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

Supplementary Material

A. Perspective Transformation

In this section, we formulate the perspective transformation [32] that we used in the detection network of MVTrajecter. As described in Sec. 3.4, the perspective transformation projects image feature maps of each view into a ground plane. This perspective transformation is defined using 3D locations (x, y, z) and 2D image pixel coordinates (u, v), as:

$$\gamma \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = P_\theta \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = A[R|T] \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} & \theta_{14} \\ \theta_{21} & \theta_{22} & \theta_{23} & \theta_{24} \\ \theta_{31} & \theta_{32} & \theta_{33} & \theta_{34} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}, \tag{12}$$

where γ is a real-valued scaling factor, and P_θ is the 3×4 transformation matrix calculated using the 3×3 intrinsic camera parameter matrix A and the 3×4 extrinsic camera parameter matrix $[R|T]$. Specifically, R represents the rotation, and T represents the translation. By setting $z = 0$, we can retrieve the correspondence between the image pixel (u, v) and the ground plane coordinates (x, y), as:

$$\gamma \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = P_{\theta,0} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{14} \\ \theta_{21} & \theta_{22} & \theta_{24} \\ \theta_{31} & \theta_{32} & \theta_{34} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \tag{13}$$

where $P_{\theta,0}$ is the 3×3 transformation matrix that is P_θ with the third column canceled. This allows the projection of image features onto the ground plane.

B. Datasets Details

In this section, we describe the datasets that we used in the experiments in detail. The statistics of each dataset are summarized in Table 9, and sample frames of each dataset are provided in Fig. 4.

Wildtrack [9] This is a real-world multi-view dataset consisting of one multi-view video sequence captured with 7 cameras. The multi-view video comprises 400 frames at a frame rate of 2 fps and covers a total of 200 seconds. Each view video is recorded at 1080×1920 resolution. A total of 313 pedestrians are contained in the multi-view video, and 20 pedestrians are contained in each frame on average. This dataset covers a $12 \times 36\,\mathrm{m}$ region quantized into a 480×1440 grid using square grid cells of $2.5\,\mathrm{cm}^2$. Each grid cell is captured by 3.74 cameras on average. Wildtrack splits them into the first 360 frames (180 seconds) for training and the remaining 40 frames (20 seconds) for testing.

MultiviewX [32] This is a synthetic dataset created using the Unity engine and captures a more crowded scene than Wildtrack. MultiviewX consists of one multi-view video sequence captured with 6 cameras. The multi-view video comprises 400 frames at a frame rate of 2 fps and covers a total of 200 seconds. Each view video is recorded at 1080×1920 resolution. A total of 350 pedestrians are contained in the multi-view video, and 40 pedestrians are contained in each frame on average. This dataset covers a $16 \times 25\,\mathrm{m}$ region quantized into a 640×1000 grid using square grid cells of $2.5\,\mathrm{cm}^2$, which is slightly smaller than Wildtrack. Each grid cell is captured by 4.41 cameras on average. MultiviewX also splits them into the first 360 frames (180 seconds) for training and the remaining 40 frames (20 seconds) for testing.

GMVD [65] This is a large-scale synthetic dataset that includes 7 scenes with varying numbers of cameras and camera layouts. Of the 7 scenes, 6 scenes are captured using Grand Theft Auto (GTA), and the remaining 1 scene using the Unity engine. In addition, each scene also contains multiple multi-view video sequences with different environmental conditions including time and weather. In total, GMVD comprises 53 multi-view video sequences and 5995 frames. Each multi-view video sequence is captured at a frame rate of 2 fps, and each view video is recorded at 1080×1920 resolution. A total of 2800 pedestrians are contained in GMVD, and each sequence contains 20–40 pedestrians in each frame on average. Each scene covers a different size of the region and quantizes the region into a grid using square grid cells of $2.5\,\mathrm{cm}^2$. Each grid cell of each sequence is captured by 2.8–6.4 cameras on average. Details of the statistics for each scene are provided in Table 9. GMVD splits them into 6 scenes with 43 sequences and 4983 frames for training and 1 scene (the bottom row in Table 9) with 10 sequences and 1012 frames for testing. To make testing difficult, this testing split also contains two different camera layouts: one with 6 cameras and the other with 8 cameras.

