Towards a Better Match in Siamese Network Based Visual Object Tracker

Anfeng He*[1], Chong Luo[2], Xinmei Tian[1], and Wenjun Zeng[2]

[1] CAS Key Laboratory of Technology in Geo-Spatial Information Processing and Application System, University of Science and Technology of China, Hefei, Anhui, China
heanfeng@mail.ustc.edu.cn, xinmei@ustc.edu.cn
[2] Microsoft Research, Beijing, China
{cluo, wezeng}@microsoft.com

Abstract. Recently, Siamese network based trackers have received tremendous interest for their fast tracking speed and high performance. Despite the great success, this tracking framework still suffers from several limitations. First, it cannot properly handle large object rotation. Second, tracking gets easily distracted when the background contains salient objects. In this paper, we propose two simple yet effective mechanisms, namely angle estimation and spatial masking, to address these issues. The objective is to extract more representative features so that a better match can be obtained between the same object from different frames. The resulting tracker, named Siam-BM, not only significantly improves the tracking performance, but more importantly maintains the realtime capability. Evaluations on the VOT2017 dataset show that Siam-BM achieves an EAO of 0.335, which makes it the best-performing realtime tracker to date.

Keywords: Realtime Tracking · Siamese Network · Deep Convolutional Neural Networks

1 Introduction

Generic visual object tracking is a challenging and fundamental task in the area of computer vision and artificial intelligence. A tracker is initialized with only the bounding box of an unknown target in the first frame. The task of the tracker is to predict the bounding boxes of the target in the following frames. There are numerous applications of object tracking, such as augmented reality, surveillance and autonomous systems. However, robust and precise tracking is still an open problem.

In the past a few years, with the penetration of deep convolutional neural networks (DCNN) in various vision problems, there emerge a large number of DCNN-based trackers [12,2,5,22,3,24,25,21,11,20,13,7], among which the siamese network based trackers have received great attention. The pioneering work in

* This work is carried out while Anfeng He is an intern in MSRA.

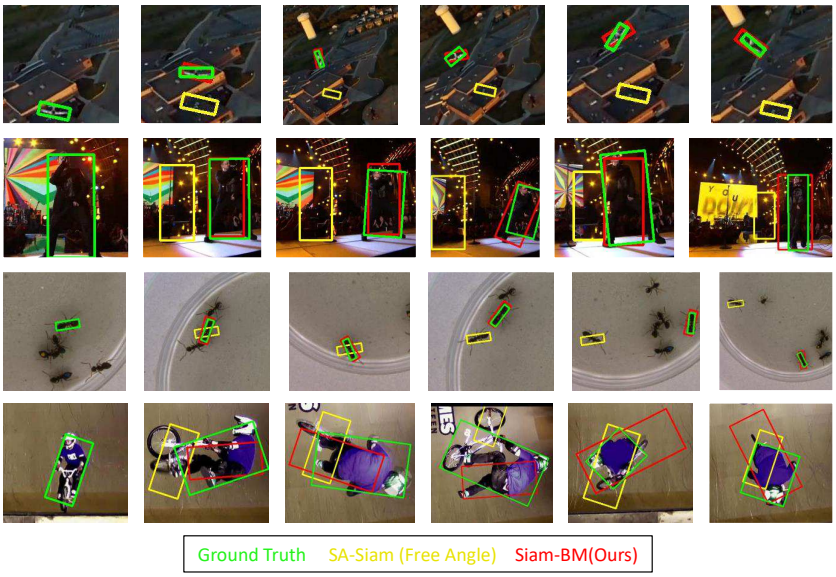

Fig. 1. Comparison with our tracker and baseline tracker. Best view in color.

this category is the SiamFC tracker [2]. The basic idea is to use the same DCNN to extract features from the target image patch and the search region, and to generate a response map by correlating the two feature maps. The position with the highest response indicates the position of the target object in the search region. The DCNN is pre-trained and remains unchanged during testing time. This allows SiamFC to achieve high tracking performance in realtime. Follow-up work of SiamFC includes SA-Siam, SiamRPN, RASNet, EAST, DSiam, CFNET and SiamDCF [12,18,26,14,10,29,25].

Despite the great success of siamese network-based trackers, there are still some limitations in this framework. First, as previous research [32,23,15] pointed out, the CNN features are not invariant to large image transformations such as scaling and rotation. Therefore, SiamFC does not perform well when the object has large scale change or in-plane rotation. This problem is exaggerated when the tracked object is non-square, because there is no mechanism in the SiamFC framework that can adjust the orientation or the aspect ratio of the tracked object bounding box. Second, it is hard to determine the spatial region from which DNN features should be extracted to represent the target object. Generally speaking, including a certain range of the surrounding context is helpful to tracking, but too many of them could be unprofitable especially when the background contains distracting objects. Recently, Wang et al. [26] also observed this problem and they propose to train a feature mask to highlight the features of the target object.

In this paper, we revisit the SiamFC tracking framework and propose two simple yet effective mechanisms to address the above two issues. The computational overhead of these two mechanisms is kept low, such that the resulting tracker, named Siam-BM, can still run in real-time on GPU.

