LETTER

Robust Visual Tracking Using Hierarchical Vision Transformer with Shifted Windows Multi-Head Self-Attention*

Peng GAO[†a)], ***Member***, **Xin-Yue ZHANG**[†], **Xiao-Li YANG**[†], **Jian-Cheng NI**[†], ***and*** **Fei WANG**[††], ***Nonmembers***

SUMMARY Despite Siamese trackers attracting much attention due to their scalability and efficiency in recent years, researchers have ignored the background appearance, which leads to their inapplicability in recognizing arbitrary target objects with various variations, especially in complex scenarios with background clutter and distractors. In this paper, we present a simple yet effective Siamese tracker, where the shifted windows multi-head self-attention is produced to learn the characteristics of a specific given target object for visual tracking. To validate the effectiveness of our proposed tracker, we use the Swin Transformer as the backbone network and introduced an auxiliary feature enhancement network. Extensive experimental results on two evaluation datasets demonstrate that the proposed tracker outperforms other baselines.

key words: *Siamese network, visual tracking, vision transformer, self-attention*

1. Introduction

Visual tracking is one of the hot research topic in the computer vision community. It has wide range of practical applications in unmanned aerial vehicles, human-computer interaction, video surveillance, and so forth. Despite various approaches achieving impressive success, due to several factors such as deformation, fast motion, and occlusion, robust tracking of target objects still remains significant challenges [1].

Over the last decade, Siamese trackers have achieved excellent results in terms of accuracy and robustness, which regards the visual tracking task as a one-shot matching problem [2]. These trackers first employ a convolution neural network (CNN) as the backbone to extract the features of a target template and a series of search candidates, where the similarity between the extracted template and candidate features then are matched in a cross-correlation fashion. The current location of the target object is determined by finding the search candidates that most similar to the target template. DiMP [3] utilizes multiple template images for training, and constantly updates the target template during tracking. To improve the discriminative ability of the tracker, a classifier is trained online using background information and initialized with the initial frame. DiMP also proposed a target prediction module to learn a more discriminative filter that performs convolution operations on the search region. However, the backbone networks employed by existing trackers are commonly built upon traditional CNNs, which may not be ideal for extracting representative features from the video sequence for visual tracking tasks, especially the global contextual references, as they can only process short-term local spatial information in the target template and search candidates.

As a relatively new architecture, Transformer [4] has demonstrated significant potential in fields such as natural language processing and speech recognition, and also received attention in the computer vision community. Transformer architectures utilize a self-attention mechanism to establish contextual relationships between inputs. However, it is not possible for the self-attention mechanism to focus on a specific input without simultaneously affecting other equally important inputs. To address this, multiple heads, known as multi-head self-attention (MSA), are employed to enhance the performance of the self-attention mechanism. More recently, MSA based Transformer architecture [5] has ignited the research passion in the visual tracking community [6]. Inspired by the idea of MSA and DiMP, we propose a novel Siamese tracker, termed SwinDiMP, that replaces the long-used off-the-shelf CNNs with a MAS based Transformer architecture to learn what is essential to the visual tracking task in a shifted windows manner. This tracker can effectively fuse hierarchical feature representation and generate more semantically meaningful contextual information than existing Siamese trackers. Experimental results on several large-scale visual tracking datasets show that the proposed SwinDiMP can achieve robust visual tracking.

2. The Proposed Approach

In this section, we provide a detailed description of the proposed SwinDiMP. The overview pipeline of SwinDiMP is illustrated in Fig. 1.

2.1 Multi-Head Self-Attention Backbone Network

The backbone network employed in SwinDiMP is based on the hierarchical vision Transformer (Swin Transformer) block [5], which is constructed by replacing the standard MSA module in a Transformer block with a windows multi-head self-attention (WMSA) module, followed by a multi-

Manuscript received August 9, 2023.
Manuscript revised September 23, 2023.
Manuscript publicized October 20, 2023.

[†]The authors are with School of Cyber Science and Engineering, Qufu Normal University, Qufu, Shandong 273165, China.
[††]The author is with School of Electronics and Information Engineering, Harbin Institute of Technology, Shenzhen, Guangdong 518055, China.

