A strong feature representation for siamese network tracker

Zhipeng Zhou¹,² · Rui Zhang¹,² · Dong Yin¹,²

Received: 15 August 2019 / Revised: 1 June 2020 / Accepted: 4 June 2020 / Published online: 8 July 2020 © Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract
Because AlexNet is too shallow to form a strong feature representation, the trackers based on the Siamese network have an accuracy gap comparing with state-of-the-art algorithms. Both deep features and appearance features benefit tracking accuracy. To combine these two kinds features, the modified pre-trained VGG16 network is fine-tuned as one branch of the backbone network. Secondly, an AlexNet branch is attached after the third convolutional layer of VGG16. Thus the response maps from both branches are merged to form a preliminary strong feature representation with deep features and shallow appearance features. Thirdly, a new mean Peak-to-side ratio (mPSR) loss is designed to help network learn target features adaptively. A channel attention block and the Average-Peak-to-Correlation Energy (APCE) are designed to help select contributed features and suppress distractors. SiamPF only takes ILSVRC2015-VID as training dataset, but it achieves excellent performance on OTB-2013 / OTB-2015 / VOT2015 / VOT2016 / VOT2017 while maintaining the real-time performance of 41FPS on the GTX 1080Ti.

Keywords Siamese network · Feature representation · mPSR

1 Introduction
Visual tracking is a fundamental topic in computer vision. It can be divided into two sub-topics base on target: single object tracking and multiple object tracking [31]. Many single object tracking methods have been studied in recent years. They are mainly based on either correlation filter framework or deep learning framework. Correlation filter was introduced to computer vision by David S. Bolme [3] who proposed a tracker named MOSSE based on correlation filter. Henriques J.F proposed a method called CSK [19], which developed the intensive sampling and the kernel trick based on MOSSE. Furthermore, he exploited multi-channel HOG feature into KCF [20], which was an enhanced vision of CSK. Similarly, Danelljan M [5] developed CSK with multi-channel color names (CN) feature. Due to
their good performances, HOG and CN have became the most popular hand-craft features in recent years. However, hand-craft features are not suitable for all targets, which limits the performance of these trackers. Thus, leveraging data-driven features seem to a better way for target representation. Combining with features extracted from CNN, the correlation filter based methods such as DeepSRDCF [6], C-COT [9], ECO [10] certainly have a higher accuracy. On the other hand, trackers mentioned above require complex setup and high computation that could hardly meet the real-time property. Therefore, end-to-end deep networks are expected to be adapted into object tracking. In this way, some trackers are proposed. For example, GOTURN [21] is proposed to learn a generic relationship between object motion and appearance with an end-to-end deep network. But it has a great accuracy gap comparing with state-of-the-art trackers. SANet [12] leverages self-structure information of object to distinguish it from distractors. It utilizes RNN to model object structure, and incorporates it into CNN to improve its robustness to similar distractors. It can only run in 1FPS though has a strong motivation and great performance. TCNN [29] employed CNNs to represent target appearances. Multiple CNNs in TCNN collaborate to estimate target states and determine the desirable paths for online model updates in the tree. As described above, TCNN needs to make several tree judgments for each frame, because only one candidate box score is generated at a time. At the same time, TCNN is time-consuming, and many features need to be stored, which takes up a lot of space. Some recent trackers [11] take advantage reinforcement learning to optimize hyperparameter and achieve the-state-of-art performance. All these methods are either time-consuming or unsatisfactory in performance. Siamese network based trackers [2, 15, 18, 27, 33] strike a balance between performance and speed while keeping a simple structure without complex setup. For example, SiamFC [2] adopts a fully convolutional siamese network to get the response map between template image and search image. It operates at frame-rates beyond real-time, despite its extreme simplicity, achieved state-of-the-art performance.