First, our tracker not only predicts the location and the scale of the target object, but also predicts the angle of the target object. This is simply achieved by enumerating several angle options and computing DCNN features for each option. However, in order to maintain the high speed of the tracker, it is necessary to trim the explosive number of (scale, angle) combinations without tampering the tracking performance. Second, we propose to selectively apply a spatial mask to CNN feature maps when the possibility of distracting background objects is high. We apply such a mask when the aspect ratio of the target bounding box is far apart from 1:1. This simple mechanism not only saves the efforts to train an object-specific mask, but allows the feature map to include a certain amount of information of the background, which is in general helpful to tracking. Last, we also adopt a simple template updating mechanism to cope with the gradual appearance change of the target object. All these mechanisms are toward the same goal to achieve a better match between the same object from different frames. Therefore, the resulting tracker is named Siam-BM.

We carry out extensive experiments for the proposed Siam-BM tracker, over both the OTB and the VOT benchmarks. Results show that Siam-BM achieves an EAO of 0.335 at the speed of 32 fps on VOT-2017 dataset. It is the best-performing realtime tracker in literature.

The rest of the paper is organized as follows. We review related work in Section 2. In Section 3, we revisit the SiamFC tracking framework and explain the proposed two mechanisms in details. Section 4 provides implementation details of Siam-BM and presents the experimental results. We finally conclude with some discussions in Section 5.

2 Related Work

Visual object tracking is an important computer vision problem. It can be modeled as a similarity matching problem. In recent years, with the widespread use of deep neural networks, there emerge a bunch of Siamese network based trackers, which performs similarity matching based on extracted DCNN features. The pioneering work in this category is the fully convolutional Siamese network (SiamFC) [2]. SiamFC extract DCNN features from the target patch and the search region using AlexNet. Then, a response map is generated by correlating the two feature maps. The object is tracked to the location where the highest response is obtained. A notable advantage of this method is that it needs no or little online training. Thus, real-time tracking can be easily achieved.

There are a large number of follow-up work [30,29,14,24,10,18,25,12,8,26] of SiamFC. EAST [14] attempts to speed up the tracker by early stopping the feature extractor if low-level features are sufficient to track the target. CFNet [29] introduces correlation filters for low level CNNs features to speed up tracking

without accuracy drop. SINT [24] incorporates the optical flow information and achieves better performance. However, since computing optical flow is computationally expensive, SINT only operates at 4 frames per second (fps). DSiam [10] attempts to online update the embeddings of tracked target. Significantly better performance is achieved without much speed drop. HP [8] tries to tune hyperparameters for each sequence in SiamFC [2] by optimizing it with continuous Q-Learning. RASNet [26] introduces three kinds of attention mechanisms for SiamFC [2] tracker. The authors share the same vision with us to look for more precise feature representation for the tracked object. SiamRPN [18] includes a region proposal subnetwork to estimate the aspect ratio of the target object. This network will generate a more compact bounding box when the target shape changes. SA-Siam [12] utilizes complementary appearance and semantic features to represent the tracked object. A channel-wise attention mechanism is used for semantic feature selection. SA-Siam achieves a large performance gain at a small computational overhead.

Apparently we are not the first who concerns transformation estimation in visual object tracking. In correlation filter based trackers, DSST [4] and SAMF [19] are early work that estimates the scale change of the tracked object. DSST [4] does so by learning separate discriminative correlation filters for translation and scale estimation. SAMF [19] uses a scale pyramid to search corresponding target scale. Recently, RAJSSC [31] proposes to perform both scale and angle estimation in a unified correlation tracking framework by using the Log-Polar transformation. In SiamFC-based trackers, while the scale estimation has been considered in the original SiamFC tracker, angle estimation has not been considered before.

There are also a couple of previous research efforts to suppress the background noise. SRDCF [6] and DeepSRDCF [5] reduce background noise by introducing the spatial regularization term in loss function during the online training of correlation filters. RASNet [26] and SA-Siam [12] are two SiamFC-based trackers. They adopt spatial attention or channel-wise attention in the feature extraction network. They both need careful training of the attention blocks.

3 Siam-BM Tracker

Our tracker Siam-BM is built upon the recent SA-Siam tracker [12], which is in turn built upon the SiamFC tracker [2]. The main difference between SA-Siam and SiamFC trackers is that the former extracts semantic features in addition to appearance features for similarity matching. In this section, we will first revisit the SiamFC tracking framework and then present the two proposed mechanisms in Siam-BM towards a better matching of object features.

