Supplementary Material for Multiview Detection with Cardboard Human Modeling

Fig. 1: The illustration of the neglected persons in Wildtrack dataset.

1 Discussion on the missing annotations

As mentioned in Section 4.6, we identify severe label omissions in the Wildtrack dataset, two typical examples are shown in Fig. 1. In the first image batch marked with green boxes, 4 persons in the bottom left corner are neglected, however, our algorithm successfully predicts these persons' locations. In the other batch of images marked with red and yellow boxes, our system demonstrates that there are more people neglected near the edge of the detection area. According to the image, these persons are standing inside the detection area bounded by the purple lines, while their labels are not provided. The lack of such annotations may cause a *'fake'* high false positive rate if the algorithm successfully makes

Fig. 2: The visualization of the masking operation applied on the Wildtrack dataset. In each camera view, the origin detection area defined in the original dataset is bounded with purple lines, while the area boundaries after masking are colored in blue. As illustrated in the BEV heatmap, we identify an enormous amount of missing annotations near the edge of the detection plane (lies in between the purple ground area and the blue masked area), the mask is applied to filter most of the missing labels and ambiguities. The masked detection area is marked by a bright blue color in the last two BEV heatmaps.

the prediction on that target. Therefore, for a fair comparison on the Wildtrack dataset, we apply a mask on the predicted BEV map for all the previous methods, specifically, we define an area where the label omission has a high occurrence rate (usually refers to the area that nears the edge of the square), and we simply ignore the prediction results in this area to avoid *'fake'* high false positive rate, the shape of the mask is shown in Fig. 2.

We further analyze the source of these unlabelled targets. As shown in the BEV map of Fig. 2, despite few errors made by human annotators, most of the ambiguities occur around the top edge of the detection square due to trivial camera calibration inaccuracy and the special architectural structure of the square, i.e., the stairs occurred at the edge of the square magnify the calibration error. As demonstrated in Fig. 1H, the defined detection plane edge colored in purple is slightly shifted from the actual edge of the plane. Coincidentally, a stair occurs at the edge of the square, which drastically enlarges the disparity between the actual edge and the shifted one by adding an extra distance in Z axis, as demonstrated in Fig. 4. In this figure, the person (or the person's lower

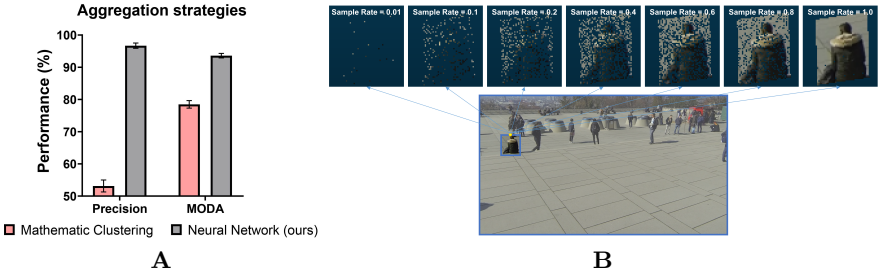

Fig. 3: (A) Comparing different aggregation strategies after projection. (B) The visualization of different point clouds sampling rates.

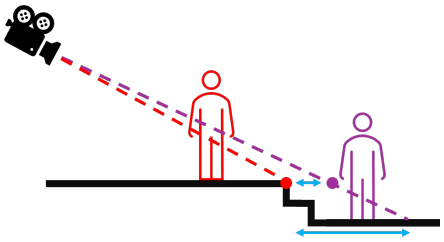

Fig. 4: A side view of the detection plane and the stairs occurred at its edge shown in Fig. 1H. This demonstrates how trivial calibration error leads to huge edge shifting of the detection plane in the Wildtrack dataset. The red point represents the actual edge of the detection plane, and the purple point stands for the erroneous edge calculated according to the camera calibration file.

leg) colored in purple is counted as a target standing inside the detection plane in our proposed system when seeing from particular views. In fact, all previous methods that utilize camera calibrations suffer from this problem to a specific extent on the Wildtrack dataset. While with the mask, these ambiguities are avoided.

