SPM-Tracker: Series-Parallel Matching for Real-Time Visual Object Tracking

Guangting Wang*[1] Chong Luo[2] Zhiwei Xiong[1] Wenjun Zeng[2]
University of Science and Technology of China[1] Microsoft Research Asia[2]
flylight@mail.ustc.edu.cn cluo@microsoft.com zwxiong@ustc.edu.cn wezeng@microsoft.com

Abstract

The greatest challenge facing visual object tracking is the simultaneous requirements on robustness and discrimination power. In this paper, we propose a SiamFC-based tracker, named SPM-Tracker, to tackle this challenge. The basic idea is to address the two requirements in two separate matching stages. Robustness is strengthened in the coarse matching (CM) stage through generalized training while discrimination power is enhanced in the fine matching (FM) stage through a distance learning network. The two stages are connected in series as the input proposals of the FM stage are generated by the CM stage. They are also connected in parallel as the matching scores and box location refinements are fused to generate the final results. This innovative series-parallel structure takes advantage of both stages and results in superior performance. The proposed SPM-Tracker, running at 120fps on GPU, achieves an AUC of 0.687 on OTB-100 and an EAO of 0.434 on VOT-16, exceeding other real-time trackers by a notable margin.

1. Introduction

Visual object tracking is one of the fundamental research problems in computer vision and video analytics. Given the bounding box of a target object in the first frame of a video, a tracker is expected to locate the target object in all subsequent frames. The greatest challenge of visual tracking can be attributed to the simultaneous requirements on robustness and discrimination power. The robustness requirement demands a tracker not to lose tracking when the appearance of the target changes due to illumination, motion, view angle, or object deformation. Meanwhile, a tracker is required to have the power to discriminate the target object from cluttered background or similar surrounding objects. These two requirements are sometimes contradictory and hard to be fulfilled at the same time.

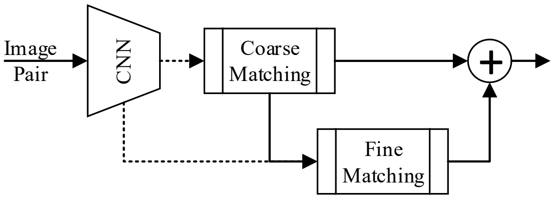


Figure 1. The series-parallel structure which connects coarse matching and fine matching stages in the proposed SPM-Tracker.

Intuitively, both requirements need to be handled through online training. A tracker keeps collecting positive and negative samples along the tracking process. For generative trackers, positive samples help to model the appearance variation of the target. For discriminative trackers, more positive and negative samples help to find a more precise decision boundary that separates the target from the background. For quite a long time, online training has been an indispensable part in tracker design. Recently, with the advancement of deep learning and convolutional neural networks, deep features have been widely adopted in object trackers [34, 9, 39, 15, 7, 30]. However, online training with deep features is extremely time consuming. Without much surprise, the deep version of many high-performance trackers [9, 7, 3, 39, 34, 48, 53] cannot run in real-time any more, even on modern GPUs.

While the excessive volume of deep features brings speed issues to online training, their strong representational power also opens up a possibility to completely give up online training. The idea is, under a given distance measure, to learn an embedding space, through offline training, that can maximize the interclass inertia between different objects and minimize the intraclass inertia for the same object [58]. Note that maximizing the interclass inertia corresponds to the discrimination power and minimizing the intraclass inertia corresponds to the robustness. The pioneering work along this research line is SiamFC [2]. In addition to the offline training, SiamFC uses cross-correlation operation to efficiently measure the distance between the target patch and all surrounding patches. As a result, SiamFC can operate at 86fps on GPU.

By design, the SiamFC framework faces challenges in balancing the robustness and the discrimination power of

*This work is done when Guangting Wang is an intern in MSRA.

the embedding and in handling the scale and aspect ratio changes of the target object. Recently, SiamRPN [26] was proposed to address the second challenge. It consists of a Siamese subnetwork for feature extraction and a region proposal subnetwork for similarity matching and box regression. In a follow-up work called DaSiamRPN [58], distractor-aware training is adopted to promote the generalization and discriminative ability of the embedding. In these two pieces of work, visual object tracking is formulated as a local one-shot detection task.

In this paper, we design a two-stage SiamFC-based network for visual object tracking, aiming to address both challenges mentioned above. The two stages are the coarse matching (CM) stage which focuses on enhancing the robustness and the fine matching (FM) stage which focuses on improving the discrimination power. By decomposing these two equally important but somewhat contradictory requirements, our proposed network is expected to achieve better performance. Moreover, both CM and FM stages perform similarity matching and bounding box regression. Thanks to the two-stage box refinement, our tracker achieves high localization precision without multi-scale test.

The key innovation in this work is the series-parallel structure that is used to connect the two stages. The schematic diagram is shown in Fig.1. Similar to the series structure which is widely adopted in two-stage object detection, the input of the second FM stage relies on the output of the first CM stage. In this sense, the CM stage is a proposal stage. Similar to the parallel structure, the final matching score as well as the box location are the fused results from both stages. This series-parallel structure brings a number of advantages which will be detailed in Section 3. In addition, we propose generalized training (where objects from the same category are all treated as the same object) to boost the robustness of the CM stage. The discrimination power of the FM stage is promoted by replacing the cross-correlation layer with a distance learning subnetwork. With these three innovations, the resulting tracker achieves superior performance on major benchmark datasets. It achieves an AUC of 0.687 on OTB-100 and EAOs of 0.434 and 0.338 on VOT-16 and VOT-17, respectively. More importantly, the inference speed is 120fps on a NVIDIA P100 GPU.

The rest of the paper is organized as follows. We discuss related work in Section 2. The proposed series-parallel framework is presented in Section 3. After describing the implementation details of SPM-Tracker in Section 4, we provide extensive experimental results in Section 5. Finally, we conclude the paper with some discussions in Section 6.