3.1 An Overview of the SiamFC Tracking Framework

Fig.2 shows the basic operations in the SiamFC tracking framework. The input of the tracker is the target object bounding box B_0 in the first frame F_1. A

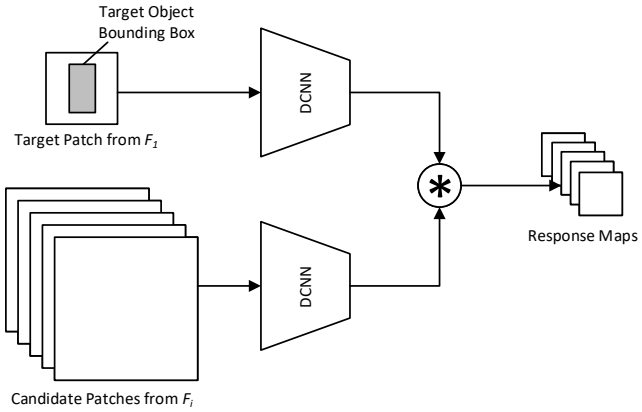

Fig. 2. The SiamFC Tracking Framework

bounding box can be described by a four-tuple (x, y, w, h), where (x, y) is the center coordinates and w, h are the width and the height, respectively. SiamFC crops the target patch T from the first frame, which is a square region covering B_0 and a certain amount of surrounding context. When the tracker comes to the i^{th} frame, several candidate patches $\{C_1, C_2, ...C_M\}$ are drawn, all of which are centered at the tracked location of the previous frame, but differ in scales. In the original SiamFC [2] work, M is set to 3 or 5 to deal with 3 or 5 different scales.

Both the target patch and the candidate patches go through the same DCNN network, which is fixed during testing time. The process of extracting DCNN features can be described by a function $\phi(\cdot)$. Then, $\phi(T)$ is correlated with $\phi(C_1)$ through $\phi(C_M)$ and M response maps $\{R_1, R_2, ...R_M\}$ are computed. The position with the highest value in the response maps is determined by:

$$(x_i, y_i, m_i) = \underset{x,y,m}{\arg\max} R_m, \quad (m = 1...M), \tag{1}$$

where x_i, y_i are the coordinates of the highest-value position and m is the index of the response map from which the highest value is found. Then, the tracking result is given by $B_i = (x_i, y_i, s_{m_i} \cdot w, s_{m_i} \cdot h)$, where s_{m_i} is the scale factor of the m_i^{th} candidate patch.

In this process, SiamFC tracker only determines the center position and the scale of the target object, but keeps the orientation and aspect ratio unchanged. This becomes a severe limitation of SiamFC tracker.

3.2 Angle Estimation

As previous research [32,15,23] has pointed out, DCNN features are not invariant to large image transformations, such as scaling and rotation. While scaling has

been handled in the original SiamFC tracker, the rotation of the target object is not considered. Ideally, the change of object angle, or object rotation, can be similarly addressed as object scaling. Specifically, one could enumerate several possible angle changes and increase the number of candidate patches for similarity matching. However, with M scale choices and N angle choices, the number of candidate patches becomes $M \times N$. It is quite clear that the tracker speed is inversely proportional to the number of candidate patches. Using contemporary GPU hardware, a SiamFC tracker becomes non-realtime even when $M = N = 3$.

Knowing the importance of realtime tracking, we intend to find a mechanism to reduce the number of candidate patches without tampering the performance of the tracker. The solution turns out to be a simple one: the proposed Siam-BM tracker adjusts the properties (scale or angle) of the tracked object only one at a time. In other words, Siam-BM can adjust both scale and angle in two frames, if necessary. As such, the number of candidate patches is reduced from $M \times N$ to $M + N - 1$. In our implementation, $M = N = 3$, so only 5 candidate patches are involved in each tracking process.

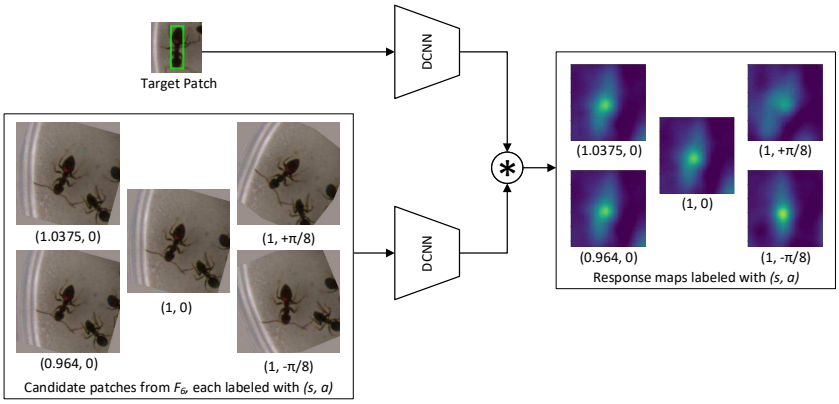

Fig. 3. Illustrating the scale and angle estimation in Siam-BM.

Mathematically, each candidate patch is associated with an (s, a) pair, where s is the scaling factor and a is the rotation angle. It is forced that $s = 1$ (no scale change) when $a \neq 0$ (angle change), and $a = 0$ when $s \neq 1$. Similarly, the tracked object is determined by:

$$(x_i, y_i, k_i) = \arg\max_{x,y,k} R_k, \quad (k = 1, 2, ...K), \tag{2}$$

where $K = M + N - 1$ is the number of candidate patches. (x_i, y_i) gives the center location of the tracked object and k_i is associated with an (s, a) pair,

giving an estimation of the scale and angle changes. Both types of changes are accumulated during the tracking process.

