Transformer Tracking with Cyclic Shifting Window Attention

Zikai Song[1] Junqing Yu[1*] Yi-Ping Phoebe Chen[2] Wei Yang[1]
[1]Huazhong University of Science and Technology, China [2]La Trobe University, Australia
{skyesong, yjqing, weiyangcs}@hust.edu.cn, phoebe.chen@latrobe.edu.au

Abstract

Transformer architecture has been showing its great strength in visual object tracking, for its effective attention mechanism. Existing transformer-based approaches adopt the pixel-to-pixel attention strategy on flattened image features and unavoidably ignore the integrity of objects. In this paper, we propose a new transformer architecture with multi-scale cyclic shifting window attention for visual object tracking, elevating the attention from pixel to window level. The cross-window multi-scale attention has the advantage of aggregating attention at different scales and generates the best fine-scale match for the target object. Furthermore, the cyclic shifting strategy brings greater accuracy by expanding the window samples with positional information, and at the same time saves huge amounts of computational power by removing redundant calculations. Extensive experiments demonstrate the superior performance of our method, which also sets the new state-of-the-art records on five challenging datasets, along with the VOT2020, UAV123, LaSOT, TrackingNet, and GOT-10k benchmarks. Our project is available at https://github.com/SkyeSong38/CSWinTT.

1. Introduction

Visual object tracking (VOT) is one of the fundamental problems in computer vision research with a wide range of applications in video surveillance, autonomous vehicles, human-machine interaction, and others. It aims to estimate the position of a target object in each video frame, commonly represented as a bounding box encapsulating the target. The target object is given as a template in the initial frame, and the tracker is required to extract proper features about the target and localize the target in the following frames. Most of the popular trackers [1, 22, 23, 37, 41] adopt the Siamese network structure, which conducts tracking by calculating the similarity between the template and search region in the current frame. The similarity metric

* Corresponding author

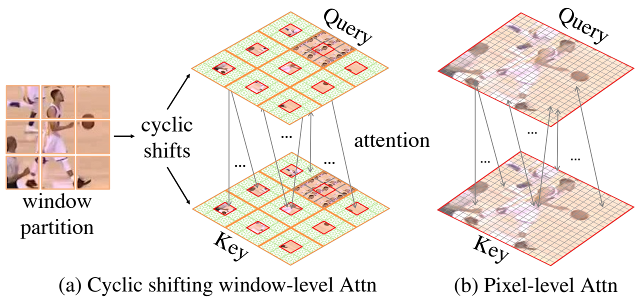

Figure 1. (a) The proposed approach firstly achieves the window-level attention between query and key through window partitioning, and then applies cyclic shifts for each window (from the base sample in the red box to generated samples surrounded by orange boxes) to greatly extend the number of window samples, while maintaining the integrity of objects. (b) Previous transformers produce pixel-level attention, which weakens the positional information between pixels and ignores the integrity of objects.

of cross-correlation used in Siamese trackers is prone to lose much semantic information for it is a single-level linear computational process. This deficiency can be well tackled by using the attention mechanism to learn the global context. Recently, transformer-based approaches [6, 12, 25, 39] have reported new state-of-the-art performance on image recognition, object detection, and semantic segmentation benchmarks. This is no wonder as transformer [36] has a powerful cross-attention mechanism to reasoning between patches [18]. Particularly, transformer trackers [7,40,45,50] have shown their great strength by introducing the attention mechanism to enhance and fuse the features of the target and the tracked object. However, we observe that these transformer trackers simply put the flattened features of the template and search region into pixel-level attention, each pixel of a flattened feature (Query) matches all pixels of another flattened feature (Key) in a complete and disordered manner, as shown in Figure 1b. This pixel-level attention destroys the integrity of the target object and leads to information loss of relative positions between pixels.

In this paper, we propose a novel multi-scale cyclic shifting window transformer for visual object tracking to further lift pixel-level attention to window-level attention, calculating attention between indivisible windows by treating each window as a whole keeps the location information within the window. The proposed method is inspired by the seminal work of the Swin Transformer [25], which adopts a hierarchical transformer structure by starting from small-sized patches and gradually increasing the size through merging to achieve a broader receptive field. Different from the Swin Transformer, we calculate the cross-window attention between the template and search region directly, which helps to discriminate the target from the background by ensuring the integrity of the object. Further, we propose a multi-head multi-scale attention where each head of the transformer measures the relevance among partitioned windows at a specific scale. The key idea here is to apply a cyclic shifting strategy on each window, as shown in Figure 1a, for generating more accurate attention results. To address performance drop around boundaries caused by the cyclic shifting operation, we design a spatially regularized attention mask which turns out to be very effective in alleviating the boundary artifacts. Finally, we present some efficient computation strategies to avoid redundant computation introduced by multi-scale cyclic shifting windows, which greatly reduce the time and computational cost. Extensive experiments demonstrate that our tracker performs remarkably better than other state-of-the-art algorithms.

To summarize, our main contributions include:

1. We propose a novel transformer architecture with multi-scale cyclic shifting window attention for visual object tracking, uplifting the original pixel-level attention to the new deliberately designed window-level attention. The cross-window attention ensures the integrity of the tracking object, and the cyclic shifts bring greater accuracy by expanding window samples.

2. We design a spatially regularized attention mask and some computational optimization strategies to improve the accuracy and speed of the window attention. Specifically, a spatially regularized attention mask is used to address performance drop around boundaries caused by the cyclic shifts, and we propose three computational optimization strategies to remove redundant computations.

