ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XI-2-2026
XXV ISPRS Congress 2026 “From Imagery to Understanding”, Commission II, 4–11 July 2026, Toronto, Canada

Beyond Centers: Bounding-Box Voxel Projection for Multi-View 3D Detection and 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

3D multi-view, multi-object tracking (3D MV-MOT) makes use of multiple cameras to reduce the number of missed detections and to mitigate occlusions. Most current 3D MV-MOT methods suffer from information loss when associating 3D locations with 2D image features via a 3D-to-2D projection, as they use a discrete grid in 3D and sample image features only at the projected centers of each grid cell. Thus, all other feature information is lost. An additional information loss commonly arises during cross-view aggregation when applying max or average pooling: these methods either overemphasize a single view or treat conflicting views, that depict different entities, e.g., due to occlusions, equally. In this work, we introduce two novel modules for 3D MV-MOT, employed to pedestrian tracking, that target these limitations: (i) *VoxROI* aggregates all image features that fall within the bounding box around a voxel’s projection into each respective image, instead of only sampling features at the projected voxel center. (ii) *SimFuse* aggregates per-view voxel features into one coherent feature representation per voxel, using similarity weights computed from re-identification (Re-ID) features. Subsequently, they are used to measure cross-view identity similarity. Views with higher Re-ID feature similarity receive larger weights, while inconsistent views are suppressed. Experimental results on the WildTrack dataset confirm our method’s effectiveness for multi-view pedestrian detection and tracking, reaching, and in particular in cross-view scenarios improving, the general state-of-the-art. The approach maintains strong performance across different camera configurations, demonstrating its generalization capability when training and testing on different camera setups.

1. INTRODUCTION

3D multi-view, multi-object tracking (3D MV-MOT) is key for a number of applications such as pedestrian safety in autonomous driving and sports analysis. Unlike monocular tracking, which often fails in the presence of occlusions, multi-view setups see the same object from different viewpoints, which helps to reduce the number of such failure cases. In 3D MV-MOT images from multiple cameras are used to detect and track objects in 3D. The observed scene is commonly represented as a bird’s-eye view (BEV) occupancy map, i.e., a fixed 2D grid on the ground, assumed to be planar and covering a defined area, where each cell is marked to indicate whether or not it is occupied. Given multi-view image sequences and the interior and exterior orientation parameters of each camera at every time step, 3D MV-MOT aims to output one trajectory with a unique identifier (ID) for each tracked object based on a BEV occupancy map as a function of time. Most recent methods addressing this task follow a common four-step pipeline (Engilberge et al., 2023a; Teepe et al., 2024a,b; Ali et al., 2025): First, a CNN backbone extracts image features. Second, a 3D voxel grid or a ground plane represented as a BEV grid, is projected into each image using the respective orientation parameters, and the image features corresponding to each voxel are sampled at the projected locations. Third, these multi-view features are aggregated into a coherent BEV representation. Finally, prediction heads are used to detect objects and to track them over time.

Most of these methods suffer from a limitation during the second step, as they rely on a center sampling scheme. In this scheme, only the centers of voxels or BEV grid cells are projected onto the image plane to sample image features, which are then assigned to their corresponding grid element. However, while the projection of a voxel/BEV grid cell typically covers many pixels in an image (up to a few thousand, depending on its distance to the image plane), using only the feature information associated with the centers creates a loss of information, as large parts of the image information is discarded (see Figure 1). The commonly applied concept of using a pooling layer for feature aggregation, the third processing step, introduces two additional limitations: On the one hand, simple pooling strategies produce unwanted results: max-pooling emphasizes the most dominant features in a voxel while ignoring potentially valuable features from other views, and average-pooling treats all features equally, which is suboptimal when they represent

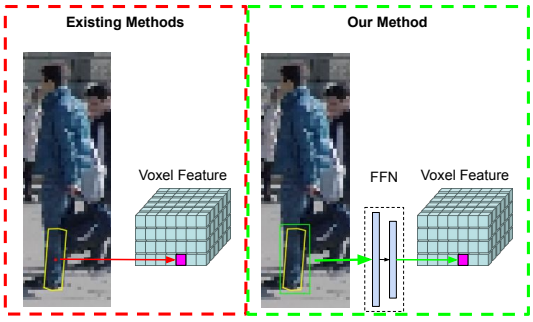

Figure 1. Illustration of the difference between center sampling (left) and our method (right). The yellow region marks an object space voxel projected into the image. Center sampling only uses the features of the voxel center, represented by the red dot, while our method uses every feature inside the voxel’s axis-aligned bounding box (shown in green) and aggregates them using a feed-forward network (FFN) to represent that voxel. Thus, our method uses all extracted image features.

* Corresponding author

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-2-2026-197-2026 | © Author(s) 2026. CC BY 4.0 License.

different objects, e.g., due to occlusions. Employing a pooling layer allows to handle a variable number of input images, whereas in our scenario a learned aggregation scheme, e.g., using a convolutional layer, has only been employed to encode a constant number of images into the architecture. Being able to process a variable number of input images is a crucial prerequisite for real-world applications, where the number of cameras in a MV-MOT setup may vary over time.

To address these limitations, we propose a novel method based on the following two main contributions:

1. **Projection (VoxROI):** For each voxel, we project its 3D axis-aligned bounding box (AABB) into each image and pool the image features within the projected AABB, in a scheme similar to region of interest align pooling (RoI Align) where for each AABB image features are sampled on a grid using max pooling per grid cell. The pooled features are fused with a linear layer, and the result is assigned to the voxel. This method replaces center sampling and retains features that would otherwise be dropped.

   Aggregation (SimFuse): We aggregate features from different views based on the cosine similarity of their Re-ID features, using the similarity scores as weights in a weighted aggregation. Re-ID features of each view are extracted in image space and aggregated into the voxel grid via VoxROI to measure cross-view identity consistency. Features from views that have higher similarity, i.e., are more likely to depict the same object, receive higher weights, while dissimilar views are down-weighted. The result is a permutation-invariant fusion that supports a variable number of images and avoids the limitations of max and average pooling.

