Supplementary Material of "UTM: A Unified Multiple Object Tracking Model with Identity-Aware Feature Enhancement"

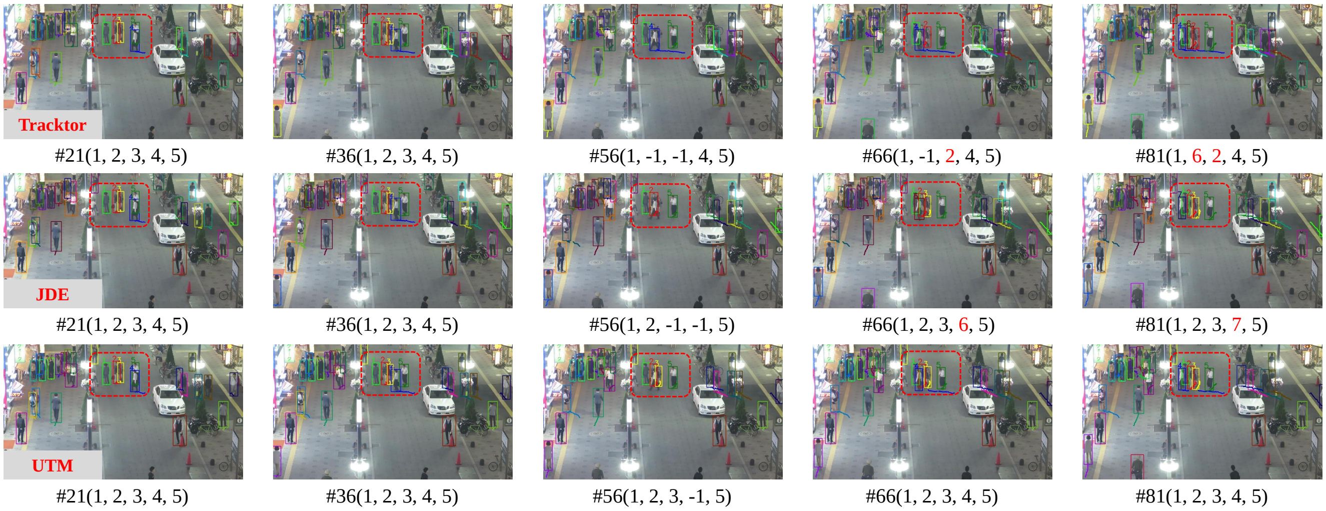
Sisi You[1] Hantao Yao[2] Bing-kun Bao[1] Changsheng Xu[2,3*]

[1]Nanjing University of Posts and Telecommunications

[2]State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences (CASIA)

[3]University of Chinese Academy of Sciences

ssyou@njupt.edu.cn, hantao.yao@nlpr.ia.ac.cn, bingkunbao@njupt.edu.cn, csxu@nlpr.ia.ac.cn



#21(1, 2, 3, 4, 5) #36(1, 2, 3, 4, 5) #56(1, -1, -1, 4, 5) #66(1, -1, 2, 4, 5) #81(1, 6, 2, 4, 5)

#21(1, 2, 3, 4, 5) #36(1, 2, 3, 4, 5) #56(1, 2, -1, -1, 5) #66(1, 2, 3, 6, 5) #81(1, 2, 3, 7, 5)

#21(1, 2, 3, 4, 5) #36(1, 2, 3, 4, 5) #56(1, 2, 3, -1, 5) #66(1, 2, 3, 4, 5) #81(1, 2, 3, 4, 5)

Figure 1. Tracking examples of different frameworks from MOT16 training dataset. Each text under image is organized as $\langle frame\_number(id1, id2, id3, id4, id5)\rangle$, which indicates the frame number of the image and the identity numbers of objects from left to right in the red dotted box. In addition, boxes with different colors in the image represent different identities, -1 and red number in the text represent missed objects and identity switches, respectively.