In this paper, inspired by transfer learning, we utilize the pretrained model (VGG16) trained on ImageNet and finetuning it with last two layers as one branch of the backbone network. Another AlexNet-like branch is attached after the third layer of VGG16. In this way, both deep features with high-level semantic information and shallow features with more spatial information can be obtained. Meanwhile, a mPSR loss is designed to help network learn discriminative features adaptively. Furthermore, we design a channel attention block to select the contributed features. At last, like what have been done in correlation filter based trackers, we adopt modified APCE called APCEP to suppress the distractions. By testing on OTB2013 /OTB2015/VOT2015/VOT2017, our tracker achieve excellent performance while remains a real-time performance of 41FPS on the GTX 1080Ti.

The rest of the paper is organized as follows. Related work is introduced in Section 2. The proposed architecture and corresponding configurations are introduced in Section 3. Section 4 presents the experimental results on five datasets and Section 5 concludes this paper. At last, discussion is presented in Section 6.

2 Related works

2.1 Siamese network trackers

Siamese network, which has two branches for template and instance, is proposed to solve similarity matching problem. Some opinions regard visual tracking as a one-shot similarity
metric task. In this way, siamese network is extremely suitable for visual tracking. SiamFC[11] is the first approach to integrate the siamese network into visual tracking. After that, CFNet [33], DSiam [15] and SA-Siam [18] develop it by developing some operations on two branches. Borrowing the idea of Region Proposal Network (RPN) in object detection, SiamRPN [25] adds two extra branches to do classification and regression. Recently, some works such as SiamVGG [26] focus on taking advantage of the capability of deeper neural network.

2.2 Average Peak-to Correlation Energy (APCE)

In correlation filter trackers, updating only happens with high confidence coefficient to prevent model from being polluted. To evaluate the confidence of tracking object and reflect the volatility of response map, APCE [35] was proposed. Nowadays, it is adopted in some correlation filter trackers as a post-process on the response map to detect tracking failures. Its formulation can be concluded as follow:

$$APCE = \frac{|F_{max} - F_{min}|^2}{mean(\sum_{w,h}(F_{w,h} - F_{min})^2)}$$

$F_{max}$ is the peak value, $F_{min}$ is the minimum value of response map, and $F_{w,h}$ is the value in the location $(w, h)$.

3 Proposed method

The main idea of this paper is to build a strong feature representation without much training data or training skill. Figure 1 shows the main structure of SiamPF. AlexNet-like branch is attached after the third layer of the modified VGG16. During training, the layer weights in modified VGG16 is frozen except for the last two layers. We also design an attention block to process the outputs of template in AlexNet-like branch. APCEP is adopted to process the final response map.

![Fig.1 The network of SiamPF. The instance and exemplar images are fed into network as inputs. Extracting the features by a modified VGG16 branch and an AlexNet-like branch which is introduced after third layer of modified VGG16. In AlexNet-like branch, a channel attention block is adopted to process the exemplar feature map. The feature maps of exemplar and instance are utilized for cross-correlation. The 17*17 response maps from both branches are combined to form the final response map. After that, APCEP is taken as a post-processing to suppress distractors.](image-url)
3.1 Analysis of SiamFC

SiamFC employs a special way to acquire the annotation. The elements \( y[u] \) of the final response map (corresponding to the final response map in Fig. 1) are considered to belong to a positive example if they are within radius \( R \) of the centre \( c \) (accounting for the stride \( k \) of the network). It can be explained by (2). In such way, every object gets the same ground-truth box though they have different sizes.

\[
y[u] = \begin{cases} 
+1 & \text{if } \frac{k}{2} ||u - c|| \leq R \\
-1 & \text{else}
\end{cases}
\]

(2)

Examples projected from annotated response map are shown in Fig. 2. From it, we can see SiamFC does not care about the edge information, it just focus on predicting the centre point of object, which would lead network to learn some background features and degrade the feature representation.