2. RELATED WORK

Multi-view multi-object tracking (MV-MOT) uses multiple cameras, with overlapping or non-overlapping fields of view, that are either stationary or mounted on moving platforms, and with known interior and exterior orientation parameters, to track objects. This task is commonly divided into three steps: (i) *2D Object detection*, (ii) *Spatial association* (match detections across views taken at the same time step), and (iii) *Temporal association* (link detections across consecutive time steps). Existing methods mainly differ in the order of steps two and three and fall into two categories: The methods of the first category initially track objects independently per view (Nguyen and Heipke, 2020; Henschel, 2021; Zhang et al., 2022), and then establish spatial associations across views. (Hu et al., 2006; Xu et al., 2016). The methods of the second category solve the spatial association first and obtain 3D detections, which are then linked over time (Ong et al., 2020; You and Jiang, 2020; Nguyen et al., 2022; Cheng et al., 2023).

Recently, end-to-end multi-view multi-object detection (MV-MOD) methods have been proposed (Hou et al., 2020; Hou and Zheng, 2021; Song et al., 2021; Qiu et al., 2022; Engilberge et al., 2023b; Lee et al., 2023; Aung et al., 2024; Alturki et al., 2025). They detect objects in each time step and handle cross-view spatial association in an end-to-end manner, commonly following a four steps taxonomy: First, image features are extracted using a CNN backbone. In the projection step, cell centers of a predefined ground plane, represented as a BEV grid, are then projected into each image, and the image features at the projected locations are sampled (Hou et al., 2020; Hou and Zheng, 2021; Vora et al., 2023). Alternatively, a 3D voxel grid is used instead of a planar 2D grid (Song et al., 2021; Qiu et al., 2022; Harley et al., 2023; Aung et al., 2024; Alturki et al., 2025). Third, in the cross-view aggregation step, the projected features are merged to form BEV features. Finally, a detection head uses these BEV features to predict pedestrian positions (as a BEV occupancy map). Some works add a 2D detection head for each input image to strengthen activations at pedestrian locations and thus to obtain better image feature maps (Hou et al., 2020; Hou and Zheng, 2021). Others use a non-parameterized BEV fusion layer (e.g., average pooling) to enable processing of a variable number of images during training and inference (Vora et al., 2023). Aung et al. (2024) weigh image features before projecting them, using a squeeze-and-excitation network (Hu et al., 2018) to focus on regions of interest. Alturki et al. (2025) apply Mask R-CNN (He et al., 2017) to each image to find object masks, then estimate the visual hull (Laurentini, 2002) of all objects and finally build a probabilistic occupancy volume consisting of an occupancy probability for every voxel. Several MV-MOT methods build on top of such MV-MOD methods (Engilberge et al., 2023a; Teepe et al., 2024a,b; Ali et al., 2025), using MV-MOD to detect and associate objects across views; the focus of these MV-MOT methods is then put on the tracking stage, i.e., linking object detections over time.

3. BEYOND CENTERS

Given a set of N synchronized cameras with known interior and exterior orientation parameters, capturing overlapping color images $\mathbf{I}_{i,t} \in \mathbb{R}^{H_{in} \times W_{in} \times 3}$ with $i = 1, 2, ..., N$ and H_{in}, W_{in} denoting the height and width of the input image, respectively, at T time steps $t = 1, 2, ..., T$, our method predicts a BEV occupancy map of pedestrians in the observed scene at each time step, along with a unique ID for each pedestrian . This allows us to associate detections across time and track pedestrians in the BEV space.

Our method extends the one presented in (Ali et al., 2025), with the use of our novel VoxROI component, instead of using center sampling in the projection step. Additionally, we introduce a novel aggregation scheme, SimFuse, which uses similarity-weighted aggregation when fusing feature information from different images. Our method can be broken down into six main stages; a comprehensive overview is given in Figure 2:

1. For each image $\mathbf{I}_{i,t}$, a feature map $\hat{\mathbf{F}}_{i,t} \in \mathbb{R}^{H \times W \times C_{img}}$ is extracted using the same encoder as employed by Teepe et al. (2024a), which is based on ResNet18 (He et al., 2016), where H, W denote the height and width of the image feature map, while C_{img} denotes the image feature dimension. ResNet18 is chosen for its small size, which keeps the computational effort and the memory footprint small. This is particularly important as our method needs to process N input images in each time step. Following Ali et al. (2025), namely the simplified directional model, image features are aggregated with viewing-direction information, yielding $\mathbf{F}_{i,t} \in \mathbb{R}^{H \times W \times C_{img}}$, before being used by VoxROI.

2. We define a 3D voxel grid $\mathbf{V} \in \mathbb{N}^{n_X \times n_Y \times n_Z}$ within the area of interest in 3D object space, where n_X and n_Y denote the discretization along the two horizontal axes of the ground plane, and n_Z denotes the discretization

