MV2GF: Multi-view Pedestrian Detection with a Visual Geometric Foundation Model

Taiga Yamane, Satoshi Suzuki, Ryo Masumura, Shota Orihashi, Tomohiro Tanaka, Mana Ihori, and Naoki Makishima

Human Informatics Laboratories, NTT, Inc., Japan
{taiga.yamane, satoshixv.suzuki}@ntt.com
https://mv2gf.github.io

Abstract. Multi-View Pedestrian Detection (MVPD) aims to detect pedestrians in the form of a bird's eye view map from multi-view images. Recent MVPD methods adopt a unified framework that projects 2D image features into a 3D world space and aggregates them into a single feature. Although they are effective, they struggle to generalize to unseen camera configurations during training due to two main issues. First, they are difficult to capture accurate visual geometry across views in unseen camera configurations. Second, they make detection models highly dependent on distortion patterns during training arising from their image feature projection. To address these, we leverage a visual geometric foundation model and propose MV2GF. This foundation model has exhibited strong generalization in capturing visual geometry across views and predicting accurate 3D attributes in diverse camera configurations. MV2GF fuses task-specific features with general-purpose geometric features extracted by the foundation model to effectively capture the visual geometry even in unseen camera configurations. Furthermore, MV2GF projects each pixel in the image features to an appropriate 3D location using 3D pointmaps predicted by the foundation model, preventing the detection model from depending on distortion patterns during training. Our experiments demonstrate the effectiveness of leveraging a visual geometric foundation model for MVPD and that MV2GF generalizes better than existing methods.

Keywords: Multi-view pedestrian detection · Bird's eye view · Visual geometric foundation model

1 Introduction

Pedestrian detection aims to find and localize pedestrians from images. This task is a crucial component of many applications, such as surveillance systems [12], autonomous driving [23], and robotics [36]. These applications often require detection in highly crowded and cluttered scenes. In such scenes, addressing occlusion, which hinders detection, is important [7]. Fortunately, multiple cameras with overlapping fields of view are often available in many applications. Therefore, extensive studies have explored multi-view pedestrian detection (MVPD)

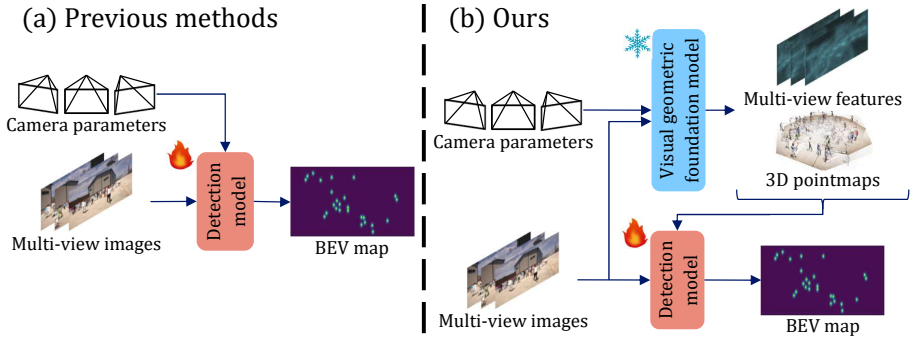

Fig. 1: (a) Previous MVPD methods. (b) Our MV2GF leverages a pretrained visual geometric foundation model. The red blocks are trainable, and the blue block is frozen.

to overcome occlusion [8,17,43]. This task aims to detect pedestrians in the form of a bird's eye view (BEV) map from multi-view images. By aggregating complementary information across multiple views, MVPD is expected to be more robust to occlusion than single-view pedestrian detection [5, 11, 57].

Recent MVPD methods adopt a unified deep learning-based framework [1–3, 13, 16, 17, 20, 34, 39, 40, 43, 50, 55]. They first extract 2D image features for individual views and project them into a 3D world space using calibrated camera parameters. Then, they aggregate projected features from multiple views into a single BEV feature based on visual geometric information across views contained in the image features, such as 3D positional relationships among pedestrians or objects in views. Finally, they predict a BEV map from the BEV feature.

While these methods have exhibited promising results, they struggle to generalize to camera configurations not included in MVPD training data [2,43], due to two main issues. Here, camera configurations include camera types, poses, and positions. The first issue is that existing methods are difficult to accurately capture visual geometry across views for unseen camera configurations during training. To accurately capture it, understanding the correspondence between 2D views and the 3D world space is important. Nonetheless, existing methods learn this correspondence solely from camera configurations in limited MVPD training data. Since the 2D-3D correspondence varies with camera configurations, for camera configurations not seen during training, existing methods often fail to understand this correspondence and to capture accurate visual geometry. This makes it difficult for detection models to aggregate features from multiple views for such camera configurations. The second issue is that existing methods make detection models heavily dependent on distortion patterns arising from image feature projection for camera configurations in MVPD training data. Existing methods employ a perspective transformation [17] or its derivative transformations [1–3,20,34,39,50] to project 2D image features into a 3D world space. These transformations cause shadow-like distortions that spread pedestrians out like shadows [16], as they project all pixels in image features onto the same planes in a 3D world space. These distortions obscure precise pedestrian locations on a

BEV map, affecting detection performance [16]. Existing methods learn distortion patterns from camera configurations in MVPD training data and implicitly compensate for the distortions. However, they often fail to compensate for them in the patterns for camera configurations not included in the training data, as distortion patterns vary significantly with camera configurations. This hinders generalization to camera configurations not seen during training.

To overcome these issues, as shown in Fig. 1 (b), we leverage a pretrained visual geometric foundation model, also known as a feed-forward 3D reconstruction transformer [6, 22, 25, 28, 41, 44–47, 51, 56]. It learns to capture visual geometry across views from large-scale multi-view datasets using a transformer [42] and demonstrates strong generalization performance in various multi-view 3D tasks across diverse camera configurations. Therefore, leveraging multi-view image features extracted by the transformer of the foundation model, which contain general-purpose geometric information across views, helps the detection model capture accurate visual geometry for camera configurations not included in MVPD training data. In addition, instead of using a perspective transformation or its derivatives, we use 3D pointmaps predicted by the foundation model to project image features into a 3D world space. These pointmaps indicate where each pixel in multi-view images is located in a 3D world space [46]. Using these pointmaps to project image features avoids shadow-like distortions because not all pixels in the image features are projected onto the same planes.