We argue that the evaluation process remains valid with this mask since the mask only covers the areas near the edge of the detection area where the crowdedness and occlusion of the pedestrians are relatively low compared with the center area. Thus, the performance of the algorithms on localizing targets under crowdedness and occlusion can be evaluated as equally as on the origin Wildtrack dataset. All the previous methods report higher performance scores after masking.

2 Different point clouds sampling rate of ROI

To clearly demonstrate the sample rate shown in Fig. 7, we visualize the results of one of the detection regions in 3D space. As Fig. 3B shows, the first row of the figure shows that we take random sampling operation. The sample rate between

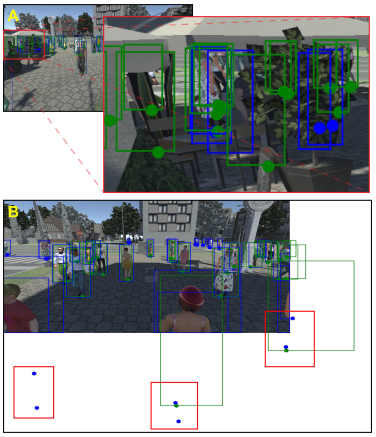

Fig. 5: Failure cases on the MultiviewX dataset. Green boxes and dot points are ground truth and blue boxes and dot points are detection results. Failure detections are caused by (A) severe occlusion and (B) the absence of pedestrians' feet.

0.4 and 0.6 not only represents sufficient appearance features of the target but also reduces the storage of the point clouds to a certain extent. In our proposed pipeline, we set 0.5 as the default sample rate.

3 Benefit of neural network for aggregation

To highlight the significance of our neural network-based point clouds aggregation procedure, we compare our method with the clustering pipeline proposed in [3]. To ensure the fairness of the experiment, we use the standing point as the position feature and the high-dimensional re-ID feature as the appearance feature. We cluster these high-dimensional features using the same clique-based clustering method introduced in [3]. The performance is shown in Fig. 3(A) well illustrates the efficiency of adopting neural networks to aggregate point clouds.

4 Depth estimation using ray tracing

Depth of localized ROI is essential for modeling cardboard humans, due to the lack of depth value labels, we adopt the ray tracing ([1]) technique to calculate the depth for each ROI localization result, namely each bounding box region. For each pedestrian detection result, we calculate the depth of the head and estimated standing point, and fill the rest of the area with interpolated depth values. With the calculated depth, we can project the 2D ROI localization results back into the 3D space to form 3D cardboards.

Given ray tracing Eq. 1, we define the standing point as $P_{\text{standpoint}} = [P^{\text{s}}_{\text{x}}, P^{\text{s}}_{\text{y}}, P^{\text{s}}_{\text{z}}]$, and head point $P_{\text{head}} = [P^{\text{h}}_{\text{x}}, P^{\text{h}}_{\text{y}}, P^{\text{h}}_{\text{z}}]$.

Depth of the standing point We first calculate the 3D coordinate of the pedestrian standing point. For each standing point, we further define the camera 3D center as $O = [O_{\mathrm{x}}, O_{\mathrm{y}}, O_{\mathrm{z}}]$, the direction of the ray direction $D_{\mathrm{standpoint}}$ from the camera center to the standing point as $D_{\mathrm{standpoint}} = [D^{\mathrm{s}}_{\mathrm{x}}, D^{\mathrm{s}}_{\mathrm{y}}, D^{\mathrm{s}}_{\mathrm{z}}]$, and the distance between the camera center and standing point on the object as t. The ray tracing formula is denoted as:

$$\begin{bmatrix} P^{\mathrm{s}}_{\mathrm{x}} \\ P^{\mathrm{s}}_{\mathrm{y}} \\ P^{\mathrm{s}}_{\mathrm{z}} \end{bmatrix} = \begin{bmatrix} O_{\mathrm{x}} \\ O_{\mathrm{y}} \\ O_{\mathrm{z}} \end{bmatrix} + t \begin{bmatrix} D^{\mathrm{s}}_{\mathrm{x}} \\ D^{\mathrm{s}}_{\mathrm{y}} \\ D^{\mathrm{s}}_{\mathrm{z}} \end{bmatrix} \quad \text{i.e.} \begin{cases} P^{\mathrm{s}}_{\mathrm{x}} = O_{\mathrm{x}} + tD^{\mathrm{s}}_{\mathrm{x}} \\ P^{\mathrm{s}}_{\mathrm{y}} = O_{\mathrm{y}} + tD^{\mathrm{s}}_{\mathrm{y}} \\ P^{\mathrm{s}}_{\mathrm{z}} = O_{\mathrm{z}} + tD^{\mathrm{s}}_{\mathrm{z}} \end{cases} \tag{1}$$

Given the premise that the standing point is on the ground plane, where $Z = 0$, we have $P^{\mathrm{s}}_{\mathrm{z}} = 0$:

$$O_{\mathrm{z}} + tD^{\mathrm{s}}_{\mathrm{z}} = 0 \tag{2}$$

hence,

$$t = -\frac{O_{\mathrm{z}}}{D^{\mathrm{s}}_{\mathrm{z}}} \tag{3}$$

substitute t into Eq. (1):

$$\begin{cases} P^{\mathrm{s}}_{\mathrm{x}} = O_{\mathrm{x}} - \frac{O_z}{D_z} D^{\mathrm{s}}_{\mathrm{x}} \\ P^{\mathrm{s}}_{\mathrm{y}} = O_{\mathrm{y}} - \frac{O_z}{D_z} D^{\mathrm{s}}_{\mathrm{y}} \\ P^{\mathrm{s}}_{\mathrm{z}} = 0 \end{cases} \tag{4}$$

Now, to determine $P^{\mathrm{s}}_{\mathrm{x}}$ and $P^{\mathrm{s}}_{\mathrm{y}}$, we need to further explore the camera position O and ray direction $D_{\mathrm{standpoint}}$. The camera position O in the world coordinate system is determined with:

$$O = -R^T T \tag{5}$$

where R and T are the rotation matrix and translation matrix that map the object from the world coordinate to the camera coordinate, and $-R^T T$ is a 3×1 matrix.

Next, to get the direction from the camera center to the standing point $D_{\mathrm{standpoint}}$ of the ray, we need to find the correlation between the ray and the world coordinates systems. We first determine the ray direction inside the camera, which is to define the ray that starts from the camera origin to the pixel coordination system, and furthermore, we translate the ray from the pixel coordinate system to the camera coordinates system using the intrinsic matrix, and finally, we project the origin-to-camera ray to an origin-to-world one. Assume the standing point in the pixel coordinate system is marked as $[u^{\mathrm{s}}, v^{\mathrm{s}}]$ and the camera has the intrinsic matrix k as:

$$k = \begin{bmatrix} fx & 0 & cx \\ 0 & fy & cy \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

where fx, fy represent the focal length in x, y direction, cx, cy are the translation between camera coordinates systems and pixel coordinate systems. We could now define the standing point in the camera coordinate system $[X^{\mathrm{s}}_{\mathrm{cam}}, Y^{\mathrm{s}}_{\mathrm{cam}}, Z^{\mathrm{s}}_{\mathrm{cam}}]$ with the following derivation:

$$Z^{\mathrm{s}}_{\mathrm{cam}} \begin{bmatrix} u^{\mathrm{s}} \\ v^{\mathrm{s}} \\ 1 \end{bmatrix} = [K|0] \begin{bmatrix} X^{\mathrm{s}}_{\mathrm{cam}} \\ Y^{\mathrm{s}}_{\mathrm{cam}} \\ Z^{\mathrm{s}}_{\mathrm{cam}} \\ 1 \end{bmatrix} \tag{7}$$