2. Related Work

Object trackers have conventionally been classified into generative trackers and discriminative trackers [24], and most modern trackers belong to the latter. A common approach of discriminative trackers is to build a binary classifier that represents the decision boundary between the object and its background [24]. It is generally believed that adaptive discriminative trackers, which continuously update the classifier during tracking, are more powerful than their static counterparts.

Correlation Filter (CF) based trackers are among the most successful and representative adaptive discriminative trackers. Bolme *et al.* [4] first proposed the MOSSE filter which is capable of producing stable CFs from a single frame and then continuously being improved during tracking. The MOSSE filter has aroused a great deal of interest and there are a bunch of follow-up work. For example, kernel tricks [19, 20, 10] were introduced to extend CF. DSST [10] and SAMF [27] enabled scale estimation in CF. SRDCF [8] was proposed to alleviate the periodic effect of convolution boundaries.

More recently, with the advancement of deep learning, the rich representative power of deep features is widely acknowledged. There is a trend to utilize deep features in CF-based trackers [31, 9, 7, 3]. However, this creates a dilemma: online training is an indispensable part of CF-based trackers, but online training with deep features is extremely slow.

In many real world applications, being real-time is mandatory for a tracker. Facing the above mentioned dilemma, many researchers resorted to another choice: static discriminative trackers. With the highly expressive deep features, it becomes possible to build high-performance static trackers. This idea was successfully realized by SiamFC [2]. SiamFC employs Siamese convolutional neural networks (CNNs) to extract features, and then uses a simple cross-correlation layer to perform dense and efficient sliding-window evaluation in the search region. Every patch of the same size as the target gets a similarity score, and the one with the highest score is identified as the new target location. There are also a great number of follow-up work [15, 52, 49], among which SA-Siam [17, 16] and SiamRPN [26, 58] are most related to ours.

The main challenge in SiamFC-based methods is to find an embedding space, through offline training, that is both robust and discriminative. Zhu et al. [58] propose distractor-aware training to emphasize these two aspects. They use diverse categories of positive still image pairs to promote the robustness, and use semantic negative pairs to improve the discriminative ability. However, it is difficult to attend to both requirements in a single network. SA-Siam [17] and Siam-BM [16] adopt a two-branch network to encode images into two embedding spaces, one for semantic similarity (more robust) and the other for appearance similarity (more discriminative). This typical parallel structure does not take advantage of the innate proposal capability of the semantic branch.

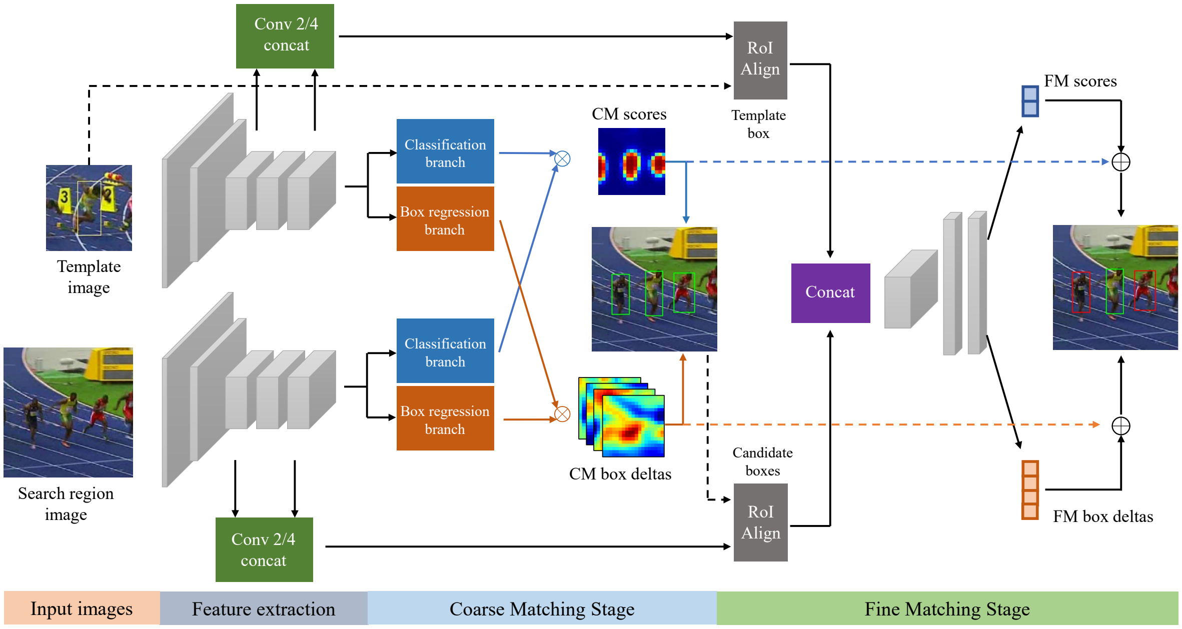


Figure 2. Details of the proposed series-parallel matching framework. We employ Siamese AlexNet [25] for feature extraction. The CM stage adopts the network structure of SiamRPN [26]. RoI Align [18] is used to generate fixed-length regional features for each proposal. The FM stage implements a relation network [50] for distance learning. Finally, results from both stages are fused for decision making.

Another challenge in SiamFC-based methods is how to handle scale and shape changes. Almost all SiamFC-based trackers adopt an awkward multi-scale test for scale adjustment, but the aspect ratio of bounding boxes remains unchanged throughout the tracking process. SiamRPN [26] addresses this issue with an elegant region proposal network (RPN). The capability to do box refinement also allows it to discard multi-scale test. In this work, we follow SiamRPN to use RPN for bounding box size adjustment. The two-stage refinement allows our SPM-Tracker to achieve an even more precise box location.