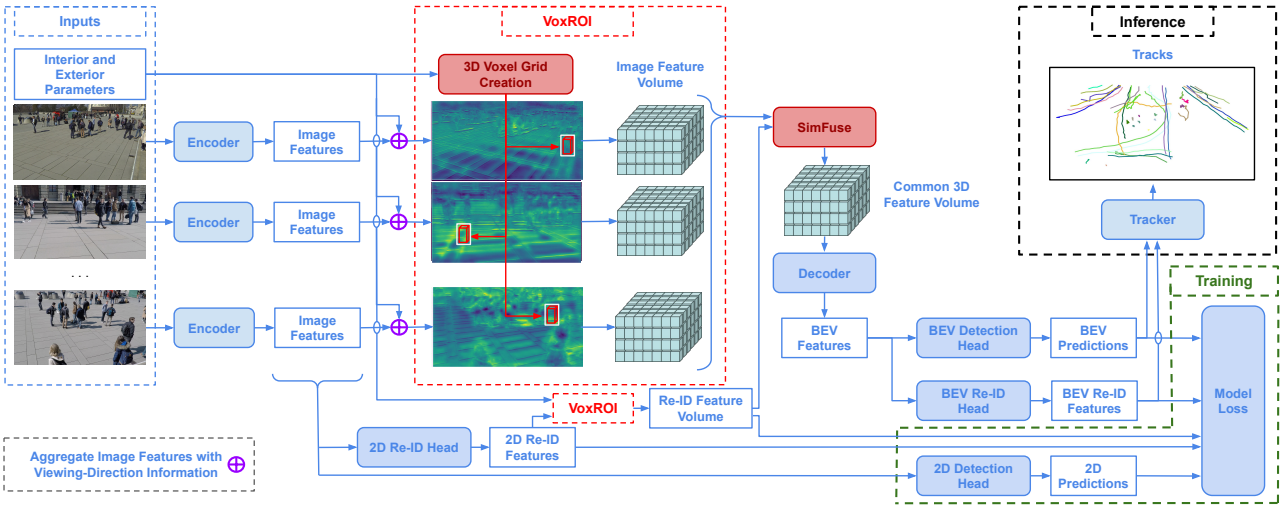

Figure 2. Overview of our method. For each image $\mathbf{I}_{i,t}$ captured by camera i at time t, a ResNet-18 encoder extracts a feature map which is aggregated with directional information similar to Ali et al. (2025). VoxROI, which is enclosed by the dashed red box, produces a per-voxel feature vector from each image by using the extracted image feature maps and the respective orientation parameters, yielding one 3D feature volume per image. Using SimFuse the per-image volumes are then aggregated into a common 3D feature volume, and a ResNet-18–based decoder extracts BEV features from this volume. Image features are used to compute 2D Re-ID features while BEV features are used to compute BEV Re-ID features. The BEV detection head uses only BEV features to predict pedestrian positions in BEV and the 2D detection head uses image features to predict pedestrians in image space, achieve higher activations at pedestrian locations and thus more meaningful feature maps are extracted. Finally, at the inference step a tracker is used to form tracklets by associating current time step BEV detections with those from the previous time step using predicted positions and BEV Re-ID features from both time steps. Components used only during training are enclosed by the dashed green box; components used only during inference are enclosed by the dashed black box. Our contributions are shown in red.

along the vertical axis. Our novel VoxROI component uses the extracted feature maps $\mathbf{F}_{i,t}$ and the interior and exterior orientation parameters of each input image to generate an image feature vector $\mathbf{f}_{v,i,t} \in \mathbb{R}^{C_{vox}}$ for each voxel $v \in \mathbf{V}$, where C_{vox} denotes the voxel feature dimension, thereby constructing a 3D image feature volume $\mathbf{G}_{i,t} \in \mathbb{R}^{n_X \times n_Y \times n_Z \times C_{vox}}$ for each input image. This processing step is described in detail in Section 3.1.

3. At this stage, SimFuse combines the individual image feature volumes $\mathbf{G}_{i,t}$ into a common feature volume $\mathbf{G}_t$ using weighted aggregation. The weighting reduces the consequences of mixing features from different objects by giving more weight to image features with similar Re-ID features (see section 3.2) and less weight to conflicting ones, e.g., due to occlusion. The weights are computed from the so-called Re-ID feature volumes $\mathbf{H}_{i,t} \in \mathbb{R}^{n_X \times n_Y \times n_Z \times C_{reid}}$, where C_{reid} denotes Re-ID feature dimension. These volumes are computed in the same way as the image feature volumes, but they are built from 2D Re-ID features instead of image features. The 2D Re-ID features are produced by the 2D Re-ID head, which takes $\hat{\mathbf{F}}_{i,t}$ as input. SimFuse is described in Section 3.2.

4. BEV features $\mathbf{B}_t \in \mathbb{R}^{n_X \times n_Y \times C_{BEV}}$ are computed from $\mathbf{G}_t$ using a ResNet-18 based decoder, where C_{BEV} denotes the BEV feature dimension.

5. The final stage of our network architecture consists of four heads: two Re-ID heads, a 2D detection head and a BEV detection head. $\hat{\mathbf{F}}_{i,t}$ feeds the 2D Re-ID head to compute 2D Re-ID features while $\mathbf{B}_t$ feeds the BEV Re-ID head to compute BEV Re-ID features. The 2D detection head uses $\hat{\mathbf{F}}_{i,t}$ to predict the pedestrian in image space of each of the input images. The BEV detection head uses $\mathbf{B}_t$ to predict the pedestrian positions in BEV. This processing step is described in detail in Section 3.3.

6. Finaly, we build tracklets by associating the BEV detections of the current time step with those from the previous time step, using predicted positions and BEV Re-ID features from both time steps. This processing step is described in detail in Section 3.4.

3.1 VoxROI

To create the image feature volumes $\mathbf{G}_{i,t}$, we project the eight corners of each voxel v into every image:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{K}\,\mathbf{R}^T \left(\begin{pmatrix} x \\ y \\ z \end{pmatrix} - \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} \right), \tag{1}$$

where x, y, z describe the position of a voxel’s corner in object space, u, v describe the corresponding point in image space, x_0, y_0, z_0 are the coordinates of the projection center in object space, $\mathbf{K}$ is the calibration matrix and $\mathbf{R}$ denotes the rotation matrix that defines the viewing direction. Using the resulting projection of a voxel, we determine its axis-aligned bounding box (AABB) in the respective image space. A voxel’s image space projection, and thus the corresponding AABB, scales inversely with the distance between the voxel and the image plane: voxels closer to a camera yield larger AABBs containing more image features, while distant voxels yield smaller AABBs with fewer features.

To enable the aggregation of feature information across images, we compute a fixed-length feature vector $\mathbf{f}_{v,i,t}$ per voxel v. To compute this feature vector, we adapt the concept of RoI Align

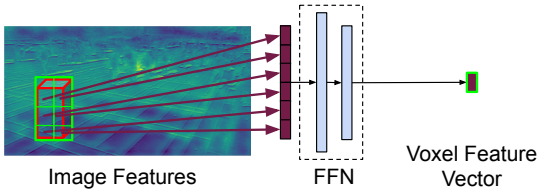

Figure 3. Visualization of our VoxROI approach for a single voxel and a single image. (1) Project the voxel's eight corners into the image plane (voxel shown as a red cuboid). (2) Form the AABB around the projected corners (shown in green). (3) Sample features inside the AABB on a grid (shown in green) by applying max pooling per cell. (4) Concatenate the pooled features and process them with a feed-forward network (FFN) to obtain the final voxel feature vector.