Based on this, we propose **M**ulti-**V**iew detection with a **V**isual **G**eometric **F**oundation model, named **MV2GF**. In this paper, we mainly employ pretrained Depth Anything 3 (DA3) [28] as a visual geometric foundation model. This is because it can predict 3D attributes at a real-world metric scale, inject camera parameters, and process more than two views in a single forward pass, all of which are suitable for MVPD. To effectively leverage multi-view image features and 3D pointmaps from DA3, we introduce two key components: Task-specific and Geometric information Fusion (TGF) and Feature Pointmap Aggregation (FPA). TGF aims to fuse general-purpose geometric information from DA3 with task-specific information and to generate informative multi-view image features. First, this extracts multi-scale image features for individual views using a trainable ResNet [15] and takes multi-view image features from the frozen DA3 transformer. By training ResNet end-to-end for MVPD, its features contain task-specific information focused on pedestrians, whereas DA3 features contain general-purpose geometric information. Subsequently, TGF fuses ResNet features with DA3 features using a feature pyramid network (FPN) [29], yielding informative high-resolution image features. TGF helps capture visual geometry across views for diverse camera configurations, even those not seen in MVPD training data, while highlighting foreground pedestrians in individual views. FPA aims to aggregate information across views and generate a BEV feature using image features from TGF and 3D pointmaps from DA3, while avoiding distortions. First, it projects each pixel in the image features to an appropriate location in a 3D world space using coordinates obtained from the 3D pointmaps. Then, FPA generates a 3D voxel feature by aggregating features from multiple views within

each voxel using max pooling and outputs a BEV feature by compressing the vertical dimension. FPA prevents the detection model from becoming dependent on distortion patterns during training.

Extensive experiments demonstrate that leveraging a visual geometric foundation model via our TGF and FPA greatly improves generalization performance to camera configurations not included in MVPD training data. Particularly on MODA metric [21], MV2GF trained on GMVD [43] training data outperforms the previous state-of-the-art method by 4.6 points for GMVD testing data, 4.7 points for MVPerception [52], and 2.2 points for Wildtrack [7].

2 Related Work

2.1 Multi-view Pedestrian Detection

There are two main paradigms in MVPD. Traditional methods [4, 8, 9, 14, 37, 49] first detect pedestrians for each view using single-view pedestrian detection [35, 58]. They then aggregate detection results for the same pedestrians across views into locations on a BEV map using techniques such as conditional random field [4], mean-field inference [14], or clustering [49]. Although these methods are important early attempts, their final detection performance is limited because occlusion severely hinders the detection in each view. To address this, recent methods [1–3, 13, 16, 17, 20, 34, 39, 40, 43, 50, 55] adopt a unified deep learning-based framework that does not rely on single-view pedestrian detection. They project image features across multiple views into a 3D world space and aggregate projected features into a single BEV feature based on visual geometric information across views contained in image features. They predict a BEV map from this BEV feature. MVDet [17] is a pioneer of these methods and projects image features onto a ground plane using a perspective transformation. To further improve detection performance, subsequent studies have extended MVDet by introducing more effective projections [1, 2, 34, 39, 50], training strategies [13, 16, 34, 40, 43, 55], and network architectures [2, 3, 16, 20, 24].

Despite these improvements, in terms of capturing visual geometry across views and representing it in image features, many existing methods [3, 16, 17, 34, 39, 43, 50] simply apply a single-view image encoder for each view, such as dilated ResNet [15, 53, 54] or FPN [29]. Some methods [2, 20, 24] have introduced effective image feature extraction tailored for MVPD. MVFP [2] highlights the features of the image foreground by combining mean and max pooling, focusing on the geometry of pedestrians. BoosterSHOT [20] selects information in image features important for capturing the geometry in scenes using a network like squeeze-and-excitation attention [18]. While they have improved detection performance, existing methods learn to capture visual geometry across views solely from camera configurations in limited MVPD training data and struggle to accurately capture it for camera configurations not seen during training. Our MV2GF leverages multi-view features extracted by a visual geometric foundation model to effectively capture the visual geometry for diverse camera configurations.

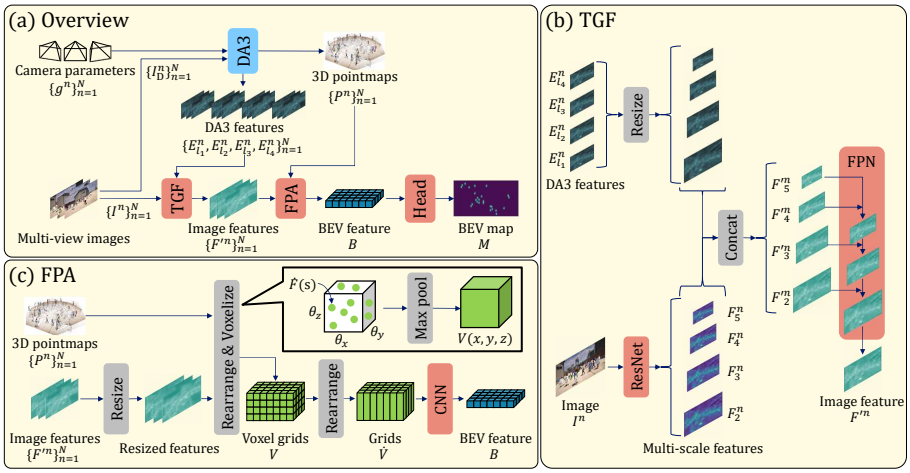

Fig. 2: (a) Overview of MV2GF. It consists of (b) TGF and (c) FPA. We set the red blocks to be trainable, and the gray blocks do not have trainable parameters.

In terms of projecting image features into a 3D world space, existing MVPD methods rely on a perspective transformation [17] or its derivative transformations [1–3, 20, 34, 39, 50]. As described in Sec. 1, these transformations introduce shadow-like distortions [16]. By applying a convolutional neural network (CNN) or deformable attention [59] to projected features, existing methods learn distortion patterns and implicitly compensate for the distortions, achieving robust detection against patterns identical to those in MVPD training data. However, this makes detection models heavily dependent on distortion patterns during training and hinders generalization to camera configurations not included in the training data, as distortion patterns vary significantly with camera configurations. In the context of autonomous driving, transformer-based projections [19, 26, 27, 31, 48] are the dominant projections and can mitigate the distortions. However, they assume that camera configurations and the size of the BEV map are identical during training and testing, making them unsuitable for practical MVPD. Our MV2GF avoids the distortions by using 3D pointmaps predicted by a visual geometric foundation model to project image features, as described in Sec. 1.

2.2 Visual Geometric Foundation Models

Recently, visual geometric foundation models [6, 22, 25, 28, 41, 44–47, 51, 56], also known as feed-forward 3D reconstruction transformers, have exhibited strong generalization abilities across various multi-view 3D tasks, such as multi-view depth estimation, 3D reconstruction, and camera pose estimation. Through training on large-scale multi-view datasets, they learn to capture accurate visual geometry across views for diverse camera configurations by applying transformers [42] to multi-view images or videos. DUSt3R [46] and MASt3R [25] are pioneers in visual geometric foundation models. They predict 3D pointmaps

between two views by applying a transformer to a pair of images and compute various 3D attributes from these pointmaps. While effective for two views, they require a time-consuming and unstable post-processing to consider more than two views. Subsequent studies have focused on overcoming this limitation [6, 22, 28, 41, 44, 45, 47, 51, 56]. Among them, VGGT [44] has achieved better results across various 3D tasks than previous methods by applying a transformer to multiple views in a single forward pass and leveraging multi-task learning. After that, MapAnything [22], Pi3X [47], and Depth Anything 3 (DA3) [28] have been extended from VGGT to predict 3D attributes at a real-world metric scale and inject camera parameters. Our MV2GF leverages DA3 because of such abilities and its high generalization performance for diverse camera configurations.