Dataset	Scenes	Sequences	Frames	Cameras	Covered region	Grid size	Pedestrians
Wildtrack [9]	Real	1	400	7	$12 \times 36\,\mathrm{m}$	480×1440	20 / frame
MultiviewX [32]	Unity	1	400	6	$16 \times 25\,\mathrm{m}$	640×1000	40 / frame
GMVD [65]	Unity	2	723	6	$16 \times 25\,\mathrm{m}$	640×1000	40 / frame
	GTA	10	1034	5	$20 \times 30\,\mathrm{m}$	800×1200	20 / frame
	GTA	10	1000	3	$30 \times 12\,\mathrm{m}$	1200×480	30 / frame
	GTA	10	1014	5	$25 \times 25\,\mathrm{m}$	1000×1000	30 / frame
	GTA	1	182	5	$28 \times 27\,\mathrm{m}$	1120×1080	20 / frame
	GTA	10	1030	7	$33 \times 31\,\mathrm{m}$	1320×1240	30 / frame
	GTA	10	1012	6, 8	$29 \times 19\,\mathrm{m}$	1160×760	30 / frame

Table 9. Statistics of Wildtrack, MultiviewX, and GMVD. The scene in the bottom row of GMVD contains sequences of the same scene captured by 6 or 8 cameras. In the experiments on GMVD, we utilized the scene in the bottom row for testing and the other scene for training.

Method	MODA↑	MODP↑	Recall↑	Precision↑
EarlyBird [62]	73.2	78.5	77.4	**94.9**
MVFlow [20]	72.4	78.2	77.7	93.5
TrackTacular [63]	68.0	**79.3**	72.9	93.7
MVTr [71]	72.7	78.9	79.0	92.6
Ours	**74.4**	79.2	**79.4**	94.0

Table 10. Comparison of detection results with previous end-to-end MVPT methods on GMVD.

	MODA↑	MODP↑	Recall↑	Precision↑
MVFP [2]	73.3	76.5	79.2	93.0
Ours w/ $\mathcal{L}_{\mathrm{det}}$ only	71.6	78.0	78.2	92.3
Ours w/ $\mathcal{L}_{\mathrm{all}}$	**74.4**	**79.2**	**79.4**	**94.0**

Table 11. Effect of $\mathcal{L}_{\mathrm{all}}$ on detection performance and comparison with the state-of-the-art MVPD method.

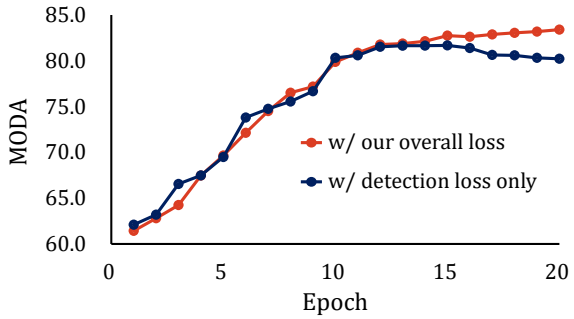

Figure 3. MODA on the validation data of models trained with our overall loss $\mathcal{L}_{\mathrm{all}}$ and with the detection loss $\mathcal{L}_{\mathrm{det}}$ only.

C. Implementation Details

During training, we applied random resizing and cropping to multi-view image sequences as the data augmentation following [31, 62]. We applied the same augmentation to all multi-view images in one multi-view image sequence within the window. We set the scale range of the resizing and cropping to $[0.8, 1.2]$. We also applied dropout at the rate of 0.1 to the attention layer. For the Adam optimizer [42], we set the optimizer momentum to $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We did not use the weight decay. We set the radius of the Gaussian kernel to 8 pixels when generating smoothed ground truth occupancy maps.

D. Analysis of Detection Results

In this section, we evaluated and analyzed the detection results of MVTrajecter. Unless otherwise stated, all experiments were conducted on GMVD.

Evaluation Metrics. Following previous studies [32, 62, 65], we used four standard metrics provided by Chavdarova et al. [9] and Kasturi et al. [37]: Multiple Object Detection Accuracy (MODA), Multiple Object Detection Precision (MODP), recall, and precision. A detected pedestrian was classified as a true positive if its distance from the ground truth was within 0.5 meters. We used MODA as the primary performance indicator following previous studies [31, 32, 62, 63, 65].