(He et al., 2017): While standard RoI Align extracts features from a 2D RoI to detect objects in image space, our key adaptation is to define the RoI as the 2D AABB of a projected 3D voxel. We subdivide each AABB using an $m_p \times n_p$ grid and apply max pooling to the image features $\mathbf{F}_{i,t}$ within each cell. Here, m_p and n_p denote the number of grid cells along the AABB height and width, respectively. The pooled features are concatenated to form an intermediate representation, which is then processed by a feed-forward network to obtain the final voxel feature vector $\mathbf{f}_{v,i,t}$. Figure 3 illustrates our VoxROI approach. Applying this process to all voxels produces a 3D image feature volume $\mathbf{G}_{i,t}$ for each image.

3.2 SimFuse

In the aggregation stage, for each voxel image feature volumes $\mathbf{G}_{i,t}$ are fused to obtain a common 3D feature volume $\mathbf{G}_t$. To reduce the consequences of mixing feature information from different pedestrians, our novel aggregation scheme SimFuse uses weighted aggregation. To do so, we use the corresponding feature vectors in the Re-ID feature volumes $\mathbf{H}_{i,t}$ within the same voxel to compute aggregation weights. The Re-ID feature volume generation is trained such that feature vectors of the same pedestrian are close in feature space and form clusters. Therefore, feature vectors that are consistent with each other receive higher weights, while inconsistent ones receive lower weights. For this purpose, we assume: (i) each voxel contains at most one pedestrian; and (ii) Re-ID features of the same pedestrian have a higher similarity than those of different pedestrians.

Since SimFuse computes aggregation weights from Re-ID feature volumes, we first describe their construction. For each view, 2D Re-ID features are extracted from $\hat{\mathbf{F}}_{i,t}$ using a dedicated 2D Re-ID head (see Section 3.3). In contrast to the image features, which capture information crucial to generally detect pedestrians, the 2D Re-ID features are used to distinguish between individual pedestrians, so that features of the same pedestrian are similar and those of different pedestrians are different. These features are then aggregated into a voxel grid via VoxROI applying the same process as described in Section 3.1, resulting in a Re-ID feature volume $\mathbf{H}_{i,t}$ per image.

Let $\mathbf{r}_{v,i,t} \in \mathbb{R}^{C_{reid}}$ be the per-voxel Re-ID feature vectors contained in $\mathbf{H}_{i,t}$. We first compute weights w based on $\mathbf{r}_{v,i,t}$ and then use these weights to fuse the image-wise feature vectors $\mathbf{f}_{v,i,t}$ per voxel to create a common feature vector $\mathbf{g}_{v,t}$ per voxel.

First, the pairwise cosine similarity $s_{v,ij,t}$,

$$s_{v,ij,t} = \max\left(0, \frac{\mathbf{r}_{v,i,t} \cdot \mathbf{r}_{v,j,t}}{|\mathbf{r}_{v,i,t}||\mathbf{r}_{v,j,t}|} - \gamma\right), \tag{2}$$

is calculated between all $i, j \in \{1, \ldots, N\}$ pairs. All values below a threshold γ are set to zero to exclude potential outliers from the subsequent feature vector fusion. Experimentally, we have found $\gamma = 0.5$ to work well and have chosen this value for all experiments. Next, we compute a cross-view-consistency score, which measures how well the features from one image agree with those from the other images. A high score indicates that features observed in one image are consistent with those of multiple other images:

$$c_{v,i,t} = \sum_{j \neq i} s_{v,ij,t}\,, \tag{3}$$

where we exclude the $j = i$ term since including self-similarity would inflate the score and would not reflect cross-view consistency. This is followed by normalizing the score across all N cameras:

$$w_{v,i,t} = \frac{c_{v,i,t}}{\sum_{k=1}^{N} c_{v,k,t} + \epsilon} \tag{4}$$

where ϵ is a small value, added to avoid dividing by 0. Lastly, a weighted aggregation of the voxel features is applied:

$$\mathbf{g}_{v,t} = \sum_{i=1}^{N} w_{v,i,t}\mathbf{f}_{v,i,t}\,, \tag{5}$$

yielding the common feature volume $\mathbf{G}_t$. Thus, image features with corresponding Re-ID features that are similar across images receive higher weights, while other image features are suppressed, reducing the consequences of mixing features associated with different pedestrians. Lastly, the feature vectors in $\mathbf{G}_t$ are concatenated along the Z-axis to create a 2D feature map of shape $n_X \times n_Y$ with feature dimension $C_{vox} \cdot n_Z$; this map is used as input to the ResNet-18–based decoder block to extract BEV features.

3.3 Prediction Heads and Loss Function

The architectures of the BEV detection, 2D detection and the re-identification heads and the loss function are defined according to (Teepe et al., 2024a) and (Ali et al., 2025).

Prediction Heads: Our method consists of four prediction heads during training: a BEV detection head, a 2D detection head, a BEV Re-ID head, and a 2D Re-ID head. The BEV detection head predicts a ground-plane occupancy map and a per-cell 2D offset to the grid cell center. The offset compensates for the quantization error introduced by the discrete grid representation, allowing the model to estimate pedestrian positions with sub-cell accuracy. Additionally, to help the network focus on pedestrian features, a 2D detection head predicts pedestrian bounding boxes in each input image. For association, the re-identification heads output Re-ID features for each pedestrian, computed for both, BEV and per-image feature maps.

Loss Function:

The loss formulation and its hyperparameters follow those of Teepe et al. (2024a) with the addition of the Re-ID feature volume loss $\mathcal{L}_{\text{3DReID}}$. Our model is trained with a multi-task

objective comprising BEV detection, 2D detection, and pedestrian re-identification in 2D and 3D. For the BEV detection head, the prediction of the ground-plane occupancy map is supervised by a focal loss, $\mathcal{L}_{\text{ctr}}^{\text{BEV}}$, and the offset regression by a masked ℓ_1 loss, $\mathcal{L}_{\text{off}}^{\text{BEV}}$. Following Kendall et al. (2018), these two terms are balanced using uncertainty-based weighting:

$$\begin{aligned}\mathcal{L}_{\text{BEV}} =& 5\exp(-s_{\text{ctr}}^{\text{BEV}})\mathcal{L}_{\text{ctr}}^{\text{BEV}} + 5\exp(-s_{\text{off}}^{\text{BEV}})\mathcal{L}_{\text{off}}^{\text{BEV}} \\ &+ \frac{1}{2}(s_{\text{ctr}}^{\text{BEV}} + s_{\text{off}}^{\text{BEV}}).\end{aligned} \tag{6}$$

where $s_{\text{ctr}}^{\text{BEV}}$ and $s_{\text{off}}^{\text{BEV}}$ are learnable uncertainty parameters.

