepe et al. (2024a). All results are in %.

the encoder and decoder networks, we use weights pre-trained on ImageNet-1K provided by PyTorch.

4.3 Experimental Setup

We evaluated our model using three distinct configurations. The first configuration serves as the baseline, where the directional feature maps are not utilized in the 3D pulling step; instead, the model relies solely on the image feature maps produced by the backbone. The second configuration employs the directional model, creating directional features after the backbone step, as described in Section 3 and as shown in Figure 1. Lastly, a simplified directional model uses the direction of the optical axis for all the pixels of an image. The goal of this configuration is to evaluate the impact of a global directional representation on the model's performance, without the complexity of pixel-specific directions.

We compare our model with the state-of-the-art multi-view multi-object detection and tracking models on the WildTrack dataset. The set includes state-of-the-art MV-MOD models including MVDet (Hou et al., 2020), MVDetr (Hou and Zheng, 2021) and SHOT (Song et al., 2021) which use CNN layer for the aggregation step, and GMVD (Vora et al., 2023) and (Aung et al., 2024) which use pooling layer for the aggregation step. Additionally, three MV-MOT models including a non-end-to-end MV-MOT model ReST (Cheng et al., 2023), EarlyBird (Teepe et al., 2024a) and TrackTacular (Teepe et al., 2024b) which use a CNN layer for the aggregation step are employed for comparison.

5. Results

We first compare our model with the state-of-the-art multi-view multi-object detection and tracking models on the WildTrack dataset, using all seven views for both training and testing. The results are shown in Table 1. Overall our model shows somewhat similar performance relative to the other detection models, while lagging behind in some metrics (e.g., MODA compared to (Aung et al., 2024)). One important point to note is that incorporating directional features or the simplified directional features enhances the performance of the proposed model across multiple key metrics, including MODA, Recall, IDF1 and MOTA. Specifically, the enhancement in MODA and Recall metrics reflects improved spatial association, while the gains in IDF1 and MOTA metrics highlight advancements in temporal association.

As GMVD (Vora et al., 2023) reported, the traditional evaluation protocol can be a little misleading because the training and test sets have significant overlap, which promotes overfitting. Therefore, to investigate the generalization capabilities of our model, we performed the same experiment as described by Vora et al. (2023), focusing on generalization to new camera configurations where camera positions are varied between the training and test sets. Figure 3 shows the used camera splits on the WildTrack dataset. It can be observe that in the set where views $(2,4,5,6)$ are used, a larger area of the scene is covered by two or three cameras. In contrast, the set with views $(1,3,4,7)$ provides a larger area where pedestrians can be seen from four different cameras. We trained all models on two sets of camera views separately and then evaluated the trained models on both sets, again separately. The results are compared to state-of-the-art results as reported by Vora et al. (2023) and EarlyBird, trained by us as per Teepe et al. (2024a). The results are shown in Table 2.

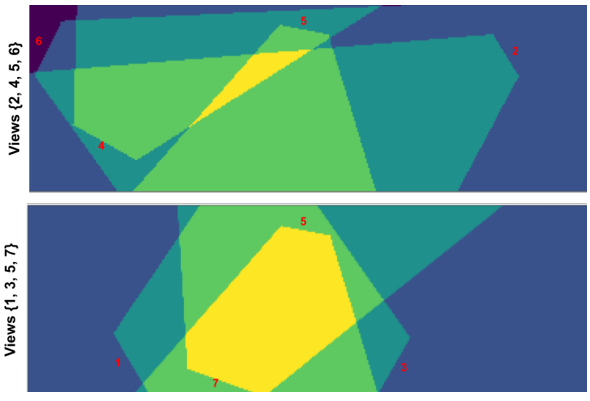

Figure 3. Camera splits of WildTrack dataset, along with the frustums of their fields of view and the overlap between the cameras fields of view in BEV.