2. Related Work

Visual object tracking. Existing visual object tracking approaches can be roughly divided into two categories, the Correlation Filter (CF) based trackers and Deep Neural Network (DNN) based trackers. CF based approaches [4, 5, 9, 11, 16, 20, 42] exploit the convolution theorem and train a filter in the Fourier domain that maps known target images to the desired output. The filter is learned through circular shifting patches around the target object to discriminate background against the target. DNN based trackers refer to the methods adopting deep neural networks in the tracking process. Many methods [19, 32, 33] treat the tracking task as a basic recognition task, i.e., using a convolutional backbone network to extract features and locate the target by classification heads in the form of fully connected layers.

Recent years, tracking algorithms that adopt a Siamese network structure [1, 2, 8, 10, 14, 22, 23, 29, 41, 44, 46, 48, 52] have shown great success. A Siamese network usually consists of two branches, one for template and the other for search regions, and similarities between them are reported through cross-correlations. However, such a strategy is unable to effectively explore the semantic correlation between template and search regions. This issue leads to the further exploration of using the powerful cross-attention mechanism of transformer structure for object tracking.

Vision transformers. Vaswani *et al.* [36] propose the very first Transformer structure for handling long-range dependencies in Natural Language Processing (NLP). The basic block in a transformer is the attention module, which takes a sequence as input and measures the relevance of different parts of the sequence, aggregating the global information from the input sequence. Transformer not only conducts the self-attention within a single input but also calculates the cross-attention between different inputs. ViT [12] first introduces transformer to image recognition tasks. Ever since, transformer has been widely applied in image classification [12, 25], object detection [6], semantic segmentation [39], visual object tracking [7, 40, 43, 45] and etc.

The seminal work of Swin Transformer [25] proposes an effective hierarchical architecture with shifted windows and achieves the state-of-the-art performance on COCO object detection [24], and ADE20K semantic segmentation [51]. Though our approach is inspired by the Swin Transformer, we have three fundamental differences: (1) where attention is applied are different. Swin Transformer partitions the image into windows and then conducts pixel-level attention inside each window, while we do window partitioning in feature maps, and calculate attention between windows by treating each window as a whole. (2) multi-scaling strategy is different. Swin transformer uses the same window size in one layer and merges windows to form a larger window in deeper layers. In contrast, we use windows with different sizes as heads for multi-scale matching. (3) window shifting is applied differently. Swin Transformer shifts the whole feature map, in order to exchange information and provide connectivity between different windows. We apply independent cyclic shifts in each window in a non-exchangeable way. Additionally, in contrast to Swin Transformer, where each window is shifted only once, in our algorithm each window is shifted multiple times depending on its size.

Recently, transformer-based visual object tracking methods have become more and more popular. TrDiMP [40] sep-

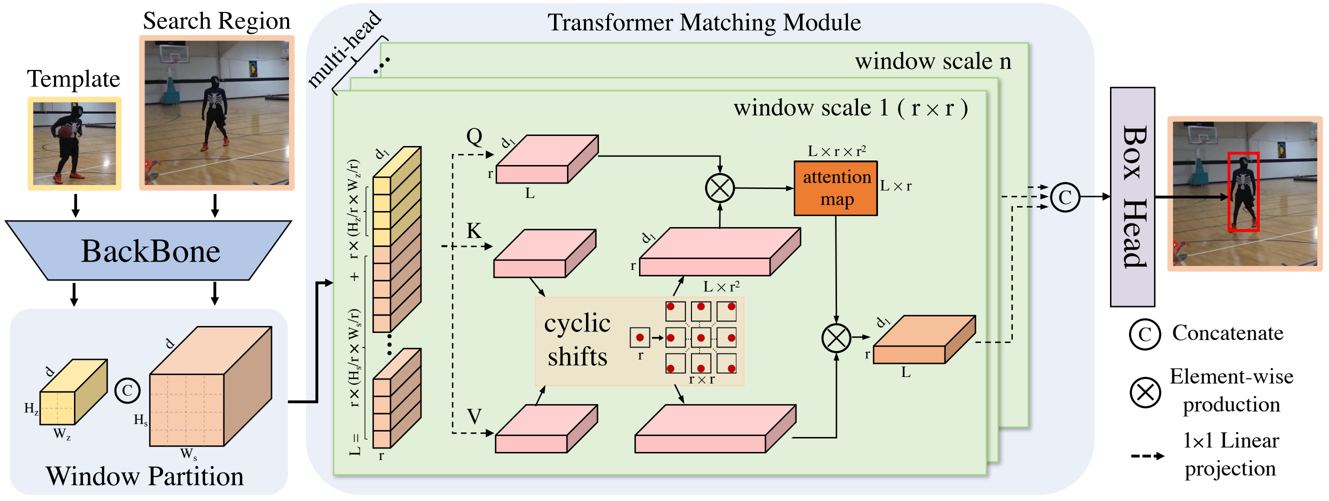

Figure 2. An overview of proposed tracking architecture. Given a template image in the first frame and a search region in the subsequent frame, we extract deep feature maps through a backbone network. Then these two features are partitioned into small windows and flattened as a window sequence. The transformer matching module applies window-level attention to the window sequence. Specifically, the Key(K)-Value(V) pair in the transformer employ the cyclic shifting strategy to generate a great number of samples. Then the transformer outputs the fused features that contain the deep matching information between template and search region, these fused features are passed through the bounding box prediction head to obtain the final tracking result.

arates the encoder-decoder transformer into two Siamese-like branches, the encoder reinforces the template features and the decoder propagates the tracking cues from previous templates to the current frame. TransT [7] proposes a feature fusion network and employs an attention mechanism to combine the features of template and search region. This feature fusion network consists of an ego-context augment module based on self-attention and a cross-feature augment module based on cross-attention. STARK [45] develops a spatial-temporal architecture based on the encoder-decoder transformer, the encoder learns the relationship between template and search region and the decoder learns a query embedding to predict the target positions. Moreover, STARK introduces a corner-based prediction head used for estimating the bounding box and a score head for controlling the updates of the template image. Most of the previous tracking algorithms such as [40, 45] use encoder-decoder structure to enhance or fuse the features, while we consider the transformer as a feature matching module to calculate the similarity between template and search region. Moreover, previous approaches use the transformer naively and do touch the attention mechanism within. On the contrary, we carefully design a multi-head multi-scale window-level attention transformer with the cyclic shifting strategy, to fully exploit the transformer structure for object tracking.

