TCCF: Tracking Based on Convolutional Neural Network and Correlation Filters

Qiankun Liu, Bin Liu, and Nenghai Yu

Key Laboratory of Electromagnetic Space Information,
Chinese Academy of Sciences,
School of Information Science and Technology,
University of Science and Technology of China

Abstract. With the rapid development of deep learning in recent years, lots of trackers based on deep learning were proposed, and achieved great improvements compared with traditional methods. However, due to the scarcity of training samples, fine-tuning pre-trained deep models can be easily over-fitted and its cost is expensive. In this paper, we propose a novel algorithm for online visual object tracking which is divided into two separate parts, one of them is target location estimation and the other is target scale estimation. Both of them are implemented with correlation filters independently while using different feature representations. Instead of fine-tuning pre-trained deep models, we update correlation filters. And we design the desired output of correlation filters for every training sample which makes our tracker perform better. Extensive experiments are conducted on the OTB-15 benchmark, and the results demonstrate that our algorithm outperforms the state-of-the-art by great margin in terms of accuracy and robustness.

Keywords: visual object tracking, correlation filter, CNN

1 Introduction

As one of the popular branches of computer vision, visual object tracking has been widely used in various fields, such as military strike, traffic control, security system, human-computer interaction and so on, and it has been rapidly developed thanks to the development of deep learning in recent years. Single-target tracking can be described as follow: an arbitrary target is given by a bounding box in the first frame and the trackers give predicted bounding box in subsequent frames. Although great progress has been made in past decades, visual object tracking is still a challenging task to handle with owning to complicated and volatile interferences like illumination variation, scale variation, partial or full occlusion, motion blur, background clutters and deformation.

The features that can effectively distinguish target from its surrounding background play significant role in visual tracking. Trackers [12, 17, 10, 16, 32, 31] based on hand-crafted features have solved mentioned challenges more or less to some extent and can run at a high speed. But hand-crafted features are

usually specially designed for certain scenarios, so they are less accurate or robust when faced with more complex scenarios, which can lead to tracking failure. Recently, lots of trackers [24, 22, 25, 28, 29, 4] based on deep learning methods have been proposed and made a great progress in terms of accuracy and robustness. These trackers all use features that extracted by Convolutional Neural Networks (CNNs) as the representation of tracking target.

Apart from advantages of CNNs, there exist some thorny issues to handle with. For example, a numerous amount of annotated samples are required for supervised training of deep CNNs which have millions of parameters. And the lack of training samples becomes more severe for online visual object tracking since there is only one annotated sample provided in the first frame of each video. What's more, the computation of features extracted by CNNs is more complex than hand-crafted features. Correlation filters, which transform convolution into multiplication to accelerate processing speed, have been widely used in tracking [10, 5, 25], but the desired output of correlation filters in these trackers are either the same for all training samples or designed improperly which reduces the correlation between filters and tracking targets.

In this paper, we divide tracking into two parts, one of them is target location estimation, and the other is target scale estimation. Both of them are implemented by correlation filters. On the training stage, we design the desired output of correlation filters carefully to get more superior filters. The contributions of this paper are summarized as follows: i) we comprehensively analyze the diversity of the features from different layers in a CNN as well as the difference between hand-crafted features and the features extracted by CNNs. And we design two correlation filters to utilize these features effectively; ii) we proposed a novel tracker based on these correlation filters for single-target tracking which just need to update correlation filters dynamically instead of fine-tuning pretrained deep models. We conducted extensive experiments on OTB-15 benchmark [30] dataset and the results demonstrate that our algorithm outperforms several state-of-the-art trackers.

The rest of this paper is organized as follows. Section 2 reviews related work about our algorithm. Section 3 introduces our algorithm thoroughly and section 4 represents the experimental results of our evaluation on different trakers. Finally, section 5 makes a conclusion on our work in this paper.