Fig.3 illustrates the scale and angle estimation in the proposed Siam-BM tracker. In the figure, each candidate patch and each response map are labeled with the corresponding (s, a) pair. We can find that, when the target object has the same orientation in the target patch as in the candidate patch, the response is dramatically increased. In this example, the highest response in the map with $(1, -\pi/8)$ is significantly higher than the top values in other maps.

3.3 Spatial Mask

Context information is helpful during tracking. However, including too much context information could be distracting when the background has salient objects or prominent features. In the SiamFC framework, the target patch is always a square whose size is determined only by the area of the target object. Fig.4 shows some examples of target patches containing objects with different aspect ratios. It can be observed that, when the target object is a square, the background is made up of narrow stripes surrounding the target object, so the chance of having an integral salient object in it is small. But when the aspect ratio of the target object is far apart from 1 (vertical or horizontal), it is more likely to have salient objects in the background area.

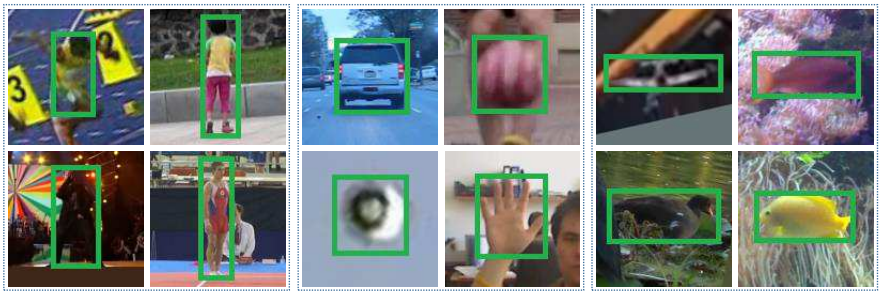

Fig. 4. Some examples of target patches containing objects with different aspect ratios. Target patches tend to include salient background objects when the object aspect ratio is far apart from 1:1.

We propose to selectively apply spatial mask to the target feature map. In particular, when the aspect ratio of the target object exceeds a predefined threshold th_r, a corresponding mask is applied. We have mentioned that the proposed Siam-BM tracker is built upon a recent tracker named SA-Siam [12]. In SA-Siam, there is an attention module which serves a similar purpose. However, we find that the spatial mask performs better and is more stable than the channel-wise attention scheme. Therefore, we replace the channel attention model in SA-Siam with spatial masking in Siam-BM.

3.4 The Siam-BM Tracker

Siam-BM is built upon SA-Siam [12], which contains a semantic branch and an appearance branch for feature extraction. The target patch has a size of 127×127 as in SiamFC, and the candidate patches have a size of 255×255. We set $M = N = 3$, so that there are 5 candidate patches and their corresponding scale and angle settings are $(s, a) = (1.0375, 0), (0.964, 0), (1, 0), (1, \pi/8), (1, -\pi/8)$. Correspondingly, five response maps are generated after combining semantic and appearance branches. Similar to SiamFC and SA-Siam, normalization and cosine window are applied to each of the five response maps. An angle penalty of 0.975 is applied when $a \neq 0$ and a scale penalty of 0.973 is applied when $s \neq 1$.

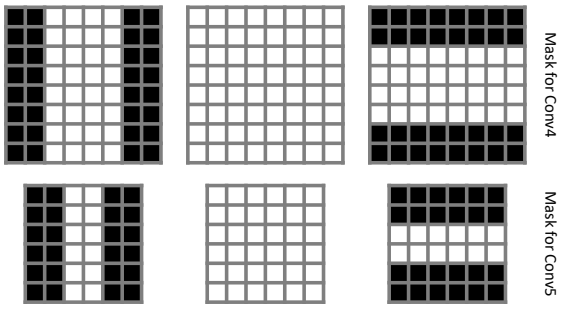

Fig. 5. Spatial feature mask when the aspect ratio of target object exceeds a predefined threshold. Left two masks: $h/w > th_r$; right two masks: $w/h > th_r$; middle two masks: $\max\{w/h, h/w\} < th_r$.

Following SA-Siam, both conv4 and conv5 features are used, and the spatial resolutions are 8×8 and 6×6, respectively. Spatial mask is applied when the aspect ratio is greater than $th_r = 1.5$. Fig.5 shows the fixed design of spatial masks when $\max\{\frac{w}{h}, \frac{h}{w}\} > th_r$. The white grids indicate a coefficient of 1 and the black grids indicate a coefficient of 0.