3. Method

In this section, we present our multi-scale **c**yclic **s**hifting **win**dow **t**ransformer **t**racker, namely CSWinTT. We use the transformer as a matching module for measuring the relevance between template and search region to fully exploit the powerful cross-attention capabilities of the transformer.

The tracking architecture is visualized in Figure 2, which consists of three major components: a feature extraction backbone, a transformer matching module, and a bounding box estimation head. We choose the ResNet-50 [15] as our backbone for feature extraction, which takes a pair of image inputs, i.e., the template image and the search region image. The pair of output features are then partitioned into window sequences and fed into the transformer matching module. The matching module concatenates the two window sequences and sends them to the multi-head 6-layer transformer. The multi-head transformer uses a specific window size for each head for scale adaption. Finally, the outputs of each transformer head are concatenated together, passed through the corner-based box estimation head as in [45] to get the result bounding box.

3.1. Multi-Scale Cyclic Shifting Window Attention

Multi-scale window partition. Give template patch z in the initial frame and an image s for search region. We pass z and s through the backbone and a bottleneck layer, obtaining the feature map $f_z \in \mathbb{R}^{d \times H_z \times W_z}$ and $f_s \in \mathbb{R}^{d \times H_s \times W_s}$ respectively. Then we extract patches of shape $r_i \times r_i$ from f_z and f_s in head i, where d_i represents the number of channels of head i. The total number of template windows is $N_z^i = \frac{H_z}{r_i} \times \frac{W_z}{r_i}$ and for search region windows is $N_s^i = \frac{H_s}{r_i} \times \frac{W_s}{r_i}$. After window extraction, the feature maps are reshaped to window sequences $f_z^i \in \mathbb{R}^{N_z^i \times d_i \times r_i \times r_i}$ and $f_s^i \in \mathbb{R}^{N_s^i \times d_i \times r_i \times r_i}$. The two window sequences are then concatenated along the spatial dimension and generate f_c^i with $(N_z^i + N_s^i) \times d^i$. Then query-key-value attention mechanism is applied with query $\mathbf{Q_i}$, key $\mathbf{K}_i$ and value $\mathbf{V}_i$. Key-value pairwise similarities are then calculated and fused using a multi-head attention mechanism as follows:

Multi-head attention. Multi-head attention is the fundamental component in our architecture. As described in [36], given queries $\mathbf{Q}$, keys $\mathbf{K}$, and values $\mathbf{V}$. The multi-head attention is computed as:

$$\begin{aligned} \text{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V}) &= \text{Concat}(\mathbf{H}_1,\ldots,\mathbf{H}_{n_h}) \\ \text{where} \quad \mathbf{H}_i &= \text{Attention}(\mathbf{Q}_i,\mathbf{K}_i,\mathbf{V}_i) \\ &= \text{softmax}(\frac{\mathbf{Q}_i\mathbf{K}_i^T}{\sqrt{d_k}})\mathbf{V}_i \end{aligned} \tag{1}$$

where n_h is the number of heads, and d_k is the dimension of key. For a clearer description of the post-order steps, we define the attention score as:

$$\text{AttnScore}(\mathbf{Q},\mathbf{K}) = \frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}} \tag{2}$$

Cyclic shifting strategy. Compared to the pixel-level attention, one problem of our window attention is the resolution of attention map reduces from $\mathbb{R}^{(H_zW_z+H_sW_s)^2}$ to $\mathbb{R}^{(N_z^i+N_s^i)^2}$. This will lead to a coarser similarity score, and it is hard to fuse the output of each head for the attention generated by different heads does not have the same size.

To address the aforementioned problems, we propose a cyclic shifting strategy on the proposed window-level attention. It enhances the effectiveness of cross-window attention while preserving positional information and the integrity of objects, as shown in Figure 3. Within a specific head, consider a window with size $r \times r$ referred to as the base sample. We define the shift operator shift(x, y) of the base sample as translating the sample by x pixels in the horizontal direction, and y pixels in the vertical direction. Our cyclic shifts of a base sample then are performed at a single-pixel distance and move the sample into bottom-right directions with boundaries being warped back to the top-left. The operation shift$(x, y), x, y \in [-r+1, r-1]$ generates $r \times r$ into $(2r-1)^2$ samples for the base sample with window size $r \times r$. Obviously, these cyclic shifts generate a lot of duplicates, we will discuss how to effectively remove duplicate computations in the section 3.2.

3.2. Efficient Computation

Spatially regularized attention mask. In practice, we find the shifted samples near the center contribute more to the final attention. This is reasonable as samples close to the boundaries are more likely to break the integrity and position information of the tracking object in the window. Hence we design a weighting scheme applied as a form of attention mask $\mathbf{M}$ in the transformer, as shown in Figure 4. The spatial weights of the mask penalize samples depending on their spatial locations, the formula for weight generation is expressed in 3. The further away the generated sample is from the base sample, the larger the penalty is and the weight is smaller.

$$\mathbf{M}(x,y) = -(\frac{x}{r})^2 - (\frac{y}{r})^2, \quad x,y \in [-r+1, r-1] \tag{3}$$

The spatial attention mask $\mathbf{M}$ is directly added onto the attention score in 2.