2 Related Work

The usage of features extracted by CNNs has shown great effectiveness for computer vision tasks in recent years, such as segmentation [8], classification [20], and gained considerable improvements. However, the computation complexity of extracting these features is much higher than hand-crafted features, and some scholars have done lots of research to improve computational efficiency. Correlation filters, which have played a significant role in signal processing since the 80's [23, 15] and solved myriad objective functions in the Fourier domain, are widely

used in visual tracking to speed up trackers owning to their high computational efficiency.

In [6], David S. Bolme *et al.* proposed a new type of correlation filter called Average of Synthetic Exact Filters (ASEF), and performed well in some specific tasks [6, 7]. However, a large number of samples are required for the training of ASEF. In the next year, David S. Bolme *et al.* modified ASEF and proposed Minimum Output Sum of Squared Error (MOSSE) filter for tracking [5], which achieved remarkable performance at a high processing speed. Both ASEF and MOSSE filters are single-channel correlation filters. Henriques J F *et al.* [14] proposed an analytic model which is named KCF for datasets consisted of thousands of translated patches using the concept of circulant matrices. For linear regression, this model is equivalent to a correlation filter, but it is also suitable for non-linear regression. What's more, KCF can be extend to multi-channel correlation filter. The work in [19] also did some research on multi-channel correlation filters which make it possible for correlation filters to be more widely used.

Martin Danelljan *et al.* proposed a concise tracker called DSST [10] based on correlation filters which inspired us to do our research. The highlight of DSST is its approach for scale estimation. But according to our observation of DSST, we found that the desired outputs of correlation filters are designed improperly, which will be explained in detail in section 3. The tracker HDT [25] exploits features from different layers in a CNN by a correlation filter for localizing tracking target. But HDT is limited to only location estimation which leads to poor performance in video sequences with severe scale variations. What's more, the desired output of correlation filter is fixed since the first frame, which worsen its performance.

3 Tracking based on CNN and Correlation Filters

Here, we will describe our algorithm TCCF (Tracking based on CNN and Correlation Filters) thoroughly. Before that, we first introduce the features used for target location estimation and scale estimation.

3.1 Feature Selection

Hand-crafted features, take HOG [9] features for example, do well in representing the texture and edge of tracking target. As shown in Fig.1, different targets are all described clearly by HOG features[1]. But the drawback of hand-crafted features is that they can not distinguish tracking target between objects that are in the same category effectively (refer to the feature map locating at the intersection of third row and second column).

Recently, some deep CNN models[26, 21, 27] trained on ImageNet [11] have been widely used in many computer vision tasks and achieved great success. The

[1] The HOG feature map is visualized with the aid of Pitor's Computer Vision Matlab Toolbox: `https://pdollar.github.io/toolbox/`

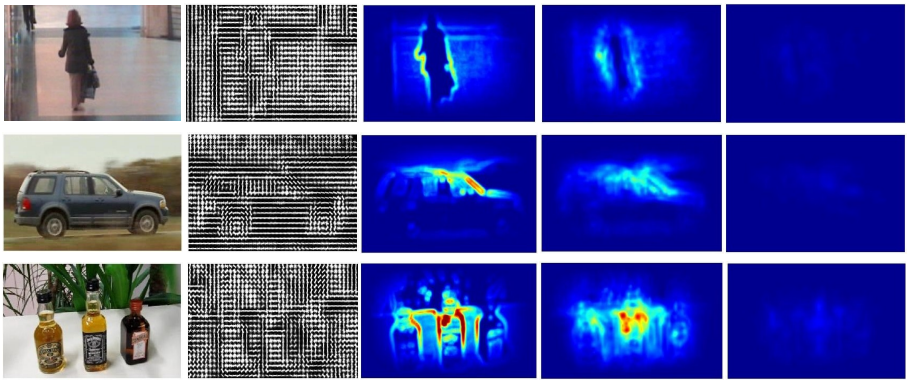

Fig. 1. Feature maps for different tracking targets. From left to right: the first column are input images, the second one are visualized HOG feature maps, the rest are feature maps extracted by VGG-16 from conv2-2, conv3-3 and conv4-3 layers respectively, and the feature map from a layer presented here is the average of all channel feature maps.

features extracted by CNNs are more discriminative than hand-crafted (refer to Fig.1). What's more, features extracted by a CNN vary from layers to layers. As shown in Fig.1, shallower layers capture generic information of the target, while deeper layers capture semantic information of the target. Wang L *et.al* also did some research on these differences between different layers [28].

