Head Point Positioning and Spatial-Channel Self-Attention Network for Multi-Object Tracking

Xin Zhang
State Key Lab of Virtual Reality
Technology and Systems
Beihang University
Beijing, China
Email: zhangxin8275@buaa.edu.cn

Song Gao
State Key Lab of Virtual Reality
Technology and Systems
Beihang University
Beijing, China
Email: gaosong@buaa.edu.cn

Yuanzhe Yang
State Key Lab of Virtual Reality
Technology and Systems
Beihang University
Beijing, China
Email: sy2006320@buaa.edu.cn

Chengxiang Chu
State Key Lab of Virtual Reality
Technology and Systems
Beihang University
Beijing, China
Email: chuchengxiang97@buaa.edu.cn

Zhong Zhou✉
State Key Lab of Virtual Reality
Technology and Systems
Beihang University
Beijing, China
Email: zz@buaa.edu.cn

***Abstract*—Multi-Object Tracking (MOT) aims to generate trajectories for multiple objects in the surveillance scene. This is a challenging task because the pedestrians in tracking video often gather together and occlude each other. Consequently, the two main problems in the popular tracking-by-detection framework are how to alleviate unreliable detection and extract robust object appearance features. In this paper, we propose a new tracking method that is composed of two novel types of modules - an object detection strategy based on pedestrian head point positioning and a Spatial-Channel Self-Attention feature extraction network (SCSAN). Specifically, the proposed detection strategy generates more accurate tracking object bounding boxes with Soft-Head-NMS, which combines the advantages of object detection and head point positioning. The head point location information is used as a guidance to screen unreliable detection. The SCSAN utilizes the Spatial-Channel Self-Attention mechanism to lead and determine the optimal attention value for each area and channel. Extensive experiments are carried out to demonstrate the proposed tracker achieves competitive results and is state-of-the-art in half metrics.**

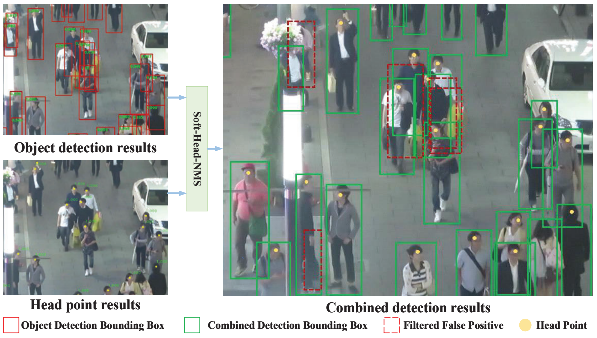
Object detection results
Soft-Head-NMS
Head point results
Combined detection results
🟥 Object Detection Bounding Box
🟩 Combined Detection Bounding Box
▨ Filtered False Positive
● Head Point

Fig. 1. Object locating with head point position information guiding. In the tracking scene, applying only one kind of detection method, the bounding boxes are unreliable due to heavy occlusion. Head point position information can help tracker filter and alleviate unreliable detection.

I. Introduction

Multi-Object Tracking(MOT) is the task of linking a number of object hypotheses detected in surveillance video to generate tracklets of different objects. It plays an essential role in many applications of computer vision, such as intelligent traffic system [1], multi-camera activity analysis [2] and intelligent surveillance [3]. Current MOT researches mainly adopt the tracking-by-detection strategy in online tracking, locating objects in each frame with a detector and associating the objects in different frames to generate trajectories. However, in return, the tracking-by-detection frameworks tend to produce ID switches and to drift under occlusion and unreliable defections because it is more dependent on detection results.

In natural tracking scenes, objects, especially pedestrians, often gather together and occlude each other. The main impact of crowd occlusion is that it severely harms the performance of the detector and significantly increases unreliable detection such as false positives, missing detection and multi objects in the same bounding box, as shown in Fig. 1. To alleviate unreliable detection interference, some studies combine detection and tracking results as the candidate set for quality evaluation and use different strategies for data association [4]. However, these tracking frameworks did not introduce other location information to screen the unreliable detection results. Furthermore, we found that the head region is visible easily in the crowded scene. It seems to be robust guidance information to direct the tracker to screen unreliable bounding boxes, as shown in Fig.1. In this paper, we propose a head point positioning based detection strategy that combines the merits of head point location and object bounding box in a unified framework to introduce head point information.

Furthermore, we propose Soft-Head-NMS detection strategy to use the pedestrian head point information to assist in alleviating unreliable detection.