3.2 mPSR Loss

To tackle the problem mentioned in 3.1. We design a new mPSR loss to help network learn more discriminative features of target. Denote the positive region of label in 3.1 as \( R \), and all pixels in \( R \) of response map are \( \{p_1, p_2, ... p_n\} \). Then \( mPSR \) and \( mPSR\_loss \) definition can be conclude as follow:

\[
mPSR = \frac{1}{n} \sum_{i=1}^{n} P_i
\]

(3)

\[
mPSR\_loss = e^{-\frac{1}{S} \sum_{i=1}^{n} P_i}
\]

(4)

Here \( S \) is the side response value except \( R \) in the response map. If the \( mPSR \) is larger, it means that target has larger response value than the distractors, and the \( mPSR\_loss \) would be smaller. The original cross-entropy loss is defined as:

\[
CE\_loss = -\frac{1}{|D|} \sum_{u \in D} \{y[u] \ast log(v[u]) + (1 - y[u]) \ast log(1 - v[u])\}
\]

(5)

Here \( v \) is the output response map of a single exemplar-instance pair and \( y \in \{+1, -1\} \) is its ground-truth label, \( D \) is the response map points set. In our experiment, we simply set a \( \beta = 1 \) to balance two objectivenesses. So the final objectiveness is:

\[
loss = CE\_loss + \beta mPSR\_loss
\]

(6)

Fig. 2 Labels projected from response map
### 3.3 Strong feature representation network

As analysis in 3.1, the current operations in SiamFC fails to capture some features (edge information, etc.). Thus, in order to obtain a strong feature representation, we choose a pretrained model as one branch of our backbone network.

Mainstream deep learning based trackers use modified AlexNet as their backbone network. As a matching task, AlexNet is not deep enough to obtain high semantic features. While in some deep networks like ResNet, some operations like padding would lead to object location preference. Therefore, we take modified VGG16 network pretrained on ImageNet and fine-tuning it as main branch of backbone network. We fine-tune modified VGG16 on last two layers and add another AlexNet-like branch after third layer of modified VGG16 as auxiliary branch. From both branches, we can obtain robust features with semantic and appearance information. The structure of modified VGG16 branch and AlexNet-like branch is detailed in Table 1.

We define the AlexNet branch and VGG16 branch as \( \varphi_A(\cdot) \) and \( \varphi_V(\cdot) \) respectively. After generating the feature map through the fully convolutional siamese network, we compute cross-correlation of two input images, which is define as Eq 7 and Eq 8, where \( z \) and \( x \) represent the exemplar image and instance image, respectively. \( \varphi_s(\cdot) \) represents a convolutional embedding function and \( b \| \) denotes the bias with value \( b \in R \). And the final response map \( S \) is defined as Eq 9. \( \lambda \) is a hyper-parameter to balance two response maps.

\[
\begin{align*}
    f_A(x_T, x_I) &= \varphi_A(z) \ast \varphi_A(x) + b\| \\
    f_V(x_T, x_I) &= \varphi_V(z) \ast \varphi_V(x) + b\| \\
    S &= \lambda \ast f_V(x_T, x_I) + (1 - \lambda) f_A(x_T, x_I)
\end{align*}
\]

### 3.4 Channel Attention Block

As a matching task, features of exemplar image play an important role. There exists many background noises in the exemplar image which would mislead tracker to drift. To further

| Modified VGG16 | AlexNet |
|----------------|---------|
| Type | K-size | C-in | C-out | Stride | Type | K-size | C-in | C-out | Stride |
| Conv1 | 3 | 3 | 64 | 1 |
| Conv2 | 3 | 64 | 64 | 1 |
| MaxPool1 | 2 | | | |
| Conv3 | 3 | 64 | 128 | 1 |
| Conv4 | 3 | 128 | 128 | 1 |
| MaxPool2 | 2 | | | |
| Conv5 | 3 | 128 | 256 | 1 |
| Conv6 | 3 | 256 | 256 | 1 |
| Conv7 | 3 | 256 | 256 | 1 |
| MaxPool3 | 2 | | | |
| Conv8 | 3 | 256 | 512 | 1 |
| Conv9 | 3 | 512 | 512 | 1 |
| Conv10 | 3 | 512 | 512 | 1 |
| Conv11 | 3 | 512 | 512 | 1 |
| Conv12 | 3 | 512 | 256 | 1 |