Here, our tracking algorithm is divided into two parts, one of them is target location estimation, and the other is target scale estimation. The two parts are implemented independently. Since features extracted by CNNs can separate target from background more effectively than hand-crafted features, and there are some diversities between features from different layers, so they will be used by a correlation filter to implement location estimation. Once the location of tracking target is determined, hand-crafted (to be exact, HOG) features are used by another correlation filter to complete scale estimation since they do better in representing the texture and edge of target than features extracted by CNNs.

3.2 Correlation Filters

The structure of our proposed method is shown in Fig.2, tracking online is divided into two parts here. Location Correlation Filter (LCF) is used for location estimation, while Scale Correlation Filter (SCF) is used for scale estimation. Both LCF and SCF are multi-channel correlation filters. Here, we make an introduction to the multi-channel correlation filter used in our algorithm.

Let x^t, which is a multi-channel signal, denote the features extracted from the given training sample, y^t denote the desired output of correlation filter and f^t denote the correlation filter we want to get. The upper case variants $X^t = \mathcal{F}(x^t)$, $Y^t = \mathcal{F}(y^t)$ and $F^t = \mathcal{F}(f^t)$, where $\mathcal{F}(\cdot)$ denote the Discrete Fourier

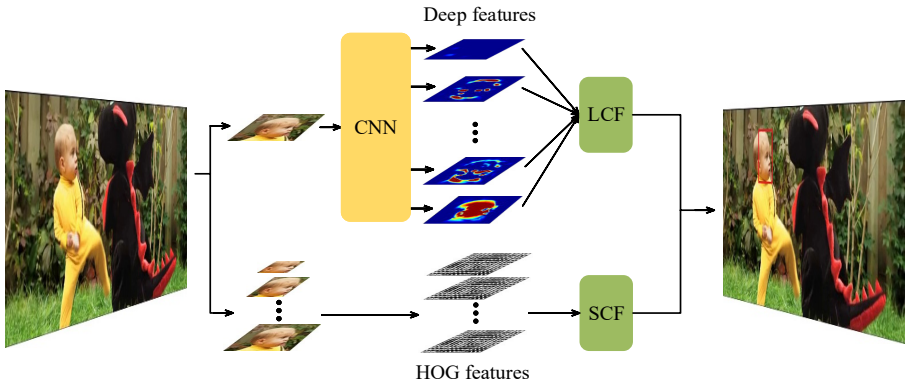

Fig. 2. The structure of TCCF.

Translation (DFT). y^t is artificially pre-defined according to the specific problem we are handling with. The correlation f^t is an ensemble of C weak filters, where C is the number of channels. In the Fourier domain, F^t can be computed by minimizing:

$$F^t = arg\min_{F^t} ||Y^t - \sum_{c=1}^{c=C} F_c^t \odot X_c^t||^2 + \lambda \sum_{c=1}^{c=C} ||F_c^t||^2 \tag{1}$$

where the subscript index c denote the component in c_{th} channel. The parameter λ in the second term on the right is the regularizer and the symbol $\odot$ denote element-wise product. The solution to equation (1) is:

$$F_c^t = \frac{Y^t \odot \bar{X}_c^t}{\sum_{c=1}^{c=C} X_c^t \odot \bar{X}_c^t + \lambda} \tag{2}$$

where the division is performed element-wise and $\bar{X}_c^t$ denote the complex conjugation of X_c^t. Obviously, the first term in the denominator is the power spectrum of x^t. From equation (2) we can find that once the training sample x^t and the regularizer λ are determined, the filter is directly controlled by y^t.

Given a testing sample t, we first transform it to the Fourier domain to obtain T, then the response of t can be computed by:

$$r = \mathcal{F}^{-1}(\sum_{c=1}^{c=C} T_c \odot F_c^t) \tag{3}$$

where $\mathcal{F}^{-1}(\cdot)$ is the inverse of DFT (IDFT).