Similar to the BEV detection head, the 2D detection head is supervised by a focal loss, $\mathcal{L}_{\text{ctr}}^{\text{2D}}$, for center prediction and ℓ_1 losses for offset, $\mathcal{L}_{\text{off}}^{\text{2D}}$, and size regression, $\mathcal{L}_{\text{size}}^{\text{2D}}$. These terms are normalized by the number of cameras S, where the hyperparameters are those of Teepe et al. (2024a):

$$\mathcal{L}_{\text{2D}} = \frac{1}{S}[\mathcal{L}_{\text{ctr}}^{\text{2D}} + \mathcal{L}_{\text{off}}^{\text{2D}} + \frac{1}{10}\mathcal{L}_{\text{size}}^{\text{2D}}]. \tag{7}$$

For pedestrian re-identification, the 2D Re-ID features are employed using two complementary terms: a cross-entropy loss, $\mathcal{L}_{\text{CE}}^{\text{2DReID}}$, for identity classification and a supervised contrastive loss, $\mathcal{L}_{\text{SupCon}}^{\text{2DReID}}$ (Khosla et al., 2020). The resulting objective is

$$\mathcal{L}_{\text{2DReID}} = \mathcal{L}_{\text{CE}}^{\text{2DReID}} + \mathcal{L}_{\text{SupCon}}^{\text{2DReID}}. \tag{8}$$

Similarly, we impose direct supervision on the 3D Re-ID feature volume, again using two complementary losses: a cross-entropy loss, $\mathcal{L}_{\text{CE}}^{\text{3DReID}}$, for identity classification, and a supervised contrastive loss, $\mathcal{L}_{\text{SupCon}}^{\text{3DReID}}$, for representation learning. This encourages the learned volumetric features to encode identity-specific re-identification information.

$$\mathcal{L}_{\text{3DReID}} = \mathcal{L}_{\text{CE}}^{\text{3DReID}} + \mathcal{L}_{\text{SupCon}}^{\text{3DReID}}. \tag{9}$$

The overall training loss is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{BEV}} + \mathcal{L}_{\text{2D}} + \mathcal{L}_{\text{2DReID}} + \mathcal{L}_{\text{3DReID}}. \tag{10}$$

3.4 Tracking

Object tracking is realized in a two-stage procedure, adopting the methodology of (Chen et al., 2018) and the parameterization of (Teepe et al., 2024a). The first stage performs initial data association, while the second stage recovers tracks lost, e.g., due to temporary occlusions or detection failures. The tracklets are initialized with the BEV detection head predictions. In successive time steps, the temporal association of detections is performed using motion and appearance information. Specifically, a constant-velocity Kalman Filter (Kalman, 1960) provides a motion prediction, while the re-identification head supplies appearance Re-ID features. The association metric, inspired by DeepSORT (Wojke et al., 2017), combines the cosine similarity D_c, computed from the Re-ID features, and the Mahalanobis distance D_m between the motion prediction and the BEV detected centers, into a distance $D = \eta D_c + (1-\eta)D_m$, where η is a hyperparameter. A threshold of $0.4m$ is applied to D_m to gate infeasible associations. Pairs exceeding this threshold are excluded from the assignment by setting their Mahalanobis distance to infinity. The optimal assignment is obtained using the Hungarian method (Kuhn, 1955). The second stage addresses previous occlusions and detection failures by attempting to re-associate unmatched detections, e.g., where the Mahalanobis distance was larger than $0.4m$ to all traklets, with an increased Mahalanobis distance threshold, for which we used the value of $2.5m$. Detections that persist without a match spawn new tracklets, while tracklets that remain unassociated for 10 frames are deleted, assuming that the corresponding pedestrian has left the scene.

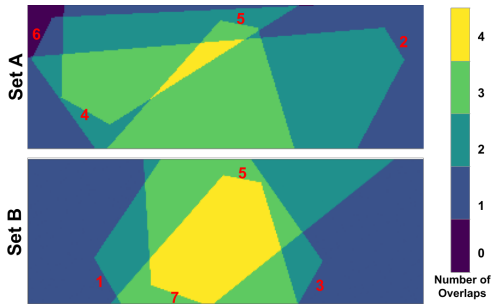

Figure 4. WildTrack camera splits with viewing frustums and BEV overlap. Red numerals mark cameras positions,

4. EXPERIMENTS

4.1 Experimental Setup

To assess the impact of the proposed novel methodological components, we evaluate three configurations of our method:

1. Ours: Our method as described in Section 3.
2. Ours (w/o VoxROI): Our method, but VoxROI is replaced by the center sampling approach from the literature.
3. Ours (w/o SimFuse): Our method, but SimFuse is replaced by average pooling for cross-view aggregation.

We compare against state-of-the-art multi-view multi-object detection and tracking methods: The set includes MV-MOD methods that use a learnable CNN for feature aggregation across views, MVDetr (Hou and Zheng, 2021), SHOT (Song et al., 2021), 3DROM (Qiu et al., 2022) and methods that use a pooling layer for aggregation, GMVD (Vora et al., 2023), MVFP (Aung et al., 2024) and (Alturki et al., 2025). We also include MV-MOT methods for comparison: ReST (Cheng et al., 2023) which uses a graph neural network for aggregation, EarlyBird (Teepe et al., 2024a) and TrackTacular (Teepe et al., 2024b), which employ CNN-based aggregation, and (Ali et al., 2025). We consider the latter as our baseline as it is the method that we build upon. All the mentioned methods use a center sampling approach in the projection step.