1. Qualitative Results.

To prove the robustness of the proposed UTM in occlusion scenes, we illustrate the tracking results of Tracktor [1], JDE method [4], and UTM on MOT16 training dataset where some continuous frames with complex scenes are applied. As shown in Figure 1, the predicted trajectories are specified by the colors of bboxes and the assigned identity numbers, and the dotted line represents the trajectory of the last 50 frames. To clearly show the tracking performance, we mark the numbers of identity switches (IDS) in red at the bottom of the image, and apply -1 to represent missed objects. It can be obviously observed that the tracking performance of Tracktor and JDE is not stable enough in occlusion scenes, *e.g.*, IDS in the 66-th and 81-th frames, and missed objects in the 56-th frame. In contrast, UTM avoids identity switches from the 21-th to 81-th frames since UTM forms a positive feedback loop with IAFE module. It is worth noting that UTM always successfully associate the tracklets before and after occlusion to maintain the complete trajectories, *e.g.,* id-4 is re-associated in the 66-th frame, while the other two methods assign an new identity number to the occluded object when it reappears, *e.g.,* id-2 and id-3 of Tracktor, id-4 of JDE.

We further illustrate some qualitative results to show that the proposed UTM can successfully recognize occluded objects in complex scenes in Figure 2. The tracked trajectories are specified by different colors, and the red arrows point to the occluded objects tracked by UTM. As shown in Figure 2, Tracktor and JDE cannot recognize the occluded objects, while UTM accurately tracks the occluded objects through the identity-aware feature enhancement. The examples in different scenes show the effectiveness of UTM

[*] indicates corresponding author: Changsheng Xu.

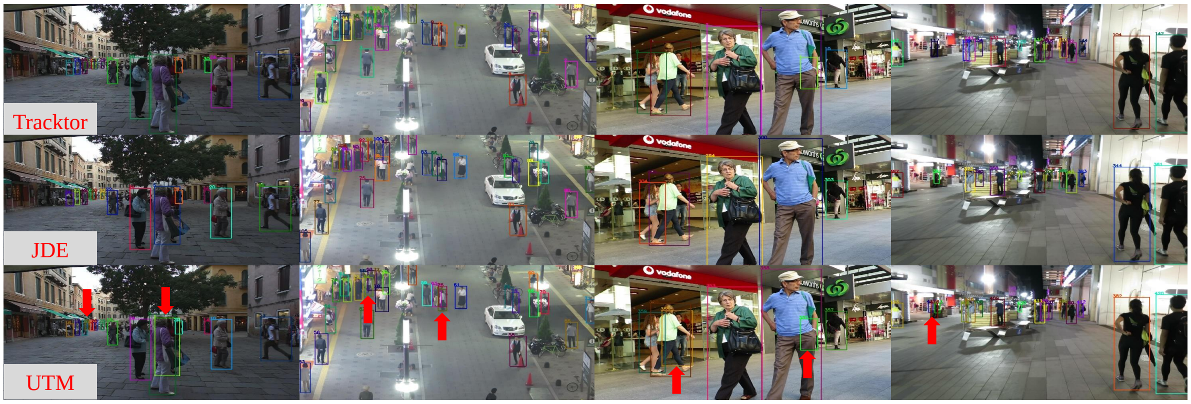


Figure 2. Visualized tracking results of the different frameworks on the MOT16 training dataset. Different colors represent different identities, and the red arrows point to the occluded objects tracked by UTM.

in generating longer and high-quality trajectories in complex scenes.

2. Effect of Different Refined Methods

In this section, we conduct several comparisons between UTM and existing methods to analyze the effect of different refined methods. As shown in Table 1, the refined methods consist of Tracktor [1] and CenterTrack [12], where Tracktor utilizes the regression head of Faster R-CNN [7] to refine the public detections and does not generate the additional detections, and CenterTrack initializes a new trajectory if the IoU between private detection and public detection box is larger than a threshold. Under the Tracktor refining protocol, the proposed method UTM achieves the better performance than existing methods on MOTA and HOTA, *e.g.,* 1.4% and 0.8% improvements. In addition, UTM gets worse performance than the offline method on IDF1, *e.g.,* 65.1% *vs* 66.8%. The reason is that offline methods utilize the global information from past to future for data association, while UTM merely adopts the past information to generate trajectories. Simultaneously, UTM obtains the better performance than existing methods on MOTA, IDF1, and HOTA metrics under the CenterTrack refining protocol. We attribute the performance improvement to that the proposed UTM leverages the identity-aware knowledge to enhance the object detection and feature embedding modules.