3 Proposed Method

This section introduces MV2GF, which leverages a visual geometric foundation model to improve the generalization performance of the detection model to camera configurations not included in MVPD training data. Figure 2 (a) shows an overview of MV2GF that contains TGF and FPA. We leverage Depth Anything 3 (DA3) [28] as a visual geometric foundation model. From multi-view images and camera parameters, DA3 extracts general-purpose geometric features that capture visual geometry across views and predicts 3D pointmaps for individual views. MV2GF leverages these geometric features and 3D pointmaps in TGF and FPA, respectively. TGF aims to fuse internally extracted task-specific features with general-purpose geometric features extracted by DA3 and to generate informative high-resolution image features. By doing so, MV2GF uses the visual geometry captured by DA3 for MVPD. FPA aims to project image features extracted by TGF into a 3D world space using 3D pointmaps predicted by DA3 and to generate a BEV feature by aggregating features across views. By using these pointmaps, MV2GF effectively avoids shadow-like distortions. Finally, a detection head predicts a BEV map from the BEV feature. Hereafter, we explain DA3, TGF, FPA, and the training and inference of MV2GF in that order.

3.1 Preliminary

In this subsection, we explain DA3 [28], which our method leverages as a visual geometric foundation model. The inputs to DA3 are multi-view images $\{I_{\mathrm{D}}^n\}_{n=1}^N$ and camera parameters $\{g^n\}_{n=1}^N$ (intrinsics and extrinsics) of N views. $I_{\mathrm{D}}^n \in \mathbb{R}^{3\times H_{\mathrm{D}}\times W_{\mathrm{D}}}$ is the n-th view image, where H_{D} and W_{D} are the height and width of the input image. $g^n \in \mathbb{R}^9$ is the n-th view camera parameters, representing the translation vector $t^n \in \mathbb{R}^3$, the rotation quaternion $q^n \in \mathbb{R}^4$, and the field of view $f^n \in \mathbb{R}^2$. First, DA3 patchifies I_{D}^n into a set of tokens $T_0^n \in \mathbb{R}^{C_{\mathrm{D}}\times(H_{\mathrm{D}}/14\cdot W_{\mathrm{D}}/14)}$ and embeds g^n into a token $c_0^n \in \mathbb{R}^{C_{\mathrm{D}}\times 1}$, where C_{D} is the channel size of DA3. Then, it captures visual geometry across views via a transformer with L blocks consisting of cross-view and within-view attention, as

$$\{c_l^n, T_l^n\}_{n=1}^N = \mathrm{TransformerBlock}_l(\{c_{l-1}^n, T_{l-1}^n\}_{n=1}^N), \tag{1}$$

where $c_l^n \in \mathbb{R}^{C_\mathrm{D} \times 1}$ and $T_l^n \in \mathbb{R}^{C_\mathrm{D} \times (H_\mathrm{D}/14 \cdot W_\mathrm{D}/14)}$ are tokens from the l-th transformer block $\mathrm{TransformerBlock}_l(\cdot)$. For simplicity, we omit the positional encoding in the transformer. DA3 predicts a 3D pointmap $P^n \in \mathbb{R}^{3 \times H_\mathrm{D} \times W_\mathrm{D}}$ for each view using tokens from four blocks $\{l_1, l_2, l_3, l_4\}$ and camera parameters, as

$$P^n = \mathrm{Prediction}(\{T_{l_1}^n, T_{l_2}^n, T_{l_3}^n, T_{l_4}^n, g^n\}), \tag{2}$$

where $\mathrm{Prediction}(\cdot)$ indicates the prediction head of DA3. For 2D image pixel coordinates (u, v), $P^n(u, v) \in \mathbb{R}^3$ represents predicted 3D world coordinates of the pixel $I_\mathrm{D}^n(u, v) \in \mathbb{R}^3$. Let $\{E_{l_1}^n, E_{l_2}^n, E_{l_3}^n, E_{l_4}^n\}_{n=1}^N$ be rearranged multi-view image features, where $E_l^n \in \mathbb{R}^{C_\mathrm{D} \times H_\mathrm{D}/14 \times W_\mathrm{D}/14}$ is rearranged from T_l^n. MV2GF uses these $\{E_{l_1}^n, E_{l_2}^n, E_{l_3}^n, E_{l_4}^n\}_{n=1}^N$ in TGF and the pointmaps $\{P^n\}_{n=1}^N$ in FPA.

3.2 Task-specific and Geometric Information Fusion

Figure 2 (b) shows our Task-specific and Geometric information Fusion (TGF). TGF generates informative high-resolution image features for individual views by fusing task-specific features extracted by a trainable ResNet [15] and general-purpose geometric features extracted by the frozen DA3. The former features highlight foreground pedestrians in each view, and the latter features help capture visual geometry across views for diverse camera configurations.

The inputs to TGF are DA3 features $\{E_{l_1}^n, E_{l_2}^n, E_{l_3}^n, E_{l_4}^n\}_{n=1}^N$ and multi-view images $\{I^n\}_{n=1}^N$. The input resolution of $I^n \in \mathbb{R}^{3 \times H \times W}$ may be different from that of I_D^n due to the pretraining setting of DA3. First, TGF extracts multi-scale features $\{F_2^n, F_3^n, F_4^n, F_5^n\}$ for each view using a trainable ResNet, where $F_k^n \in \mathbb{R}^{C_k \times H/2^k \times W/2^k}$ and C_k are the n-th view feature and the channel size of the k-th stage in ResNet, respectively. By making ResNet trainable for MVPD, these features contain task-specific information focused on pedestrians at various scales. Then, TGF resizes $E_{l_1}^n$, $E_{l_2}^n$, $E_{l_3}^n$, and $E_{l_4}^n$ to the same resolution as F_2^n, F_3^n, F_4^n, and F_5^n using a bilinear interpolation, respectively, and concatenates the same resolution features along the channel direction. Let these concatenated features be $\{F_2^{'n}, F_3^{'n}, F_4^{'n}, F_5^{'n}\}$, where $F_k^{'n} \in \mathbb{R}^{(C_k + C_\mathrm{D}) \times H/2^k \times W/2^k}$. Finally, by applying FPN [29] to these features, TGF gradually fuses task-specific information with general-purpose geometric information from low-resolution to high-resolution, resulting in informative high-resolution image feature $F^{'n} \in \mathbb{R}^{C \times H/4 \times W/4}$. Here, C is the channel size in FPN. By performing these processes for all views, TGF outputs $\{F^{'n}\}_{n=1}^N$. These features contain not only task-specific information but also general-purpose geometric information extracted by DA3, helping capture visual geometry across views for diverse camera configurations, even those not seen in MVPD training data.