4.2 Datasets

We use the WildTrack dataset (Chavdarova et al., 2018) in all our experiments. It provides a real-world multi-view setting with pedestrian annotations in both, image and object space. WildTrack contains image sequences from seven calibrated cameras at a resolution of 1920×1080 pixels, each with 400 frames annotated at 2 fps. The scene spans $36\,m \times 12\,m$. Following Chavdarova et al. (2018) and Vora et al. (2023), we use the first 90% of frames of each sequence for training and the last 10% for testing.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-2-2026-197-2026 | © Author(s) 2026. CC BY 4.0 License.

Method	MODA%	MODP%	Prec.%	Recall%	IDF1%	MOTA%
MVDeTr (Hou and Zheng, 2021)	91.5	82.1	**97.4**	94.0	-	-
SHOT (Song et al., 2021)	90.2	76.5	96.1	94.0	-	-
3DROM (Qiu et al., 2022)	93.5	75.9	97.2	96.2	-	-
GMVD (Vora et al., 2023) *	86.7	76.2	95.1	91.4	-	-
MVFP (Aung et al., 2024) *	**94.1**	78.8	96.4	**97.7**	-	-
(Alturki et al., 2025) *	93.6	82.4	96.6	97.0	-	-
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**
(Ali et al., 2025) *	89.3	81.7	93.7	95.7	92.0	87.4
Ours (w/o VoxROI) *	89.6±0.6	81.8±0.3	94.8±0.9	94.8±0.5	88.1±0.7	86.4±0.2
Ours (w/o SimFuse) *	91.4±1.5	**83.0**± 0.6	95.3 ±0.6	96.1±1.1	94.7±0.4	90.4±1.4
Ours *	91.0±1.0	82.6±0.2	94.3±0.7	96.9±0.5	92.5±1.2	89.6±0.7

Table 1. Comparison with state-of-the-art detection and tracking methods trained and evaluated on all seven views. * indicates methods that can handle a variable number of input images. Our results are reported as mean and standard deviation over three runs.

As noted by Vora et al. (2023), using all seven views for training can be misleading, because the training and test sets share camera views, which encourages overfitting. To test the generalization capability, we follow their setup and vary the camera configuration between training and testing. Figure 4 shows the two WildTrack splits we use. The set (2,4,5,6), which we call set A, covers a larger part of the scene where there is an overlap between two or three cameras, while the set (1,3,4,7), which we call set B, has a larger region seen by four cameras. We refer to training and testing on different camera subsets as *Cross-View Evaluation*. In contrast, we call the conventional approach of using all views for both training and testing *Full-View Evaluation*.

The WildTrack dataset is annotated in a different manner than typical 2D datasets. Typically, annotations are performed directly in 2D image space, and by definition, occluded pedestrians are not annotated. In contrast, WildTrack annotations are the BEV position of the pedestrian, produced by human annotators, while the bounding boxes are automatically projected from this BEV annotation into image space. Thus, the WildTrack dataset does not distinguish between occluded and non-occluded pedestrians in image space. This is why we define pedestrians as occluded if their intersection over union (IoU) with a pedestrian closer to the camera exceeds a threshold of 0.7. This choice is consistent with standard non-maximum suppression (NMS) thresholds used in modern object detectors, where boxes with IoU $>$0.7 are treated as redundant and suppressed (Ren et al., 2016). Hence, instances beyond this overlap level cannot be expected to be detected reliably as separate objects. This is important for our method, as the 2D detection head operates in image space and should not be penalized for pedestrians that are not visible from a given camera view.

4.3 Metrics

Detection Metrics: The detection performance of our method is assessed following the established protocol of (Chavdarova et al., 2018). Comparisons with the ground truth are based on the Euclidean distance in BEV space. A prediction is classified as a true positive if it falls within a circle of radius $r = 0.5\,m$ centered on a ground truth pedestrian. Our key evaluation metric is Multiple Object Detection Accuracy (MODA), which combines false positives (FP) and false negatives (FN), normalized by the total number of ground truth instances (GT):

$$MODA = 1 - \frac{FP + FN}{GT} \tag{11}$$

Additionally, to gauge the quality of the localization, we employ Multiple Object Detection Precision (MODP). This metric averages the spatial precision of all true positives (TP):

$$MODP = \frac{\sum 1 - p[p < s]/s}{TP} \tag{12}$$

where p is the distance from a detection to its ground truth, s is the threshold for true positives (set to $s = 0.5\,m$), and TP is the number of true positives. The condition $[p < s]$ ensures the summation is done only over detections classified as true positives. Finally, we also report the conventional Precision and Recall scores to offer a more complete picture of the detection performance.

Tracking Metrics: The tracking performance is evaluated using two standard metrics: Multiple Object Tracking Accuracy (MOTA) (Bernardin and Stiefelhagen, 2008) and the Identity-F1 score (IDF1) (Ristani et al., 2016). MOTA aggregates false negatives (FN), false positives (FP), and identity switches (IDSW), normalized by the total number of ground truth objects (GT):

$$\text{MOTA} = 1 - \frac{FN + FP + IDSW}{GT} \tag{13}$$

While MOTA captures overall tracking robustness, IDF1 specifically measures the accuracy of identity preservation over time. It is defined as the harmonic mean of Identity Precision (IDP) and Identity Recall (IDR):

$$\text{IDF1} = \frac{2 \cdot IDP \cdot IDR}{IDP + IDR} \tag{14}$$

where IDP is the proportion of true positive identifications among all predicted identifications, and IDR is the proportion of true positive identifications among all ground truth identifications. A detection is considered correctly identified if it is matched to the same ground truth trajectory over time.