Recall that the sizes of query $\mathbf{Q}_i$ and key $\mathbf{K}_i$ in the i-th head are $\mathbb{R}^{(N_z+N_s)\times d_i \times r_i \times r_i}$ respectively. With $r_i \times r_i$ as the window size, and each window is expanded into $[(2r_i-1)^2 \times d_i \times r_i \times r_i]$ with cyclic shifts. The size of attention score on each window is $[(2r_i-1)^2 \times (2r_i-1)^2]$, this is formed by the window of query and key. Since the query and key are also cyclic shifting, we add the spatial weights of size $[(2r_i-1) \times (2r_i-1)]$ to the last dimension of attention score (dimension of keys), achieving the effect of positional penalty.

Computational optimization. Intuitively, the cyclic shifts increase the computational cost greatly, especially when the window size is large. To achieve computational efficiency, we optimize in three ways: (i) eliminating the cyclic shifts of the Query; (ii) halving the duplicated shifting periods;

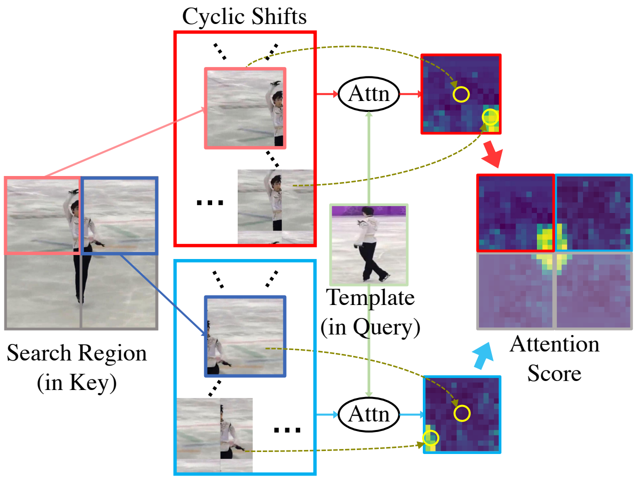

Figure 3. Cyclic shifting window attention. We illustrate that when the search region in the Key is partitioned into 4 windows and the template in the Query becomes one window. The above two windows in the search region are selected as examples to do attention with the template after cyclic shifts, and attention scores are obtained respectively, then they are combined to form the final attention score.

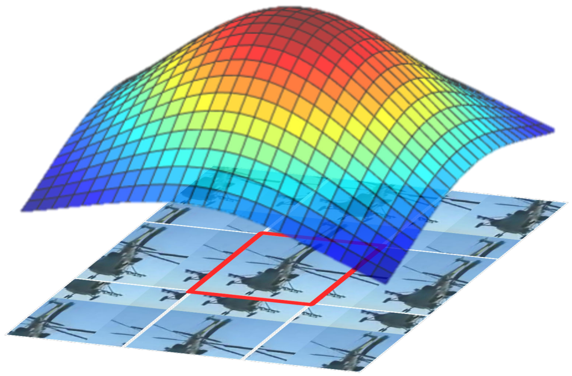

Figure 4. Visualization of the spatial regularization weights designed to alleviate the boundary artifacts. The red box in the center is the base sample, generated samples are penalized more if they are further from the center base sample (i.e., the smaller the value of their applied attention mask).

and (iii) adopting the programming optimization for matrix translation.

Suppose we have $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ of size (H, W, d). Standard transformer flattens the features to (HW, d), two parts account for the time cost of attention computation are the attention score computation $O(HW \times dHW)$ and fusion feature computation $O(HW \times HW \times d)$. After applying the cyclic shifts, the size of $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ are $(\frac{H}{r}, \frac{H}{r}, 2r-1, 2r-1, r, r, d)$ with r as window size, and the complexity of computing the attention score increases to $O((\frac{H}{r}\frac{W}{r}(2r-1)(2r-1))^2 \times r^2 d)$. We observe if Q and K perform the same shifting, computing attention scores is meaningless, so we just need to perform a cyclic operation on K and keep Q unchanged to achieve the same effect. In addition, note that the cyclic generated samples to the bottom-right and the top-left directions are repeated, we reduce the number of shifting periods by half for better efficiency. And we also apply a programming trick to improve the tracking speed, which is using permutations of matrix coordinates to perform cyclic shifts instead of direct translations on the matrix.

3.3. Tracking with Window Transformer

Multi-scale window transformer facilitates the tracking process by conducting accuracy-aware attention with windows at different scales. Therefore, the choice of the window sizes is extremely important. In our implementation, we set the number of heads n_h to 8 with window size $r_i = [1, 2, 4, 8, 1, 2, 4, 8]$ for head i. Notice the second half of the heads have the same window size, that's because we adopt feature map translation which displaces the backbone feature of the search image by $(\frac{r_i}{2}, \frac{r_i}{2})$ pixels. In this way, when the windows are partitioned in a non-overlapping manner, the contents of windows are complemented by each other to avoid the situation that the object has been segmented all the time.

In further, to improve the robustness of the tracking algorithm, we use two templates of the same size as the input of the transformer. One of which is fixed using the initial template, the other is online updated to the latest tracking result with high confidence, a score head is employed to control the updates, as designed in STARK [45].

In the training stage, we use the L1 loss and the generalized IoU loss [34] to train the overall architecture in an end-to-end manner. During the inference, the template image and its corresponding backbone features are initialized in the first frame, and the search region is used as the input to the tracker during the tracking process in subsequent frames, with the predicted bounding box returned by the network as the final result.