References

[1] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *ICCV*, pages 941–951, 2019. 1, 2, 3

[2] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *CVPR*, pages 6247–6257, 2020. 3

[3] Peng Dai, Renliang Weng, Wongun Choi, Changshui Zhang, Zhangping He, and Wei Ding. Learning a proposal classifier for multiple object tracking. In *CVPR*, pages 2443–2452, 2021. 3

[4] Song Guo, Jingya Wang, Xinchao Wang, and Dacheng Tao. Online multiple object tracking with cross-task synergy. In *CVPR*, pages 8136–8145, 2021. 1, 3

[5] Jiawei He, Zehao Huang, Naiyan Wang, and Zhaoxiang Zhang. Learnable graph matching: Incorporating graph partitioning with deep feature learning for multiple object tracking. In *CVPR*, pages 5299–5309, 2021. 3

[6] Andrea Hornakova, Roberto Henschel, Bodo Rosenhahn, and Paul Swoboda. Lifted disjoint paths with application in multiple object tracking. In *ICML*, pages 4364–4375, 2020. 3

[7] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *PAMI*, 39(6):1137–1149, 2017. 2

[8] Fatemeh Saleh, Sadegh Aliakbarian, Hamid Rezatofighi, Mathieu Salzmann, and Stephen Gould. Probabilistic tracklet scoring and inpainting for multiple object tracking. In *CVPR*, pages 14329–14339, 2021. 3

[9] Daniel Stadler and Jurgen Beyerer. Improving multiple pedestrian tracking by track management and occlusion handling. In *CVPR*, pages 10958–10967, 2021. 3

[10] Pavel Tokmakov, Jie Li, Wolfram Burgard, and Adrien Gaidon. Learning to track with object permanence. In *ICCV*, pages 10860–10869, 2021. 3

[11] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *ECCV*, pages 1–21. Springer, 2022. 3

[12] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *ECCV*, pages 474–490. Springer, 2020. 2, 3

Methods	Refined	MOTA↑	IDF1↑	HOTA↑	AssA↑	DetA↑	MT↑	ML ↓	FP↓	FN↓	IDS↓
LifTsI(O) [6]	Tracktor	58.2	65.2	50.7	54.9	47.1	28.6	33.6	16,850	217,944	**1,022**
MPNT(O) [2]	Tracktor	58.8	61.7	49.0	51.1	47.3	28.8	33.5	17,413	213,594	1,185
LPC(O) [3]	Tracktor	59.0	**66.8**	51.7	**56.0**	47.7	27.3	35.0	23,102	206,947	1,122
GMTsI(O) [5]	Tracktor	59.0	65.9	51.1	55.1	47.6	29.0	33.6	20,395	209,553	1,105
GMT [5]	Tracktor	56.2	63.8	49.1	53.9	44.9	21.0	35.5	**8,719**	236,541	1,778
Tracktor [1]	Tracktor	56.3	55.1	44.8	45.1	44.9	21.1	35.3	8,866	235,449	1,987
ArTIST [8]	Tracktor	56.7	57.5	-	-	-	22.4	37.5	12,353	230,437	1,756
TADAM [4]	Tracktor	59.7	58.7	-	-	-	-	-	9,676	216,029	1,930
TMOH* [9]	Tracktor	62.1	62.8	50.4	50.9	50.2	26.9	31.4	10,951	201,195	1,897
UTM	Tracktor	**63.5**	65.1	**52.5**	53.2	**52.5**	**37.4**	**27.0**	33,683	**170,352**	1,686
CenterTrack [12]	CenterTrack	61.5	59.6	48.2	47.8	49.0	26.4	31.9	14,076	200,672	2,583
ByteTrack [11]	CenterTrack	67.4	70.0	56.1	57.5	54.9	31.0	31.2	**9,939**	172,636	**1,331**
PermaTrack [10]	CenterTrack	73.1	67.2	54.2	51.2	58.0	**42.3**	**19.1**	24,577	**123,508**	3,571
UTM	CenterTrack	**74.0**	**75.5**	**61.0**	**62.3**	**60.0**	41.7	22.5	14,198	130,212	2,389

Table 1. Comparison with different refined methods under the public detection protocol on MOT17 dataset. Best results are marked in **BLOD**. "O" and * indicate the offline methods and post processing methods.