3.3 Feature Pointmap Aggregation

Figure 2 (c) shows our Feature Pointmap Aggregation (FPA). FPA projects image features extracted by TGF into a 3D world space based on 3D pointmaps

predicted by DA3 and aggregates them into a single BEV feature. Using these pointmaps, each pixel in the TGF features is projected to an appropriate 3D location, thereby avoiding shadow-like distortions. This makes the detection model more robust to camera configurations not included in the training data than employing a perspective transformation or its derivatives used in existing methods.

The inputs to FPA are 3D pointmaps $\{P^n\}_{n=1}^N$ from DA3 and image features $\{F^{'n}\}_{n=1}^N$ from TGF. First, FPA resizes each of $\{F^{'n}\}_{n=1}^N$ to the same resolution as P^n using a bilinear interpolation and rearranges the N pointmaps and resized features into $\dot{P} \in \mathbb{R}^{3\times(N\cdot H_\mathrm{D}\cdot W_\mathrm{D})}$ and $\dot{F} \in \mathbb{R}^{C\times(N\cdot H_\mathrm{D}\cdot W_\mathrm{D})}$. After this, we obtain the 3D world location for each C-channel feature vector $\dot{F}(i) \in \mathbb{R}^C$ by $\dot{P}(i) \in \mathbb{R}^3$. Next, FPA creates C-channel voxel grids $V \in \mathbb{R}^{C\times X/4\times Y/4\times Z}$ and initializes each voxel with a C-channel zero vector. Here, X and Y are the height and width of a BEV map defined by the datasets, and Z is the assumed vertical dimension. We set the height and width of V to one-fourth those of the original BEV map for computational efficiency. Then, for each voxel in V, FPA performs max pooling to feature vectors whose 3D locations lie within that voxel and uses the pooling result as a feature vector of the corresponding voxel. Let the voxel size of V be $\theta_x \times \theta_y \times \theta_z$. This process is formulated, as

$$S(x,y,z) = \left\{ s \in \{1,\dots,N\cdot H_\mathrm{D}\cdot W_\mathrm{D}\} \;\middle|\; \begin{pmatrix}\theta_x x\\ \theta_y y\\ \theta_z z\end{pmatrix} \le \dot{P}(s) < \begin{pmatrix}\theta_x(x+1)\\ \theta_y(y+1)\\ \theta_z(z+1)\end{pmatrix} \right\}, \tag{3}$$

$$V(x,y,z) = \max_{s\in S(x,y,z)} \dot{F}(s), \tag{4}$$

where (x,y,z) indicates the voxel coordinates in V. If $S(x,y,z) = \varnothing$, we keep $V(x,y,z)$ as the zero vector. Finally, we compress Z dimension in V by rearranging it to $\dot{V} \in \mathbb{R}^{(C\cdot Z)\times X/4\times Y/4}$ and applying a CNN, resulting in a BEV feature $B \in \mathbb{R}^{C\times X/4\times Y/4}$. This series of processes projects the most relevant and informative features from multiple views into each 3D voxel and aggregates information across views into a BEV feature. FPA projects each feature vector to an appropriate 3D location, avoiding shadow-like distortions and improving generalization to camera configurations not included in the training data.

3.4 Training and Inference

The detection head, followed by the sigmoid function, predicts a BEV pedestrian occupancy map $M \in \mathbb{R}^{1\times X/4\times Y/4}$ from a BEV feature B. Because M has a lower resolution than the ground truth BEV map, the detection head also regresses the offset map $O \in \mathbb{R}^{2\times X/4\times Y/4}$ to recover the discretization error, as in [16].

During training, we compute Focal loss [30] for M and L1 loss [58] for O, following [16]. Let $\mathcal{L}_\mathrm{det}$ and $\mathcal{L}_\mathrm{off}$ be losses for M and O, respectively. They are defined using the down-sampled and Gaussian-smoothed ground truth BEV map $\bar{M} \in \mathbb{R}^{1\times X/4\times Y/4}$ and the ground truth offset map $\bar{O} \in \mathbb{R}^{2\times X/4\times Y/4}$, as

$$\mathcal{L}_\mathrm{det} = \mathrm{FocalLoss}(M,\bar{M}),\ \mathcal{L}_\mathrm{off} = \mathrm{L1Loss}(O,\bar{O}), \tag{5}$$

where FocalLoss$(\cdot,\cdot)$ and L1Loss$(\cdot,\cdot)$ denote Focal loss and L1 loss, respectively. In addition, as in [17], we predict occupancy maps of pedestrian heads and feet for each view from TGF image feature F'^{n} and compute Focal loss for these predictions as an auxiliary loss. Let $M^n_{\text{heads}} \in \mathbb{R}^{1\times H/4\times W/4}$ and $M^n_{\text{feet}} \in \mathbb{R}^{1\times H/4\times W/4}$ be predicted occupancy maps of pedestrian heads and feet for n-th view. We compute a loss $\mathcal{L}^n_{\text{view}}$ for them, as

$$\mathcal{L}^n_{\text{view}} = \text{FocalLoss}(M^n_{\text{heads}}, \bar{M}^n_{\text{heads}}) + \text{FocalLoss}(M^n_{\text{feet}}, \bar{M}^n_{\text{feet}}), \tag{6}$$

where $\bar{M}^n_{\text{heads}} \in \mathbb{R}^{1\times H/4\times W/4}$ and $\bar{M}^n_{\text{feet}} \in \mathbb{R}^{1\times H/4\times W/4}$ are the down-sampled ground truth heads and feet map for n-th view smoothed with a Gaussian kernel, respectively. We optimize MV2GF using the following overall loss $\mathcal{L}$, as

$$\mathcal{L} = \mathcal{L}_{\text{det}} + \mathcal{L}_{\text{off}} + \frac{1}{N}\Sigma^N_{n=1}\mathcal{L}^n_{\text{view}}. \tag{7}$$

During inference, we extract pedestrians over the threshold from a predicted BEV occupancy map M and convert their locations to those in a $X \times Y$ BEV map using a predicted offset map O, in the same way as [16].

4 Experiments

4.1 Datasets

In this subsection, we explain the three MVPD datasets used in our experiments. Note that the following datasets are not used for pretraining of DA3.

Wildtrack [7]. This is a real-world dataset comprising 400 multi-view frames at 1080×1920 resolution captured by 7 cameras. Wildtrack covers a $12\,\text{m} \times 36\,\text{m}$ region quantized into 480×1440 grids using square grid cells of $2.5\,\text{cm} \times 2.5\,\text{cm}$. Each frame includes 20 pedestrians in the region on average. The first 360 frames are used for training, and the last 40 frames are used for testing.