On the other hand, the occlusion between objects and partial loss in tracking will reduce the discriminative ability of object appearance features, as shown in Fig.1. To alleviate such issues, [5] introduced the attention mechanism to guide the feature extraction network to focus on the object areas of detection images and tracklet images. Additionally, inspired by [6], we propose a Spatial-Channel Self-Attention feature extraction network (SCSAN). Specifically, the original self-attention mechanism only considers global information, resulting in a significant background pixel weight, which brings the noise to interfere with feature extraction. In addition, it cannot calculate the attention value of different channels in the feature map. Therefore, Our model introduces the Spatial-Weight learning module and the Channel-Attention module in the Self-Attention mechanism to determine the optimal self-attention maps for object images.

The main contributions of this paper can be summarized as follows:

1. A new detection strategy is proposed to combine object detection and head point position results. The strategy takes advantage of both object detection and head point position to alleviate unreliable detection in the online multi-object tracking framework.

2. We design a Spatial-Channel Self-Attention feature extraction network (SCSAN), which introduces the Spatial-Weight Learning module and the Channel-Attention module in the Self-Attention mechanism to allocate different attention values to each location and channel in the object feature map.

3. Experimental results demonstrate that our tracker achieves competitive performance on the MOT benchmark dataset and is state-of-the-art in some metrics.

II. Related Work

A. Detection Strategy

Given detection by the detector at each frame, the tracking-by-detection framework locally associates detections frame-by-frame to generate long trajectories in general. Recent approaches have focused on improving the performance of detection to improve tracking [7]–[10]. Chen et al. in [4] combined detection and predicted bounding boxes from tracklet as candidates set for quality evaluation and used different methods for data association. Voigtlaender et al. in [11] introduced top-down segmentation information instead of detection information to locate the tracking object. Additionally, some recent works have directly used the dense detection output, before the non-maximum suppression, as the input to their tracker [12]. This is primarily to overcome the limitations of detectors and non-maximum suppression algorithms when objects are occluding each other or are too close to each other. Shu et al. in [13] proposed an extension to deformable part-based human detector, which can handle occlusion up to a scale. However, the above methods only applied one kind of location information to determine the bounding box of the tracking objects and did not introduce additional information for guidance, which cannot effectively alleviate unreliable detection results in tracking. On the contrary, we propose the Soft-Head-NMS detection strategy to introduce object head location information. The motivation behind introducing object head point information is that the head region is visible easily in crowds and thus can be a piece of powerful guidance information to direct the detector to screen unreliable detection and improve the detection results in MOT. Furthermore, our detection strategy can be easily integrated into other tracking frameworks to improve the tracking performance further.

B. Feature Extraction

Object appearance is an important clue for identifying cross-frame observations. Inspired by the recent advances in deep learning, several Multi-Object Tracking frameworks [14]–[16] using person re-identification networks [17], [18] to extract object features. These methods have shown that discriminative generic features can be trained using deep learning module. Chu et al. in [5] introduced a Spatial-Temporal Attention Mechanism (STAM) to handle the tracking drift caused by the occlusion and interaction among objects. Zhu et al. in [19] proposed a Dual Matching Attention Networks (DMAN), which introduced spatial and temporal attention mechanism to extract robust features against appearance variations and cluttered backgrounds. In this paper, we introduce the self-attention mechanism with the Spatial-Weight Learning module and the Channel-Attention module. Our feature extraction network differs from the DMAN method. First, the attention feature map in the DMAN corresponds to the detection image and trajectory images. Since the attention feature map is affected by different trajectory images, it becomes unreliable when other objects appear in the trajectory image. In contrast, we exploit the image itself to generate the self-attention map, which is demonstrated to be more robust to noisy detections and occlusions. Second, we introduce the Spatial-Weight Learning module and the Channel-Attention module to alleviate the noise introduced by the background pixels and learn discriminative feature representations at multiple channels. Third, compared with DMAN, our network is end-to-end, which can alleviate the complexity of training and extracting features.

III. Proposed Method

Three main components of our tracking framework are a proposed detection strategy, a designed feature extraction network and a trajectory management method. The tracking framework introduces object head point information and improves detection performance by the Soft-Head-NMS detection strategy. Then we use the Spatial-Channel Self-Attention Network to compute the attention values of different areas and channels to extract features. Finally, we update the tracking state of objects and trajectories.