$$= \begin{bmatrix} fx & 0 & cx & 0 \\ 0 & fy & cy & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X^{\mathrm{s}}_{\mathrm{cam}} \\ Y^{\mathrm{s}}_{\mathrm{cam}} \\ Z^{\mathrm{s}}_{\mathrm{cam}} \\ 1 \end{bmatrix} \tag{8}$$

$$\Rightarrow \begin{cases} Z^{\mathrm{s}}_{\mathrm{cam}} u^{\mathrm{s}} = X^{\mathrm{s}}_{\mathrm{cam}} fx + Z^{\mathrm{s}}_{\mathrm{cam}} cx \\ Z^{\mathrm{s}}_{\mathrm{cam}} v^{\mathrm{s}} = Y^{\mathrm{s}}_{\mathrm{cam}} fy \ + Z^{\mathrm{s}}_{\mathrm{cam}} cy \\ Z^{\mathrm{s}}_{\mathrm{cam}} = Z^{\mathrm{s}}_{\mathrm{cam}} \end{cases} \tag{9}$$

$$\Rightarrow \begin{cases} X^{\mathrm{s}}_{cam} = \frac{Z^{\mathrm{s}}_{cam}(u^{\mathrm{s}} - cx)}{fx} \\ Y^{\mathrm{s}}_{cam} = \frac{Z^{\mathrm{s}}_{cam}(v^{\mathrm{s}} - cy)}{fy} \\ Z^{\mathrm{s}}_{\mathrm{cam}} = Z^{\mathrm{s}}_{\mathrm{cam}} \end{cases} \tag{10}$$

We observe that $Z^{\mathrm{s}}_{\mathrm{cam}}$ is still unknown. However, since the calculation target is the ray direction, which is not affected by the length of the ray, we divide the $Z^{\mathrm{s}}_{\mathrm{cam}}$ in each line on the right of the equations to obtain the normalized origin-to-camera direction D_{o2c}, denoted as:

$$D_{\mathrm{o2c}} = \begin{bmatrix} X^{\mathrm{s}}_{cam} \\ Y^{\mathrm{s}}_{cam} \\ Z^{\mathrm{s}}_{cam} \end{bmatrix} = \begin{bmatrix} \frac{u^{\mathrm{s}} - cx}{fx} \\ \frac{v^{\mathrm{s}} - cy}{fy} \\ 1 \end{bmatrix} \tag{11}$$

Finally, we project the origin-to-camera ray direction to the origin-to-world direction using the inverse of rotation matrix M ([5]). Therefore the final origin-to-world direction $D_{\mathrm{standpoint}}$ is represented as:

$$D_{\mathrm{standpoint}} = \begin{bmatrix} D^s_{\mathrm{x}} \\ D^s_{\mathrm{y}} \\ D^s_{\mathrm{z}} \end{bmatrix} = D_{o2c} \cdot M^{-1} = \begin{bmatrix} \frac{u^{\mathrm{s}} - cx}{fx} \\ \frac{v^{\mathrm{s}} - cy}{fy} \\ 1 \end{bmatrix} \cdot M^{-1} \tag{12}$$

With $O, D_{\mathrm{standpoint}}, P^{\mathrm{s}}_{\mathrm{z}}, P^{\mathrm{s}}_{\mathrm{y}}$ in Eq. (1) can be determined. We now know the exact 3D world coordinate $P_{\mathrm{standpoint}} = [P^s_{\mathrm{x}}, P^s_{\mathrm{y}}, P^s_{\mathrm{z}}]^{\mathrm{T}}$ of the standing point. Lastly, we project the standing point $P_{\mathrm{standpoint}}$ to the camera coordinate system using the extrinsic matrix to obtain $[X^{\mathrm{s}}_{cam}, Y^{\mathrm{s}}_{cam}, Z^{\mathrm{s}}_{cam}]$ in which $Z^{\mathrm{s}}_{\mathrm{cam}}$ is not divided, and the $Z^{\mathrm{s}}_{\mathrm{cam}}$ value is the *depth* of the standing point.