All the convolutional layers are integrated with BatchNorm and ReLU except the last one working for generating outputs. ‘K-size’ means ‘Kernel Size’, ‘C-in’ means ‘Channel in’, ‘C-out’ means ‘Channel out’
strengthen the current object feature representation, we add an attention block to the output feature map of exemplar images. We have different process in two branches. We adopt a channel attention block (see Fig. 3) for AlexNet branch while takes no measures for VGG16 branch. From our experiment, adding an attention block in VGG16 branch would hurt the performance, might because the VGG16 output feature map has high semantic information, attention block would hurt the relationship among these channels.

3.5 Post-process

Refer to the correlation filter based trackers, our tracker also benefits from post-process. Since the resized output response map has the same resolution as the input, APCE would be effective to suppress distractors. But APCE is still too compact to distinguish success and failure during tracking. So we modify Eq 1 to enlarge the output range of APCE. It can be concluded as follow:

\[
APCEP = \left( \frac{F_{\text{max}}^2 - F_{\text{min}}^2}{\text{mean}(\sum_{w,h}(F_{w,h} - F_{\text{min}})^2)} \right)^2
\]  

(10)

Figure 4 shows the comparison between APEC and APCEP in normal and occluded situation. From Fig. 4, we can see in the normal scene, APCEP gets a much higher score, but in occluded scene, APCEP degrade a lot and approximates as APCE. That means APCEP is more discriminating.
4 Experiment

Experiments are performed on five public datasets: OTB2013, OTB2015, VOT2015, VOT2016, VOT2017. OTB contains 100 videos while VOT contains 60 videos.

4.1 Implementation details

The experimental platform is CPU: i7-7700, GPU: GTX1080Ti, Memory: 16G, Operating system: Ubuntu 16.04. We train our model with straightforward SGD using PyTorch. Training is performed with 50 epochs. The gradients for each iteration are estimated using mini-batches of size 8, and the learning rate is annealed at every 20 epochs from $10^{-1}$ to $10^{-3}$. $\lambda$ is set to 0.75. The other hyper-parameters are set the same as SiamFC. During training, the VGG pre-trained part is frozen. The code would be released at https://github.com/zzpustc/SiamPF soon.

4.2 Evaluation

4.2.1 Results on OTB2013/OTB2015

OTB [37] contains 100 sequences that are collected from commonly used tracking sequences. The evaluation is based on two metrics: success and precision plot. When the overlap between groundtruth box and generated box is larger than a given threshold in a certain frame, we call it a successful frame. The precision plot shows the percentages of frames that tracking results are within 20 pixels from the target. So we test our tracker on the benchmark comparing with MCPF [38], ECOhc [10], CFNet [33], SiamFC [2], Staple [1]. Figures 5 and 6 show the results on OTB2013, OTB2015 respectively. From Figs. 5 and 6, we can see that SiamPF rank top among these trackers both in success plot and precision plot. SiamPF ranks second on both on OTB2013 and OTB2015. In fact, SiamPF shares the same rules with SiamFC: Both methods do not update the template and just search on a small region around the target. ECO-hc and MCPF are both correlation filter based methods, they update the template while searching on a larger region. That is, they are more...

![Success plot and precision plot of SiamPF on the OTB2013](image-url)
Fig. 6 Success plot and precision plot of SiamPF on the OTB2015

robust when the target is moving fast. SiamPF can extract more robust features comparing to ECO-hc and MCPF though it fails to solve the problem of fast movement. On the other hand, SiamPF is superior to MCPF and ECOhc on VOT2017.