In addition, we perform template updating in Siam-BM. The template for frame t, denoted by $\phi(T_t)$ is defined as followings:

$$\phi(T_t) = \lambda_S \times \phi(T_1) + (1 - \lambda_S) \times \phi(T_t^u), \tag{3}$$

$$\phi(T_t^u) = (1 - \lambda_U) \times \phi(T_{t-1}^u) + \lambda_U \times \hat{\phi}(T_{t-1}). \tag{4}$$

where $\hat{\phi}(T_{t-1})$ is the feature of the tracked object in frame $t-1$. It can be cropped from the feature map of candidate regions of frame $t-1$. $\phi(T_t^u)$ is the moving average of feature maps with updating rate λ_U. λ_S is the weight of the first frame. In our implementation, we set $\lambda_S = 0.5$, $\lambda_U = 0.006$.

Note that the spatial mask is only applied to the semantic branch. This is because semantic responses are more sparse and centered than appearance

responses, and it is less likely to exclude important semantic responses with spatial masks. The attention module in SA-Siam is removed.

4 Experiments

In this section, we evaluate the performance of Siam-BM tracker against state-of-the-art realtime trackers and carry out ablation studies to validate the contribution of angle estimation and spatial masking.

4.1 Datasets and Evaluation Metrics

OTB: The object tracking benchmarks (OTB) [27,28] consist of two major datasets, namely OTB-2013 and OTB-100, which contain 51 and 100 sequences respectively. The two standard evaluation metrics on OTB are success rate and precision. For each frame, we compute the IoU (intersection over union) between the tracked and the groundtruth bounding boxes, as well as the distance of their center locations. A success plot can be obtained by evaluating the success rate at different IoU thresholds. Conventionally, the area-under-curve (AUC) of the success plot is reported. The precision plot can be acquired in a similar way, but usually the representative precision at the threshold of 20 pixels is reported.

VOT: We use the recent version of the VOT benchmark, denoted by VOT2017 [17]. The VOT benchmarks evaluate a tracker by applying a reset-based methodology. Whenever a tracker has no overlap with the ground truth, the tracker will be re-initialized after five frames. Major evaluation metrics of VOT benchmarks are accuracy (A), robustness (R) and expected average overlap (EAO). A good tracker has high A and EAO scores but low R scores.

In addition to the evaluation metrics, VOT differs from OTB in groundtruth labeling. In VOT, the groundtruth bounding boxes are not always upright. Therefore, we only evaluate the full version of Siam-BM on VOT. OTB is used to validate the effectiveness of spatial mask.

4.2 Training Siam-BM

Similar to SA-Siam, the appearance network and the fuse module in semantic branch are trained using the ILSVRC-2015 video dataset (only color images are used). The semantic network uses the pretrained model for image classification on ILSVRC. Among a total of more than 4,000 sequences, there are around 1.3 million frames and about 2 million tracked objects with groundtruth bounding boxes. We strictly follow the separate training strategy in SA-Siam and the two branches are not combined until testing time.

We implement our model in TensorFlow [1] 1.7.0 framework in Python 3.5.2 environment. Our experiments are performed on a PC with a Xeon E5-2690 2.60GHz CPU and a Tesla P100 GPU.

4.3 Ablation Analysis

Angle estimation: We first evaluate whether angle estimation improves the performance on the VOT benchmark. Spatial masking is not added, so our method is denoted by Siam-BM (w/o mask). There are two baseline methods. In addition to vanilla SA-Siam, we implement a variation of SA-Siam, denoted by SA-Siam (free angle). Specifically, when the bounding box of the tracked object is not upright in the first frame, the reported tracking results are tilted by the same angle in all the subsequent frames. Table 1 shows the EAO as well as accuracy and robustness of the three comparing schemes. Note that the performance of SA-Siam is slightly better than that reported in their original paper, which might due to some implementation differences. We can find that angle estimation significantly improves the tracker performance even when it is compared with the free angle version of SA-Siam.

Table 1. Comparison between Siam-BM (w/o mask) and two baseline trackers shows the effectiveness of angle estimation.

Trackers	EAO	Accuracy	Robustness
SA-Siam (vanilla)	0.261	0.505	1.276
SA-Siam (free angle)	0.287	0.529	1.234
Siam-BM (w/o mask)	0.301	0.544	1.305

Spatial mask: We use the OTB benchmark for this ablation study. Angle estimation is not added to the trackers evaluated in this part, therefore our method is denoted by Siam-BM (mask only). For all the 100 sequences in OTB benchmark, we compute the aspect ratio of the target object using $r = \max(\frac{h}{w}, \frac{w}{h})$, where w and h are the width and height of the groundtruth bounding box in the first frame. We set a threshold th_r, and if $r > th_r$, the object is called an *elongated object*. Otherwise, we call the object a *mediocre object*. At the testing stage, Siam-BM (mask only) applies spatial mask to elongated objects. At the training stage, we could either use the full feature map or the masked feature map for elongated objects. For mediocre objects, mask is not applied in training or testing. The comparison between different training and testing choices are included in Table 2. Comparing (3)(4) with (5)(6) in the Table, we can conclude that applying spatial mask significantly improves the tracking performance for elongated objects. Comparison between (3) and (4) shows that training with spatial mask will further improve the performance for elongated objects, which agrees with the common practice to keep the consistency of training and testing. An interesting finding is obtained when we comparing (1) with (2). If we apply spatial mask to elongated objects during training, the Siamese network seems to be trained in a better shape and the tracking performance for mediocre objects is also improved even though no spatial mask is applied during testing time.