Comparison of detection results. We compared the detection results of MVTrajecter with EarlyBird [62], MVFlow [20], TrackTacular [63], and MVTr [71], which are state-of-the-art end-to-end MVPT methods and were compared in Sec. 4.4 for their tracking performances. Table 10 shows the comparison results. MVTrajecter achieved the best MODA and recall and the second-best MODP and precision. This indicates that MVTrajecter is superior to the other methods in terms of detection. Compared to the other methods, our overall loss $\mathcal{L}_{\mathrm{all}}$ imposed large temporal constraints on MVTrajecter, which is presumably what suppressed the overfitting in the detection and improved its detection performance.

Effect of $\mathcal{L}_{\mathrm{all}}$ on the detection. We investigated the effect of our overall loss $\mathcal{L}_{\mathrm{all}}$ on the detection performance. Table 11 shows a comparison of the detection performance between MVFP [2], which is the state-of-the-art detection

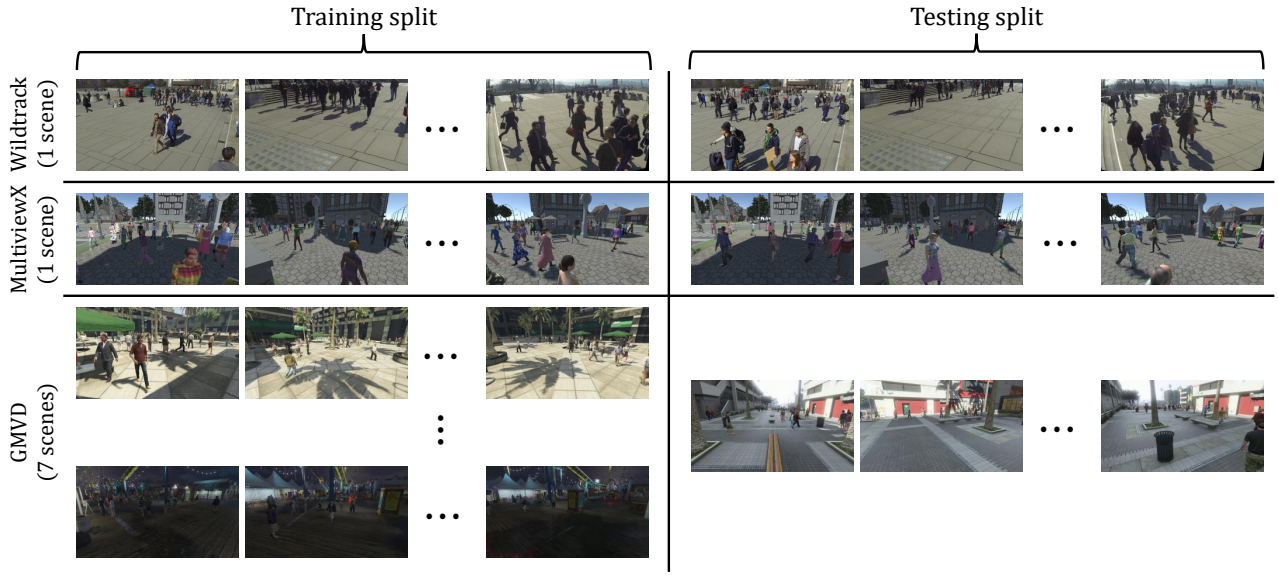

Figure 4. Sample frames of Wildtrack [9], MultiviewX [32], and GMVD [65]. Top to bottom rows represent Wildtrack, MultiviewX, and GMVD, respectively. Left and right columns show the training split and testing split of each dataset, respectively. While GMVD contains 7 scenes, we visualize only 3 of them here.

	IDF1↑	MOTA↑	MOTP↑	MT↑	ML↓
Two models	76.5	77.4	87.7	65.4	8.7
Two branches	**77.2**	**78.1**	87.7	**66.0**	8.7

Table 12. Effect of having two branches in one model.

α	IDF1↑	MOTA↑	MOTP↑	MT↑	ML↓
0.97	73.9	73.3	**90.4**	61.8	0.0
0.98	**74.8**	**75.8**	90.3	61.8	0.0
0.99	74.6	**75.8**	90.1	61.8	0.0

Table 13. Impact of weighting parameter α on tracking performances for the validation data.