SiamRPN and DaSiamRPN [58] pose the tracking problem as a local single-stage object detection problem. Some recent empirical studies [22] on object detection show that two-stage design is often more powerful than one-stage design. This may be related to hard example mining [28] and regional feature alignment [18]. In the tracking community, Zhang *et al.* [55] adopt a two-stage design for long-term tracking. However, the series structure they adopted demands for a very powerful second stage. They use MDNet [34] for the second stage, which greatly slows down the inference speed to 2fps.

3. Our Approach

3.1. Series-Parallel Matching Framework

We propose a framework for robust and discriminative visual object tracking. The proposed SPM framework is depicted in Fig.2. We employ a Siamese network to extract features from the target patch and the local search region. This is followed by two matching stages, namely coarse matching stage and fine matching stage, organized in a series-parallel structure.

Both the CM and FM stages produce similarity scores of proposals and box location deltas. We let the CM stage to focus on the robustness, i.e. to minimize the intraclass inertia for the same object. It is expected to propose the target object even when it is experiencing huge appearance changes. A number of proposals which get the top matching scores in the CM stage are then passed to the FM stage and fixed-size regional features are extracted through RoI Align [18]. The FM stage is designed to focus on discrimination, i.e. to maximize the interclass inertia between different objects. It is expected to discriminate the true target from background or surrounding similar objects. Eventually, the matching scores and box locations from both matching stages are fused to make the final decision.

The proposed SPM framework brings a number of advantages as outlined below.

- The robustness and the discrimination requirements are decomposed and emphasized in separate stages. It is easier to train two networks to achieve their respective goals than to train a single network that simultaneously achieves the goals for both requirements..

- The input proposals of the FM stage are all high-score candidates from the CM stage. FM stage training benefits from a balanced positive-negative ratio and hard negative mining to enhance the discrimination power.

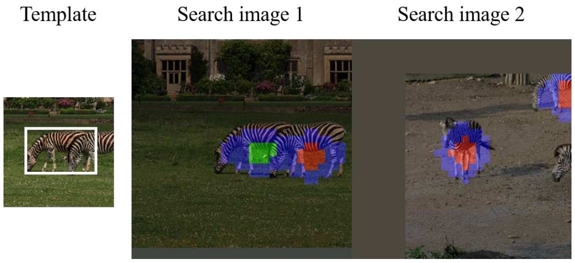


Figure 3. Illustration of the generalized training (GT) strategy for the CM stage. Given a template as shown on the left, the green blocks in search image 1 indicate the positive samples used in conventional training. The red blocks are the locations of other objects of the same category. GT takes both green and red blocks as positive samples. (The blue blocks indicate the ignored region.) Best viewed in color.

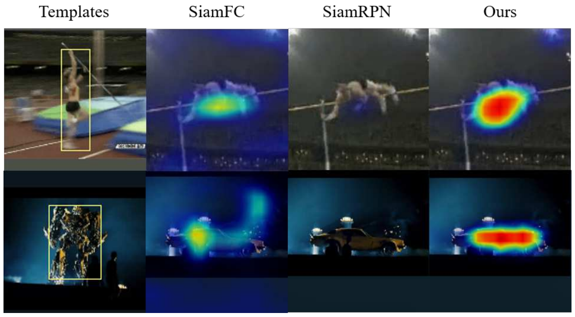


Figure 4. Visualization of the cross-correlation response maps generated by SiamFC [2], SiamRPN [26], and the CM stage of our tracker. Our tracker can robustly highlight the target object even when it has severe deformation. Best viewed in color.

- Box regression in the CM stage allows the FM stage to evaluate aligned patches with different scale or even different aspect ratio from the target object. Fusion of two-stage box regressions leads to a higher precision.
- Since only a few proposals are passed to the FM stage, it is not necessary to use cross-correlation operation to compute distance. We could adopt a trainable distance measure for the FM stage.

In the following two subsections, we will discuss the CM and FM stages in more details.

3.2. Coarse Matching Stage

The coarse matching stage looks in the search region for candidate patches which are similar to the target patch. It is expected to be very robust such that the target object will not be missed even when it is experiencing drastic appearance changes due to intrinsic or extrinsic factors. We adopt the region proposal subnetwork as introduced in SiamRPN [26] for this stage. Given the features extracted by a Siamese network, pair-wise correlation feature maps are computed for the classification branch and the regression branch. The classification branch produces the similarity scores for the candidate boxes while the regression branch generates the box deltas. Similar to SiamRPN, we can discard multi-scale test since the proposal network handles scale and shape changes in a graceful manner.

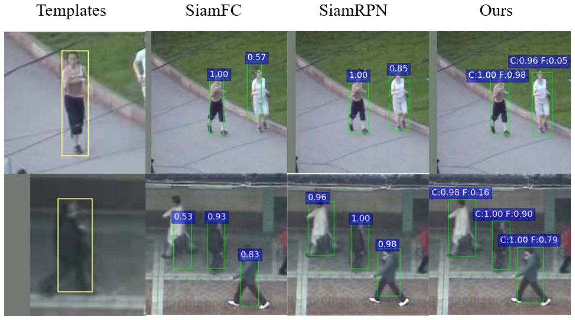


Figure 5. Visualization of the top-K matched boxes and their similarity scores output by SiamFC [2], SiamRPN [26], and our SPM-Tracker. Our tracker generates two scores, corresponding to the CM stage (**C**) and the FM stage (**F**). Objects of the same category get high C-scores but only the true target gets high F-scores. It shows that SPM-Tracker achieves the design goal.

For the CM stage, we propose generalized training (GT) to improve the robustness. Conventionally, image pairs of the same object drawn from two frames of a video are used as positive samples. In DaSiamRPN [58], still images from detection datasets are used to generate positive image pairs through data augmentation. In this work, we additionally treat some image pairs containing different objects as positive samples when the two objects belong to the same category. Fig. 3 illustrates the classification labels used in our CM stage and in other SiamFC-based trackers. This training strategy leads to exceedingly generalized embeddings which capture high-level semantic information and therefore are insensitive to object appearance changes.