In order to simplify our proposed model and reduce the cost of computation, we adopt an incremental update method as other researchers do in [25, 5, 10], which only use current frame to partially update previous correlation filters when tracking online. Given the t_{th} frame in a video sequence, let p^t and s^t denote

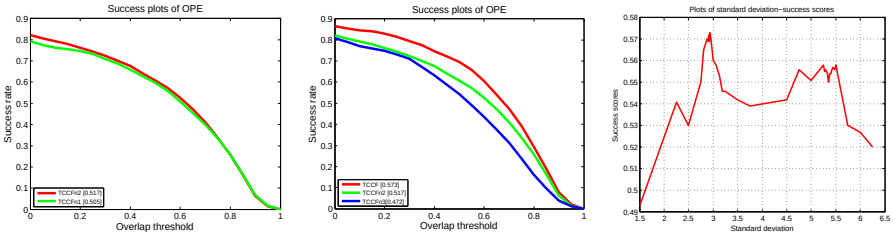

Fig. 3. Left: average success plots of two trackers. Middle: average success plots of three trackers. Right: the curve between average success scores and standard deviation of y_s^t.

the position and size of target in this frame, which are predicted by the tracker. F^t is updated as follows:

$$F_c^t = \frac{A^t}{B^t} = \frac{(1-\eta)A^{t-1} + \eta\hat{A}^t}{(1-\eta)B^{t-1} + \eta\hat{B}^t} \tag{4}$$

where

$$\hat{F}_c^t = \frac{\hat{A}^t}{\hat{B}^t} = \frac{Y^t \odot \bar{X}_c^t}{\sum_{c=1}^{c=C} X_c^t \odot \bar{X}_c^t + \lambda} \tag{5}$$

and the parameter η is the learning rate of correlation filters.

Location Correlation Filter: Since features extracted by a pre-trained CNN are used in LCF, so x^t and f^t are three dimensional, which means $x^t, f^t \in \Re^{M\times N\times C}$. Let $y_l^t \in \Re^{M\times N}$ denote the desired output of LCF and it is a 2-D Gaussian shape distribution which is determined by the mean μ_l^t and standard deviation δ_l^t. Suppose features from K convolution layers are used in our algorithm, there will be K independent correlation filters in LCF, which means:

$$\text{LCF} = \{F^{k,t} | k = 1, 2, ..., K\} \tag{6}$$

each $F^{k,t}$ has a weight w^k, and $\sum_{k=1}^{k=K} w^k = 1$. The location of target predicted by $F^{k,t}$ is the coordinate (m^k, n^k) of the maximum value in r^k. The ultimate location of target is computed as follows:

$$(m, n) = \sum_{k=1}^{k=K} w^k \cdot (m^k, n^k) \tag{7}$$

the symbol $\cdot$ denote the product of two scalars. Once the ultimate location of target is predicted, there will be a loss between (m^k, n^k) and (m, n), which implies the stability of $F^{k,t}$. And the weight w^k is updated according to the stability of $F^{k,t}$. Please refer to [25] for more information.

It should be noted that the mean of y_l^t is set to 0 and the standard deviation δ_l^t is proportional to the target size s^t, i.e.:

$$\delta_l^t \propto s^t \tag{8}$$

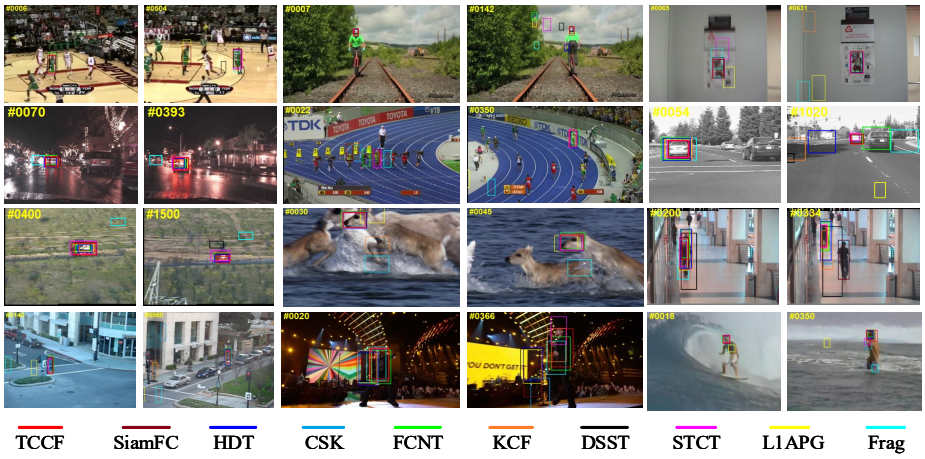

Fig. 4. Qualitative results of the proposed TCCF tracker and other 9 trackers on a subset of OTB-15 benchmark. From left to right and top to bottom: *Basketball*, *Biker*, *BlurOwl*, *CarDark*, *Bolt*, *Car*1, *RedTeam*, *Deer*, *Walking*2, *Human*4, *Singer*2, *Surfer*. Two frames of each video are presented here.

which means the desired output y_l^t of location correlation filter is controlled by δ_l^t and it is dynamically updated to adjust to the scale variation of target. While in HDT[25] and DSST[10], the desired outputs of correlation filters are fixed since the first frame in a video sequence, which has a negative impact on the performance of trackers. Suppose we choose a reference system ϕ in the image from the perspective of tracking target and the target make a translation distance D in ϕ. Now we jump out of the image and choose a reference system ϕ' in the screen from the respective of observer and the target make a translation distance D' in ϕ'. Since the location estimation is completed in ϕ' and it's a common sense that the larger s^t is, the larger D' will be when D is a constant and vice versa, which means the location estimation is relevant to the size of target.

Scale Correlation Filter: In order to implement scale estimation, we predefine a set of scale factors $\{\alpha_l = \theta^{\lceil \frac{L}{2} \rceil - l} | l = 1, 2, ..., L\}$, where θ>1 is the step for scale transformation. Given a training sample, we first extract L rectangles of interest with the size $\alpha_l \cdot s^t$, where s^t denote the size of target in this training sample. Then we get a feature map $M^t \in \Re^{C \times L}$ from these rectangles of interest with each collum in M^t corresponding to one rectangle. Let $x_c^t \in \Re^{1 \times L}$ denote the c_{th} row vector in M^t, and y_s^t denote the desired output of SCF, then SCF can be obtained by equation (2). y_s^t is 1-D Gaussian shaped distribution with its mean $\mu_s^t = 0$. And the target size s' in testing sample is determined by:

$$s' = \alpha_i \cdot s^t \tag{9}$$

where α_i is the scale factor and i is the index of the maximum value in the response r.

Table 1. Average precision scores on different attributes: Illumination Variation (IV), Occlusion (OCC), Deformation (DEF), Out-of-Plane Rotation (OPR), Background Clutters (BC), Scale Variation (SV), Motion Blur (MB), Fast Motion (FM), Out-of-View (OV), Low Resolution (LR), In-Plane Rotation (IPR).

Attributes \ Trackers	CSK	Frag	L1APG	Staple	DSST	KCF	FCNT	HDT	SiamFC	STCT	TCCF
IV	0.405	0.256	0.295	0.251	0.545	0.667	0.712	0.803	0.686	0.737	0.815
OCC	0.368	0.336	0.392	0.323	0.546	0.609	0.693	0.743	0.629	0.732	0.738
DEF	0.341	0.289	0.338	0.286	0.487	0.582	0.688	0.760	0.560	0.734	0.731
OPR	0.363	0.342	0.340	0.314	0.505	0.598	0.740	0.745	0.678	0.717	0.759
BC	0.460	0.317	0.366	0.308	0.565	0.623	0.689	0.766	0.635	0.762	0.797
SV	0.348	0.306	0.347	0.298	0.567	0.568	0.709	0.787	0.717	0.761	0.755
MB	0.325	0.302	0.342	0.247	0.670	0.573	0.698	0.780	0.703	0.768	0.757
FM	0.302	0.321	0.317	0.279	0.630	0.529	0.635	0.763	0.697	0.719	0.745
OV	0.252	0.330	0.350	0.207	0.475	0.441	0.635	0.651	0.688	0.594	0.611
LR	0.380	0.306	0.409	0.360	0.509	0.558	0.716	0.756	0.859	0.741	0.805
IPR	0.389	0.320	0.405	0.329	0.566	0.587	0.763	0.803	0.697	0.705	0.753
Overall	0.406	0.342	0.392	0.328	0.579	0.611	0.742	0.804	0.693	0.770	0.796