Table 2. Comparison between training and testing choices with or without spatial mask.

Training \ Testing	mediocre object	elongated object	
	no mask	w/ mask	w/o mask
w/ mask	0.681 (1)	0.654 (3)	0.581 (5)
w/o mask	0.665 (2)	0.644 (4)	0.609 (6)

We then compare the performance of Siam-BM (mask only) with the state-of-the-art realtime trackers on OTB-2013 and OTB-100, and the results are shown in Table 3 and Fig.6. The improvement of Siam-BM (mask only) with respect to SA-Siam demonstrates that the simple spatial masking mechanism is indeed effective.

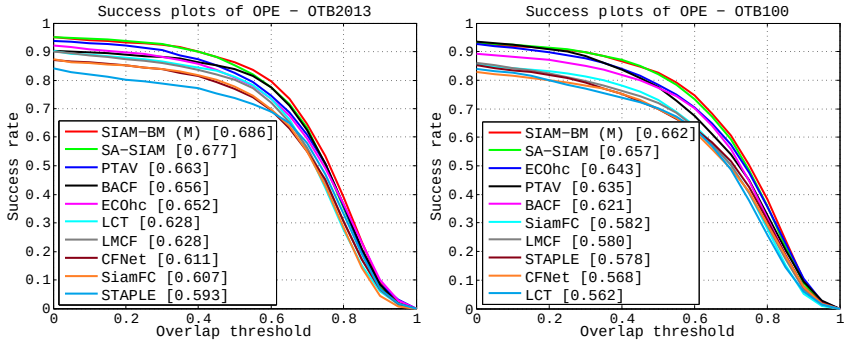

Fig. 6. Comparing SiamBM (Mask only) with other high performance and real-time trackers

Fig. 7 shows the relationship between the object aspect ratio and the performance gain of spatial masking. Consistent with our observation, when the aspect ratio is far apart from 1, doing spatial masking is helpful. However, when the object is a mediocre one, masking the features is harmful. In general, the performance gain of feature masking is positively correlated with the deviation of aspect ratio from 1.

Siam-BM Finally, we show in Tabel 4 how the performance of Siam-BM is gradually improved with our proposed mechanisms. The EAO of the full-fledged Siam-BM reaches 0.335 on VOT2017, which is a huge improvement from 0.287 achieved by SA-Siam. Of course, as we add more mechanisms in Siam-BM, the tracking speed also drops, but the full-fledged Siam-BM still runs in realtime.

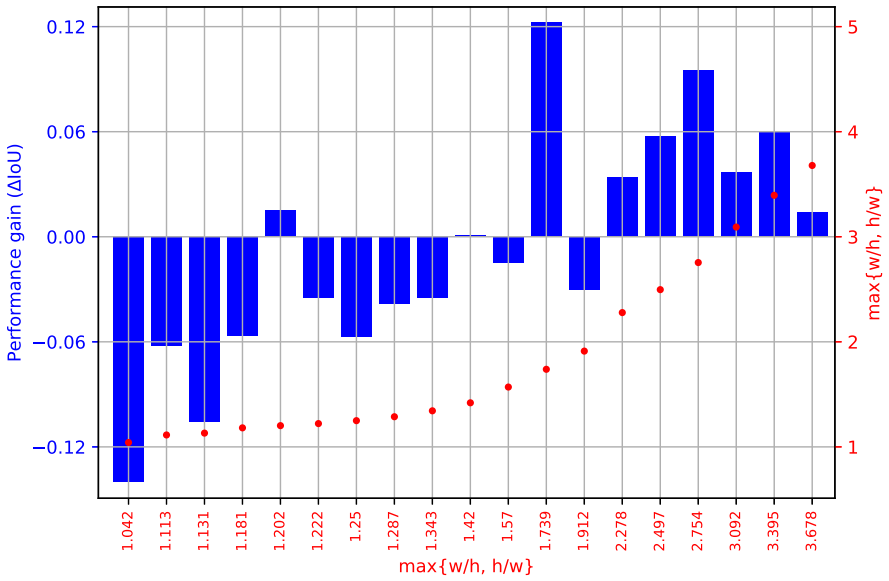

Fig. 7. Performance gain of feature masking is positively correlated with the deviation of aspect ratio from 1.

4.4 Comparison with the State-of-the-Art Trackers

We evaluate our tracker in VOT2017 main challenge and realtime subchallenge. The final model in this paper combines all components mentioned in previous section. We do not evaluate the final model in OTB because the groundtruth labeling in OTB is always upright bounding boxes and applying rotation does not produce a higher IoU even when the tracked bounding box is more precise and tight.