MVPerception [52]. This is a synthetic dataset comprising 800 multi-view frames captured by 6 cameras and closely follows the style of Wildtrack. This dataset covers a $16\,\text{m} \times 25\,\text{m}$ region quantized into 640×1000 grids. Each frame includes 80 pedestrians in the region on average, making this dataset more crowded than the other two datasets. The first 720 frames are used for training, and the remaining 80 frames are used for testing.

GMVD [43]. This is a large-scale synthetic dataset. While the other two datasets consist of a single scene and camera configuration, GMVD comprises 7 scenes and 13 camera configurations. In addition, this dataset contains diverse environmental conditions, such as time and weather. Therefore, GMVD is more challenging than the other two datasets. A total of 5997 multi-view frames at 1080×1920 resolution are included in this dataset. Each scene covers a different-sized region, and each region is quantized into grids using the same grid cell size as Wildtrack. Each frame includes 20-40 pedestrians on average. In our experiments, we split

Method	GMVD-D				MVPerception				Wildtrack			
	MODA	MODP	Prec.	Rec.	MODA	MODP	Prec.	Rec.	MODA	MODP	Prec.	Rec.
MVDet [17]	69.3	76.8	94.9	73.2	65.5	71.1	90.1	73.6	60.7	72.5	82.6	76.9
SHOT [39]	71.5	77.9	93.7	76.7	69.4	73.1	91.1	76.8	70.8	73.1	82.3	90.2
MVDeTr [16]	73.8	81.7	96.1	76.9	71.3	78.0	92.8	77.3	72.4	78.1	85.6	87.1
3DROM [34]	74.5	77.8	93.7	80.0	74.1	75.6	91.1	82.1	76.2	75.7	85.2	92.1
BoosterSHOT [20]	74.9	80.2	96.2	77.8	74.5	76.4	87.4	87.1	78.6	79.0	87.4	91.8
OmniOcc [3]	75.1	76.9	92.3	82.0	76.6	74.0	90.3	85.7	82.0	77.9	90.5	91.7
MVFP [2]	75.7	78.2	94.3	80.5	79.9	74.8	91.9	87.6	85.2	78.3	92.2	93.1
MSMVD [50]	80.2	81.3	95.7	83.9	81.8	79.4	94.2	87.2	85.7	79.3	92.2	**93.6**
Ours	**84.8**	**83.1**	**96.8**	**87.7**	**86.5**	**80.6**	**98.5**	**87.8**	**87.9**	**79.6**	**94.7**	93.2

Table 1: Comparison with previous methods under settings where camera configurations during testing are different from those during training. All models are trained on GMVD-D training split.

GMVD in two ways. First, we split GMVD into 6 scenes with 11 camera configurations for training and the remaining 1 scene with 2 camera configurations for testing. We refer to this split as GMVD-D and use it to evaluate detection performance when camera configurations during testing differ from those during training. Second, we split GMVD into the first 80% frames of each scene and camera configuration for training and the remaining 20% frames for testing. We refer to this split as GMVD-S and use it to evaluate detection performance when camera configurations during testing are the same as those during training.

4.2 Implementation Details and Evaluation Metrics

Unless otherwise mentioned, we used ResNet18 [15] in TGF and pretrained DA3 Giant [28] as a visual geometric foundation model. ResNet18 was pretrained on ImageNet [10]. DA3 Giant consists of 40 transformer blocks (*i.e.*, $L = 40$) and uses tokens from 20th, 28th, 34th, and 40th blocks for prediction (*i.e.*, $\{l_1, l_2, l_3, l_4\} = \{20, 28, 34, 40\}$). The channel size C_{D} in DA3 was 1536. Since the grid cell size of the ground truth BEV map is 2.5 cm × 2.5 cm for all datasets, we set θ_x and θ_y in FPA to 10 cm. We set θ_z to 50 cm and Z of voxel grids V to 4. This means using features up to 2.0 m in the vertical direction, covering the height of most pedestrians. We set the channel size C in FPN to 256 and the detection threshold to 0.4. We used a convolutional layer for the detection head.

We resized input images to 720×1280 for TGF and to 280×504 for DA3. We optimized the model using the Adam optimizer [32]. We set the batch size to 1 and accumulated gradients over 16 batches. We initialized the learning rate to 1.0×10^{-3} and decayed it to 1.0×10^{-6} following a cosine schedule. We trained the model for 10 epochs on GMVD-D and GMVD-S, and for 50 epochs on Wildtrack and MVPerception. More detailed implementations are provided in Supp. A.

Following previous studies [17,43], we used four common metrics proposed by Chavdarova *et al.* [7] and Kasturi *et al.* [21]: multiple object detection accuracy (MODA), multiple object detection precision (MODP), precision (Prec.), and recall (Rec.). A detected pedestrian was classified as a true positive if its distance from the ground truth was within 0.5 meters. MODA was used as the primary metric because it considers both false positives and false negatives.

Method	GMVD-S				MVPerception				Wildtrack			
	MODA	MODP	Prec.	Rec.	MODA	MODP	Prec.	Rec.	MODA	MODP	Prec.	Rec.
MVDet [17]	82.8	80.4	97.1	85.3	90.0	81.2	94.3	95.8	88.2	75.7	94.7	93.6
SHOT [39]	84.5	79.6	97.5	86.7	92.3	85.5	95.2	97.1	90.2	76.5	96.1	94.0
MVDeTr [16]	86.9	84.6	97.4	89.3	93.0	90.2	96.5	96.6	91.5	82.1	97.4	94.0
3DROM [34]	88.1	84.7	97.0	90.5	94.1	85.5	97.4	96.7	93.5	75.9	97.2	96.2
BoosterSHOT [20]	88.7	84.9	97.0	91.5	94.5	88.5	98.3	96.1	92.8	**84.9**	**97.5**	95.3
OmniOcc [3]	89.8	82.7	97.2	92.4	94.2	86.4	97.4	96.8	93.5	81.5	94.9	97.8
MVFP [2]	88.6	81.0	97.4	91.0	95.0	85.7	98.4	96.5	94.1	78.8	96.4	97.7
MSMVD [50]	91.1	**86.6**	**98.0**	93.0	96.4	**91.1**	**99.7**	96.8	94.6	83.3	95.9	98.8
Ours	**91.3**	86.1	97.2	**93.9**	**96.5**	90.5	98.6	**97.8**	**94.7**	82.9	95.8	**98.9**

Table 2: Comparison with previous methods under settings where camera configurations during testing are identical to those during training. All models are trained on the training split of the same dataset used for testing.