A. Soft-Head-NMS Object Detection Strategy

After obtaining a new tracking frame, we generate bounding boxes and head point position of each object through the

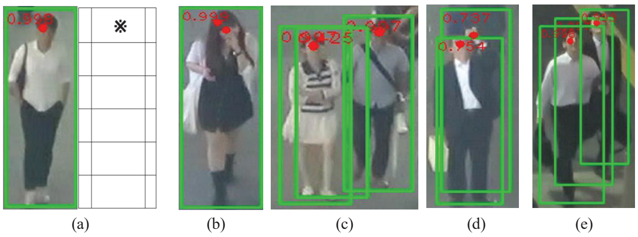

Fig. 2. The bounding box and head point positioning results. (a) shows the head point position when it matches the bounding box. (b) shows the false-positive results of head point. (c) shows the false-positive results of the bounding box. (d) and (e) shows matching results of multiple bounding boxes corresponding to multiple head points.

object detection network and head point regression network, respectively. It is necessary to generate a sufficient number of detection bounding boxes to filter and obtain accurate tracking bounding boxes. Therefore, we set detection threshold $T_{detcon} = 0.5$ to generate the object detection bounding boxes.

In order to measure trakcing objects bounding box set B_{track}, we perform preliminary matching on the head points and bounding boxes firstly. We divide all the bounding boxes into six parts evenly horizontally and divided into three parts according to the ratio $1:4:1$ vertically, as shown in Fig.2(a). If the position of the head point is in the middle grid at the top of the horizontal, it indicates that the head point matches the bounding box. In this way, we generate the preliminary matching result.

Furthermore, we conduct a preliminary screen of the preliminary matching results obtained in the previous step. For the first situation, when a bounding box corresponds to multiple head points and these head points do not correspond to other bounding boxes, as shown in Fig.2(b). We only retain the head point with the highest confidence. Secondly, because we lowered the confidence threshold, some redundant bounding boxes are generated. Therefore, for multiple bounding boxes that associate with the same head point, we only keep the bounding box with the highest confidence, as shown in Fig.2(c). After the initial screen, most of the bounding boxes and head points can be a one-to-one correspondence. However, the issue of multiple head points corresponding to multiple bounding boxes still exists, as shown in Fig.2(d) and (e).

In the following step, we perform detailed matching with KM algorithm [20] to deal with such issue. Specifically, we calculate the weight between the bounding box and the head point for bipartite graph matching firstly. For a correctly matching bounding box, the head point should be at the horizontal center of the bounding box. Therefore, we define the distance ratio between the head point and the center point of the bounding box as C_M which can accurately describe the matching degree between the head point and the bounding box. The following formula can calculate the C_M:

$$C_M = \frac{|x_p - x_{center}|}{W/2} \tag{1}$$

where x_p is the abscissa of the head point, x_{center} is the abscissa of the bounding box center point and W is the width of the bounding box. For a correct matching bounding box, the head point should be at the horizontal center of the bounding box, so C_M can accurately describe the matching degree between the head point and the bounding box. In addition, we introduce the confidence of bounding box C_B and the confidence of head point C_H to calculate the weight between the bounding box and the head point.

We observe that the issue of multiple head points and multiple bounding boxes correspondence can be divided into the following three situations:

Situation 1: the two head points correspond to two bounding boxes. It is necessary to judge whether the head points and the bounding boxes are accurate. The calculation formula for the weights in the matching bipartite graph is as follows:

$$W_{s_1} = 3\left(C_B + C_H\right) + C_M \tag{2}$$

Situation 2: for the issue of two head points corresponding to three bounding boxes, it is due to redundant boxes. The matching degree between head point and bounding box becomes the main factor that needs to be considered and the weights in the matching bipartite graph can be given by:

$$W_{s_2} = C_B + C_H + 3C_M \tag{3}$$

Situation 3: the situation of three head points corresponding to three bounding boxes is similar to the two head points corresponding to two bounding boxes. Therefore the weights in the matching bipartite graph can be defined as:

$$W_{s_3} = 3\left(C_B + C_H\right) + C_M \tag{4}$$

Moreover, when the weight of the head point and the bounding box is inferior, it can be directly considered that the head point and the bounding box cannot match. We analyze the weight of head point and bounding box in training data and set weight threshold $T_{W_{s_1}} = 2$, $T_{W_{s_2}} = 3$ and $T_{W_{s_3}} = 2$ for the three kinds of multiple head points and multiple bounding boxes correspondence issues, respectively.

Last, we apply the KM algorithm to associate the head point and the bounding box to generate head point-optimized bounding box set B_H. In order to highlight the guidance of the head point, we define the confidence of ith head point-optimized B_{Hi} as:

$$CB_{Hi} = C_B + C_H \tag{5}$$

where CB_{Hi} is the confidence of the ith head point-optimized bounding box.