As shown in Fig.8, our Siam-BM tracker is among the best trackers even when non-realtime trackers are considered. From Fig.9, we can see that Siam-

Table 3. Comparing SiamBM (Mask only) with other high performance and real-time trackers

Trackers	OTB2013		OTB100		FPS
	AUC	Prec.	AUC	Prec.	
ECOhc [3]	0.652	0.874	0.643	0.856	60
BACF [16]	0.656	0.859	0.621	0.822	35
PTAV [9]	0.663	0.895	0.635	0.849	25
SA-Siam [12] (baseline)	0.677	0.896	0.657	0.865	50
Siam-BM (mask only)	**0.686**	**0.898**	**0.662**	**0.864**	48

Table 4. Analysis of our tracker Siam-BM on the VOT2017. The impact of progressively integrating one contribution at a time is depicted.

	Baseline SA-Siam	⟹	Angle Estimation	⟹	Spatial Mask	⟹	Template Updating
EAO	0.287		0.301		0.322		**0.335**
Accuracy	0.529		0.544		0.551		**0.563**
Robustness	1.234		1.305		1.07		**0.977**
FPS	50		35		34		32

BM outperforms all realtime trackers in VOT2017 challenge by a large margin. The Accuracy-Robustness plot in Fig.10 also shows the superiority of our tracker.

We also compare the EAO value of our tracker with some of the latest trackers. RASNet [26] achieves an EAO number of 0.281 in the main challenge and 0.223 in the realtime subchallenge. SiamRPN [18] achieves an EAO number of 0.243 in the realtime subchallenge. The EAO number achieved by Siam-BM is much higher.

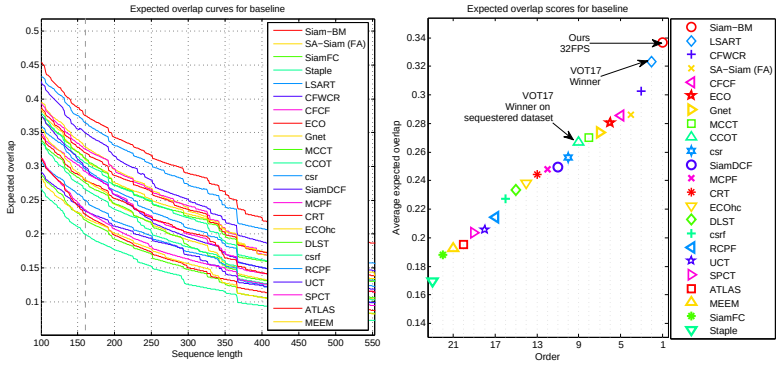

Fig. 8. EAO curves and rank in VOT17 main challenge. FA represents Free Angle here.

5 Conclusion

In this paper, we have designed a SiamFC-based visual object tracker named Siam-BM. The design goal is to achieve a better match between feature maps of the same object from different frames. In order to keep the realtime capability of the tracker, we propose to use low-overhead mechanisms, including parallel scale and angle estimation, fixed spatial mask and moving average template updating. The proposed Siam-BM tracker outperforms state-of-the-art realtime trackers by a large margin on the VOT2017 benchmark. It is even comparable to the best

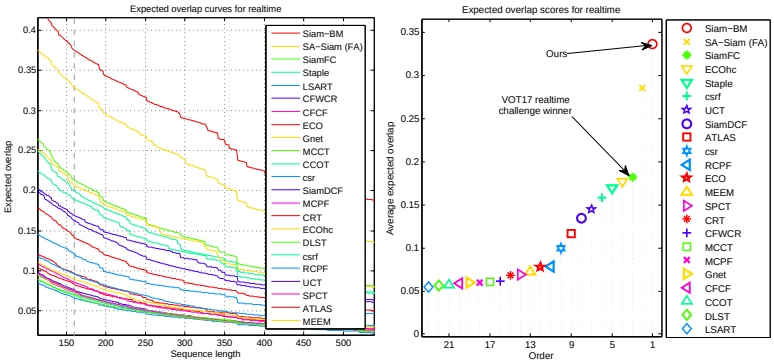

Fig. 9. EAO curves and rank in VOT17 realtime challenge. FA represents Free Angle here.

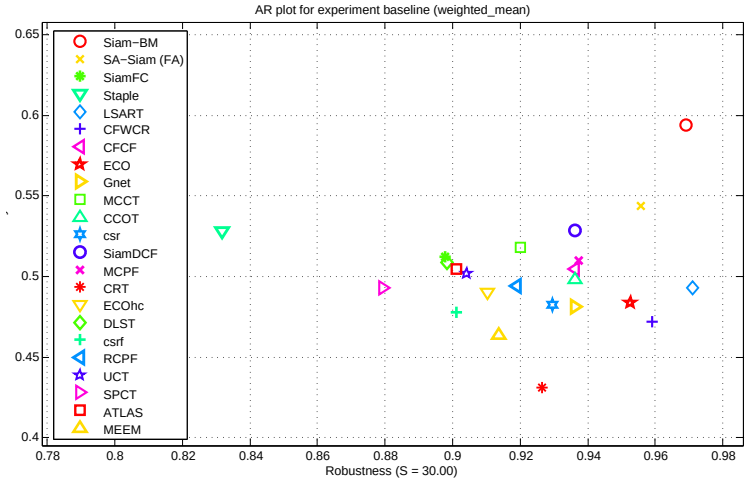

Fig. 10. Accuracy and robustness plots in VOT17 main challenge. Best trackers are closer to the topright corner. FA represents Free Angle here.

non-realtime trackers. In the future, we will investigate the adaptation of object aspect ratio during tracking.