	IDF1↑	MOTA↑	MOTP↑	MT↑	ML↓
EMA	74.8	76.1	87.0	64.2	8.9
TAC	**77.2**	**78.1**	**87.7**	**66.0**	**8.7**

Table 14. Comparison between TAC and the exponential moving average (EMA) appearance feature aggregation.

Pooling	IDF1↑	MOTA↑	MOTP↑	MT↑	ML↓
Mean	76.3	77.5	87.6	65.2	8.8
Max	**77.2**	**78.1**	**87.7**	**66.0**	**8.7**

Table 15. Comparison between mean pooling and max pooling in the detection network.

method, and MVTrajecter trained with $\mathcal{L}_{\text{all}}$ and with $\mathcal{L}_{\text{det}}$ only. When MVTrajecter was trained only with $\mathcal{L}_{\text{det}}$, only its detection network was optimized. Using $\mathcal{L}_{\text{all}}$ for training greatly improved the detection performance. In addition, surprisingly, MVTrajecter outperformed MVFP even though MVTrajecter did not utilize the complex projections and detection modules used in MVFP. Figure 3 shows the MODA for the validation data at each epoch of the models trained with $\mathcal{L}_{\text{all}}$ and with $\mathcal{L}_{\text{det}}$ only. While MODA of the model trained only with $\mathcal{L}_{\text{det}}$ decreased from epoch 12, MODA of the model trained with $\mathcal{L}_{\text{all}}$ continuously increased. This indicates that $\mathcal{L}_{\text{all}}$ suppressed the overfitting of the detection.

E. Additional Ablation Study

In this section, we conducted ablation studies to investigate the effect of having two branches in one model, the impact of weighting parameter α, the value ranges of C_{TMC} and C_{TAC}, the comparison of TAC with EMA aggregation, and the pooling choice. Unless otherwise stated, all experiments were conducted on GMVD.

Advantage of having two branches. MVTrajecter has the motion and appearance branches in one model, as shown in Fig. 2. To verify the effectiveness of having two branches in one model, we compared it with the case of tracking performed by two models (*i.e.*, one has the motion branch and the other has the appearance branch). According to the comparison results in Table 12, having two branches achieved a slightly better performance. This indicates the advantage of simultaneously modeling both motion and appearance in an end-to-end manner.

Impact of weighting parameter α between TMC and TAC. As described in Sec. 4.3, we tuned α on the validation data. Here, we show the impact of α on the tracking

Method	IDF1↑	MOTA↑	MOTP↑
w/ PE in motion branch	76.9	78.0	**87.9**
w/ PE in appearance branch	76.7	77.9	87.6
w/o PE (Ours)	**77.2**	**78.1**	87.7

Table 16. Effect of positional embeddings. "PE" means positional embeddings.

Method	HOTA↑
EarlyBird [62]	64.5
MVFlow [20]	66.1
TrackTacular [63]	68.2
MVTr [71]	70.7
Ours	**75.0**

Table 17. Comparison with previous methods on the HOTA metric when all models used their own detection results for tracking.

K	FPS
Detection network	6.4
1	5.9
3	5.8
5	5.6
7	5.4

Table 18. Effect of past trajectory length K on inference speed.

Method	FPS
EarlyBird [62]	6.0
MVFlow [20]	6.1
TrackTacular [63]	6.2
MVTr [71]	6.1
Ours	5.4

Table 19. Comparison of inference speed with previous methods.

performances for the validation data in Table 13. Increasing α improved MOTA while it worsened MOTP. We therefore set $\alpha = 0.98$ on the basis of these results.

Value ranges of C_{TMC} and C_{TAC}. We investigated the value ranges of C_{TMC} and C_{TAC} during inference to verify the validity of the extreme weighting of α. On GMVD test split and for $K = 7$, the value range of C_{TMC} was from 0.3 to 1.2×10^{3}, and the value range of C_{TAC} was from -6.2 to -4.7×10^{-25}. Because the value ranges of C_{TMC} and C_{TAC} differed greatly, the weighted sum using $\alpha = 0.98$ worked effectively.

Comparison between TAC and EMA appearance feature aggregation. We aggregated appearance features over multiple past timestamps by calculating the probabilities for each timestamp (Eq. 2) and summing the probabilities for all timestamps (Eq. 3) in TAC. In contrast to our approach, some monocular tracking methods aggregate the appearance features by updating the latest appearance features using the exponential moving average (EMA) [1, 17, 18, 67, 73]. We compared TAC with EMA appearance feature aggregation, and the results shown in Table 14 indicate that leveraging TAC outperformed leveraging EMA aggregation. Since EMA aggregation directly fuses the appearance features of pedestrians recognized as identical, we presume that it is more affected by association errors than TAC.