Depth of the head point To calculate the 3D world coordinate of the head, we leverage the assumption that the head and standing point of the same pedestrian lie on the same vertical line, therefore, both head and standing point share the same P_{x} and P_{y}. The actual height in the real world is the calculation target, and we regard the top of each bounding box as the head of the pedestrian in each 2D image. Therefore. Recall the definition in Sec. 4, we have:

$$\begin{bmatrix} P_{\mathrm{x}}^{\mathrm{s}} \\ P_{\mathrm{y}}^{\mathrm{s}} \\ P_{\mathrm{z}}^{\mathrm{s}} \end{bmatrix} = \begin{bmatrix} O_{\mathrm{x}} \\ O_{\mathrm{y}} \\ O_{\mathrm{z}} \end{bmatrix} + t \begin{bmatrix} D_{\mathrm{x}}^{\mathrm{h}} \\ D_{\mathrm{y}}^{\mathrm{h}} \\ D_{\mathrm{z}}^{\mathrm{h}} \end{bmatrix} \text{ i.e. } \begin{cases} P_{\mathrm{x}}^{\mathrm{h}} = O_{\mathrm{x}} + tD_{\mathrm{x}}^{\mathrm{h}} \\ P_{\mathrm{y}}^{\mathrm{h}} = O_{\mathrm{y}} + tD_{\mathrm{y}}^{\mathrm{h}} \\ P_{\mathrm{z}}^{\mathrm{h}} = O_{\mathrm{z}} + tD_{\mathrm{z}}^{\mathrm{h}} \end{cases} \tag{13}$$

In this case, $P_{\mathrm{x}}^{\mathrm{h}}, O_{\mathrm{x}}, O_{\mathrm{y}}$ are known and $D_{\mathrm{x}}^{\mathrm{h}}$ can be calculated according to Eq. (10) - Eq. (12). Thus, the only unknown t can be calculated by substituting $P_{\mathrm{x}}^{\mathrm{h}} = P_{\mathrm{x}}^{\mathrm{s}}$ into Eq. (3). Hence, with all the calculated variables, $P_{\mathrm{z}}^{\mathrm{h}}$ is obtained. The depth of the head is then acquired following the same steps as in the depth calculation for the standing point.

5 Point clouds generation for each ROI area

We fill each localized ROI region with interpolated depth values according to the calculated depth of the standing point and head. For each pixel in the localized ROI area, the 3D coordinate(point cloud) $P_{\mathrm{all}} = [P_{\mathrm{x}}^{\mathrm{a}}, P_{\mathrm{x}}^{\mathrm{a}}, P_{\mathrm{x}}^{\mathrm{a}}]$ is obtained with the following formula:

$$\begin{bmatrix} P_{\mathrm{x}}^{\mathrm{all}} \\ P_{\mathrm{y}}^{\mathrm{all}} \\ P_{\mathrm{z}}^{\mathrm{all}} \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}_{4\times4}^{-1} \cdot \begin{bmatrix} X_{cam}^{\mathrm{all}} \\ Y_{cam}^{\mathrm{all}} \\ Z_{cam}^{\mathrm{all}} \\ 1 \end{bmatrix} \tag{14}$$

Where

$$\begin{cases} X_{cam}^{\mathrm{all}} = \frac{Z_{cam}^{\mathrm{all}}(u^{\mathrm{all}} - cx)}{fx} \\ Y_{cam}^{\mathrm{all}} = \frac{Z_{cam}^{\mathrm{all}}(v^{\mathrm{all}} - cy)}{fy} \\ Z_{\mathrm{cam}}^{\mathrm{all}} = \text{interpolated depth} \end{cases} \tag{15}$$

R, t is the rotation and translation matrix of the camera.

We now have the 3D world coordinates(point clouds) for every pixel in a localized ROI area.

6 The differences in the supplemented dataset.

As shown in Table 1, the additional annotations are highlighted in the Wildtrack+ and MultiviewX+ datasets. In the supplemented dataset, we only added 2D annotations outside the playground (indicated by the blue numbers) without altering the camera information, ensuring a fair comparison.