Inspired by the effectiveness of dynamical update of y_l^t, we keep y_s^t dynamically updated like equation (8), but experimental results demonstrate that the dynamical update of y_s^t reduces the performance of tracker which is opposite of what we have expected.

Here we give an explanation. Unlike location estimation which is implemented in ϕ', the scale estimation is just to find an optimal scale factor α_i which is independent with ϕ and ϕ'. Since the scale variation between two consecutive frames is small, which means the probability of severe scale variation between two consecutive frames is much lower and vice versa, so y_s^t is independent with the size of target but relative to the number of scale factors L:

$$\delta_s^t \propto L \tag{10}$$

4 Experiments

The proposed algorithm is implemented in MATLAB with Caffe framework [18] and runs at 3.5 fps on a Ubuntu 14.04.3 machine with a 3.0GHz Intel i7-5960x CPU and a Nvidia GM2000 TITAN X GPU. The VGG-16 is used as the pre-trained CNN in our experiments, and the last 6 convolutional layers are used to extract features. We use $L = 33$ and $\theta = 1.02$ for scale estimation. And the learning rate η is set to 0.00902.

We use one-passe-evaluation (OPE) metric on the first 50 video sequences in OTB-15 benchmark [30] to evaluate different trackers. According to different challenging factors, such as illumination variation, occlusion, deformation and so on, there are 11 attributes tagged to these video sequences, which make it possible to evaluate these trackers thoroughly.

Inspired by DSST[10], we first construct two naive trackers TCCFn1 and TCCFn2 based on LCF to illustrate the effectiveness of the dynamical update

Table 2. Average success scores on different attributes: Illumination Variation (IV), Occlusion (OCC), Deformation (DEF), Out-of-Plane Rotation (OPR), Background Clutters (BC), Scale Variation (SV), Motion Blur (MB), Fast Motion (FM), Out-of-View (OV), Low Resolution (LR), In-Plane Rotation (IPR).

Attributes \ Trackers	CSK	Frag	L1APG	Staple	DSST	KCF	FCNT	HDT	SiamFC	STCT	TCCF
IV	0.318	0.201	0.235	0.172	0.399	0.433	0.486	0.500	0.484	0.570	0.587
OCC	0.269	0.252	0.277	0.190	0.377	0.396	0.473	0.486	0.458	0.514	0.510
DEF	0.267	0.235	0.248	0.178	0.357	0.400	0.482	0.487	0.407	0.544	0.506
OPR	0.271	0.261	0.248	0.197	0.355	0.396	0.503	0.486	0.487	0.513	0.532
BC	0.344	0.246	0.294	0.215	0.416	0.417	0.474	0.501	0.466	0.579	0.574
SV	0.266	0.238	0.253	0.192	0.402	0.352	0.474	0.479	0.528	0.545	0.541
MB	0.274	0.265	0.263	0.196	0.495	0.395	0.505	0.524	0.536	0.587	0.571
FM	0.257	0.268	0.252	0.202	0.477	0.370	0.469	0.510	0.536	0.553	0.557
OV	0.214	0.258	0.253	0.146	0.365	0.327	0.459	0.441	0.509	0.455	0.457
LR	0.242	0.201	0.267	0.217	0.308	0.297	0.415	0.415	0.616	0.491	0.568
IPR	0.296	0.265	0.305	0.216	0.397	0.389	0.519	0.523	0.519	0.507	0.552
Overall	0.310	0.265	0.292	0.200	0.421	0.403	0.508	0.514	0.509	0.567	0.573