*This work was supported in part by the China Postdoctoral Science Foundation under Grant 2023M732022.

a) E-mail: pgao@qfnu.edu.cn (Corresponding author)

DOI: 10.1587/transinf.2023EDL8053

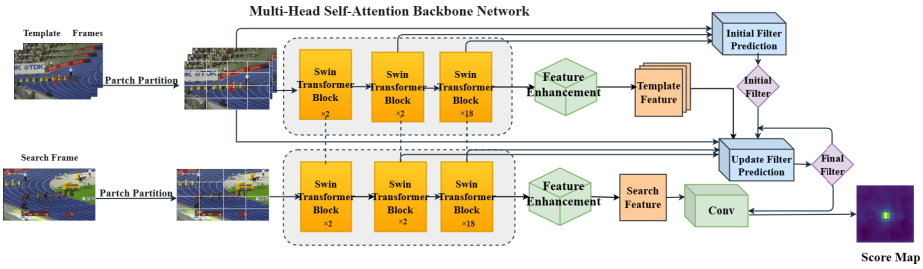

Fig. 1 Overview of the proposed SwinDiMP. Our approach consists of three main components: a multi-head self-attention backbone network, an auxiliary feature enhancement network, and a target prediction module inherited from DiMP [3].

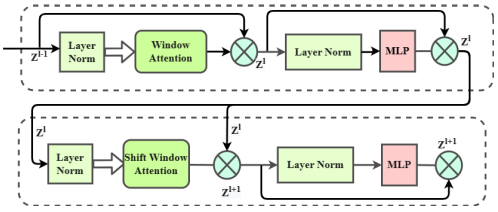

Fig. 2 The pipeline of a Swin Transformer block.

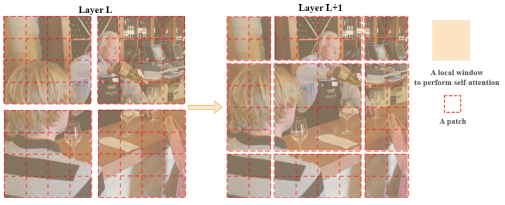

Fig. 3 An illustration of the shifted windows manner for computing self-attention in Swin Transformer [5].

layer perceptron (MLP) with GELU non-linearity in between, while keeping the other layers the same, as depicted in Fig. 2. Before each MSA and MLP, a LayerNorm (LN) layer is placed, and a residual connection is utilized after each module.

WMSA module proposes to compute self-attention in non-overlapping local windows instead of global self-attention. To achieve this, the image is divided evenly and non-overlapping into windows. Because the number of patches in each window is much smaller than that of the entire image, and the number of windows remains the same, the computational complexity of WMSA has a linear relationship with the image size, which greatly reduces the overall computational complexity of the model. While WMSA reduces the computational complexity from quadratic to linear, it still lacks cross-window connection, which limits its modeling and representation ability. To address this issue and facilitate better interaction between windows, Swin Transformer introduces shifted windows multi-head self-attention (SWMSA). In Fig. 3, layer L evenly shifts the local window, whereas layer $L+1$ shifts the window shape across the feature map, resulting in a new distribution. This new configuration allows for windows in subsequent layers to overlap, promoting connectivity between them. Specifically, the first module partitions the 8×8 feature map into uniformly divided 2×2 windows of size 4×4 (at this point, the local window size is $M = 4$). Then, the following module utilizes shifted windows configuration by displacing the windows by $(\lfloor \frac{M}{2} \rfloor, \lfloor \frac{M}{2} \rfloor)$ pixels from the regularly partitioned windows, as shown in the red box. By adopting this shifted windows partition method, the calculation of two consecutive Swin Transformer blocks can be expressed as,

$$\hat{\mathbf{Z}}^{L} = \mathtt{WMSA}(\mathtt{LN}(\mathbf{Z}^{L-1})) + \mathbf{Z}^{L-1} \tag{1}$$

$$\mathbf{Z}^{L} = \mathtt{MLP}(\mathtt{LN}(\hat{\mathbf{Z}}^{L})) + \hat{\mathbf{Z}}^{L} \tag{2}$$

$$\hat{\mathbf{Z}}^{L+1} = \mathtt{SWMSA}(\mathtt{LN}(\mathbf{Z}^{L})) + \mathbf{Z}^{L} \tag{3}$$

$$\mathbf{Z}^{L+1} = \mathtt{MLP}(\mathtt{LN}(\hat{\mathbf{Z}}^{L+1})) + \hat{\mathbf{Z}}^{L+1} \tag{4}$$

where $\hat{\mathbf{Z}}^{L}$ and $\mathbf{Z}^{L}$ represent the output feature maps of (S)WMSA and MLP modules of the L^{th} block, respectively. The shifted windows partition approach groups patches that do not belong to the same window into a single computational attention. This strengthens the connection between the windows, leading to further improves modeling and representation ability.