Fig.4 shows the response map of the CM stage and compares it with that of SiamFC and SiamRPN (with distractor-aware training). It is observed that our tracker is able to generate strong responses even when the target object has significant deformation. By contrast, SiamRPN [26, 58] barely produce any response and SiamFC does not have a precise localization.

3.3. Fine Matching Stage

The fine matching stage is expected to capture fine-grained appearance information so that the true target can be distinguished from background or similar surrounding objects. The FM stage only evaluates the top K highest-score patches from the CM stage.

As illustrated in Fig. 2, the FM stage shares features with the CM stage. For each proposal, the regional features are directly cropped from the shared feature maps. Considering the fact that shallow features contain detailed appearance information and also result in high localization precision, we take both deep and shallow features and fuse them by concatenation. Then, RoI Align operation [18] creates fixed-size feature maps for each proposal.

Since there are only a limited number of patches to be evaluated in this stage, we can afford to use a more powerful distance learning network, instead of the cross-correlation layer, to measure the similarity. Additionally, such a network could be trained to generate a complementary score to the CM similarity scores. We adopt a light-weight relation network as proposed in [50] for the FM stage. The input of the relation network is the concatenated feature from the image pairs. A 1×1 convolution layer is followed by two fully connected layers which generate feature embedding for classification and box regression.

Finally, the similarity scores and the box deltas from two stages are fused by weighted sum. The candidate box with the highest similarity score is identified as the target object. Fig. 5 shows the top-K candidates and their similarity scores output by different trackers. Our tracker is associated with two scores corresponding to the CM and FM stages. The high C-scores for all the foreground objects suggest the robustness of SPM-Tracker and the low F-scores for non-target objects demonstrate the discrimination power.

4. Implementation

4.1. Network Structure and Parameters

The CNN backbone used for feature extraction is the standard AlexNet [25]. It is pre-trained on the ImageNet dataset. Unlike other SiameFC-based trackers, we keep the padding operations in the backbone network. This is because the RoI Align operation needs pixel alignment between feature maps and source images. The CM stage still uses the central features without padding. In our implementation, the target image has a size of $127 \times 127 \times 3$. The size of its last-conv-layer feature map with padding is $16 \times 16 \times 256$. Only the central $6 \times 6 \times 256$ features are used for the CM stage, which is consistent with the original SiamFC. The FM stage extracts regional features from *conv2* (384 channels) and *conv4* (256 channels) layers and concatenates them. We use RoI Align operation to pool regional features of size $6 \times 6 \times 640$ for each proposal, where 6×6 is the spatial size and 640 is the number of channels. The two fully-connected layers in the FM stage are lightweight, with only 256 neurons per layer.

4.2. Training

The entire network can be trained end-to-end. The overall loss function is composed of four parts: classification loss and box regression loss in both the CM stage and FM stage. For the CM stage, an anchor box is assigned a positive (or negative) label when its intersection-over-union (IoU) overlap with the ground-truth box is greater than 0.6 (or less than 0.3). Other patches whose IoU overlap falls in between are ignored. For the FM stage, positive (or negative) labels are assigned to candidate boxes whose IoU overlaps are greater (or less) than 0.5. Same as in the Faster R-CNN object detection framework [37], box regression loss is added to positive samples in both stages. We adopt cross-entropy loss for classification and smooth L1 loss [14] for box regression. The overall loss function can be written as:

$$L = \lambda_1 L_{cm\_cls} + \lambda_2 L_{cm\_b} + \lambda_3 L_{fm\_cls} + \lambda_4 L_{fm\_b}, \quad (1)$$

where L_{cls} denotes the classification loss and L_b denotes the box regression loss. We set $\lambda_2 = 2$ and $\lambda_1 = \lambda_3 = \lambda_4 = 1$ since the box regression loss of CM module is much smaller than the others.

The training image pairs are extracted from both videos and still images. The video datasets include VID [38] and the training set of Youtube-BB [35]. Following DaSiamRPN [58], we also make use of still image datasets, including COCO [29], ImageNet DET [38], Cityperson [56] and WiderFace [51]. The sampling ratio between videos and still images is $4 : 1$. There are three types of image pairs, denoted by same-instance, same-category, and different-category. They are sampled at a ratio of $2 : 1 : 1$.

The standard SGD optimizer is adopted for training. In each step, the CM stage produces hundreds of candidate boxes, among which 48 boxes are selected to train the FM stage. The positive-negative ratio is set to $1 : 1$. The learning rate is decreased from 10^{-2} to 10^{-4}. We train the network for 50 epochs and 160,000 image pairs are sampled in each epoch.

4.3. Inference

During inference, we crop the template image patch from the first frame and feed it to the feature extraction network. The template features are cached so that we do not need to compute it in the subsequent frames.

Given the tracking box in the last frame, a search image patch surrounding the box location is cropped and resized to 271×271. The CM stage takes the search image as input and then outputs a number of boxes. The candidate box that has the largest overlap with the tracking box in the previous frame will be reserved to increase the stability. Other boxes go through the standard proposal processing in RPN [37]. First, boxes with low scores are filtered. Then non-maximum suppression (NMS) is applied. The NMS threshold is 0.5. Finally, K candidate boxes with top scores are selected and passed to the FM stage. In this step, we do not add shape penalties or cosine window penalties in order to aggressively propose boxes. The number of candidate boxes K is set to 9, which is further analyzed in Section 5.2. We use five anchors whose aspect ratios are $[0.33, 0.5, 1.0, 2.0, 3.0]$.