4.3 Comparison with Previous Methods

To verify the effectiveness of MV2GF, we compared its generalization ability to camera configurations not included in MVPD training data with that of previous state-of-the-art methods. Especially, we employed MVDet [17], SHOT [39], MVDeTr [16], 3DROM [34], BoosterSHOT [20], OmniOcc [3], MVFP [2], and MSMVD [50]. For the first five methods, we added max pooling after the projection to handle any number of views by modifying their official implementations. For the other methods, we re-implemented them from scratch based on the original papers. These previous methods rely on a perspective transformation or its derivatives to project image features and learn to capture visual geometry across views solely from camera configurations in MVPD training data. Table 1 shows the comparison results when models were trained on GMVD-D training split and tested on GMVD-D testing split, MVPerception, and Wildtrack. MV2GF outperformed previous methods on all metrics for GMVD-D and MVPerception. It also achieved the highest MODA, MODP, and precision, and the second-highest recall for Wildtrack. In particular, it exceeded the previous highest MODA by 4.6 points for GMVD-D, 4.7 points for MVPerception, and 2.2 points for Wildtrack. These results show that MV2GF generalizes better to camera configurations not included in MVPD training data than previous methods and demonstrate the effectiveness of leveraging a visual geometric foundation model for MVPD. We visualized the detection results in Supp. E.

While our primary objective is to improve the generalization ability of the detection model, detection performance when camera configurations during testing are identical to those during training is also important. Table 2 shows the comparison results when models were trained on GMVD-S training split, MVPerception training split, or Wildtrack training split and tested on the corresponding testing split. For all datasets, MV2GF achieved superior MODA and recall to all previous methods and was comparable to MSMVD. Particularly on MODA, it outperformed MSMVD by 0.2, 0.1, and 0.1 points for GMVD-S, MVPerception, and Wildtrack, respectively. These results demonstrate that MV2GF is effective even when camera configurations during testing are identical to those during training. On the other hand, MV2GF underperformed MSMVD on MODP and

TGF	FPA	MODA	MODP	Prec.	Rec.
		73.4	78.5	93.2	79.1
✓		77.8	79.5	93.8	83.3
	✓	81.3	82.6	95.5	85.4
✓	✓	**84.8**	**83.1**	**96.8**	**87.7**

(a)

ResNet	DA3	MODA	MODP	Prec.	Rec.
✓		81.3	82.6	95.5	85.4
	✓	77.5	77.9	95.1	81.7
✓	✓	**84.8**	**83.1**	**96.8**	**87.7**

(b)

	MODA	MODP	Prec.	Rec.
Only 20th	83.6	82.7	96.5	86.7
Only 28th	83.6	83.0	96.4	86.8
Only 34th	83.9	83.0	96.5	87.1
Only 40th	84.0	**83.1**	96.0	87.6
All	**84.8**	**83.1**	**96.8**	**87.7**

(c)

θ_z (cm)	MODA	MODP	Prec.	Rec.
12.5	81.6	81.7	96.3	84.8
25	84.5	82.8	96.6	87.5
50	**84.8**	**83.1**	**96.8**	**87.7**
100	84.3	**83.1**	96.3	87.6
200	83.7	82.9	96.1	87.3

(d)

	MODA	MODP	Prec.	Rec.
DINOv2	82.9	82.2	95.6	86.9
DINOv3	83.2	82.5	96.6	86.2
DA3	**84.8**	**83.1**	**96.8**	**87.7**

(e)

	MODA	MODP	Prec.	Rec.
MapAnything	83.0	82.3	96.5	86.2
Pi3X	82.3	82.6	96.4	85.5
DA3	**84.8**	**83.1**	**96.8**	**87.7**

(f)

Table 3: (a) Effect of TGF and FPA. (b) Effect of using features from ResNet and DA3 in TGF. (c) Effect of using features from multiple transformer blocks in DA3. (d) Effect of θ_z in FPA. (e) Effect of using features from a visual geometric foundation model. (f) Effect of the visual geometric foundation model choice. Default settings in MV2GF are marked in gray. All models are trained on GMVD-D training split and tested on its testing split.

precision. This is because MSMVD uses multi-scale BEV features with multi-scale image features, enabling precise pedestrian localization. In Supp. C, we investigated the effect of this multi-scale BEV design.

4.4 Ablation Studies

In this subsection, we investigated the effects of each component in MV2GF from various viewpoints. We conducted more detailed investigations in Supp. B. Unless otherwise mentioned, all models were trained on GMVD-D training split and tested on GMVD-D testing split.

Effect of TGF and FPA. To investigate the effects of TGF and FPA on detection performance, we incrementally added each of them to a baseline model. As a baseline model, we used MVDet [17] trained with our loss and employed ResNet18 [15] and FPN [29] for its image feature extraction. Table 3 (a) shows the effects of TGF and FPA. Adding TGF improved detection performance on all metrics, and the improvement was 4.4 points on MODA. The baseline with TGF outperformed all previous methods in Tab. 1 on MODA, other than MSMVD. This result demonstrates the effectiveness of using geometric features from DA3 via TGF to capture visual geometry across views. Adding FPA also improved detection performance, and the improvement was greater than that by TGF. In addition, the baseline with FPA outperformed all previous methods in Tab. 1 on MODA. This result highlights the effectiveness of projecting image features via

FPA while avoiding shadow-like distortions. The combined use of TGF and FPA further improved detection performance, and the improvement from the baseline was 11.4 points on MODA. This result indicates that both TGF and FPA are important for better generalization to camera configurations not included in MVPD training data.

Effect of using features from ResNet and DA3 in TGF. In TGF, we utilized both task-specific ResNet features and general-purpose geometric DA3 features. To verify the advantage of this, we compared cases where both features were used with only ResNet features or DA3 features were used, as shown in Tab. 3 (b). Using both features outperformed using either on all metrics. In particular, using both features outperformed using only ResNet features and only DA3 features by 3.5 points and 7.3 points on MODA, respectively. These results demonstrate that both task-specific ResNet features and general-purpose geometric DA3 features are important for better detection performance in camera configurations not included in the training data.

Effect of using features from multiple transformer blocks in DA3. In TGF, we used geometric DA3 features from four transformer blocks (20th, 28th, 34th, and 40th blocks) that were also used for predicting 3D pointmaps in DA3. To verify the advantage of this, we compared cases where features from all four blocks were used with features only from one block were used, as shown in Tab. 3 (c). Using features from four blocks yielded better detection performance across all metrics than using features from a single block. This result demonstrates the effectiveness of capturing visual geometry across views using the features from multiple transformer blocks in DA3.

Effect of θ_z in FPA. We compared different θ_z in FPA, as shown in Tab. 3 (d). Note that we set θ_z to 50 cm by default, and all models used features up to 2.0 m in the vertical direction. From 200 cm to 50 cm, smaller θ_z achieved better detection performance. This is because a too large θ_z compresses features too much in the vertical direction, preventing the use of the entire pedestrian's body information in predicting a BEV map. On the other hand, from 50 cm to 12.5 cm, employing smaller θ_z degraded detection performance. This is because a too small θ_z leaves many grids in V as zero vectors, making V too redundant. These results show that our default $\theta_z = 50$ cm is the best choice.