4. Experiments

4.1. Implementation Details

We train our model on the LaSOT [13], GOT-10k [17], and TrackingNet [31] datasets. The image pairs are directly sampled from the same sequence and common data augmentation operations including brightness jitter and horizontal flip are applied. The size of the input template is 128$\times$128 pixels, the search region is 5^2 times of the target box area and further resized to 384$\times$ 384 pixels. We use ResNet-50 [15] as the backbone, the parameters of which are initialized with ImageNet pretrained [35] model. Other parameters in our model are initialized with Xavier Uniform. We use the $\lambda_{l1} = 5$ and $\lambda_{giou} = 2$ as the loss weight for l1 loss and giou loss [34]. The AdamW optimizer [26] is employed with initial learning rates of 1e-5 and 1e-4 for backbone parameters and other parameters, respectively, and weight decay is set to 1e-4 for every 10 epochs after 500 epochs. We train our model on two Nvidia Tesla T4 GPUs for a total of 600 epochs, each epoch uses 4×10^4 images. The mini-batch size is set to 64 images with each GPU hosting 32 images. The training process of the update module is the same as [45]. Our approach is implemented in Python 3.7 using PyTorch 1.6. CSWinTT operates about 12 frames per second (FPS) on a single GPU during the online tracking process.

4.2. State-of-the-art Comparison

We compare our proposed CSWinTT algorithm with the state-of-the-art trackers on five tracking benchmarks, including UAV123 [30], LaSOT [13], TrackingNet [31], GOT-10k [17], and VOT2020 [21].

UAV123 [30]: UAV123 gathers an application-specific collection of 123 sequences and captures from unmanned aerial vehicles video dataset. It adopts the Area Under the Curve (AUC) and Precision (P) as the evaluation metrics.

Table 1. Comparisons on four tracking benchmarks. The **red**, green and blue indicate performances ranked at first, second, and third places

Method	Year	UAV123 [30]		LaSOT [13]			TrackingNet [31]			GOT-10k [17]		
		AUC	P	AUC	P_{Norm}	P	AUC	P_{Norm}	P	AO	$SR_{0.5}$	$SR_{0.75}$
SiamFC [1]	2016	49.2	72.7	33.6	42.0	33.9	57.1	66.3	53.3	34.8	35.3	9.8
ECO [9]	2017	52.5	74.1	32.4	33.8	30.1	55.4	61.8	49.2	31.6	30.9	11.1
ATOM [8]	2019	61.7	82.7	51.5	57.6	50.5	70.3	77.1	64.8	55.6	63.4	40.2
DiMP [2]	2019	64.2	84.9	57.7	66.4	57.9	74.0	80.1	68.7	61.1	71.7	49.2
SiamRPN++ [22]	2019	64.2	84.0	49.6	56.9	49.1	73.3	80.0	69.4	51.7	61.6	32.5
SiamFC++ [44]	2020	61.8	80.4	54.4	62.3	54.7	75.4	80.0	70.5	59.5	69.5	47.9
D3S [27]	2020	-	-	-	-	-	72.8	76.8	66.4	59.7	67.6	46.2
MAML [38]	2020	-	-	52.3	-	53.1	75.7	82.2	72.5	-	-	-
SiamAttn [46]	2020	65.0	84.5	56.0	64.8	-	75.2	81.7	-	-	-	-
KYS [3]	2020	-	-	55.4	63.3	55.8	74.0	80.0	68.8	63.6	75.1	51.5
PrDiMP [10]	2020	66.6	87.2	59.9	68.8	60.8	75.8	81.6	70.4	63.4	73.8	54.3
Ocean [49]	2020	62.1	82.3	51.6	60.7	52.6	69.2	79.4	68.7	61.1	72.1	47.3
SiamRCNN [37]	2020	-	-	64.8	72.2	-	81.2	85.4	80.0	64.9	72.8	59.7
SiamGAT [14]	2021	64.6	84.3	53.9	63.3	53.0	-	-	-	62.7	74.3	48.8
AutoMatch [47]	2021	64.4	83.8	58.2	67.5	59.9	76.0	82.4	72.5	65.2	76.6	54.3
TrDiMP [40]	2021	67.0	87.6	64.0	73.2	66.6	78.4	83.3	73.1	68.8	**80.5**	59.7
TransT [7]	2021	68.1	87.6	64.9	73.8	69.0	81.4	**86.7**	**80.3**	67.1	76.8	60.9
STARK-ST50 [45]	2021	69.2	88.2	66.0	**75.5**	70.8	81.3	86.1	-	68.0	77.7	62.3
CSWinTT	Ours	**70.5**	**90.3**	**66.2**	75.2	**70.9**	**81.9**	**86.7**	79.5	**69.4**	78.9	**65.4**

Table 2. Result comparisons on VOT2020 [21], where trackers only predict bounding boxes rather than reporting masks.

	EAO↑	Accuracy↑	Robustness↑
KCF [16]	0.154	0.407	0.430
SiamFC [1]	0.179	0.418	0.502
CSR-DCF [28]	0.193	0.406	0.582
ATOM [8]	0.271	0.462	0.734
DiMP [2]	0.274	0.457	0.740
UPDT [4]	0.278	0.465	0.755
TrDiMP [40]	0.300	0.471	0.782
TransT [7]	0.293	0.477	0.754
STARK [45]	0.303	**0.481**	0.775
CSWinTT(Ours)	**0.304**	0.480	**0.787**

The precision is used to measure the center distance and the AUC plot computes the intersection-over-union (IoU) score between the estimated bounding box and the ground-truth. As shown in Table 1, where the previous state-of-the-art trackers such as TrDiMP [40], TransT [7], and START [45] are included for comparison, note that STARK-ST50 is chosen for the reason that it uses the same ResNet-50 backbone as our algorithm, which can more fairly compare the performance of transformer structure. Our CSWinTT outperform the aforementioned methods by a considerable margin and exhibits very competitive performance (70.5% AUC and 90.3% Precision) when compared to the best previous tracker STARK (69.2% AUC and 88.2% Precision)

LaSOT [13]: LaSOT is a large-scale long-term dataset including 1400 sequences and distributed over 14 attributes, the testing subset of LaSOT contains 280 sequences with an average length of 2448 frames. Methods are ranked by the AUC, Precision, and Normalized Precision (P_{Norm}). The evaluation results of compared tracking algorithms are shown in Table 1. Our model achieves the top-rank AUC score (66.2%) and Precision score (70.9%), which outperforms the previous best result by STARK-ST50, and also surpasses the other two transformer trackers TransT [7]/TrDiMP [40] for 1.3%/2.2% AUC score, respectively.