In order to determine the tracking object bounding boxes set B_{track}, we sort all the bounding boxes in the current frame according to the confidence to generate candidate bounding boxes set B_{can} and output the bounding box B_{max} with the maximum confidence to B_{track} as tracking object bounding box. Then, we re-assign the confidence of remaining bounding boxes as:

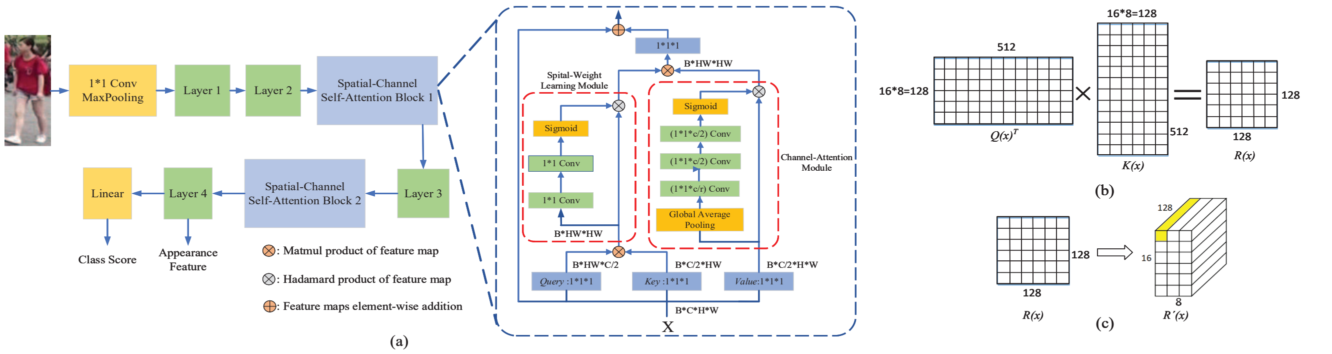

Fig. 3. (a) shows the architecture of the proposed Spatial-Channel Self-Attention Network (SCSAN). The SCSAN includes two modules, namely the Spatial-Weight Learning module and the Channel-Attention module. The weights of different pixels are produced in the Spatial-Weight Learning module to reduce background noise, while the Channel-Attention module is calculating the different attention values in channels to enhance the discriminative of the feature map. We give an object detection or trajectory image as input. (b) and (c) illustrates calculation and transformation of self-attention map. (b) shows the process of calculating the self-attention map in the original self-attention mechanism. (c) shows we transform the obtained self-attention map to a three-dimensional matrix with the size of $16*8*128$.

$$C_{cani} = C_{cani} \cdot exp\left(-\frac{IoU_i * IoU_i}{\delta}\right) \tag{6}$$

where C_{cani} indicates the confidence of ith bounding box in B_{can}, IoU_i indicates the IoU of bounding box B_{max} and B_{cani}. Finally, we delete the candidate that confidence less than the confidence threshold T_{con}, until B_{can} is empty.

B. Spatial-Channel Self-Attention Network

The appearance feature is vital in calculating the similarity score between the objects and the trajectories. To get a robust appearance feature for tracking object, we design a Spatial-Channel Self-Attention feature extraction network (SCSAN), as shown in Fig.3(a). In SCSAN, we exploited the ResNet50 [21] as the backbone network and propose Spatial-Channel Self-Attention mechanism to extract object appearance feature. Specifically, we design the Spatial-Weight Learning model to alleviate the issue of image background noise in the self-attention mechanism. Meanwhile, we introduce Channel-Attention model to guide the network to determine the attention value of each channel in the feature map. We describe the two modules as follows.

In the original self-attention mechanism, the process of calculating the self-attention map is shown in Fig.3(b). We find that the feature map $Q(x)^T$ and $K(x)$ are learned through two different $1{*}1$ convolutional layers, respectively. As a result, the similarity weight between each pixel is completely dependent on the channel features of each pixel, and the positional relationship between the pixels is ignored. Therefore, although the self-attention mechanism considers global information, it does not perform weight learning and lacks the adaptability to pixel positions. This causes the weight of background pixels with similar features to increase and introduce abundant noise in object feature maps.

To alleviate such issue, we design the Spatial-Weight Learning module to optimize self-attention mechanism. First, we transform the obtained self-attention map $R(x)$, as shown in Fig.3(c). Each row vector in $R(x)$ represents the influence of the overall 128 pixels on the i_{th} pixel and the arrangement of $R(x)$ conforms to the spatial position of the original feature map. Therefore, we transform it to a three-dimensional feature map $R'(x)$ of size $16*8*128$. Then, we use the Spatial-Weight Learning module to guide the network to learn the weights of pixels at different locations, as shown in Fig.3(a). Specifically, we regard 128 in the third dimension as the number of channels. In this way, the yellow vector in $R'(x)$ represents the influence of the overall pixels to the (i, j) pixel. Then, we can apply the $1*1$ convolution layer to learn the influence weight of 128 pixels in the object feature map to the (i, j) pixel. Furthermore, in order to enhance the ability of the network, we introduce two $1*1$ convolutional layers in the Spatial-Weight Learning module. Last, we generate the spatial weight map by a sigmoid activation function, so that the network can extract more robust feature under the guidance of spatial information.