We select the basic Swin Transformer block with the window size of 7 and the patch size of 4 as our backbone network. To better fit visual tracking requirements, we make the following modifications:

1. We adjust the total strides of the Swin Transformer to 16 and reduce the number of stages from four to three. In each stage, the number of Swin Transformer blocks is set to $(2, 2, 18)$.
2. To compare qualitatively with ResNet-50, we add a convolution layer after the backbone network for upsampling operation. Although this layer may result in the loss of some important features, our experiments indicate that the WMSA backbone network still outperforms ResNet-50 [7] in terms of representation capability despite such loss during tracking.

2.2 Auxiliary Feature Enhancement Network

While the proposed WMSA backbone network can extract powerful features, it does not perform dimensional separation, and thus may be limited in its visual representation of the perceptual domain. To address this issue, we incorporated an auxiliary feature enhancement network to learn global attention across dimensions. The auxiliary network

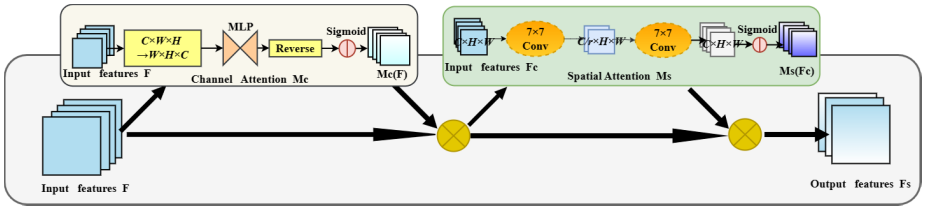

Fig. 4 The overview of the proposed auxiliary feature enhancement network.

consists of a channel attention subnetwork and a spatial attention subnetwork, as shown in Fig. 4.

The channel attention subnetwork preserves channel information using a 3-D arrangement, and uses a two-layer MLP to amplify the spatial dependence of channels across dimensions. Equation (5) shows the computational process of channel attention M_c,

$$\mathbf{F}_c = M_c(\mathbf{F}) \otimes \mathbf{F} \tag{5}$$

where $\otimes$ indicates the element-wise multiplication.

In the spatial attention subnetwork, two convolutional layers are used to fuse the spatial information. It is worth noting that in order to prevent a significant increase in parameters, we perform channel reduction with a coefficient of r in the first convolutional layer and recover the channel numbers in the second convolutional layer. Equation (6) shows the calculation process of spatial attention M_s,

$$\mathbf{F}_s = M_s(\mathbf{F}_c) \otimes \mathbf{F}_c \tag{6}$$

2.3 Tracking with Enhanced Features

During online tracking, given the first annotated frame, we use 15 different data enhancement schemes based on DiMP to create a template frame containing 15 samples. The initial template features are extracted through our proposed WMSA backbone network and then processed using the auxiliary network to obtain the final template feature maps. The filter-based prediction model [3] is initialized and continuously updated with the template feature maps. Similarly, the search candidates obtain initial search features using the same WMSA backbone network, and these features are then enhanced by the auxiliary network to obtain the final search features. The prediction model then provides the target location, which is used to update the target templates. Additionally, the oldest templates are discarded once the number reaches a certain threshold of 50.

During offline training, SwinDiMP is trained using discriminative learning loss [3]. Multiple image pairs are used for training, and a Hinge-like loss function is employed to penalize background information to improve the tracking robustness.

3. Experiments

SwinDiMP is implemented in Python using PyTorch with 2 Intel® Xeon® E5-2698 v4 CPU @ 2.2 GHz CPU with 240 GB RAM, and 4 NVIDIA® Tesla® V100 GPU with 128 GB VRAM. We utilize a modified WMSA based Transformer as our backbone network that accounts for window-to-window information interaction. Additionally, we perform auxiliary feature enhancement operations for template features to reduce dispersion of important information and amplify global interaction representation. The input size of the target template and search candidate are set to 127×127 and 255×255 pixels, respectively. The backbone network is initialized using pre-trained weights. Other parameters and experimental settings are as same as DiMP [3] and Swin Transformer [5].