4.2.2 Results on VOT2015/VOT2016/VOT2017

VOT [23] contains totally 60 sequences. Its metrics consist of accuracy and robustness. And the overall performance is evaluated using Expected Average Overlap (EAO) which takes account of both accuracy and robustness. Besides, a new real-time experiment is conducted. Figures 7, 8 and 9 show order of trackers (MDNet [30], DeepSRDCF [7], EBT [41], srdcf [8], sPST [22], scebt [34], nsamf [24], struck [17], CFCF [16], mcct [36], csr

Fig. 7 Expected overlap scores of SiamPF on the VOT2015/VOT2016, larger is better
Fig. 8 Expected overlap scores of SiamPF on the VOT2017, larger is better

[28], MCPF [38], CRT [4], ECOhc [10] etc.) by evaluating their EAO respectively on VOT2015, VOT2016 and VOT2017. Tables 2, 3 and 4 contain more qualitative details (Red means ranking first, blue means ranking second, green means ranking third).

From the present results above, we can see that different trackers have different advantages, but SiamPF is always among the top tier over all the evaluation metrics. Specially, SiamPF rank 1st in the real-time experiment on VOT2017, which means our tracker has the best balance on EAO and speed. MDNet learns multi-domain features to help itself adapt to a new domain easily while SiamPF does not have specific operation for domain information. CCOT provides a fine combination of deep and shallow features and its continuous convolution operators can achieve accurate sub-pixel positioning. But it fails to realize fast tracking with a high computation. Also according to Table 3, SiamPF performs better than CCOT on Accuracy. TCNN uses a tree structure combined with multiple CNNs to represent multiple appearances of the model, which can adapt to multiple transformations of the target, avoiding overfitting only the nearest frame. But it needs to make multiple tree judgments for each frame, because only one score for a candidate box is generated at a time. Thus it’s time-consuming and some appearance model would hurt robustness sometimes. SiamPF performs better than TCNN on Failures. Some other trackers, such as gnet, ssat, etc. are similar to TCNN and CCOT, although their comprehensive performance is better than that of SiamPF, there is always one worse than that of SiamPF in terms of Accuracy, Failures and Speed.

4.3 Qualitative performance

In this section, we present more detailed comparisons on OTB2015, and their success plots and precision plots are shown in Figs. 9 and 10, respectively. We mainly divide the sequences in OTB2015 into 11 kinds: Motion Blur, Out-of-Plane Rotation, Scale Variation, Background Clutters, Deformation, Illumination Variation, Out-of-View, Low Resolution, In-Plane Rotation, Fast Motion and Occlusion.
As expected, in most cases, SiamPF ranks top on both success plots and precision plots and has a little gap compared to MCPF and ECO-hc. But in the case of **Low Resolution**, SiamPF has a poor ranking and the same thing happens to MCPF and ECO-hc. One reason is that SiamPF which is a deeper CNN compared to SiamFC requires richer input information. That means, SiamPF maybe overfitting on some kind deep pattern. On the other hand, SiamPF is robust to **Illumination Variation** which is a serious problem among many computer vision tasks.

![Success plot in different scenarios](image)
### Table 2: Detail information about several state-of-the-art trackers performance on the VOT2015

| Tracker | EAO   | Accuracy | Failures  | A-R rank |
|---------|-------|----------|-----------|----------|
| MDNet   | 0.3783| 0.5991   | 13.1519   | 1        |
| DeepSRDCF| 0.3181| 0.5565   | 16.9525   | 3        |
| EBT     | 0.3130| 0.4596   | 15.3702   | 4        |
| srdcf   | 0.2877| 0.5529   | 21.2642   | 5        |
| LDP     | 0.2785| 0.4841   | 23.8973   | 6        |
| sPST    | 0.2767| 0.5479   | 26.2529   | 7        |
| scebt   | 0.2548| 0.5423   | 31.8157   | 8        |
| nsamf   | 0.2536| 0.5246   | 25.6161   | 9        |
| struck  | 0.2458| 0.4537   | 27.1530   | 10       |
| SiamPF  | 0.3263| 0.5905   | 18.6719   | 2        |