In the FM stage, similarity scores and refined box positions are predicted by the classification head and the box regression head. Let u_c, u_f be the scores predicted by CM

	CM Only	CM+FM Series	CM+FM Series-Parallel
OTB-100 (AUC)	0.643	0.632	**0.670**
VOT-17 (EAO)	0.279	0.296	**0.323**
VOT-16 (EAO)	0.359	0.343	**0.391**

Table 1. Ablation analysis of different architectures. Results on three benchmark datasets are consistent, and they demonstrate the superiority of the series-parallel structure.

and FM stages, respectively. Let $\mathbf{B_c}, \mathbf{B_f}$ be the bounding box locations after the adjustment of the CM and FM stages, respectively. The final score and box coordinates are the weighted sum of the results from the two modules:

$$\begin{aligned} u &= (1 - W_{cls})u_c + W_{cls}u_f \\ \mathbf{B} &= \frac{u_c}{W_{box}u_f + u_c}\mathbf{B_c} + \frac{W_{box}u_f}{W_{box}u_f + u_c}\mathbf{B_f}, \end{aligned} \tag{2}$$

where W_{cls}, W_{box} are weights of the FM module for similarity score and box coordinates. We find that good tracking results are usually achieved when W_{cls} takes a value around 0.5 and W_{box} takes a value of 2 or 3.

After applying cosine windows [2], the candidate box with the highest score is selected and its size is updated by linear interpolation with the result in the previous frame. Our tracker can run inference at 120fps with a single NVIDIA P100 GPU and an Intel Xeon E5-2690 CPU.

5. Experiments

The three main contributions in this work are: 1) using series-parallel structure to connect two matching stages; 2) adopting generalized training for the CM stage; and 3) adopting a relation network for distance measurement in the FM stage. In this section, we will first perform ablation analyses which support our contributions, and then carry out comparison studies with the state-of-the-art trackers on major benchmark datasets.

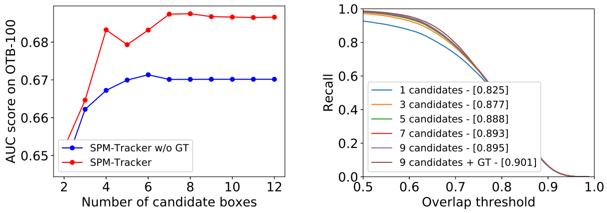
5.1. Analysis of the Series-Parallel Structure

We corroborate the effectiveness of the series-parallel structure by comparing it with two alternatives. The baseline scheme, denoted by “CM only” in Table 1 is actually SiamRPN [26]. Our implementation achieves a slightly better performance than reported in their original paper (0.279 vs 0.244 on VOT-17 benchmark) because we have included additional still images in the training. When the FM stage is added in series, the performance (denoted by “CM+FM Series” in Table 1) is similar to the baseline (better on VOT-17 and worse on OTB-100 and VOT-16).

The proposed “CM+FM Series-Parallel” method, which performs two-stage fusion, significantly outperforms the other two schemes, as Table 1 shows. The reason why fusion plays an important role is that the two stages pay attention to different aspects of tracker capabilities: robustness at the CM stage and discrimination power at the FM stage. The matching score produced by one stage does not reflect the other capability. The idea of fusion has been practiced in many trackers [47, 13, 6, 17] and has shown effectiveness.

	OTB-100 AUC	VOT-17 EAO	VOT-16 EAO
S-P model	0.670	0.323	0.391
S-P model + GT	**0.687**	**0.338**	**0.434**

Table 2. Generalized training (GT) for the CM stage significantly improves the performance.



(a)

(b)

Figure 6. CM module analysis: (a) AUC score vs. number of candidate boxes; (b) recall rate vs. overlap thresholds (the values in brackets indicate the mean recalls over thresholds 0.5:0.05:0.7). All experiments are carried out on OTB-100 dataset.

5.2. Analysis of the CM Stage

Generalized Training Strategy: To make the CM module more robust to object appearance change, we propose to take image pairs in the same category as positive samples during training. This is referred to as the generalized training (GT) strategy. We compare the performance of SPM-Tracker when it is trained with or without the GT strategy for the CM stage. Improvements achieved on all three benchmark datasets, as shown in Table 2, confirm the effectiveness of this strategy. Some of the visualization results have already been presented in Fig. 4 to show that the GT strategy helps to locate objects with large deformation.

Number of Candidate Boxes: During inference, the CM stage passes K top-scored candidate boxes to the FM stage. On the one hand, the larger the K is, the higher the probability that the true target is included in the final evaluation. On the other hand, a larger K means more false positives will be evaluated in the FM stage, which reduces the speed and might decrease the accuracy as well. In order to determine K, we investigate the relationship between the tracking performance and the number of candidate boxes. Fig. 6(a) shows how the AUC on OTB-100 changes with K. We find that when K is larger than 7, the performance tends to flatten. Therefore, we choose $K = 9$ in experiments.

Recall: Recall of candidate boxes can be used to measure the robustness. We use recall to further validate the GT strategy and K selection in the CM stage. To ensure fairness, we crop the template from the first frame and the search region in the current frame is generated according

	Tracker	AUC score (OPE)			Speed
		OTB-2013	OTB-50	OTB-100	(FPS)
CF-based Trackers	LCT [32]	0.628	0.492	0.568	27
	Staple [1]	0.593	0.516	0.582	80
	LMCF [46]	0.628	0.533	0.580	85
	CFNet [44]	0.611	0.530	0.568	75
	BACF [12]	0.656	0.570	0.621	35
	ECO-hc [7]	0.652	0.592	0.643	60
	MKCFup [42]	0.641	-	-	150
	MCCT-H [48]	0.664	-	0.642	45
SiamFC-based Trackers	SiamFC [2]	0.607	0.516	0.582	86
	DSiamM [15]	0.656	-	-	25
	RASNet [49]	0.670	-	0.642	83
	SiamRPN [26]	0.658	0.592	0.637	200
	SA-Siam [17]	0.677	0.610	0.657	50
	StructSiam [57]	0.637	-	0.621	45
	MemTrack [52]	0.642	-	0.626	50
	DaSiamRPN [58]	0.656	0.602	0.658	160
	Siam-BM [16]	0.684	0.617	0.662	48
Misc.	EAST [21]	0.638	-	0.629	159
	PTAV [11]	0.663	0.581	0.635	25
	ACT [5]	0.657	-	0.625	30
	RT-MDNet [23]	-	-	0.650	46
	Ours	**0.693**	**0.653**	**0.687**	120

Table 3. Comparison with state-of-the-art real-time trackers on OTB dataset. Trackers are grouped into CF-based methods, SiamFC-based methods and miscellaneous. Numbers in red and blue are the best and the second best results, respectively.

to the ground-truth box in the previous frame. Fig. 6 (b) shows the recall vs. overlap threshold. The mean recall for the overlap thresholds $[0.5, 0.7]$ is also computed and listed in brackets. It is obvious that using a single candidate box results in significantly lower recall than using multiple candidates. As the number of candidate boxes increases, the recall also increases before it saturates at around $K = 9$. At the saturation point, applying the GT strategy still can boost recall. This double confirms the power of the GT strategy.