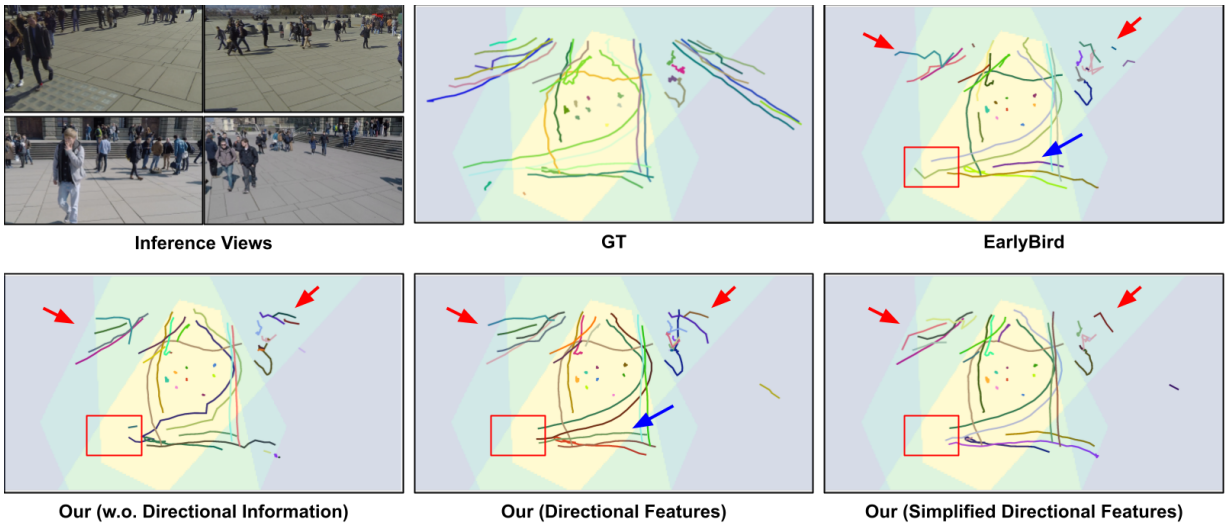

Figure 4. Qualitative results showing five BEV perspectives of pedestrian trajectories, represented by colored lines, as predicted by our different models, and the EarlyBird model for comparison. Additionally, we illustrate the overlap between the cameras fields of view $(1, 3, 5, 7)$ in BEV for each depiction. Trajectory colors indicate pedestrian IDs; however, the same pedestrian may have different IDs across depictions. All the models where trained on views $(2, 4, 5, 6)$ and tested on views $(1, 3, 5, 7)$. The top-left image provides an example of the input images used during inference. The red arrows highlight trajectories that were incorrectly tracked by our model, while the blue arrows indicate tracks that our model successfully followed. The red rectangle marks an area where four views captured the pedestrians, yet our model failed to detect them.

The comparison between our three configurations shows that using directional features improves the model across most metrics. It is notable that directional features show the most improvement when the training set differs from the inference set, especially for both, MODA and Recall. Additionally, it is generally observed that the use of directional features produces slightly better results compared to the simplified directional features. This suggests that the granularity provided by pixel-specific directional information might be beneficial for the model's performance. During inference with the same camera setup, all models show similar performance, with EarlyBird performing the best and our model closely behind, especially the variant with directional features. However, when tested with a different camera configuration, MVDet, MVDeTR, and SHOT experience a significant performance drop. GMVD is more robust to changes in camera setup, however it shows a larger drop when tested on views $(2, 4, 5, 6)$. Our model generally achieved second-best results after the EarlyBird model, except when tested on views $(1, 3, 5, 7)$ where it preforms slightly better. Furthermore, the results indicate that our model is more conservative in its predictions, as shown by the relatively high MODP and the relatively low Recall compared to EarlyBird. A high MODP indicates that the spatial overlap between predicted pedestrians and ground truth pedestrians is accurate. The low Recall demonstrates that the model concentrates on detecting pedestrians with high confidence and precise localization; however, it sacrifices Recall by failing to detect more ground truth pedestrians.