Pooling choice. We performed max pooling to aggregate projected features from multiple views, as described in Sec. 3.4. Another option is to perform mean pooling instead of max pooling. When we compared these two pooling operations, as shown in Table 15, max pooling achieved a better tracking performance. Max pooling extracts the most relevant and informative features from different view perspectives for the subsequent modules, which is presumably what resulted in better tracking performance.

Effect of positional embedding. While we used temporal embeddings (see Sec. 3.4), we did not use positional embeddings because the BEV features implicitly contain the positional information. To justify this, we implemented learnable positional embeddings in the motion and appearance branches, as shown in Table 16. We did not observe the improvement from our original implementation. Therefore, positional embeddings are not crucial for the motion and appearance branches.

Evaluation on HOTA metric. While we did not use Higher Order Tracking Accuracy (HOTA) [43] because comparison methods were not evaluated on it in their original papers, it is one of the important evaluation metrics in the field of monocular tracking. Therefore, we compared the models in Table 1 using the HOTA metric. Table 17 shows the comparison results. Our method also significantly outperformed previous methods on the HOTA metric. This also demonstrates the effectiveness of our proposed method.

F. Analysis of Inference Speed

Effect of past trajectory length K. Introducing more past information inevitably causes a decrease in inference speed. We measured the inference speed of the models in Table 5 and the detection network of our method on an Nvidia A100 GPU. The FPS of the models at $K = 1, 3, 5, 7$ and the detection network are shown in Table 18. Since the detection network occupies most of the runtime, the effect of increasing K on inference speed is very small.

Comparison with previous methods. We also measured the inference speed of the models in Table 1. The

FPS of EarlyBird [62], MVFlow [20], TrackTacular [63], MVTr [71], and ours are shown in Table 19. Our method is slightly slower than the others because it uses more past information. However, this speed gap is very small compared to the tracking performance improvement.

G. Qualitative Results

To visually verify the effectiveness of our MVTrajecter, we visualized the detection and tracking results of MVTrajecter, TrackTacular [63], and the ground truth. Figures 5 and 6 show the comparison of the detection and tracking results on one multi-view video sequence (consisting of 100 timestamps) of GMVD, respectively. In Fig. 5, our MVTrajecter reduced the number of missed detections and false positive detections, which are denoted by red and green circles, respectively. This demonstrates that MVTrajecter is superior to TrackTacular in the detection. In Fig. 6, we can see that MVTrajecter correctly tracked pedestrians that TrackTacular failed to track, as indicated by the red squares. This demonstrates that MVTrajecter is also superior to TrackTacular in tracking. While MVTrajecter improved the detection and tracking performance, it still struggled to detect and track pedestrians near the boundaries of the occupancy maps. Therefore, in order to improve performance further, methods that can accurately handle pedestrians near the boundaries are needed.

To validate the versatility of MVTrajecter, which is not limited to GMVD, we also visualized tracking results on Wildtrack and MultiviewX. Figures 7 and 8 show the comparison of tracking results on Wildtrack and MultiviewX, respectively. In both figures, we can see that even in cases where TrackTacular failed to track pedestrians, MVTrajecter correctly tracked them, as indicated by the red squares. These results demonstrate that MVTrajecter is effective for various datasets.