### Table 3: Detail information about several state-of-the-art trackers performance on the VOT2016

| Tracker | EAO   | Accuracy | Failures  | A-R rank |
|---------|-------|----------|-----------|----------|
| CCOT    | 0.3310| 0.5332   | 16.5817   | 1        |
| TCNN    | 0.3249| 0.5470   | 17.9393   | 2        |
| SSAT    | 0.3207| 0.5703   | 19.2720   | 3        |
| MLDF    | 0.3106| 0.4873   | 15.0437   | 4        |
| Staple  | 0.2952| 0.5433   | 23.8950   | 6        |
| DDC     | 0.2929| 0.5337   | 20.9812   | 7        |
| EBT     | 0.2913| 0.4529   | 15.1935   | 8        |
| STAPLEp | 0.2862| 0.5523   | 24.3165   | 9        |
| DNT     | 0.2783| 0.5087   | 19.5438   | 10       |
| SiamPF  | 0.3049| 0.5454   | 17.3332   | 5        |

### Table 4: Detail information about several state-of-the-art trackers performance on the VOT2017

| Tracker | EAO(baseline) | Accuracy | Failures  | A-R rank | EAO(real-time) |
|---------|---------------|----------|-----------|----------|---------------|
| CFCF    | 0.2857        | 0.5049   | 19.6495   | 1        | 0.0587        |
| gnet    | 0.2737        | 0.4999   | 17.3674   | 2        | 0.0599        |
| mctt    | 0.2703        | 0.5228   | 19.4526   | 3        | 0.0605        |
| csr     | 0.2561        | 0.4846   | 23.5731   | 4        | 0.0993        |
| MCPF    | 0.2478        | 0.5035   | 25.9600   | 6        | 0.0602        |
| CRT     | 0.2441        | 0.4639   | 21.0611   | 7        | 0.0683        |
| ECOhc   | 0.2384        | 0.4893   | 28.7674   | 8        | 0.1767        |
| DLST    | 0.2332        | 0.5038   | 24.6046   | 9        | 0.0568        |
| DACF    | 0.2285        | 0.4494   | 25.2403   | 10       | 0.2120        |
| SiamPF  | **0.2554**    | **0.5006**| **23.3362**| **5**    | **0.2376**    |
4.4 Ablation analysis

In this section, we analyse the influence of each operation to the final performance, and it can be concluded in Table 5. From Table 5, it can be observed that pretrained model bring the greatest promotion. AlexNet-like branch and mPSR loss also carry out 0.8% and 0.5% improvement, respectively. Both channel attention block and APCEP benefit for our tracker.
Influence analysis of each operation

| mPSR loss | Frozen Pretrained model | AlexNet branch | Channel attention | APCEP | OTB2013 AUC |
|-----------|-------------------------|----------------|------------------|-------|-------------|
| ✓         | ✓                       | ✓              | ✓                | ✓     | 0.6610      |
| ✓         | ✓                       | ✓              | ✓                | ✓     | 0.6565      |
| ✓         | ✓                       | ✓              | ✓                | ✓     | 0.6511      |
| ✓         | ✓                       | ✓              | ✓                | ✓     | 0.6431      |
| ✓         | ✓                       | ✓              | ✓                | ✓     | 0.6080      |

✓ means adding this operation into tracker

5 Conclusion

In this work, we propose a tracker called SiamPF, which finetunes pretrained VGG16 model in different stages to obtain multi-layer features. We design an attention block to strengthen the feature representation. At last, we modified APCE to process the score map. In this way, SiamPF achieve top performance on OTB2013/OTB2015/VOT2015/VOT2016/VOT2017. But there still exists many problems remains to be solved. The shallow features are not fully explored. Spatial-temporal property should be considered in long-term tracking. Our next work would focus on solving these problems.