We also present qualitative results for different configurations of our model, illustrating the BEV perspective of pedestrian trajectories, represented by colored lines, as predicted by the various models. The models were trained on views $(2, 4, 5, 6)$ and tested on views $(1, 3, 5, 7)$, with the results shown in Figure 4. In the figure, the blue arrow illustrates an example where using directional features improved both the detection and, subsequently, the tracking of pedestrians in the scene. At the same time, it is evident that all the models struggle to detect pedestrians in areas where the overlap between the camera fields of view arises from cameras with similar viewing axes, or where just one camera can see that area, as depicted by the red arrows. The latter is understandable, as none of the models can estimate the position of pedestrians using one camera. Furthermore, this shows that camera positions play a crucial role in the BEV prediction, as the ability to effectively localize and track pedestrians is significantly dependent on camera placement and coverage provided by multiple views. Lastly, the red rectangle denotes an area where all four camera views overlap, yet our model still struggles to detect pedestrians in that region. This could be due to the used aggregation step as it uses average pooling which means that the features of multiple pedestrians are averaged, potentially leading to misdetections or confusion between pedestrians.

6. Conclusion and Future Works

In this paper, we introduce a novel method for end-to-end 3D multi-view multi-object detection and tracking (MV-MOD and MV-MOT), which integrates directional features to enhance generalization and performance across diverse camera setups. The proposed model, tested on the WildTrack dataset, achieves promising results, particularly in scenarios where the camera configurations used for training and testing differ from each other. We conducted experiments to evaluate three distinct model configurations, which demonstrated that incorporating directional feature maps yielded slightly better results compared to the baseline method, in which no directional information was used.

Nevertheless, there are some limitations that present challenges for the model. The number of views is limited. To overcome this limitation and improve generalization, augmentation could be applied, involving shifting and rotation of all views and the voxel grid together in the global coordinate system. Furthermore, we believe that the aggregation method used is suboptimal and could be improved. It could be replaced with a layer

that performs a weighted average, where the features are weighted based on whether they describe pedestrian or the background and whether they describe the same pedestrian or not. This would help to differentiate between pedestrians more effectively, particularly in cases of occlusion or close proximity. Additionally, the conservative nature of the model leads to more precise detections at the cost of false negatives, which is problematic in the context of autonomous driving, where it is essential to detect and localize all pedestrians in the scene in 3D. To address this limitation, several improvements could be implemented: incorporating temporal information across consecutive frames to maintain detection consistency and using data augmentation techniques for MV-MOD methods.

7. ACKNOWLEDGEMENTS

This work was supported by the German Research Foundation (DFG) as a part of the Research Training Group i.c.sens [GRK2159].

References

Ali, R., Mehltretter, M., Heipke, C., 2023. Integrating Motion Priors For End-To-End Attention-Based Multi-Object Tracking. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 48, 1619–1626.

Aung, S., Park, H., Jung, H., Cho, J., 2024. Enhancing multi-view pedestrian detection through generalized 3d feature pulling. *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 1185–1194.

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

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., 2020. End-to-end object detection with transformers. *European Conference on Computer Vision*, Springer, 213–229.

Chavdarova, T., Baqué, P., Bouquet, S., Maksai, A., Jose, C., Bagautdinov, T., Lettry, L., Fua, P., Van Gool, L., Fleuret, F., 2018. Wildtrack: A multi-camera hd dataset for dense unscripted pedestrian detection. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5030–5039.

Chen, L., Ai, H., Zhuang, Z., Shang, C., 2018. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. *ICME*, IEEE, 1–6.

Cheng, C.-C., Qiu, M.-X., Chiang, C.-K., Lai, S.-H., 2023. Rest: A reconfigurable spatial-temporal graph model for multi-camera multi-object tracking. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 10051–10060.

Engilberge, M., Liu, W., Fua, P., 2023. Multi-view tracking using weakly supervised human motion prediction. *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 1582–1592.

Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J., 2021. YOLOX: Exceeding YOLO Series in 2021. *arXiv preprint arXiv:2107.08430*.