Our experiments involve training the tracker on four datasets including GOT-10k [8], COCO [9] and TrackingNet [10], and evaluating it against other state-of-the-art trackers on GOT-10k and TrackingNet.

Table 1 Comparison of different backbone networks.

Tracker	GOT-10k		
	AO	$SR_{0.50}$	$SR_{0.75}$
DiMP	0.611	0.717	0.492
$DiMP_{BN}$	0.638	0.748	0.515
SwinDiMP	**0.642**	**0.769**	**0.522**

3.1 Ablation Studies

We conduct extensive performance studies of various trackers, including the baseline DiMP [3], DiMP with the proposed MSA backbone network ($DiMP_{BN}$), DiMP with the introduced feature enhancement network ($DiMP_{FE}$), and the proposed SwinDiMP, on the GOT-10k dataset. The average overlap (AO), $SR_{0.50}$ and $SR_{0.75}$ scores provided by the official toolkit were used as evaluation indices.

Backbone network: We first evaluate the effectiveness of the proposed WMSA based Transformer backbone network. We tested DiMP, $DiMP_{BN}$, and SwinDiMP. Table 1 presents the tracking results. The AO score of $DiMP_{BN}$ improved by 2.7% compared to the baseline, indicating that Transformer outperforms ResNet50 in extracting features. Meanwhile, our proposed tracker, SwinDiMP, achieved an AO score of 64.2%, improved by 3.1% compared to $DiMP_{BN}$, and demonstrated superior tracking performance.

Feature Enhancement: We also test the effect of the proposed auxiliary feature enhancement netowkr. We compared two trackers on GOT-10k, one using the auxiliary feature enhancement network on both template and search features, and the other only on the template feature. As shown in Table 2, using feature enhancement on both template and candidate yield a 1.2% improvement in AO score

Table 2 Comparison of different feature enhancement strategies.

Tracker	Feature Enhancement		GOT-10k		
	Template	Candidate	AO	$SR_{0.50}$	$SR_{0.75}$
DiMP	-	-	0.611	0.717	0.492
$DiMP_{FE}$	✓	-	0.612	0.718	0.492
$DiMP_{FE}$	✓	✓	0.623	0.723	0.512
SwinDiMP	✓	✓	**0.642**	**0.769**	**0.522**

Table 3 Comparison of different convolutional connections.

Tracker	Convolutional Connection		GOT-10k		
	Conv 1×1	Conv 3×3	AO	$SR_{0.50}$	$SR_{0.75}$
$DiMP_{BN}$	✓	-	0.628	0.738	0.504
$DiMP_{BN}$	-	✓	0.638	0.748	0.515
SwinDiMP	-	✓	**0.642**	**0.769**	**0.522**

Table 4 Comparisons with state-of-the-art trackers on GOT-10k.

Tracker	KYS	Ocean	SiamRPN++	D3S	SiamTPN	SiamCAR	DiMP	SwinDiMP
AO	0.636	0.611	0.518	0.597	0.576	0.569	0.611	**0.642**
$SR_{0.50}$	0.751	0.721	0.618	0.676	0.686	0.670	0.717	**0.769**
$SR_{0.75}$	0.515	0.473	0.329	0.462	0.441	0.415	0.492	**0.522**

compared to using it on only one branch. Therefore, we chose to use the auxiliary feature enhancement network on both template and candidate in SwinDiMP, which achieved the best tracking results.

Convolutional Connection: We further investigate the effectiveness of the backbone network with a convolution connection for feature enhancement. We tested two different convolutional kernel sizes, 1×1 and 3×3, and found that the 3×3 kernel size yield a 1% higher AO score and higher tracking accuracy, as shown in Table 3. Therefore, we used a convolutional kernel size of 3×3 in our experiments.

In summary, our SwinDiMP tracker outperforms other variants of DiMP on the GOT-10k dataset, demonstrating the effectiveness of the proposed backbone network and feature enhancement network.

3.2 Comparison with the State-of-the-Art

We compare our proposed SwinDiMP tracker with the state-of-the-art tracking approaches, including KYS [11], Ocean [12], SiamRPN++ [13], D3S [14], SiamTPN [15] SiamCAR [16], and DiMP [3], on two public large-scale challenging benchmarks.