5.3. Analysis of the FM Stage

Multi-Layer Feature Fusion: The FM stage takes regional features cropped from the shared backbone network as inputs. Generally speaking, deep features are rich in high-level semantic information and shallow features are rich in low-level appearance information. As suggested in many previous works [43, 45, 39, 3], multi-layer features can be fused to achieve better performance. We follow this common practice and use *conv2* + *conv4* features for the FM stage. To demonstrate the advantage of multi-layer feature fusion, we compare the performance of SPM-Tracker with alternative implementations which only use single layer features. We train and test models which use *conv2*, *conv3*, or *conv4* only. On OTB-100 benchmark, these three models achieve AUC scores of 0.666, 0.675, and 0.676, respectively, while our final model using *conv2* + *conv4* achieves an AUC of 0.687. This experiment demonstrates that the FM stage benefits from multi-layer feature fusion.

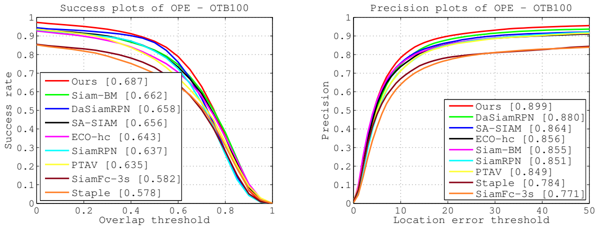


Figure 7. The success plot and precision plot on OTB-100.

Replace Cross-Correlation Layer with the Relation Network: An important innovation that contributes to the high efficiency of SiamFC tracker is the cross-correlation layer that achieves dense and efficient sliding-window evaluation in the search region. Almost all SiamFC-based trackers have followed this usage. We also use cross-correlation layer in our CM stage for similarity matching and box regression. But in the FM stage, there are much fewer candidate boxes scattered in the search region. There is not much advantage in using cross-correlation operation. Therefore, we replace the cross-correlation layer with a more powerful relation network as described in [50]. Experimental results verify our design choice. When models are trained without the GT strategy, using cross-correlation layer in the FM stage results in an AUC of 0.647 on OTB-100, which is slightly better than the single stage baseline SiamRPN (0.643), but is notably inferior to the relation-network-based model (0.670). In addition, when GT is adopted, the AUC score of the cross-correlation-based model is 0.655 while that of the relation-network-based model is 0.687.

5.4. Comparison with State-of-the-Arts

Evaluation on OTB: Our SPM-Tracker is first compared with the state-of-the-art real-time trackers on OTB 2013/50/100 benchmarks. The detailed AUC scores are summarized in Table 3. Due to space limitation, we only show the success plot and the precision plot of one pass evaluation (OPE) on OTB-100 in Fig. 7. The SPM-Tracker outperforms other real-time trackers on all three OTB benchmarks by a large margin.

We also compare SPM-Tracker with some non-real-time top-performing trackers, including C-COT [9], ECO [7], MDNet [34], ADNet [54], TCCN [33], LSART [41], VITAL [40], RTINet [53], and DRL-IS [36]. The AUC score vs. speed curve on OTB-100 is shown in Fig. 8. SPM-Tracker strikes a very good balance between tracking performance and inference speed.

Evaluation on VOT: SPM-Tracker is evaluated on two VOT benchmark datasets, VOT-16 and VOT-17. Table 4 shows the comparison with almost all the top-performing trackers despite their speed. Among the real-time trackers, SPM-Tracker is by far the best performing one with su-

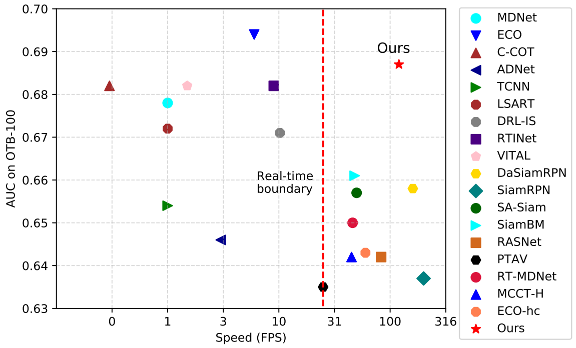


Figure 8. Performance-speed trade-off of top-performing trackers on OTB-100 benchmark. The speed axis is logarithmic.

Tracker	VOT-16			VOT-17			FPS
	A	R	EAO	A	R	EAO	
CREST	0.51	0.25	0.283	-	-	-	1
MDNet	0.54	0.34	0.257	-	-	-	1
C-COT	0.54	0.24	0.331	-	-	-	0.3
LSART	-	-	-	0.49	0.22	0.323	1
ECO	0.55	**0.20**	0.375	0.48	0.27	0.280	8
UPDT	-	-	-	0.53	**0.18**	**0.378**	-
SiamFC	0.53	0.46	0.235	0.50	0.59	0.188	86
Staple	0.54	0.38	0.295	0.52	0.69	0.169	80
ECO-hc	0.54	0.30	0.322	0.49	0.44	0.238	60
SA-Siam	0.54	-	0.291	0.50	0.46	0.236	50
Siam-BM	-	-	-	0.56	0.26	0.335	32
SiamRPN	0.56	0.26	0.344	0.49	0.46	0.244	200
DaSiamRPN	0.61	0.22	0.411	0.56	0.34	0.326	160
Ours	**0.62**	0.21	**0.434**	**0.58**	0.30	0.338	120

Table 4. Comparison with state-of-the-art trackers on VOT benchmark. Both non-real-time methods (top rows) and real-time methods (bottom rows) are included. "A" and "R" denote accuracy and robustness. EAO stands for expected average overlap. The numbers in red and blue indicate the best and the second best results.

perior accuracy and EAO. Even when compared with non-real-time trackers, SPM-Tracker achieves the best accuracy and the EAO performance is among the best.