In addition, the channels of the object feature map also have different meanings. Therefore, inspired by SENet [22], we design the Channel-Attention module to enable the self-attention mechanism to learn the attention value of different channels adaptively. Moreover, in order to ease the amount of calculation and enable the network to converge, we add the channel attention module after the $Value$ convolutional layer in the self-attention module, as shown in Fig.3(a).

First, in the Channel-Attention module, the object feature performs the global average pooling to enable the network to focus on the channel information of the feature map. The difference from SENet is that we apply three convolutional layers instead of fully connected layers to learn the attention value of the channel, which can reduce the amount of network calculations. Finally, we use the Sigmoid activation function to compress the weight of each channel to $(0, 1)$, which represents the attention value of the different channels. We multiply the obtained channel attention map with the original feature map to generate the channel attention feature map.

We learn appearance feature through a Re-ID task. Therefore, we use the softmax loss to improve prediction accuracy of the network. Simultaneously, the object feature map representing different IDs is adjusted by the triplet loss and center loss, aiming to learn the similarities and differences between input image pairs. The training objective of Spatial-Channel Self-Attention network can be written as a weighted linear sum of losses:

$$L_{total} = \alpha L_{softmax} + \beta L_{triplet} + \gamma L_{center} \tag{7}$$

where α, β and γ are loss weights. We utilize the ground-truth bounding boxes in the MOT16 training set to generate training data for the feature extract network training.

C. Data Association and Trajectory Management

We follow the standard online tracking algorithm to associate detection bounding boxes and trajectories. We calculate the similarity score of the detection and tracklet feature maps firstly, by the cosine distance. Then tracker generates the affinity matrix with similar scores. Meanwhile, we apply the Hungarian algorithm [20] for bipartite graph matching with obtained similarity matrix and link the detected boxes to the existing tracklets. Last, we also use Kalman Filter to predict the locations of the tracklets in the current frame. The tracker associates the remaining detection with unassociated traklet based on IoU between detection and predict locations of unassociated traklet, with a threshold T_{IoUa}. For trajectory management, we initial the trajectory for object detection, which is not associated with any trajectory in any of the first T_{init} frames. The Trajectory is terminated if they are not associated with T_{term} frames.

IV. Experiments

We first verify the effectiveness of the proposed detection strategy and object feature extraction network by applying them for a Multi-Object Tracking problem. Then, we analyze the performance improvement in MOT by the proposed tracking framework in detail.

A. Experiment Setup

Implementation details. To evaluate the performance of the proposed online tracking method, we conduct extensive experiment on the MOT16 dataset [23]. We employ Faster R-CNN [24] to generate the object bounding boxes, and use RAZNet [25] to estimate object head point position information. We set T_{con}=0.5 to generate the tracking object set B_{track} and set T_{IoUa}=0.7 for data association. For trajectory management, we set threshold T_{init}=3 and T_{term}=10 for trajectory initialization and trajectory termination, respectively.

Evaluation metrics. In order to measure the accuracy of tracking results, we adopt multiple metrics used in the MOT benchmark to evaluate the proposed tracking method, including Multiple Object Tracking Accuracy (MOTA), the ratio of correct detections over the average number of ground-truth and computed detections (IDF1 score), the ratio of Mostly Tracked objects (MT), the ratio of Mostly Lost Objects (ML), the number of False Negatives (FN), the number of False Positives (FP), the number of ID Switches (IDs), the number of fragments (Frag). Table III and Table IV present the tracking performance on the MOT 16 and MOT 17 dataset.

B. Ablation Studies

In order to verify the effectiveness of the proposed detection strategy and evaluate its contribution, we conduct ablation experiments on MOT16. The results are shown in Table I. We compare our detection strategy with SDP [26], Faster R-CNN [24] and Mask R-CNN [27]. In addition, to exclude the disturbance of other factors, we use PCB [28] to extract object feature and DeepSORT [14] to generate object trajectory.