Dataset Type	Dataset	Number of BBox (per frame)
Real-World Dataset	Wildtrack	15
	Wildtrack+ (*ours*)	**49** (+34)
Synthesis Dataset	MultiviewX	26
	MultiviewX+ (*ours*)	**55** (+29)

Table 1: Average number of Bounding Boxes (BBox) per frame across different datasets. Blue numbers indicate the additional BBoxes in our enhanced datasets.

7 Limitation and future work

Even though the accuracy drops slightly on the Multiview and MultiviewX+ datasets with more severe occlusion, our methods still have very competitive performance. Our analysis suggests the reason is inaccurate standing point detection, causing the cardboard human to be poorly constructed. On the one hand, compared to the Wildtrack dataset, cameras in MultiviewX are placed lower (1.8 meters in height), which leads to the absence of pedestrian feet when they are close to the cameras. On the other hand, a high level of occlusions in MultiviewX also results in missing pedestrian feet. Failure cases are shown in Fig. 5 in the supplementary materials. Although our method may encounter difficulties due to the potential invisibility of the standing point, it is noteworthy that the trained estimator is still able to make reasonable estimations of the standing location, even when it lies outside the boundaries of the captured image. Additionally, the missing or truncated standing point can be compensated by other camera views.

There are multiple possible directions along the track of our research in future works: First, as discussed above, our method relies too much on pedestrian detection and keypoint detection. We speculate this problem can be alleviated if more accurate mapping of the human body from 2D to 3D could be established. In this regard, existing works in 3D human modeling offer a valuable source of ideas [2,4]. Second, instead of modeling the scene with explicit point cloud representation, it is possible to model the entire 3D space with implicit representation (NeRF-base methods [5]). This paper offers a brand-new insight that the coarse but correct reconstruction of scenes can effectively integrate multiview clues and accurately locate targets. We hope that our findings will motivate the progress of multiview detection.

References

1. Appel, A.: Some techniques for shading machine renderings of solids. In: Proceedings of the April 30–May 2, 1968, spring joint computer conference. pp. 37–45 (1968)
2. Güler, R.A., Neverova, N., Kokkinos, I.: Densepose: Dense human pose estimation in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7297–7306 (2018)

3. Lima, J.P., Roberto, R., Figueiredo, L., Simoes, F., Teichrieb, V.: Generalizable multi-camera 3d pedestrian detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 1232–1240 (June 2021)
4. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: Smpl: A skinned multi-person linear model. ACM transactions on graphics (TOG) **34**(6), 1–16 (2015)
5. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European conference on computer vision. pp. 405–421. Springer (2020)