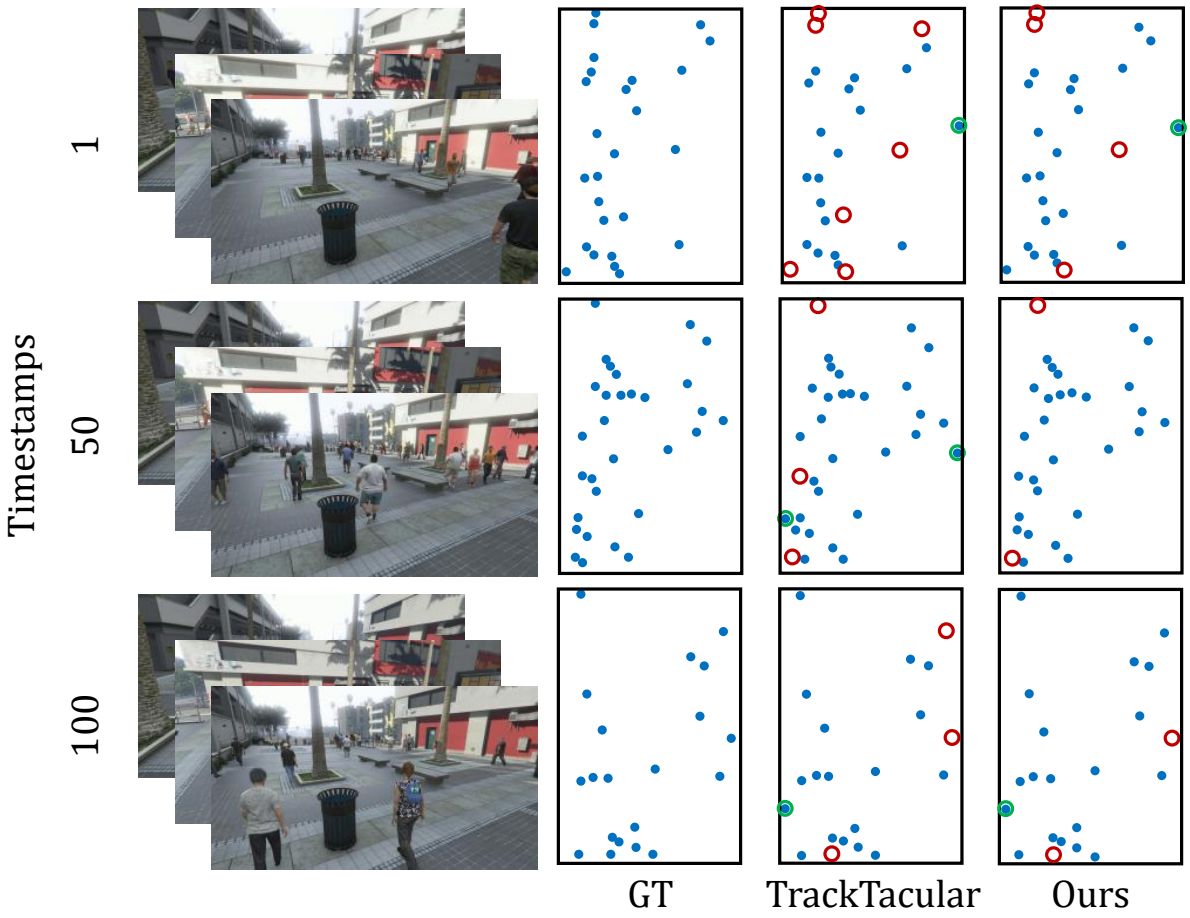

Figure 5. Qualitative comparison of detection results between ground truth (GT), TrackTacular [63], and the proposed MVTrajecter on GMVD. Blue filled circles represent detected pedestrians, red circles represent missed detections, and green circles represent false positives.