TABLE I
Evaluation results on MOT16 with different detection method.The arrow each metric indicates that the higher (↑) or lower (↓) value is better

Detector	MOTA↑	IDF1↑	FP↓	FN↓
Mask R-CNN + PCB	40.2	52.6	14426	51234
Faster R-CNN + PCB	60.6	63.0	14801	56143
SDP + PCB	58.6	58.0	**8461**	66295
Ours +PCB	**63.1**	**64.2**	12516	**50423**

The experiment results as shown in Table I. The comparison between our detection strategy and object detection methods confirms that our detection strategy performs best. Compare with the Faster R-CNN, our detection strategy improves 2.5 in MOTA, 1.2 in IDF1 and effectively reduced FN, which demonstrates the merits of our detection strategy in locating objects. By locating object bounding boxes with head point guidance, the detection strategy reduces unreliable detection and improves the MOTA, as shown in Fig.4(a).

TABLE II
Evaluation results on MOT16 with different feature representations.

Method	MOTA↑	IDF1↑	IDs↓
SDP + ResNet50	62.2	63.7	850
SDP + PCB	62.1	58.9	1061
SDP + StrongBaseline	63.3	64.9	757
SDP + Ours(SCSAN)	**64.6**	**65.3**	**711**

To demonstrate the contribution of the proposed SCSAN network in our method, we compare representations learned by SCSAN with ResNet50, PCB and StrongBaseline. Moreover, we use SDP [26] detection result, provide by MOT16 officially, and DeepSORT for tracking. The experiment results are shown in Table II. It can be seen that the MOTA, IDF1 and IDs of SCSAN are better than other methods. Our tracker upgrades MOTA to 64.6, IDF1 to 65.3 and reduces IDs to 711, which demonstrates the effectiveness of our feature extraction network. The higher IDF1 value indicates that the proposed SCSAN network focuses on more explicitly object regions and channels which enhance the power of extracting discriminative feature.

TABLE III

TRACKING PERFORMANCE ON MOT16 DATASET. THE ARROW EACH METRIC INDICATES THAT THE HIGHER (↑) OR LOWER (↓) VALUE IS BETTER.

	Tracker	MOTA↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDs↓	Frag↓
Batch	QuadMOT [29]	44.1	38.3	14.6%	44.9%	6388	86245	745	1096
	EDMT [30]	45.3	47.9	17.0%	39.9%	11122	87899	639	946
	LMP [15]	48.8	51.3	18.2%	40.1%	6654	86245	481	595
Online	Tracktor++ [31]	56.2	54.9	20.7%	35.8%	**2394**	76844	617	1068
	MPNTrack [32]	58.6	61.7	27.3%	34.0%	4949	70252	**354**	**684**
	DeepSortv2 [14]	61.4	52.2	32.8%	18.2%	5119	63352	781	2008
	TMOH [33]	63.2	63.5	27.0%	31.0%	3122	63376	635	1486
	Tube_TK [34]	64.0	59.4	33.5%	19.4%	10962	53626	1117	1366
	CNNMTT [35]	65.2	62.2	32.4%	21.3%	6578	55896	946	2283
	POI [29]	66.1	65.1	34.0%	20.8%	5061	55914	805	3093
	CTracker [36]	67.6	57.2	32.9%	19.0%	8934	48305	1897	3112
	FairMOT [37]	67.7	68.4	**37.5%**	19.0%	13501	49653	953	2399
	Ours	**67.9**	**69.0**	28.7%	**17.9%**	12455	**47931**	597	2406

TABLE IV

TRACKING PERFORMANCE ON MOT17 DATASET.

	Tracker	MOTA↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDs↓	Frag↓
Batch	EDMT [30]	50.0	51.3	21.6%	36.3%	32279	247297	2264	3260
	MHT_DAN [38]	50.7	47.2	20.8%	36.9%	22875	252889	2314	2865
	NOTA [39]	51.3	54.7	17.1%	35.4%	20148	252531	2285	4080
Online	Tracktor++ [31]	56.3	55.1	20.1%	35.3%	**8866**	235449	1987	3763
	TMOH [33]	62.1	62.8	26.9%	31.4%	10951	201195	1897	4622
	POI [29]	63.0	58.6	31.2%	**19.9%**	27060	177483	4137	5727
	CTracker [36]	66.6	57.4	32.2%	24.2%	22284	160491	5529	9114
	CTTrack17 [40]	**67.8**	64.7	34.6%	24.6%	18498	**160332**	3039	6102
	Ours	66.9	**68.3**	**35.7%**	21.5%	23587	193286	**1868**	**2683**

Fig.4(b) shows the visualization results of the self-attention feature map form SCSAN. In Fig.4(b), the top row shows images from the object detection or trajectory, while the bottom row presents corresponding self-attention feature maps. It can be seen that our self-attention feature map focus more explicitly on object regions and suppress noise and occlusion, which enhances the power of extracting discriminative feature.