6 Discussion

At present, the-state-of-art trackers mainly borrow ideas from popular object detectors. In fact, the target we want to track is usually a salient object in a video. Thus, we can also be inspired by some salient object methods [13, 39]. For example, as we have mentioned above, SiamFC does not care about the edge information of target which makes it unavailable for accurate tracking. Therefore some works [40] that focus on fine-grain detection can be helpful. On the other hand, improving visual attention mechanism [14, 32] to further learn a robust representation should be paid attention to.

Acknowledgements This work is supported by the National Natural Science Foundation of China (NSFC) under Grant No. 61671423 and Grant No. 61271403.

Compliance with Ethical Standards

Conflict of interests The authors declare that they have no conflict of interest.

References

1. Bertinetto L, Valmadre J, Golodetz S, Miksik O, Torr PH (2016) Staple: Complementary learners for real-time tracking. In: CVPR, pp 1401–1409
2. Bertinetto L, Valmadre J, Henriques J, Vedaldi A, Torr PH (2016) Fully-convolutional siamese networks for object tracking. In: ECCV, pp 850–865
3. Bolme DS, Beveridge JR, Draper BA, Lui YM (2010) Visual object tracking using adaptive correlation filters. In: CVPR, pp 2544–2550
4. Chen K, Tao W (2018) Convolutional regression for visual tracking. IEEE Trans Image Process 27(7):3611–3620
5. Danelljan M, Shahbaz Khan F, Felsberg M, Van de Weijer J (2014) Adaptive color attributes for real-time visual tracking. In: CVPR, pp 1090–1097
6. Danelljan M, Hager G, Shahbaz Khan F, Felsberg M (2015) Convolutional features for correlation filter based visual tracking. In: ICCV Workshops, pp 58–66
7. Danelljan M, Hager G, Shahbaz Khan F et al (2015) Convolutional features for correlation filter based visual tracking. In: CVPR Workshops, pp 58–66
8. Danelljan M, Hager G, Shahbaz Khan F et al (2015) Learning spatially regularized correlation filters for visual tracking. In: ICCV, pp 4310–4318
9. Danelljan M, Robinson A, Khan FS, Felsberg M (2016) Beyond correlation filters: Learning continuous convolution operators for visual tracking: In ECCV, pp 472–488
10. Danelljan M, Bhat G, Shahbaz Khan F, Felsberg M (2017) ECO: efficient convolution operators for tracking. In: In CVPR, pp 6638–6646
11. Dong X, Shen J, Wang W, Shao L, Ling H, Porikli F (2019) Dynamical Hyperparameter Optimization via Deep Reinforcement Learning in Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence
12. Fan H, Ling H (2017) Sanet: Structure-aware network for visual tracking. In: CVPR Workshops, pp 42–49
13. Fan DP, Cheng MM, Liu JJ, Gao SH, Hou Q, Borji A (2018) Salient objects in clutter: Bringing salient object detection to the foreground. In: ECCV, pp 186–202
14. Fan DP, Wang W, Cheng MM, Shen J (2019) Shifting more attention to video salient object detection. In: CVPR, pp 8554–8564
15. Guo Q, Feng W, Zhou C, Huang R, Wan L, Wang S (2017) Learning dynamic siamese network for visual object tracking. In: ICCV, pp 1763–1771
16. Gundogdu E, Alatan AA (2018) Good features to correlate for visual tracking. IEEE Trans Image Process 27(5):2526–2540
17. Hare S, Golodetz S, Saffari A, Vineet V, Cheng MM, Hicks SL, Torr PH (2015) Struck: Structured output tracking with kernels. IEEE Trans Pattern Anal Mach Intell 38(10):2096–2109