Effect of using features from a visual geometric foundation model. To verify that the performance improvement by using DA3 features stems from its characteristic tailored to capturing visual geometry across views rather than merely from a large amount of pretrained data, we implemented cases where DA3 features in TGF were replaced with features of DINOv2 [33] or DINOv3 [38], as shown in Tab. 3 (e). When using DINOv2 or DINOv3 features, superior detection performance was achieved compared to using only ResNet features (see Tab. 3 (b)), but it was inferior to that of using DA3 features. This result demonstrates the advantage of using DA3 features that focus on capturing visual geometry across views.

	MSMVD				Ours			
N	MODA	MODP	Prec.	Rec.	MODA	MODP	Prec.	Rec.
6	82.7	81.1	95.2	87.1	85.9	82.0	96.8	88.9
5	79.9	79.0	93.0	86.3	84.1	80.5	96.8	87.0
4	70.5	78.7	88.6	80.9	80.0	80.3	95.5	84.0
3	61.4	72.6	85.0	74.5	73.8	78.0	93.7	79.1
2	34.0	63.0	76.0	49.8	57.6	73.3	86.3	67.6

Table 4: Effect of the number of input views N for MSMVD [50] and our MV2GF. Both models are trained on GMVD-D training split and tested on data comprising 6 views in GMVD-D testing split.

Effect of the visual geometric foundation model choice. In the experiments, MV2GF employed DA3 as a visual geometric foundation model because it can predict 3D attributes at a real-world metric scale, inject calibrated camera parameters, and process more than two views in a single forward pass. MapAnything [22] and Pi3X [47] also have such abilities, while DA3 predicts 3D attributes more accurately than them. We investigated the effect of the visual geometric foundation model choice by replacing geometric features in TGF and 3D pointmaps in FPA with those of MapAnything or Pi3X, as shown in Tab. 3 (f). Employing DA3 achieved better detection performance than employing the others. This result indicates that employing a stronger visual geometric foundation model yields better detection performance. On the other hand, even when MapAnything or Pi3X was employed, our method outperformed all previous methods in Tab. 1. This demonstrates that our method generalizes better than previous methods, regardless of the visual geometric foundation model choice.

Effect of the number of views. We investigated the effect of the number of input views N on detection performance. Table 4 shows the detection performance of MSMVD [50] and our MV2GF for different N on data comprising 6 views in GMVD-D testing split. Reducing N negatively affected detection performance across all metrics for both MSMVD and MV2GF. Meanwhile, MV2GF exhibited less performance degradation than MSMVD and achieved superior detection performance across all N. This result demonstrates that MV2GF is more robust than MSMVD with respect to the number of input views.

5 Limitations

Although leveraging a visual geometric foundation model yields high generalization performance for MV2GF to camera configurations not included in MVPD training data, it has a negative impact on inference speed. To investigate this impact, we measured the inference speed of our MV2GF and previous methods using an Nvidia 80GB A100 GPU. Table 5 shows the inference speed of each model, including data loading and preprocessing. These inference speeds were measured on data comprising 6 views in GMVD-D testing split. MV2GF was slightly slower than previous methods. In particular, the gap in inference speed between MV2GF and previous methods was from 1.2 FPS to 1.9 FPS. While

	FPS
MVDet [17]	5.2
SHOT [39]	4.8
MVDeTr [16]	4.6
3DROM [34]	4.6
BoosterSHOT [20]	4.7
OmniOcc [3]	4.6
MVFP [2]	4.6
MSMVD [50]	4.5
Ours	3.3

Table 5: Comparison of inference speed. These inference speeds are measured on data comprising 6 views in GMVD-D testing split using an Nvidia 80GB A100 GPU.

the inference speed required for practical application has not been explicitly discussed in the field of MVPD, most MVPD datasets comprise multi-view videos at a frame rate of 2.0 FPS [7,17,43,52]. Therefore, the inference speed of MV2GF is sufficient to process these datasets. In future work, we will explore an effective way to accelerate the inference of MV2GF by employing a model compression technique (*e.g.*, knowledge distillation).

6 Discussion and Conclusion

We propose a new multi-view pedestrian detection (MVPD) method, named MV2GF. It leverages a visual geometric foundation model to improve the generalization ability of the detection model to camera configurations not included in MVPD training data. In particular, MV2GF leverages multi-view features and 3D pointmaps from this foundation model. The former contains general-purpose geometric information and helps capture visual geometry across views for diverse camera configurations, even those not seen in MVPD training data. The latter is used to project each pixel in image features to an appropriate location in a 3D world space, thereby avoiding shadow-like distortions that occur in previous methods. Extensive experiments demonstrated the advantage of leveraging a visual geometric foundation model for MVPD, and MV2GF achieved significantly superior generalization performance compared to previous methods. We hope that our work will broaden the scope of MVPD applications.

When MV2GF was trained on data with different camera configurations from those in the testing data (see Tab. 1), it generalized better than previous methods. However, its detection performance still underperformed previous methods trained on data with the same camera configurations as the testing data (see Tab. 2). Optimizing detection performance for specific camera configurations is important for some applications. In future work, we will explore a way to further improve detection performance in such situations while maintaining generalization ability to camera configurations not included in MVPD training data.