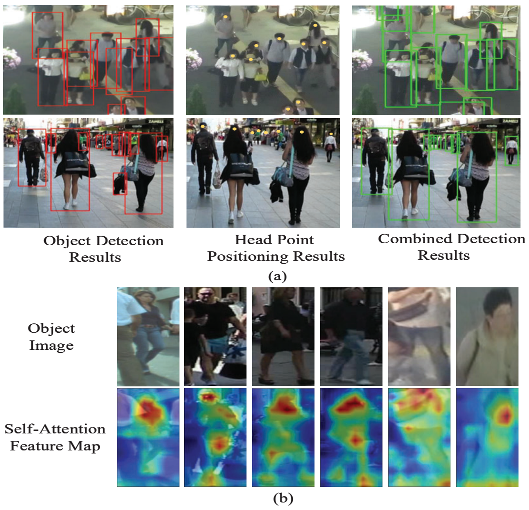
Object Detection Results
Head Point Positioning Results
Combined Detection Results
(a)
Object Image
Self-Attention Feature Map
(b)

Fig. 4. Visualization results. (a) shows the visualization of object locating results. (b) shows the object image and object feature map.

C. Performance on MOT Benchmark Datasets

We report evaluation on the test set of MOT16 and compare our tracker with other offline and online trackers in Table III. As shown in Table III, our tracking method achieves a comparable MT, FP, IDs, Frag score and performs favourably against the state-of-the-art methods in terms of MOTA, IDF1, ML and FN on the MOT16 dataset. Our tracker upgrades MOTA to 67.9, IDF1 to 69.0 and reduces ML to 17.9, FN to 47931. Meanwhile, our tracker achieves the competitive performance in IDs and best performance in IDF1 among online and batch methods, demonstrating the merits of our tracker in object identity matching and the stability of multi-object tracking. MOTA and FN correspond to the object detection capability. Therefore, the improvement of MOTA and FN demonstrates the merits of our Soft-Head-NMS detection strategy in object locating for MOT. In addition, IDF1 and IDs can reflect the quality of the object feature extracted by the tracker. The improved performance of IDF1 and IDs demonstrates the merits of the Spatial-Channel Self-Attention network in feature extraction for MOT. Similarly, Table IV shows that our tracker outperforms existing online trackers on half of the metrics and achieves the best performance in terms of IDF1, MT, IDs and Frag on the MOT17 dataset. In addition, we achieve the best IDF1 score among all the online and batch trackers on both the MOT16 and MOT17 datasets.

However, as shown in Table III and Table IV, our tracker has a high FP. This is because the Soft-Head-NMS detection strategy not only can effectively alleviate unreliable detections but also complement missing detections. Therefore, our tracker can detect and track these pedestrians who are not recorded as tracking objects. Therefore, our detection strategy will cause the phenomenon of high FP, and the similar situation exists in work [4], [30] too. Additionally, this phenomenon also reflects the effectiveness of the detection strategy proposed in this paper.

V. CONCLUSION

We present a novel framework that improves two main components of most online trackers, detection and feature extraction. The tracker combines the advantage of object detection and head point positioning. Then generating optimal object bounding boxes by the proposed Soft-Head-NMS detection strategy, which also helps alleviate typical difficulties in tracking such as occlusion handling and track drifting. In this work, for calculating more accurate similarity scores, the tracker learns the discriminative feature map from object image with Spatial-Channel Self-Attention module. We have proved that the proposed tracking framework leads to competitive performance improvement through extensive experiments.

ACKNOWLEDGMENT

This work was supported by the National Key R&D Program of China (Grant No.2018YFB2100603) and the National Natural Science Foundation of China (Grant No.61872024). The authors would like to thank the anonymous reviewers for their critical and constructive comments and suggestions.

References

[1] Y. Rasekhipour, A. Khajepour, S. Chen, and B. Litkouhi. A potential field-based model predictive path-planning controller for autonomous road vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1255–1267, 2016.

[2] N. Ran, L. Kong, Y. Wang, and Q. Liu. A robust multi-athlete tracking algorithm by exploiting discriminant features and long-term dependencies. In *International Conference on Multimedia Modeling*, pages 411–423. Springer, 2019.

[3] Z. Li, X. Yu, P. Li, and M. Hashem. Moving object tracking based on multi-independent features distribution fields with comprehensive spatial feature similarity. *The Visual Computer*, 31(12):1633–1651, 2015.

[4] L. Chen, H. Ai, Z. Zhuang, and C. Shang. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2018.

[5] Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, and H. Yu. Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4836–4845, 2017.

[6] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pages 7354–7363. PMLR, 2019.

[7] W. Choi. Near-online multi-target tracking with aggregated local flow descriptor. In *Proceedings of the IEEE international conference on computer vision*, pages 3029–3037, 2015.