Excluding extra data: Compared with DaSiamRPN [58], our tracker has used two more datasets (Cityperson [56] and WiderFace [51]) in training. For fair comparison, we have also trained a model excluding these two datasets. The AUC on OTB-100 slightly drops to 0.671, but still outperforms DaSiamRPN and Siam-BM. The EAO on VOT-16 becomes 0.432 and that on VOT-17 slightly increases to 0.347.

5.5. Qualitative Results

Successful Cases: In Fig. 9, we visualize three successful tracking cases, including the very challenging *jump* and *diving* sequences. Owing to the robustness of the CM stage, our tracker is able to detect targets with huge deformation. The region proposal branch allows SPM-Tracker to fit to the varying object shapes. In these two sequences, some of the best trackers such as ECO [7] and MDNet [34] also fail. DaSiamRPN [58] barely follows the target, but the box locations are less precise. This demonstrates the advantage of our two-stage box refinement.

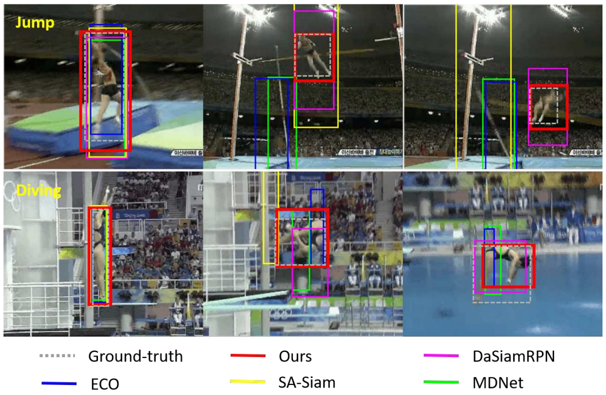


Figure 9. Visualization of three successful tracking sequences from OTB-100.

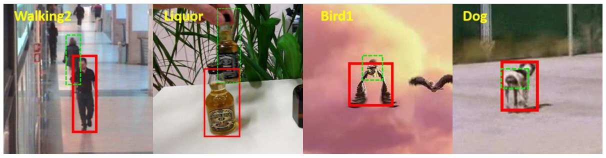


Figure 10. Visualization of failure cases. The green box is ground-truth and the red box is our tracking result.

Failure Cases: We observe two types of failures in SPM-Tracker, as shown in Fig. 10. In *walking2* and *liquor* sequences, when the target is occluded by a similar object, the tracking box may drift. The other type of failure occurs when the ground-truth target is only a part of an object, as in sequences *bird1* and *dog*. SPM-Tracker seems to have a strong sense of objectness and tends to track the entire object even when the template only contains a part of it.

6. Conclusion

We have presented the design and implementation of a static discriminative tracker named SPM-Tracker. SPM-Tracker adopts a novel series-parallel structure for two-stage matching. Evaluations on OTB and VOT benchmarks show its superior tracking performance. In the future, we plan to explore solutions to the drifting problem when the target is occluded by similar objects. Possible choices include template update and forward-backward verification. We believe that the series-parallel matching framework has great potential and is worthy of further investigation.

Acknowledgement

We acknowledge funding from National Key R&D Program of China under Grant 2017YFA0700800.

References

[1] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip HS Torr. Staple: Complementary learners for real-time tracking. In *CVPR*, pages 1401–1409, 2016.

[2] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *ECCV*, pages 850–865, 2016.

[3] Goutam Bhat, Joakim Johnander, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Unveiling the power of deep tracking. In *ECCV*, 2018.

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

[5] Boyu Chen, Dong Wang, Peixia Li, Shuang Wang, and Huchuan Lu. Real-time actor-critictracking. In *ECCV*, pages 328–345, 2018.

[6] Dapeng Chen, Zejian Yuan, Gang Hua, Yang Wu, and Nanning Zheng. Description-discrimination collaborative tracking. In *ECCV*, pages 345–360, 2014.

[7] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, Michael Felsberg, et al. Eco: Efficient convolution operators for tracking. In *CVPR*, pages 6931–6939, 2017.

[8] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Learning spatially regularized correlation filters for visual tracking. In *ICCV*, pages 4310–4318, 2015.

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

[10] Martin Danelljan, Fahad Shahbaz Khan, Michael Felsberg, and Joost Van de Weijer. Adaptive color attributes for real-time visual tracking. In *CVPR*, pages 1090–1097, 2014.

[11] Heng Fan and Haibin Ling. Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking. In *ICCV*, pages 5487–5495, 2017.

[12] H Kiani Galoogahi, Ashton Fagg, and Simon Lucey. Learning background-aware correlation filters for visual tracking. In *CVPR*, pages 1144–1152, 2017.

[13] Jin Gao, Haibin Ling, Weiming Hu, and Junliang Xing. Transfer learning based visual tracking with gaussian processes regression. In *ECCV*, pages 188–203, 2014.

[14] Ross Girshick. Fast R-CNN. In *ICCV*, pages 1440–1448, 2015.

[15] Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang. Learning dynamic siamese network for visual object tracking. In *ICCV*, pages 1763–1771, 2017.

[16] Anfeng He, Chong Luo, Xinmei Tian, and Wenjun Zeng. Towards a better match in siamese network based visual object tracker. In *ECCV Workshop*, 2018.