TrackingNet [31]: TrackingNet is a large-scale tracking dataset consisting of 511 sequences for testing. The evaluation is performed on the online server. 1 shows that, compared with SOTA models, our CSWinTT performs better visual tracking quality and ranks at the first in AUC score of 81.9% and normalized precision of 86.7%. The specific gain is 0.7% relative improvement of the AUC score when compared with the TransT [7], which represents the previous best algorithm on this benchmark.

GOT-10k [17]: GOT-10k is a large-scale dataset containing over 10k videos for training and 180 for testing. It forbid the trackers to use external datasets for training. We follow this protocol by retraining our trackers using only the GOT10k train split. As can be seen from Table 1, among previous transformer trackers, TrDiMP [40] and STARK-ST50 [45] provides the best performance, with an AUC score of 68.8% and 68.0%. Our approach has remarked improvement and obtains an AUC score of 69.4%, significantly outperforming the best existing tracker (TrDiMP) by 0.6%.

VOT2020 [21]: VOT2020 benchmark contains 60 challenging videos. The performance on this dataset is evaluated using the expected average overlap (EAO), which takes both accuracy (A) and robustness (R) into account. In addition, a new anchor-based evaluation protocol is proposed in VOT2020, the segmentation mask is adopted as the groundtruth. However, since our algorithm does not output a segmentation mask, trackers only predict bounding boxes are chosen as the comparisons to ensure a fair evaluation. It can be seen from the data in Table 2 that CSWinTT obtains an EAO of 0.304, ranking first in previous trackers.

4.3. Ablation Study

We conduct ablation analysis to evaluate the different components in our CSWinTT and evaluate the performance of diverse window sizes using the UAV123 dataset [30]. Besides, we show the superiority of the three previously mentioned computational optimization strategies.

Table 3. Ablation Study on UAV123 [30]. **Win** represents the multi-scale window transformer. **CS** denotes the proposed cyclic shifting strategy. **SR** means to apply the spatially regularized attention mask. **Pos** represents the relative position encoding.

#	Win	CS	SR	Pos	AUC	Prec.
1	Original Transformer				66.2	86.6
2	✓				54.4	70.8
3	✓	✓			69.7	89.2
4	✓	✓	✓		70.1	89.8
5	✓	✓		✓	69.8	89.6
6	✓	✓	✓	✓	70.5	90.3

Effects of different components in our method. We evaluate the effect of components including multi-scale window attention (Win), cyclic shifts (CS), spatially regularized attention mask (SR), and relative position encoding (Pos) employed in our method. The ablation study result is shown in Tab. 3, #1 represents the performance of the original transformer. We can see that window-level attention alone (#2) is very ineffective as it greatly reduces the resolution of the attention mechanism, however, combining window-level attention with the cyclic shifting strategy can handle this drawback. It can be seen in #3, there is a 15.3% improvement in the AUC score after applying the cyclic shifts, and it outperforms the original transformer by 3.5%, which illustrates that the cyclic shifting strategy plays a key role on the window-level attention. #4 shows that the AUC score can be improved by 0.4% when employing the spatially regularized mask to cyclic shifting samples, which demonstrates that spatial regularity can alleviate the boundary artifacts to a certain extent and improve the performance of window attention. In addition, we test the effectiveness of relative position encoding in our method following the way of [25]. The performance improves by 0.1% when the relative position encoding (#5) is used, the small improvement indicates that the position encoding in window-level attention is not very important, and confirms that window-level attention itself contains rich position information.

Table 4. Comparison between the different window sizes on UAV123 [30]. The first four items are adopting the same window size in each head. Multi-scale denotes the proposed CSWinTT.

Window Size	AUC	Prec.
1×1	66.2	86.6
2×2	68.3	87.9
4×4	70.0	89.6
8×8	69.3	89.2
Multi-scale	70.5	90.3

Different window sizes for our transformer. To explore the performance of diverse window sizes on cyclic shifting window attention, we designed a quantitative analysis experiment as shown in Table 4. The first four rows indicate that the same window size is used in all 8 heads in the case that cyclic shifting strategy is employed. It can see from the experimental results that the highest 70.0% AUC score is obtained in size 4×4 when using a single window size, as a matter of fact, the performance is really closer for all windows sizes. When adopting the multi-scale window size, the best AUC score of 70.5 is achieved, demonstrating that multi-scale windows can fuse information from different scales to improve the performance of the tracker.

Table 5. Comparison about the tracking speed for three computation optimization. **RMQ** represents removing the cyclic shifts of Query. **Peri** denotes halving shifting periods. **Prog** means adopting the programming optimization for matrix translation.