4.4 Implementation Details

Following (Hou et al., 2020), we resize images to 1280×720 pixels, before using them as input to our method, to reduce

	Method	Inference on Set A				Inference on Set B			
		MODA%	MODP%	Prec%	Recall%	MODA%	MODP%	Prec%	Recall%
Trained on Set A	MVDeTr	75.4	79.5	**96.9**	77.9	41.7	73.7	92.0	45.7
	SHOT	81.9	74.1	94.1	87.4	51.4	72.5	94.4	54.6
	GMVD *	84.0	72.9	92.4	91.6	75.1	71.1	94.3	79.9
	EarlyBird	91.0	81.5	96.8	94.1	78.1	79.9	94.9	82.5
	(Ali et al., 2025) *	86.9	79.8	96.1	90.6	78.8	79.9	**96.4**	81.8
	Ours (w/o VoxROI) *	89.0±1.7	80.9±0.2	95.7±0.4	93.2±2.0	80.8±1.4	**80.7**±0.7	95.5±1.5	84.9±2.3
	Ours (w/o SimFuse) *	90.2±1.4	**81.8**±0.1	96.2±0.8	93.8±1.0	**82.3**±0.9	80.5±0.5	94.6±1.3	87.3±0.9
	Ours *	**91.6**±0.6	81.4±0.4	96.3±0.7	**95.3**±0.9	82.1±1.2	80.5±0.3	94.0±0.4	**87.8**±1.5
Trained on Set B	MVDeTr	5.6	65.5	62.4	14.0	72.5	78.9	95.0	76.5
	SHOT	15.3	62.9	89.2	17.4	79.7	76.4	95.7	83.5
	GMVD *	62.6	67.4	86.7	73.9	80.8	74.0	94.2	86.0
	EarlyBird	**85.9**	78.5	**96.9**	88.4	85.9	78.5	**96.1**	89.5
	(Ali et al., 2025) *	77.5	77.7	92.0	84.8	85.3	79.7	95.5	89.6
	Ours (w/o VoxROI) *	70.3±0.9	79.6±0.2	93.8±0.9	75.3±1.0	85.9±1.4	81.6±0.6	96.4±1.0	89.1±0.5
	Ours (w/o SimFuse) *	82.5±0.6	80.0±0.3	93.9±1.2	88.3±1.4	86.8±1.2	81.7±0.3	95.7±0.9	91.0±0.5
	Ours *	85.4±1.6	**80.6**±0.5	96.4±0.7	**88.8**±2.4	**87.5**±0.8	**82.2**±0.4	96.0±0.6	**91.3**±0.6

Table 2. Comparison of methods trained on different camera sets, including results for MVDeTR, SHOT, and GMVD as reported by Vora et al. (2023), and Ali et al. (2025). All results are in %. * indicates methods that can handle a variable number of input images. Our results are reported as mean and standard deviation over three runs.

memory usage. Additionally, the scene is discretized into a $360 \times 120 \times 4$ voxel grid with cell sizes of 2.5 cm in X/Y and 50 cm in Z direction; the Z extent is set to 2 m to cover maximum adult pedestrian height. For data augmentation, as in (Hou and Zheng, 2021; Harley et al., 2023), we apply random resize-and-crop to the RGB inputs and adjust the interior orientation parameters K accordingly. We train using the Adam optimizer (Kingma, 2014) and a one-cycle learning-rate schedule with a maximum learning rate of 10^{-3}, where the learning rate is increased and then decreased during training, following the OneCycleLR policy, see Teepe et al. (2024a). We train for 50 epochs and a batch size of 1 with gradient accumulation, thus the effective batch size becomes 8. Encoder and decoder weights are initialized from the ImageNet-1K pretrained PyTorch model, while all other network weights are randomly chosen using the default PyTorch initialization.

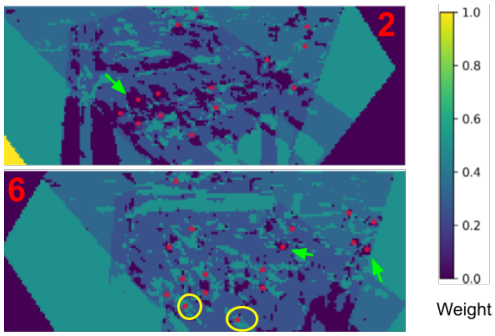

Figure 5. BEV visualization of aggregation weights for two views at two different time stamps. Red numerals mark camera positions, red dots show pedestrian ground truth positions. Some island-like weighting patterns are denoted by green arrows, the yellow ellipses highlight some pedestrians without such patterns. The color bar shows the weight magnitude.

5. RESULTS

We train each model three times and report the mean and standard deviation across runs for each metric.

5.1 Full-View Evaluation

We first compare our method with the state-of-the-art multi-view multi-object detection and tracking methods on the WildTrack dataset, using all seven views for both training and testing. The results are shown in Table 1. Our method demonstrates highly competitive performance, achieving results on par with state-of-the-art methods. As shown in Table 1, our approach achieves a MODA score of 91.0%, and 96.9% Recall closely matching the best performing method MVFPl with a Recall of 97.7%. This strong performance is consistent across all evaluation metrics. Moreover, our proposed method yields a clear performance gain over the baseline (Ali et al., 2025), upon which it is built: We observe an approximate increase of 1.7% in MODA and 1.2% in IDF1. Comparing the results of the different variants of our method, it is evident that our VoxROI module (cf. variant Ours (w/o SimFuse)) is the most critical contribution to our method’s performance. Removing VoxROI causes a large performance drop, with IDF1 decreasing by 4% to a level below the original baseline. This suggests that VoxROI is the dominant component in terms of performance impact. The effect of SimFuse is more moderate and configuration-dependent.

Figure 5 visualizes the per-voxel weights produced by SimFuse for two views. The weights form localized island-like structures centered at pedestrian positions. For voxels corresponding to visible pedestrians, higher weights are assigned, while voxels at pedestrian locations that are occluded in a given view receive lower weights, some of which are denoted with green arrows, effectively suppressing their contribution. This object-centric weighting behavior indicates that similarity computed from Re-ID feature volumes promotes identity-consistent aggregation

	Method	Inf. on set A		Inf. on set B	
		IDF1%	MOTA%	IDF1%	MOTA%
Tr. set A	EarlyBird	87.9	82.7	78.8	**81.8**
	(Ali et al., 2025)	84.5	85.2	78.6	74.5
	Ours (w/o VoxROI)	87.8±3.2	87.1±1.7	83.2±2.0	77.9±1.9
	Ours (w/o SimFuse)	87.5±3.6	87.7±1.8	**84.4**±2.2	78.4±1.0
	Ours	**89.5**±1.0	**89.0**±1.0	83.5±2.5	78.3±1.6
Tr. set B	EarlyBird	**83.6**	75.3	86.8	**88.0**
	(Ali et al., 2025)	74.5	75.0	87.8	83.6
	Ours (w/o VoxROI)	66.4±2.1	64.7±1.6	86.3±3.3	83.0±1.1
	Ours (w/o SimFuse)	78.3±5.3	78.7±2.7	86.0±0.6	83.7±1.1
	Ours	76.0±4.6	**80.5**±3.5	**88.3**±0.5	84.7±1.4

Table 3. Comparison of tracking 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 %. Our results are reported as mean and standard deviation over three runs.

across views. However, some pedestrians that clearly stand behind others do not consistently receive reduced weights, denoted with yellow ellipses in the figure, suggesting that occlusion relationships are still not fully resolved. This indicates that, despite improved identity modeling, our similarity-based weighting alone remains limited in handling complex inter-object occlusions.