[8] K. Fragkiadaki and J. Shi. Detection free tracking: Exploiting motion and topology for segmenting and tracking under entanglement. In *CVPR 2011*, pages 2073–2080. IEEE, 2011.

[9] S. Tang, B. Andres, M. Andriluka, and B. Schiele. Subgraph decomposition for multi-target tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5033–5041, 2015.

[10] A. Milan, S.H. Rezatofighi, A. Dick, I. Reid, and K. Schindler. Online multi-target tracking using recurrent neural networks. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[11] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B.B.G. Sekar, A. Geiger, and B. Leibe. Mots: Multi-object tracking and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7942–7951, 2019.

[12] S. Tang, M. Andriluka, B. Andres, and B. Schiele. Multiple people tracking by lifted multicut and person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3539–3548, 2017.

[13] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah. Part-based multiple-person tracking with partial occlusion handling. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1815–1821. IEEE, 2012.

[14] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[15] S. Tang, M. Andriluka, B. Andres, and B. Schiele. Multiple people tracking by lifted multicut and person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3539–3548, 2017.

[16] X. Zhang, X. Wang, and C. Gu. Online multi-object tracking with pedestrian re-identification and occlusion processing. *The Visual Computer*, 37:1089–1099, 2021.

[17] J.H. Zhou, B. Su, and Y. Wu. Online joint multi-metric adaptation from frequent sharing-subset mining for person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2909–2918, 2020.

[18] T. L. Chen, S. J. Ding, J. Y. Xie, Y. Yuan, W. Y. Chen, Y. Yang, Z. Ren, and Z. Y. Wang. Abd-net: Attentive but diverse person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8351–8361, 2019.

[19] J. Zhu, H. Yang, Nian Liu, M. Kim, W. Zhang, and M. Yang. Online multi-object tracking with dual matching attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 366–382, 2018.

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

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

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

[23] M. Anton, L.T. Laura, R. Ian, R. Stefan, and S. Konrad. MOT16: A Benchmark for Multi-Object Tracking. *arXiv e-prints*, page arXiv:1603.00831, March 2016.

[24] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

[25] C. Liu, X. Weng, and Y. Mu. Recurrent attentive zooming for joint crowd counting and precise localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1217–1226, 2019.

[26] F. Yang, W. Choi, and Y. Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2129–2137, 2016.

[27] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[28] Y. Sun, L. Zheng, Y. Yang, Q. Tian, and S. Wang. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 480–496, 2018.

[29] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan. Poi: Multiple object tracking with high performance detection and appearance feature. In *European Conference on Computer Vision*, pages 36–42. Springer, 2016.

[30] J. Chen, H. Sheng, Y. Zhang, and Z. Xiong. Enhancing detection model for multiple hypothesis tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 18–27, 2017.

[31] P. Bergmann, T. Meinhardt, and L. Leal-Taixe. Tracking without bells and whistles. In *Proceedings of the IEEE international conference on computer vision*, pages 941–951, 2019.

[32] G. Brasó and L. Leal-Taixé. Learning a neural solver for multiple object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6247–6257, 2020.

[33] S. Daniel and B. Jurgen. Improving multiple pedestrian tracking by track management and occlusion handling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10958–10967, 2021.

[34] B. Pang, Y. Li, Y. Zhang, M. Li, and C. Lu. Tubetk: Adopting tubes to track multi-object in a one-step training model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6308–6318, 2020.

[35] N. Mahmoudi, S.M. Ahadi, and M. Rahmati. Multi-target tracking using cnn-based features: Cnnmtt. *Multimedia Tools and Applications*, 78(6):7077–7096, 2019.

[36] P. Jinlong, W. Changan, W. Fangbin, W. Yang, W. Yabiao, T. Ying, W. Chengjie, L. Jilin, H. Feiyue, and F. Yanwei. Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In *European Conference on Computer Vision*, pages 145–161. Springer, 2020.

[37] Y.F. Zhang, C.Y. Wang, X.G. Wang, W.J. Zeng, and W.Y. Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129(11):3069–3087, 2021.

[38] C. Kim, F. Li, A. Ciptadi, and J.M. Rehg. Multiple hypothesis tracking revisited. In *IEEE International Conference on Computer Vision*, 2015.

[39] C. Long, A. Haizhou, C. Rui, and Z. Zijie. Aggregate tracklet appearance features for multi-object tracking. *IEEE Signal Processing Letters*, 26(11):1613–1617, 2019.

[40] X. Zhou, V. Koltun, and P. Krhenbühl. Tracking objects as points. *arXiv*, 2020.