#	RMQ	Peri	Prog	Speed(FPS)
1				1.0
2	✓			8.2
3	✓	✓		10.9
4	✓	✓	✓	12.4
5	Original Transformer			14.9

Computation optimization and speed analysis. Cyclic shifting strategy brings a large computational burden, we improve the tracking speed by applying some optimization strategies, including removing the cyclic shifts of Query (RMQ), halving the shifts periods (Peri), and adopting the programming optimization for matrix translation (Prog), Table 5 shows the effect of each optimization method. The tracking speed is around 1 FPS with no optimization

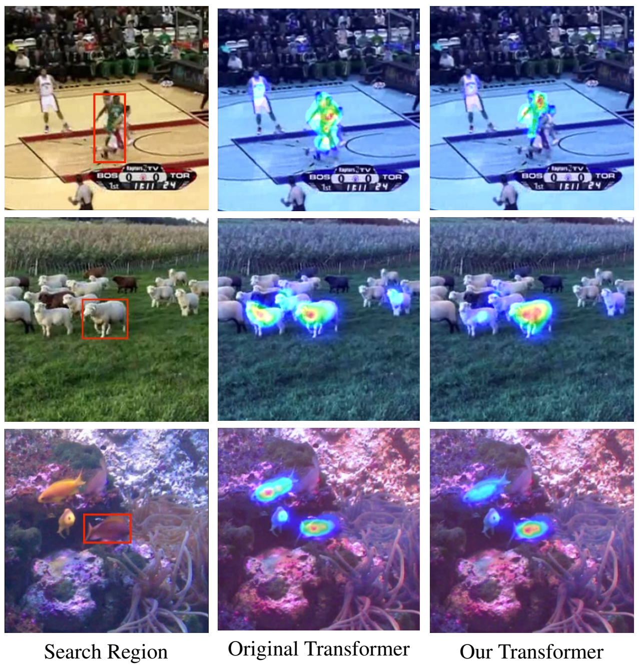

Figure 5. Visual heat maps of the attention obtained by the original transformer (middle), and our proposed transformer (right). The red box indicates the target object in the search region (left).

adopted, as shown in #1, which is almost an unusable state. With the cyclic shifts of Query are removed (#2), the tracking speed is greatly improved to 8.2 FPS, and it can be further improved by halving the shifts periods (#3). In addition, we also apply a PyTorch programming trick to use permutations of matrix coordinates to perform cyclic shifts instead of direct translations on the matrix, which also improves the tracking speed to some extent as shown in #4. Due to the absolute amount of computation introduced by the cyclic shifting window attention, the computing efficiency of our method is not as good as the original transformer (#5), but a satisfactory tracking speed of 12.4 FPS is achieved after our computational optimization.

4.4. Qualitative Analysis

Figure 5 shows the visual heat map of the attention, which exhibits the attention score of the last layer in the transformer matching module. The red area in the heat map indicates a high attention degree, while the blue area indicates a low attention degree. The first row shows the situation where the target object is obscured, the second and third rows show the scenario where the target is surrounded by similar distractors. From the visualization we can see that, compared to the pixel-level attention, the cyclic shifting window attention has a stronger discrimination ability of visual tracking, especially when the occlusion occurs or when there are similar distractors around the target object.

We further discuss why our proposed CSWinTT works. The strong discriminative ability mainly comes from two strategies: multi-scale window partition and cyclic shifts. After window partition, the target is split into multiple small blocks and each block contains the indivisible information of the object part. These blocks do not disrupt the pixels inside during attention, when some blocks are obscured and not visible, another part of the block can do attention without interference. Although there is no information exchange between different windows, the fusion of multi-scale windows can alleviate the problem, as well as be more robust to diverse sizes of occlusion areas. Additionally, the cyclic shifts can generate a more accurate attention score. For example, after window partition for a human body, there are two windows needed to do the attention. Suppose the first one is a window in the template that contains a head of the human body, which is in the center of the window. The second one is in the search region that contains the same head, as the human movement through the sequence, the head translates from the center to the edge of the window. At this point, a lower matching score will be obtained by window-level attention, which does not fully utilize the information in the windows. After employing the cyclic shifts, as shown in Figure 3, the head at the center of the template window and the head at the edge of the search region window can be finely matched. In addition, the position information in the attention can be obtained by the shift size, and this window-level position can better assist the tracking algorithm to distinguish the target object from the distractors.

5. Conclusion

In this work, we propose a transformer tracker with multi-scale cyclic shifting window attention, which is able to keep the integrity of the object and retain more location information when calculating the cross-window attention between the tracking target and the search area. Moreover, this new window attention is deliberated designed with two improvement schemes including spatially regularized attention mask and redundant computation removal to fully exploit the transformer structure for object tracking. Numerous experimental results on five challenging benchmarks demonstrate that our tracker performs better than previous state-of-the-art trackers. The proposed cyclic shifting window attention has stronger discrimination than the original pixel-level attention in the tracking field. Many other applications like image recognition and stereo matching may benefit from this window attention too.

Acknowledgement

This work is supported by the national key research and development program of China under Grant No.2020YFB1805601.

References

[1] Luca Bertinetto, Jack Valmadre, João F. Henriques, Andrea Vedaldi, and Philip H. S. Torr. Fully-convolutional siamese networks for object tracking. In *Proceedings of the ECCV*, pages 850–865. Springer, 2016. 1, 2, 6

[2] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *Proceedings of the ICCV*, pages 6182–6191. IEEE, October 2019. 2, 6

[3] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Know your surroundings: Exploiting scene information for object tracking. In *Proceedings of the ECCV*. Springer, 2020. 6

[4] Goutam Bhat, Joakim Johnander, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Unveiling the power of deep tracking. In *Proceedings of the ECCV*, pages 483–498. Springer, September 2018. 2, 6

[5] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *Proceedings of the CVPR*, pages 2544–2550. IEEE, June 2010. 2

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020. 1, 2

[7] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *Proceedings of the CVPR*, pages 8126–8135, 2021. 1, 2, 3, 6

[8] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In *Proceedings of the CVPR*, pages 4660–4669. IEEE, June 2019. 2, 6

[9] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *Proceedings of the CVPR*, pages 6638–6646. IEEE, July 2017. 2, 6

[10] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *Proceedings of the CVPR*, pages 7183–7192, 2020. 2, 6

[11] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *Proceedings of the ECCV*, pages 472–488. Springer, 2016. 2