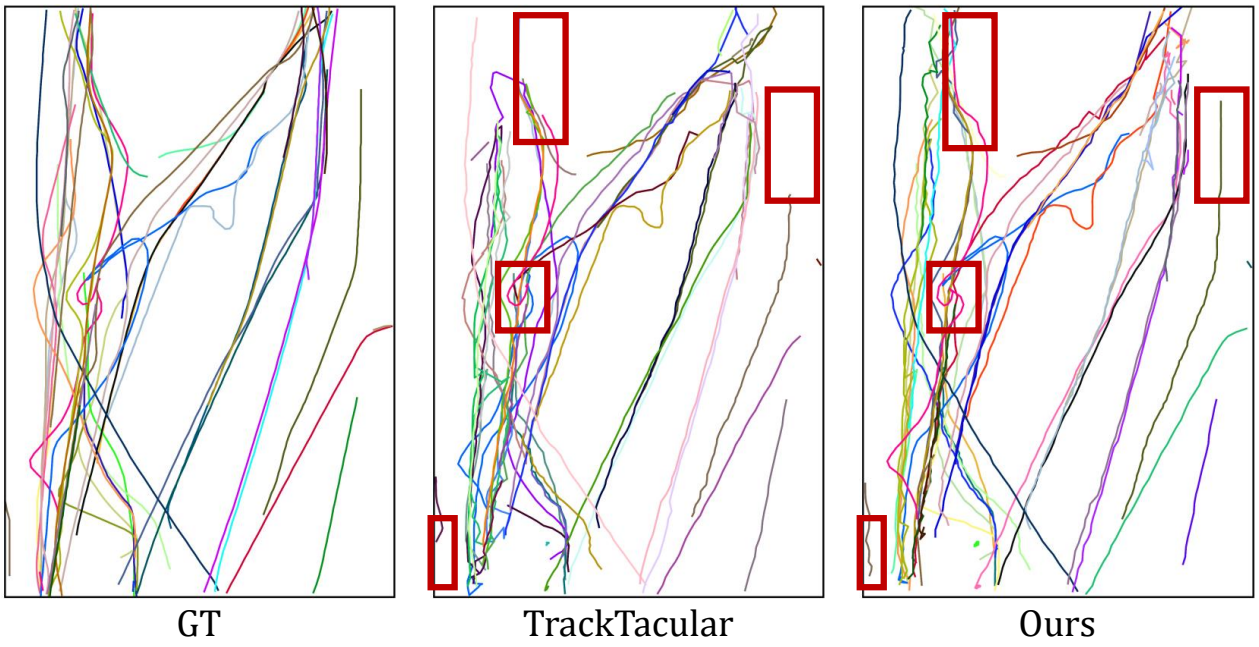

Figure 6. Qualitative comparison of tracking results between ground truth (GT), TrackTacular [63], and the proposed MVTrajecter on GMVD. Each line represents a pedestrian track. Red squares indicate examples of different results between TrackTacular and our MVTrajecter.