References

1. Alturki, R., Hilton, A., Guillemaut, J.Y.: Enhanced multi-view pedestrian detection using probabilistic occupancy volume. In: CVPRW (2025)
2. Aung, S., Park, H., Jung, H., Cho, J.: Enhancing multi-view pedestrian detection through generalized 3D feature pulling. In: WACV (2024)
3. Aung, S., Sagong, M.c., Cho, J.: Multi-view pedestrian occupancy prediction with a novel synthetic dataset. In: AAAI (2025)
4. Baqué, P., Fleuret, F., Fua, P.: Deep occlusion reasoning for multi-camera multi-target detection. In: ICCV (2017)
5. Braun, M., Krebs, S., Flohr, F., Gavrila, D.M.: EuroCity Persons: A novel benchmark for person detection in traffic scenes. TPAMI (2019)
6. Cabon, Y., Stoffl, L., Antsfeld, L., Csurka, G., Chidlovskii, B., Revaud, J., Leroy, V.: MUSt3R: Multi-view network for stereo 3d reconstruction. In: CVPR (2025)
7. Chavdarova, T., Baqué, P., Bouquet, S., Maksai, A., Jose, C., Bagautdinov, T., Lettry, L., Fua, P., Van Gool, L., Fleuret, F.: Wildtrack: A multi-camera hd dataset for dense unscripted pedestrian detection. In: CVPR (2018)
8. Chavdarova, T., Fleuret, F.: Deep multi-camera people detection. In: ICMLA (2017)
9. Daryani, A.E., Bhutta, M., Hernandez, B., Medeiros, H.: CaMuViD: Calibration-free multi-view detection. In: CVPR (2025)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: CVPR (2009)
11. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: An evaluation of the state of the art. TPAMI (2011)
12. Elhoseny, M.: Multi-object detection and tracking (MODT) machine learning model for real-time video surveillance systems. CSSP (2020)
13. Engilberge, M., Shi, H., Wang, Z., Fua, P.: Two-level data augmentation for calibrated multi-view detection. In: WACV (2023)
14. Fleuret, F., Berclaz, J., Lengagne, R., Fua, P.: Multicamera people tracking with a probabilistic occupancy map. TPAMI (2008)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
16. Hou, Y., Zheng, L.: Multiview detection with shadow transformer (and view-coherent data augmentation). In: ACMMM (2021)
17. Hou, Y., Zheng, L., Gould, S.: Multiview detection with feature perspective transformation. In: ECCV (2020)
18. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: CVPR (2018)
19. Huang, Y., Zheng, W., Zhang, Y., Zhou, J., Lu, J.: Tri-perspective view for vision-based 3d semantic occupancy prediction. In: CVPR (2023)
20. Hwang, J., Benz, P., Kim, P.: Booster-SHOT: Boosting stacked homography transformations for multiview pedestrian detection with attention. In: WACV (2024)
21. Kasturi, R., Goldgof, D., Soundararajan, P., Manohar, V., Garofolo, J., Bowers, R., Boonstra, M., Korzhova, V., Zhang, J.: Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. TPAMI (2008)
22. Keetha, N., Müller, N., Schönberger, J., Porzi, L., Zhang, Y., Fischer, T., Knapitsch, A., Zauss, D., Weber, E., Antunes, N., et al.: MapAnything: Universal feed-forward metric 3d reconstruction. arXiv:2509.13414 (2025)

23. Khan, A.H., Nawaz, M.S., Dengel, A.: Localized semantic feature mixers for efficient pedestrian detection in autonomous driving. In: CVPR (2023)
24. Lee, W.Y., Jovanov, L., Philips, W.: Multi-view target transformation for pedestrian detection. In: WACVW (2023)
25. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with MASt3R. In: ECCV (2024)
26. Li, J., He, X., Zhou, C., Cheng, X., Wen, Y., Zhang, D.: ViewFormer: Exploring spatiotemporal modeling for multi-view 3d occupancy perception via view-guided transformers. In: ECCV (2024)
27. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Qiao, Y., Dai, J.: BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In: ECCV (2022)
28. Lin, H., Chen, S., Liew, J.H., Chen, D.Y., Li, Z., Shi, G., Feng, J., Kang, B.: Depth Anything 3: Recovering the visual space from any views. arXiv:2511.10647 (2025)
29. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR (2017)
30. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV (2017)
31. Liu, H., Wang, H., Chen, Y., Yang, Z., Zeng, J., Chen, L., Wang, L.: Fully sparse 3d panoptic occupancy prediction. arXiv:2312.17118 (2023)
32. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv:1711.05101 (2017)
33. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: DINOv2: Learning robust visual features without supervision. arXiv:2304.07193 (2023)
34. Qiu, R., Xu, M., Yan, Y., Smith, J.S., Yang, X.: 3D random occlusion and multi-layer projection for deep multi-camera pedestrian localization. In: ECCV (2022)
35. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS (2015)
36. Ribeiro, D., Mateus, A., Miraldo, P., Nascimento, J.C.: A real-time deep learning pedestrian detector for robot navigation. In: ICARSC (2017)
37. Roig, G., Boix, X., Shitrit, H.B., Fua, P.: Conditional random fields for multi-camera object detection. In: ICCV (2011)
38. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., et al.: DINOv3. arXiv:2508.10104 (2025)
39. Song, L., Wu, J., Yang, M., Zhang, Q., Li, Y., Yuan, J.: Stacked homography transformations for multi-view pedestrian detection. In: ICCV (2021)
40. Suzuki, S., Tora, S., Masumura, R.: Scene generalized multi-view pedestrian detection with rotation-based augmentation and regularization. In: ICIP (2024)
41. Tang, Z., Fan, Y., Wang, D., Xu, H., Ranjan, R., Schwing, A., Yan, Z.: MV-DUSt3R+: Single-stage scene reconstruction from sparse views in 2 seconds. In: CVPR (2025)
42. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NIPS (2017)
43. Vora, J., Dutta, S., Jain, K., Karthik, S., Gandhi, V.: Bringing generalization to deep multi-view pedestrian detection. In: WACVW (2023)
44. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: VGGT: Visual geometry grounded transformer. In: CVPR (2025)
45. Wang, Q., Zhang, Y., Holynski, A., Efros, A.A., Kanazawa, A.: Continuous 3d perception model with persistent state. In: CVPR (2025)

46. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: DUSt3R: Geometric 3d vision made easy. In: CVPR (2024)
47. Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., He, T.: π^3: Permutation-equivariant visual geometry learning. arXiv:2507.13347 (2025)
48. Wei, Y., Zhao, L., Zheng, W., Zhu, Z., Zhou, J., Lu, J.: SurroundOcc: Multi-camera 3d occupancy prediction for autonomous driving. In: ICCV (2023)
49. Xu, Y., Liu, X., Liu, Y., Zhu, S.C.: Multi-view people tracking via hierarchical trajectory composition. In: CVPR (2016)
50. Yamane, T., Suzuki, S., Masumura, R., Orihashi, S., Tanaka, T., Ihori, M., Makishima, N., Kawata, N.: MSMVD: Exploiting multi-scale image features via multi-scale bev features for multi-view pedestrian detection. arXiv:2508.20447 (2025)
51. Yang, J., Sax, A., Liang, K.J., Henaff, M., Tang, H., Cao, A., Chai, J., Meier, F., Feiszli, M.: FASt3R: Towards 3d reconstruction of 1000+ images in one forward pass. In: CVPR (2025)
52. Yang, Y., Xu, M., Ralph, J.F., Ling, Y., Pan, X.: An end-to-end tracking framework via multi-view and temporal feature aggregation. CVIU (2024)
53. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: ICLR (2016)
54. Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks. In: CVPR (2017)
55. Zhang, Q., Zhang, K., Chan, A.B., Huang, H.: Mahalanobis distance-based multi-view optimal transport for multi-view crowd localization. In: ECCV (2024)
56. Zhang, S., Wang, J., Xu, Y., Xue, N., Rupprecht, C., Zhou, X., Shen, Y., Wetzstein, G.: FLARE: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In: CVPR (2025)
57. Zhang, S., Benenson, R., Schiele, B.: CityPersons: A diverse dataset for pedestrian detection. In: CVPR (2017)
58. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv:1904.07850 (2019)
59. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: Deformable transformers for end-to-end object detection. arXiv:2010.04159 (2020)