of y_l^t. The y_l^t in TCCFn1 is fixed since the first frame, and the y_l^t in TCCFn2 is dynamically updated according to equation (8). As shown on the left in Fig.3, there are 1.2% improvements in TCCFn2 which demonstrates the effectiveness of dynamical update of y_l^t. We also construct another tracker TCCFn3 where the y_l^t and y_s^t both are dynamically updated. The success scores of TCCFn2 and TCCFn3 are shown in the middle in Fig.3, from where we can figure out that the dynamical update of y_s^t reduces the performance of tracker. To find the optimal y_s^t according to equation (10), we conduct extensive experiments using variable-controlling method and get a graphic which is shown on the right in Fig.3, from where we find the optimal stanard deviation of y_s^t and then we construct the optimal tracker TCCF as depicted in the middle in Fig.3.

We compare our proposed TCCF tracker with other ten trackers, CSK [13], Frag [1], L1APG [2], Staple [3], DSST [10], KCF [14], FCNT [28], HDT [25], SiamFC [4], STCT [29]. And We do qualitative and quantitative evaluation on these trackers. Among them, qualitative results are shown in Fig.4, from where we can figure out that our approach efficiently handles some challenging factors, such as deformation, motion blur, scale variation, background cluster and so on. Quantitative results are shown in Table 1 and Table 2. We compared these trackers for every attribute. In Table 1, all the values are obtained at the threshold of 20 pixels. In Table 2, all the values are computed using the metric AUC (Area Under Curve). The first, second and third best trackers are highlighted in red, green and blue, respectively. From Table 1 and Table 2, we can find that our TCCF performs well in different attributes, which demonstrates the effectiveness of our correlation filters.

We also use the precision and success plots to evaluate all trackers in Fig.5. The precision plots demonstrate the percentage of frames where the distance between the ground truth center of target and the predicted center of target is within a given threshold. The success plots demonstrate the percentage of frames where the overlap ratio between the ground truth bounding box and the

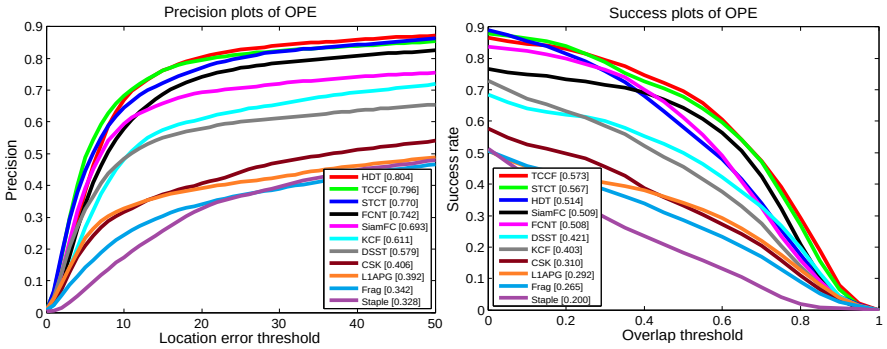

Fig. 5. Average precision plots and success plots of different trackers tested over 50 video sequences. On the left, trackers are ranked according to the precision score at the threshold of 20 pixels. On the right, trackers are ranked according to the area under curve.

predicted bounding box is higher than a given threshold. Comparing TCCF with DSST, we can figure out that there are 21.7% and 15.2% improvements in the precision and success scores. While compared with STCT, TCCF gets 2.6% and 0.6% improvements in the precision and success scores. When comparing TCCF with HDT, although HDT gets 0.8% improvements in the precision scores, TCCF gets 5.9% improvements in the success scores. The plots in Fig.5 demonstrate that our TCCF tracker achieves the best overall performance than other trackers.

5 Conclusion

In this paper, we proposed a novel algorithm for online visual object tracking based on CNN and correlation filters (TCCF). The pre-trained VGG-16 [26] is the only one CNN used in our algorithm and it is kept fixed when tracking online, so the algorithm just need to update correlation filters dynamically instead of fine-tuning pre-trained deep models, which means the structure of our algorithm is simple and compact. TCCF is consisted with two separate component entities: location estimation and scale estimation. Both of them are implemented by correlation filters independently while using different feature representations. The results of extensive experiments demonstrate that our algorithm outperform the state-of-the-art by a great margin in terms of accuracy and robustness.