Harley, A. W., Fang, Z., Li, J., Ambrus, R., Fragkiadaki, K., 2023. Simple-bev: What really matters for multi-sensor bev perception? *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2759–2765.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 770–778.

Henschel, R. D., 2021. Higher-order multiple object tracking. PhD thesis, Leibniz University Hannover.

Hou, Y., Zheng, L., 2021. Multiview Detection with Shadow Transformer (and View-Coherent Data Augmentation). *ACM International Conference on Multimedia*, 1673–1682.

Hou, Y., Zheng, L., Gould, S., 2020. Multiview detection with feature perspective transformation. *European Conference on Computer Vision*, 16, Springer, 1–18.

Hu, W., Hu, M., Zhou, X., Tan, T., Lou, J., Maybank, S., 2006. Principal axis-based correspondence between multiple cameras for people tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4), 663–671.

Kalman, R. E., 1960. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1), 35–45.

Kendall, A., Gal, Y., Cipolla, R., 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7482–7491.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D., 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33, 18661–18673.

Kuhn, H. W., 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2), 83–97.

Nassar, A. S., Lefèvre, S., Wegner, J. D., 2020. Multi-view instance matching with learned geometric soft-constraints. *ISPRS International Journal of Geo-Information*, 9(11), 687.

Nguyen, D. M., Henschel, R., Rosenhahn, B., Sonntag, D., Swoboda, P., 2022. Lmgp: Lifted multicut meets geometry projections for multi-camera multi-object tracking. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8866–8875.

Nguyen, U., Heipke, C., 2020. 3D Pedestrian tracking using local structure constraints. *ISPRS Journal of Photogrammetry and Remote Sensing*, 166, 347-358.

Ong, J., Vo, B.-T., Vo, B.-N., Kim, D. Y., Nordholm, S., 2020. A Bayesian filter for multi-view 3D multi-object tracking with occlusion handling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5), 2246–2263.

Philion, J., Fidler, S., 2020. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D. *European Conference on Computer Vision*, 16, Springer, 194–210.

Ren, S., He, K., Girshick, R., Sun, J., 2016. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149.

Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C., 2016. Performance measures and a data set for multi-target, multi-camera tracking. *European Conference on Computer Vision*, Springer, 17–35.

Song, L., Wu, J., Yang, M., Zhang, Q., Li, Y., Yuan, J., 2021. Stacked homography transformations for multi-view pedestrian detection. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6049–6057.

Teepe, T., Wolters, P., Gilg, J., Herzog, F., Rigoll, G., 2024a. EarlyBird: Early-Fusion for Multi-View Tracking in the Bird’s Eye View. *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, 102-111.

Teepe, T., Wolters, P., Gilg, J., Herzog, F., Rigoll, G., 2024b. Lifting Multi-View Detection and Tracking to the Bird’s Eye View. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 667–676.

Vora, J., Dutta, S., Jain, K., Karthik, S., Gandhi, V., 2023. Bringing generalization to deep multi-view pedestrian detection. *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, 110–119.

Wojke, N., Bewley, A., Paulus, D., 2017. Simple online and realtime tracking with a deep association metric. *2017 IEEE international conference on image processing (ICIP)*, IEEE, 3645–3649.

Xu, Y., Liu, X., Liu, Y., Zhu, S.-C., 2016. Multi-view people tracking via hierarchical trajectory composition. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4256–4265.

You, Q., Jiang, H., 2020. Real-time 3d deep multi-camera tracking. *arXiv preprint arXiv:2003.11753*.

Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., Wang, X., 2022. Bytetrack: Multi-object tracking by associating every detection box. *European Conference on Computer Vision*, Springer, 1–21.

Zhang, Y., Wang, C., Wang, X., Zeng, W., Liu, W., 2021. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129, 3069–3087.

Zhou, X., Koltun, V., Krähenbühl, P., 2020. Tracking objects as points. *European Conference on Computer Vision*, Springer, 474–490.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J., 2020. Deformable DETR: Deformable Transformers for End-to-End Object Detection. *CoRR*, abs/2010.04159. https://arxiv.org/abs/2010.04159.