5.2 Cross-View Evaluation

We train each method separately on both camera sets and then evaluate them on both sets (cross-split and same-split). We report results alongside the state of the art from (Hou and Zheng, 2021; Song et al., 2021; Vora et al., 2023; Teepe et al., 2024a; Ali et al., 2025) in Table 2 for the detection performance.

Our method achieves state-of-the-art performance or remains within 1% of the top score on both sets A and B across all configurations. Compared to methods that can handle a variable number of input views (Vora et al., 2023; Ali et al., 2025), our approach consistently outperforms them in both, same-split and cross-split evaluation. Comparing to the baseline method (Ali et al., 2025) we can see a consistent improvement in both MODA and Recall, with an improvement of up to 8% in MODA when training on set B and testing on Set A. Similarly to the scenario using all views for training, the results demonstrate that the VoxROI module is the most critical component. The variant using only VoxROI achieves 3-4% higher MODA and Recall compared to the baseline method. In contrast, using SimFuse alone yields results comparable to, or below, the baseline. A possible explanation is that the weighting mechanism relies on re-identification features. When combined with VoxROI, these features are aggregated over the full voxel extent and are therefore more representative. However, when center sampling is used instead, the Re-ID features are restricted to a single projected point, limiting their discriminative capacity and reducing the effectiveness of the similarity-based weighting.

Additionally, we report tracking results in Table 3. The tracking performance generally follows the detection trends, as expected for a detection-based tracker. The variance across runs is slightly higher than for the detection results, since tracking performance is influenced not only by detection accuracy but also by temporal association errors. When trained on Set A, our full model achieves the best performance on Set A in both IDF1 (89.5%) and MOTA (89.0%), clearly outperforming the baseline (Ali et al., 2025). When trained on Set B, our method attains the highest IDF1 on Set B (88.3%) and competitive MOTA (84.7%). While Teepe et al. (2024a) achieve higher MOTA (81.8%) and (88.0%) when evaluated on Set B, this difference could be explained by the learned feature aggregation used in Teepe et al. (2024a). In Set B, the camera views have higher overlap, meaning that more cameras observe the same pedestrian simultaneously. In such scenarios, using weighted averaging for the feature vectors from different viewpoints, as done in our approach, can dilute discriminative appearance cues that are important for reliable re-identification. In contrast, learned aggregation can better adapt to viewpoint differences and preserve identity-relevant information.

This behavior is also reflected in the limitations of the SimFuse module. Despite the use of supervised Re-ID feature volumes, SimFuse currently remains subject to several limitations. First, similarity is computed independently per frame and does not incorporate temporal context, which could otherwise stabilize weighting decisions across time. Second, weights are computed independently for each voxel, without spatial regularization, which may lead to locally inconsistent aggregation patterns. Finally, the weighted averaging of features implicitly assumes that different views provide complementary representations of the same appearance. However, for substantially different viewpoints (e.g., frontal versus rear views), simple averaging may dilute discriminative information rather than preserve it, suggesting that more expressive fusion mechanisms could further improve identity consistency.

6. CONCLUSION AND FUTURE WORKS

We introduced Beyond Centers, a novel method for 3D multi-view multi-object tracking that preserves spatial information during feature lifting into 3D voxel space and performs identity-aware cross-view aggregation for robust multi-view fusion. Our experiments demonstrate that incorporating VoxROI yields significantly better results than the baseline. The SimFuse module provides further improvement when fewer camera views are available, while its benefit saturates as more cameras observe the scene. Together, the approach reaches, and in some metrics improves, the state of the art.

Despite these improvements, limitations remain in the aggregation stage. The current fusion strategy performs independent, per-voxel weighting without explicitly modeling spatial or temporal relationships between views. As a result, complex occlusion configurations and strong viewpoint variations are not always fully resolved. Moreover, weighted averaging assumes that features from different views surviving the SimFuse checks can be combined linearly, which may not optimally preserve viewpoint-specific information. Future work will investigate attention-based fusion mechanisms that learn voxel descriptors from multi-view features within a spatial voxel neighborhood, instead of relying on independent per-voxel weighted averaging. Additionally, future work will investigate improved loss weighting strategies for better balancing the different objectives.

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., 2025. Integrating Viewing Direction and Image Features for Robust Multi-View Multi-Object 3D Pedestrian Tracking. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-G-2025, 47–55.

Alturki, R., Hilton, A., Guillemaut, J.-Y., 2025. Enhanced multi-view pedestrian detection using probabilistic occupancy volume. *Proceedings of the Computer Vision and Pattern Recognition Conference*, 3377–3386.

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.

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

Engilberge, M., Shi, H., Wang, Z., Fua, P., 2023b. Two-level data augmentation for calibrated multi-view detection. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 128–136.

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., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2980-2988.

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, J., Shen, L., Sun, G., 2018. Squeeze-and-excitation networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141.

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.

Kingma, D. P., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

Laurentini, A., 2002. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on pattern analysis and machine intelligence*, 16(2), 150–162.

Lee, W.-Y., Jovanov, L., Philips, W., 2023. Multi-view target transformation for pedestrian detection. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 90–99.

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.

Qiu, R., Xu, M., Yan, Y., Smith, J. S., Yang, X., 2022. 3d random occlusion and multi-layer projection for deep multi-camera pedestrian localization. *European Conference on Computer Vision*, Springer, 695–710.

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.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-2-2026-197-2026 | © Author(s) 2026. CC BY 4.0 License.