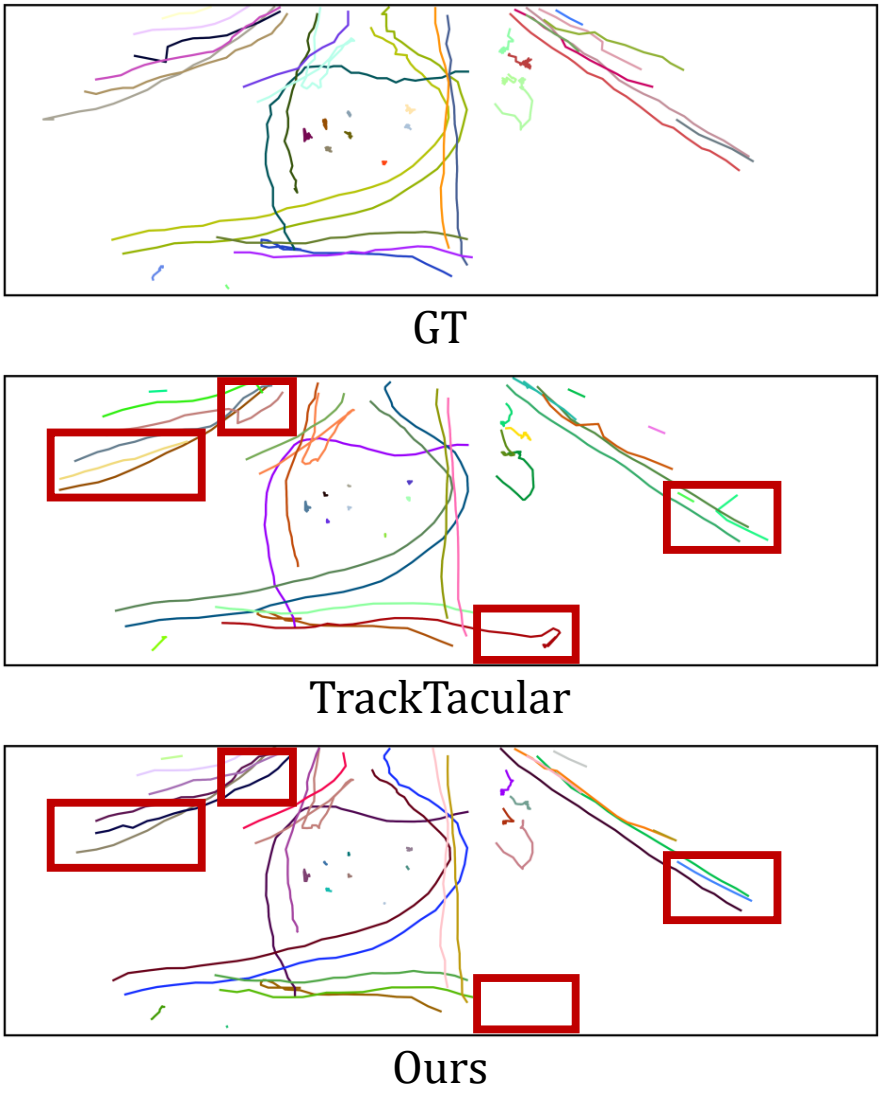

Figure 7. Qualitative comparison of tracking results between ground truth (GT), TrackTacular [63], and the proposed MVTrajecter on Wildtrack. Each line represents a pedestrian track. Red squares indicate examples of different results between TrackTacular and our MVTrajecter.

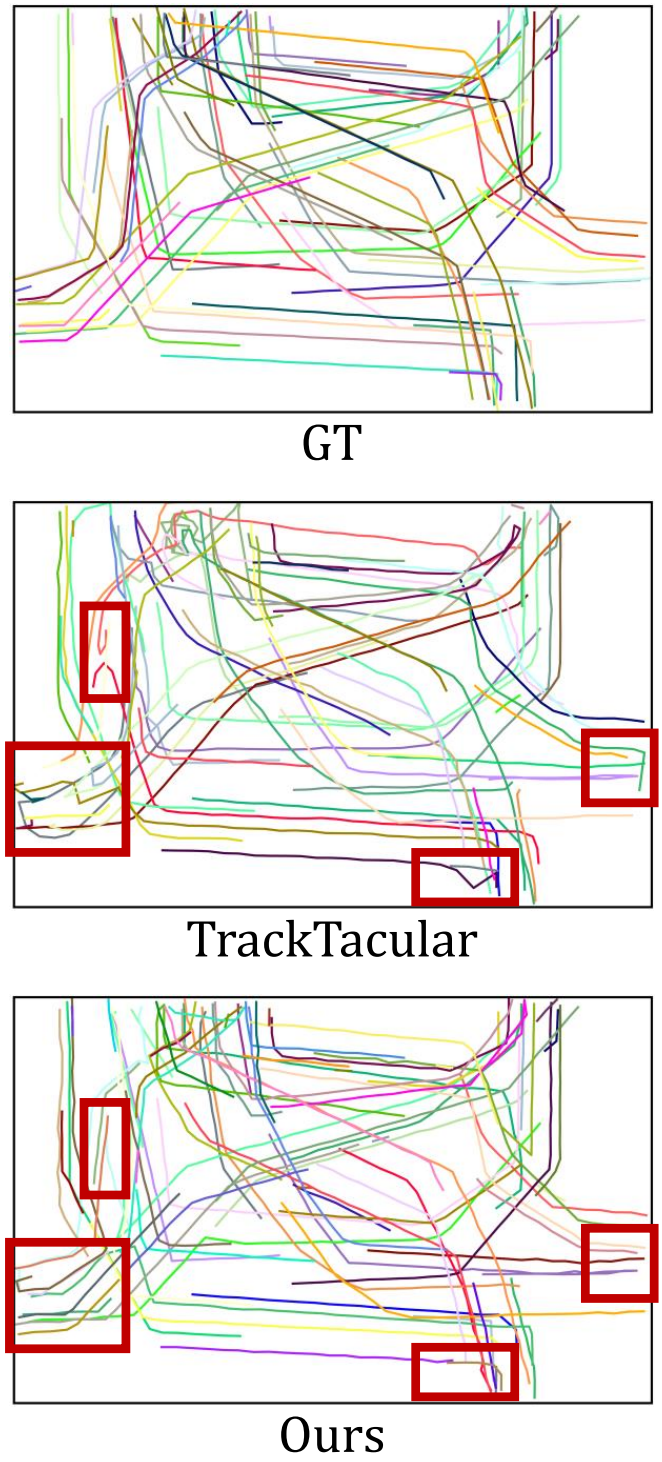

Figure 8. Qualitative comparison of tracking results between ground truth (GT), TrackTacular [63], and the proposed MVTrajecter on MultiviewX. Each line represents a pedestrian track. Red squares indicate examples of different results between TrackTacular and our MVTrajecter.