Acknowledgement This work is supported by the National Natural Science Foundation of China (Grant No. 61371192), the Key Laboratory Foundation of the Chinese Academy of Sciences (CXJJ-17S044) and the Fundamental Research Funds for the Central Universities (WK2100330002).

References

1. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on. vol. 1, pp. 798–805. IEEE (2006)
2. Bao, C., Wu, Y., Ling, H., Ji, H.: Real time robust l1 tracker using accelerated proximal gradient approach. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. pp. 1830–1837. IEEE (2012)
3. Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., Torr, P.H.: Staple: Complementary learners for real-time tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1401–1409 (2016)
4. 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. pp. 850–865. Springer (2016)
5. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. pp. 2544–2550. IEEE (2010)
6. Bolme, D.S., Draper, B.A., Beveridge, J.R.: Average of synthetic exact filters. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 2105–2112. IEEE (2009)
7. Bolme, D.S., Lui, Y.M., Draper, B.A., Beveridge, J.R.: Simple real-time human detection using a single correlation filter. In: Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on. pp. 1–8. IEEE (2009)
8. Caelles, S., Maninis, K.K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: One-shot video object segmentation. arXiv preprint arXiv:1611.05198 (2016)
9. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. vol. 1, pp. 886–893. IEEE (2005)
10. Danelljan, M., H?ger, G., Khan, F.S., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: British Machine Vision Conference. pp. 65.1–65.11 (2014)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 248–255. IEEE (2009)
12. Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M.M., Hicks, S.L., Torr, P.H.: Struck: Structured output tracking with kernels. IEEE transactions on pattern analysis and machine intelligence 38(10), 2096–2109 (2016)
13. Henriques, J., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. Computer Vision–ECCV 2012 pp. 702–715 (2012)
14. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Transactions on Pattern Analysis and Machine Intelligence 37(3), 583–596 (2015)
15. Hester, C.F., Casasent, D.: Multivariant technique for multiclass pattern recognition. Applied Optics 19(11), 1758–1761 (1980)
16. Hong, Z., Chen, Z., Wang, C., Mei, X., Prokhorov, D., Tao, D.: Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In: Computer Vision and Pattern Recognition. pp. 749–758 (2015)

17. Jia, X.: Visual tracking via adaptive structural local sparse appearance model. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1822–1829 (2012)
18. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on Multimedia. pp. 675–678. ACM (2014)
19. Kiani Galoogahi, H., Sim, T., Lucey, S.: Multi-channel correlation filters. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3072–3079 (2013)
20. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
21. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
22. Li, H., Li, Y., Porikli, F.: Deeptrack: Learning discriminative feature representations online for robust visual tracking. IEEE Transactions on Image Processing 25(4), 1834–1848 (2016)
23. Mahalanobis, A., Kumar, B.V., Casasent, D.: Minimum average correlation energy filters. Applied Optics 26(17), 3633–3640 (1987)
24. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. Computer Science (2015)
25. Qi, Y., Zhang, S., Qin, L., Yao, H., Huang, Q., Lim, J., Yang, M.H.: Hedged deep tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4303–4311 (2016)
26. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
27. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–9 (2015)
28. Wang, L., Ouyang, W., Wang, X., Lu, H.: Visual tracking with fully convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3119–3127 (2015)
29. Wang, L., Ouyang, W., Wang, X., Lu, H.: Stct: Sequentially training convolutional networks for visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1373–1381 (2016)
30. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence 37(9), 1834–1848 (2015)
31. Zhang, K., Zhang, L., Yang, M.H., Zhang, D.: Fast tracking via spatio-temporal context learning. Computer Science (2013)
32. Zhang, L., Lu, H., Du, D., Liu, L.: Sparse hashing tracking. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society 25(2), 840–849 (2016)