GOT-10k [8]: GOT-10k is a comprehensive tracking dataset that covers 560 common outdoor sports objects. The comparison relust of the participants are shown in Table 4. Our SwinDiMP outperforms all other trackers with the highest AO score of 64.2%. Moreover, compared to the baseline DiMP, SwinDiMP improves the AO, $SR_{0.50}$, and $SR_{0.75}$ scores by 0.6%, 1.8%, and 0.7%, respectively.

TrackingNet [10]: TrackingNet dataset contains various natural scenes, with diverse frame rates, resolutions, background surrounding, and object classes. As shown in Table 5, our tracker achieves a 74.6% success rate (SR), which is 0.6% higher than the baseline DiMP. Moreover, SwinDiMP outperforms the other state-of-the-art trackers in the evaluation.

Table 5 Comparisons with state-of-the-art trackers on TrackingNet.

Tracker	KYS	Ocean	SiamRPN++	D3S	SiamTPN	SiamCAR	DiMP	SwinDiMP
SR	0.740	0.692	0.733	0.728	0.708	0.740	0.740	**0.746**
Precision	0.688	0.687	0.694	0.664	0.651	0.684	0.687	**0.692**
Pre_{Norm}	0.800	0.794	0.800	0.768	0.771	0.804	0.801	**0.810**

4. Conclusion

This work presents SwinDiMP, a simple yet effective framework for robust visual tracking. While it eliminates most of the specialization in current Siamese trackers on two tracking datasets, architecture and training techniques can be optimized for further performance improvements.

References

[1] P. Gao, Q. Zhang, F. Wang, L. Xiao, H. Fujita, and Y. Zhang, "Learning reinforced attentional representation for end-to-end visual tracking," Information Sciences, vol.517, pp.52–67, 2020.
[2] L. Bertinetto, J. Valmadre, J.F. Henriques, A. Vedaldi, and P.H.S. Torr, "Fully-convolutional Siamese networks for object tracking," ECCV, pp.850–865, 2016.
[3] G. Bhat, M. Danelljan, L.V. Gool, and R. Timofte, "Learning discriminative model prediction for tracking," ICCV, pp.6182–6191, 2019.
[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," NeurIPS, vol.30, 2017.
[5] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," ICCV, pp.10012–10022, 2021.
[6] J. Thangavel, T. Kokul, A. Ramanan, and S. Fernando, "Transformers in single object tracking: An experimental survey," arXiv preprint arXiv:2302.11867, 2023.
[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," CVPR, pp.770–778, 2016.
[8] L. Huang, X. Zhao, and K. Huang, "Got-10k: A large high-diversity benchmark for generic object tracking in the wild," IEEE Trans. Pattern Anal. Mach. Intell., vol.43, no.5, pp.1562–1577, 2021.
[9] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C.L. Zitnick, "Microsoft COCO: Common objects in context," ECCV, pp.740–755, 2014.
[10] M. Müller, A. Bibi, S. Giancola, S. Alsubaihi, and B. Ghanem, "TrackingNet: A large-scale dataset and benchmark for object tracking in the wild," ECCV, pp.310–327, 2018.
[11] G. Bhat, M. Danelljan, L. Van Gool, and R. Timofte, "Know your surroundings: Exploiting scene information for object tracking," ECCV, pp.205–221, 2020.
[12] Z. Zhang, H. Peng, J. Fu, B. Li, and W. Hu, "Ocean: Object-aware anchor-free tracking," ECCV, pp.771–787, 2020.
[13] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of Siamese visual tracking with very deep networks," CVPR, pp.4282–4291, 2019.
[14] A. Lukežič, J. Matas, and M. Kristan, "D3S — A discriminative single shot segmentation tracker," CVPR, pp.7133–7142, 2020.
[15] D. Xing, N. Evangeliou, A. Tsoukalas, and A. Tzes, "Siamese transformer pyramid networks for real-time UAV tracking," WACV, pp.2139–2148, 2022.
[16] Y. Cui, D. Guo, Y. Shao, Z. Wang, C. Shen, L. Zhang, and S. Chen, "Joint classification and regression for visual tracking with fully convolutional Siamese networks," International Journal of Computer Vision, vol.130, no.2, pp.550–566, 2022.