18. He A, Luo C, Tian X, Zeng W (2018) A twofold siamese network for real-time object tracking. In: CVPR, pp 4834–4843
19. Henriques J, Caseiro R, Martins P, Batista J (2012) Exploiting the circulant structure of tracking-by-detection with kernels. In: ECCV, pp 702–715
20. Henriques J, Caseiro R, Martins P, Batista J (2014) High-speed tracking with kernelized correlation filters. IEEE Trans Pattern Anal Mach Intell 37(3):583–596
21. Held D, Thrun S, Savarese S (2016) Learning to track at 100 fps with deep regression networks. In: ECCV, pp 749–765
22. Hu Y, Alahari K, Schmid C (2015) Online object tracking with proposal selection. In: ICCV, pp 3092–3100
23. Kristan M, Matas J, Leonardis A, Vojíř T, Pflugfelder R, Fernandez G, Čehovin L (2016) A novel performance evaluation methodology for single-target trackers. IEEE Trans Pattern Anal Mach Intell 38(11):2137–2155
24. Li Y, Zhu J (2014) A scale adaptive kernel correlation filter tracker with feature integration. In: ECCV, pp 254–265
25. Li B, Yan J, Wu W, Zhu Z, Hu X (2018) High performance visual tracking with siamese region proposal network. In: CVPR, pp 8971–8980
26. Ly Y, Zhang X (2019) SiamVGG: Visual Tracking using Deeper Siamese Networks. arXiv:1902.02804
27. Liang Z, Shen J (2019) Local Semantic Siamese Networks for Fast Tracking. IEEE Transactions on Image Processing
28. Lukezic A, Vojir T, Čehovin Zaje L, Matas J, Kristan M (2017) Discriminative correlation filter with channel and spatial reliability. In: CVPR, pp 6309–6318
29. Nam H, Baek M, Han B (2016) Modeling and propagating cnns in a tree structure for visual tracking. arXiv:1608.07242
30. Nam H, Han B (2016) Learning multi-domain convolutional neural networks for visual tracking. In: CVPR, pp 4293–4302
31. Shen J, Liang Z, Liu J, Sun H, Shao L, Tao D (2018) Multiobject tracking by submodular optimization. IEEE Trans Cybern 49(6):1990–2001
32. Shen J, Tang X, Dong X, Shao L (2019) Visual object tracking by hierarchical attention siamese network. IEEE transactions on cybernetics
33. Valmadre J, Bertinetto L, Henriques J, Vedaldi A, Torr PH (2017) End-to-end representation learning for correlation filter based tracking. In: CVPR, pp 2805–2813
34. Wang N, Yeung D-Y (2015) Ensemble-based tracking: Ensemble-based Aggregating crowdsourced structured time series data. In: ICML, pp 1107–1115
35. Wang M, Liu Y, Huang Z (2017) Large margin object tracking with circulant feature maps. In: CVPR, pp 4021–4029
36. Wang N, Zhou W, Tian Q, Hong R, Wang M, Li H (2018) Multi-Cue Correlation filters for robust visual tracking. In: CVPR, pp 4844–4853
37. Wu Y, Lim J, Yang MH (2013) Online object tracking: A benchmark. In: CVPR, pp 2411–2418
38. Zhang T, Xu C, Yang MH (2017) Multi-task correlation particle filter for robust object tracking. In: CVPR, pp 4335–4343
39. Zhao JX, Cao Y, Fan DP, Cheng MM, Li XY, Zhang L (2019) Contrast prior and fluid pyramid integration for RGBD salient object detection. In: CVPR, pp 3927–3936
40. Zhao JX, Liu JJ, Fan DP, Cao Y, Yang J, Cheng MM (2019) EGNet: Edge guidance network for salient object detection. In: ICCV, pp 8779–8788
41. Zhu G, Porikli F, Li H (2016) Beyond local search: Tracking objects everywhere with instance-specific proposals. In: CVPR, pp 943–951

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.