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 1, 2

[13] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *Proceedings of the CVPR*. IEEE, June 2019. 5, 6

[14] Dongyan Guo, Yanyan Shao, Ying Cui, Zhenhua Wang, Liyan Zhang, and Chunhua Shen. Graph attention tracking. In *Proceedings of the CVPR*, pages 9543–9552, 2021. 2, 6

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the CVPR*, pages 770–778. IEEE, June 2016. 3, 5

[16] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE TPAMI*, 37(3):583–596, March 2015. 2, 6

[17] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE TPAMI*, 2019. 5, 6

[18] Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. Cotr: Correspondence transformer for matching across images. In *Proceedings of the ICCV*, 2021. 1

[19] Ilchae Jung, Jeany Son, Mooyeol Baek, and Bohyung Han. Real-time mdnet. In *Proceedings of the ECCV*, pages 83–98. Springer, September 2018. 2

[20] Hamed Kiani Galoogahi, Ashton Fagg, and Simon Lucey. Learning background-aware correlation filters for visual tracking. In *Proceedings of the ICCV*, pages 1135–1143. IEEE, Oct 2017. 2

[21] Matej Kristan, Aleš Leonardis, Jiří Matas, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kämäräinen, Martin Danelljan, Luka Čehovin Zajc, Alan Lukežič, Ondrej Drbohlav, et al. The eighth visual object tracking vot2020 challenge results. In *ECCV*, pages 547–601. Springer, 2020. 5, 6, 7

[22] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *Proceedings of the CVPR*, pages 4282–4291. IEEE, June 2019. 1, 2, 6

[23] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the CVPR*, pages 8971–8980. IEEE, June 2018. 1, 2

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 2

[25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the ICCV*, 2021. 1, 2, 7

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the ICLR*, 2018. 5

[27] Alan Lukezic, Jiri Matas, and Matej Kristan. D3s-a discriminative single shot segmentation tracker. In *Proceedings of the CVPR*, pages 7133–7142, 2020. 6

[28] Alan Lukezic, Tomas Vojir, Luka ˇCehovin Zajc, Jiri Matas, and Matej Kristan. Discriminative correlation filter with channel and spatial reliability. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6309–6318, 2017. 6

[29] Ziang Ma, Linyuan Wang, Haitao Zhang, Wei Lu, and Jun Yin. Rpt: Learning point set representation for siamese visual tracking. In *Proceedings of the ECCVW*. Springer, 08 2020. 2

[30] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. In *Proceedings of the ECCV*, pages 445–461. Springer, 2016. 5, 6, 7

[31] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *Proceedings of the ECCV*, September 2018. 5, 6

[32] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the CVPR*, pages 4293–4302. IEEE, June 2016. 2

[33] Shi Pu, Yibing Song, Chao Ma, Honggang Zhang, and Ming-Hsuan Yang. Deep attentive tracking via reciprocative learning. In *NeurIPS*, pages 1931–1941. 2018. 2

[34] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the CVPR*, pages 658–666, 2019. 5

[35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 5

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017. 1, 2, 4

[37] Paul Voigtlaender, Jonathon Luiten, Philip HS Torr, and Bastian Leibe. Siam r-cnn: Visual tracking by re-detection. In *Proceedings of the CVPR*, pages 6578–6588, 2020. 1, 6

[38] Guangting Wang, Chong Luo, Xiaoyan Sun, Zhiwei Xiong, and Wenjun Zeng. Tracking by instance detection: A meta-learning approach. In *Proceedings of the CVPR*, pages 6288–6297, 2020. 6

[39] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of the CVPR*, pages 5463–5474, 2021. 1, 2

[40] Ning Wang, Wengang Zhou, Jie Wang, and Houqiang Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *Proceedings of the CVPR*, pages 1571–1580, 2021. 1, 2, 3, 6

[41] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip H.S. Torr. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the CVPR*, pages 1328–1338. IEEE, June 2019. 1, 2

[42] Tianyang Xu, Zhen-Hua Feng, Xiao-Jun Wu, and Josef Kittler. Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking. *IEEE TIP*, 28(11):5596–5609, 2019. 2

[43] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. Transcenter: Transformers with dense queries for multiple-object tracking. *arXiv preprint arXiv:2103.15145*, 2021. 2

[44] Yinda Xu, Zeyu Wang, Zuoxin Li, Ye Yuan, and Gang Yu. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In *Proceedings of the AAAI*, volume 34, pages 12549–12556, 2020. 2, 6

[45] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. In *Proceedings of the ICCV*, 2021. 1, 2, 3, 5, 6

[46] Yuechen Yu, Yilei Xiong, Weilin Huang, and Matthew R Scott. Deformable siamese attention networks for visual object tracking. In *Proceedings of the CVPR*, pages 6728–6737, 2020. 2, 6

[47] Zhipeng Zhang, Yihao Liu, Xiao Wang, Bing Li, and Weiming Hu. Learn to match: Automatic matching network design for visual tracking. In *Proceedings of the ICCV*, pages 13339–13348, 2021. 6

[48] Zhipeng Zhang and Houwen Peng. Deeper and wider siamese networks for real-time visual tracking. In *Proceedings of the CVPR*, pages 4591–4600, 2019. 2

[49] Zhipeng Zhang, Houwen Peng, Jianlong Fu, Bing Li, and Weiming Hu. Ocean: Object-aware anchor-free tracking. In *Proceedings of the ECCV*, pages 771–787, 2020. 6

[50] Moju Zhao, Kei Okada, and Masayuki Inaba. Trtr: Visual tracking with transformer. *arXiv preprint arXiv:2105.03817*, 2021. 1

[51] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *IJCV*, 127(3):302–321, 2019. 2

[52] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *Proceedings of the ECCV*, pages 101–117. Springer, September 2018. 2