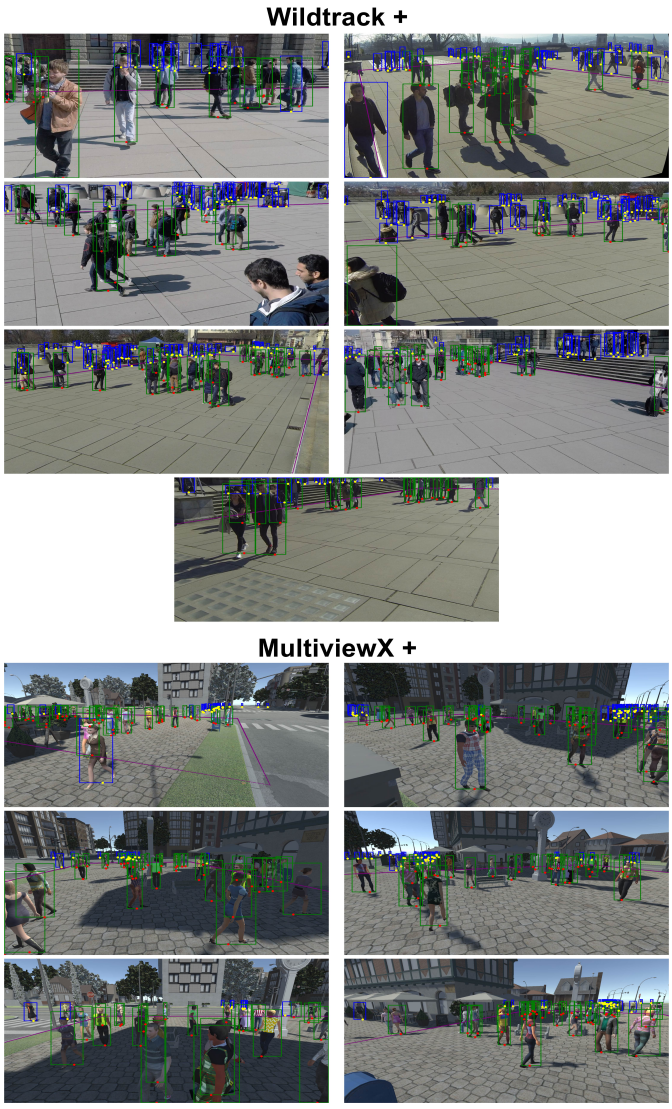

Fig. 6: Label visualization of the Wildtrack+ and MultiviewX+ dataset. For official Wildtrack and MultiviewX, we use green bounding box and red dot point to visualize the region of interest (ROI). For the Wildtrack+ and MultiviewX+ we proposed, we additionally annotate the pedestrians outside the detection area (bounded by purple lines). The supplementary labels are painted as blue bounding boxes and yellow dot point.

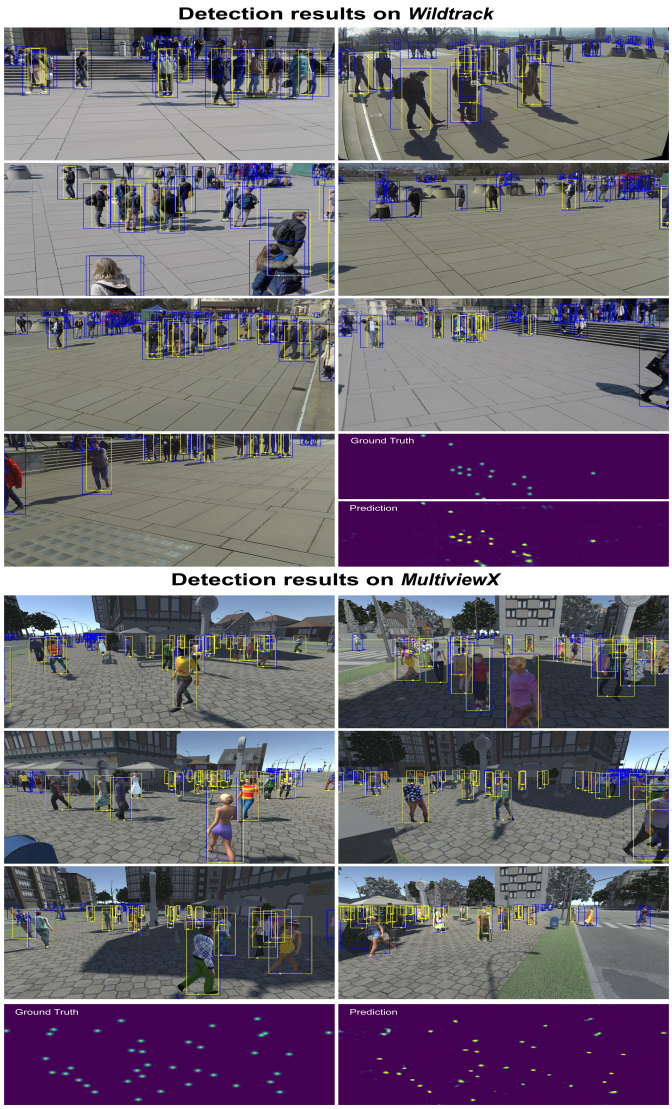

Fig. 7: Detection results on Wildtrack and MultiviewX dataset. Ground truth including standing points and bounding boxes are labeled by yellow color. And the detection results are labeled by blue color.