Acknowledgement

This work was supported in part by National Key Research and Development Program of China 2017YFB1002203, NSFC No.61572451, No.61390514, and No.61632019, Youth Innovation Promotion Association CAS CX2100060016, and Fok Ying Tung Education Foundation WF2100060004.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 (2016) 9
2. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. In: European Conference on Computer Vision Workshop. pp. 850–865. Springer (2016) 1, 2, 3, 4, 5
3. Danelljan, M., Bhat, G., Shahbaz Khan, F., Felsberg, M.: Eco: Efficient convolution operators for tracking. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017) 1, 12
4. Danelljan, M., Häger, G., Khan, F., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: British Machine Vision Conference, Nottingham, September 1-5, 2014. BMVA Press (2014) 4
5. Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M.: Convolutional features for correlation filter based visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 58–66 (2015) 1, 4
6. Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4310–4318 (2015) 4
7. Danelljan, M., Robinson, A., Khan, F.S., Felsberg, M.: Beyond correlation filters: Learning continuous convolution operators for visual tracking. In: European Conference on Computer Vision. pp. 472–488. Springer (2016) 1
8. Dong, X., Shen, J., Wang, W., Liu, Y., Shao, L., Porikli, F.: Hyperparameter optimization for tracking with continuous deep q-learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018) 3, 4
9. Fan, H., Ling, H.: Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017) 12
10. Guo, Q., Feng, W., Zhou, C., Huang, R., Wan, L., Wang, S.: Learning dynamic siamese network for visual object tracking. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017) 2, 3, 4
11. Han, B., Sim, J., Adam, H.: Branchout: Regularization for online ensemble tracking with convolutional neural networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017) 1
12. He, A., Luo, C., Tian, X., Zeng, W.: A twofold siamese network for real-time object tracking. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018) 1, 2, 3, 4, 7, 8, 12
13. Held, D., Thrun, S., Savarese, S.: Learning to track at 100 fps with deep regression networks. In: European Conference on Computer Vision. pp. 749–765. Springer (2016) 1
14. Huang, C., Lucey, S., Ramanan, D.: Learning policies for adaptive tracking with deep feature cascades. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017) 2, 3
15. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: Advances in neural information processing systems. pp. 2017–2025 (2015) 2, 5
16. Kiani Galoogahi, H., Fagg, A., Lucey, S.: Learning background-aware correlation filters for visual tracking. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017) 12

17. Kristan, M., et al.: The visual object tracking vot2015 challenge results. In: Proceedings of the IEEE international conference on computer vision workshops (2017) 9
18. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X.: High performance visual tracking with siamese region proposal network. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018) 2, 3, 4, 13
19. Li, Y., Zhu, J.: A scale adaptive kernel correlation filter tracker with feature integration. In: European Conference on Computer Vision. pp. 254–265. Springer (2014) 4
20. Lukezic, A., Vojir, T., Cehovin Zajc, L., Matas, J., Kristan, M.: Discriminative correlation filter with channel and spatial reliability. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017) 1
21. Ma, C., Huang, J.B., Yang, X., Yang, M.H.: Hierarchical convolutional features for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3074–3082 (2015) 1
22. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4293–4302 (2016) 1
23. Shen, X., Tian, X., He, A., Sun, S., Tao, D.: Transform-invariant convolutional neural networks for image classification and search. In: Proceedings of the 2016 ACM on Multimedia Conference. pp. 1345–1354. ACM (2016) 2, 5
24. Tao, R., Gavves, E., Smeulders, A.W.: Siamese instance search for tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1420–1429 (2016) 1, 3, 4
25. Wang, Q., Gao, J., Xing, J., Zhang, M., Hu, W.: Dcfnet: Discriminant correlation filters network for visual tracking. arXiv preprint arXiv:1704.04057 (2017) 1, 2, 3
26. Wang, Q., Teng, Z., Xing, J., Gao, J., Hu, W., Maybank, S.: Learning attentions: Residual attentional siamese network for high performance online visual tracking. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018) 2, 3, 4, 13
27. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2411–2418 (2013) 9
28. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(9), 1834–1848 (2015) 9
29. Xu, H., Gao, Y., Yu, F., Darrell, T.: End-to-end learning of driving models from large-scale video datasets. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017) 2, 3
30. Yang, T., Chan, A.B.: Recurrent filter learning for visual tracking. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017) 3
31. Zhang, M., Xing, J., Gao, J., Shi, X., Wang, Q., Hu, W.: Joint scale-spatial correlation tracking with adaptive rotation estimation. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 32–40 (2015) 4
32. Zhou, Y., Ye, Q., Qiu, Q., Jiao, J.: Oriented response networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4961–4970. IEEE (2017) 2, 5