[17] Anfeng He, Chong Luo, Xinmei Tian, and Wenjun Zeng. A twofold siamese network for real-time object tracking. In *CVPR*, pages 4834–4843, 2018.

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2980–2988, 2017.

[19] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV*, pages 702–715, 2012.

[20] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *T-PAMI*, 37(3):583–596, 2015.

[21] Chen Huang, Simon Lucey, and Deva Ramanan. Learning policies for adaptive tracking with deep feature cascades. In *ICCV*, pages 105–114, 2017.

[22] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *CVPR*, pages 7310–7311, 2017.

[23] Ilchae Jung, Jeany Son, Mooyeol Baek, and Bohyung Han. Real-time mdnet. In *ECCV*, pages 83–98, 2018.

[24] Zdenek Kalal, Krystian Mikolajczyk, Jiri Matas, et al. Tracking-learning-detection. *T-PAMI*, 34(7):1409, 2012.

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[26] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, pages 8971–8980, 2018.

[27] Yang Li and Jianke Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *ECCV*, pages 254–265, 2014.

[28] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. In *ICCV*, pages 2980–2988, 2017.

[29] 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, 2014.

[30] Xiankai Lu, Chao Ma, Bingbing Ni, Xiaokang Yang, Ian Reid, and Ming-Hsuan Yang. Deep regression tracking with shrinkage loss. In *ECCV*, pages 353–369, 2018.

[31] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Hierarchical convolutional features for visual tracking. In *ICCV*, pages 3074–3082, 2015.

[32] Chao Ma, Xiaokang Yang, Chongyang Zhang, and Ming-Hsuan Yang. Long-term correlation tracking. In *CVPR*, pages 5388–5396, 2015.

[33] Hyeonseob Nam, Mooyeol Baek, and Bohyung Han. Modeling and propagating cnns in a tree structure for visual tracking. *arXiv preprint*, 2016.

[34] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, pages 4293–4302, 2016.

[35] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *CVPR*, pages 7464–7473, 2017.

[36] Liangliang Ren, Xin Yuan, Jiwen Lu, Ming Yang, and Jie Zhou. Deep reinforcement learning with iterative shift for visual tracking. In *ECCV*, pages 684–700, 2018.

[37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

[38] 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.

[39] Yibing Song, Chao Ma, Lijun Gong, Jiawei Zhang, Rynson WH Lau, and Ming-Hsuan Yang. Crest: Convolutional residual learning for visual tracking. In *ICCV*, pages 2574–2583, 2017.

[40] Yibing Song, Chao Ma, Xiaohe Wu, Lijun Gong, Linchao Bao, Wangmeng Zuo, Chunhua Shen, Rynson Lau, and Ming-Hsuan Yang. Vital: Visual tracking via adversarial learning. In *CVPR*, 2018.

[41] Chong Sun, Huchuan Lu, and Ming-Hsuan Yang. Learning spatial-aware regressions for visual tracking. In *CVPR*, pages 8962–8970, 2018.

[42] Ming Tang, Bin Yu, Fan Zhang, and Jinqiao Wang. High-speed tracking with multi-kernel correlation filters. In *CVPR*, pages 4874–4883, 2018.

[43] Ran Tao, Efstratios Gavves, and Arnold W M Smeulders. Siamese instance search for tracking. In *CVPR*, pages 1420–1429, 2016.

[44] Jack Valmadre, Luca Bertinetto, João Henriques, Andrea Vedaldi, and Philip HS Torr. End-to-end representation learning for correlation filter based tracking. In *CVPR*, pages 5000–5008, 2017.

[45] Lijun Wang, Wanli Ouyang, Xiaogang Wang, and Huchuan Lu. Visual tracking with fully convolutional networks. In *ICCV*, pages 3119–3127, 2015.

[46] Mengmeng Wang, Yong Liu, and Zeyi Huang. Large margin object tracking with circulant feature maps. In *CVPR*, pages 21–26, 2017.

[47] Naiyan Wang, Siyi Li, Abhinav Gupta, and Dit-Yan Yeung. Transferring rich feature hierarchies for robust visual tracking. *arXiv preprint*, 2015.

[48] Ning Wang, Wengang Zhou, Qi Tian, Richang Hong, Meng Wang, and Houqiang Li. Multi-cue correlation filters for robust visual tracking. In *CVPR*, pages 4844–4853, 2018.

[49] Qiang Wang, Zhu Teng, Junliang Xing, Jin Gao, Weiming Hu, and Stephen Maybank. Learning attentions: residual attentional siamese network for high performance online visual tracking. In *CVPR*, pages 4854–4863, 2018.

[50] Flood Sung Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018.

[51] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *CVPR*, pages 5525–5533, 2016.

[52] Tianyu Yang and Antoni B Chan. Learning dynamic memory networks for object tracking. In *ECCV*, 2018.

[53] Yingjie Yao, Xiaohe Wu, Lei Zhang, Shiguang Shan, and Wangmeng Zuo. Joint representation and truncated inference learning for correlation filter based tracking. In *ECCV*, pages 552–567, 2018.

[54] SYJCY Yoo, Kimin Yun, Jin Young Choi, K Yun, and JY Choi. Action-decision networks for visual tracking with deep reinforcement learning. In *CVPR*, pages 1349–1358, 2017.

[55] Lijun Wang Jinqing Qi Huchuan Lu Yunhua Zhang, Dong Wang. Learning regression and verification networks for long-term visual tracking. In *arXiv preprint arXiv:1809.04320*, 2018.

[56] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. Citypersons: A diverse dataset for pedestrian detection. In *CVPR*, pages 3213–3221, 2017.

[57] Yunhua Zhang, Lijun Wang, Jinqing Qi, Dong Wang, Mengyang Feng, and Huchuan Lu. Structured siamese network for real-time visual tracking. In *ECCV*, pages 351–366, 2018.

[58] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV*, pages 103–119, 2018.