



Robust and Real-Time Visual Tracking Based on Single-Layer Convolutional Features and Accurate Scale Estimation

Runling Wang¹(✉), Jiancheng Zou¹, Manqiang Che²,
and Changzhen Xiong²

¹ School of Sciences, North China University of Technology,
Beijing 100144, China
1573112241@qq.com

² Beijing Key Laboratory of Urban Intelligent Control Technology,
North China University of Technology, Beijing 100144, China
1229462669@qq.com

Abstract. Visual tracking is a fundamental problem in computer vision. Recently, some methods have been developed to utilize features learned from a deep convolutional neural network for visual tracking and achieve record-breaking performances. However, deep trackers suffer from efficiency. In this paper, we propose an object tracking method combining the single-layer convolutional features with correlation filter to locate and speed up. Meanwhile accurate scale prediction and high-confidence model update strategy are adopted to solve the scale variation and similarity interfere problems. Extensive experiments on large scale benchmarks demonstrate the effectiveness of the proposed algorithm against state-of-the-art trackers.

Keywords: Object tracking · Correlation filter · Convolutional features
Scale pyramid · Model update

1 Introduction

Visual tracking addresses the problem of identifying and localizing an unknown target in a video given the target specified by a bounding box in the first frame. It has attracted increasing interest in the past decades due to its importance in numerous applications, such as intelligent video surveillance, vehicle navigation, and human-computer interaction. Despite the significant effort that has been made to develop algorithms [1–4] and benchmark evaluations [5, 6] for visual tracking, it is still a challenging task owing to complicated interfering factors like heavy illumination changes, shape deformation, partial and full occlusion, large scale variations, to name a few.

Owing to the high complexity of deep learning, most deep trackers suffer from low tracking speed, and thus are impractical in many real-world applications. Some new deep trackers with smaller network structure achieve high efficiency while at the cost of significant decrease on precision. In Fig. 1, we display the relationship between tracking speed and accuracy of some deep state-of-the-art trackers [1–4, 7–12]. For better illustration, only those trackers with accuracy higher than 0.82 are reported.

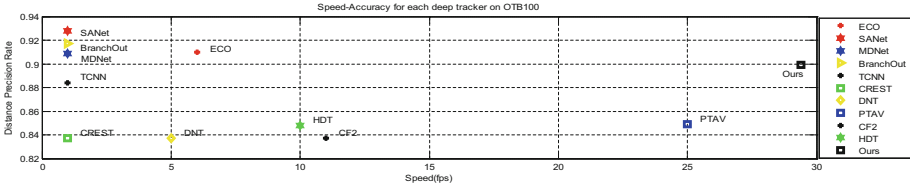


Fig. 1. Speed and accuracy plot of deep state-of-the-art visual tracking on OTB100

Obviously, SANet [2], MDNet [3] and BranchOut [4] utilizing robust deep features for appearance representation obtain highest accuracies than 0.9, but the speeds are around 1 fps; ECO [1] introduces factorized convolution operator to reduce the number of model’s parameters but only gets slight increase in speed; CF2 [11] combines the hierarchical features from VGG-19 [13] network with fast shallow tracker based on correlation filters, and achieves high accuracy but 11 fps in speed which is far from practical; PTAV [10] runs in real-time and the performances are barely satisfactory.

Though afore mentioned progresses in either accuracy or speed, real-time and robust trackers remain rare. In this paper, we consider the problems mentioned above and propose an algorithm based on single-layer convolutional features and accurate scale estimation to seek a trade-off between speed and accuracy. The main contributions of our work can be summarized below:

- We decrease the hierarchical layers and adopt a single-layer convolutional features to speed up.
- We change the Gaussian distribution of the samples to match the selected layer by tinkering with the Gaussian bandwidth of label function for training samples.
- We introduce an accurate scale estimation method to predict the scale variation of the object, expecting to further improve the performance.
- We utilize the high-confidence model update strategy, which is beneficial to precision improvement, to prevent our proposed model from drifting due to serious occlusion or interference of similar objects.

The framework of our tracker is shown in Fig. 2, which consists of translation prediction and scale estimation.

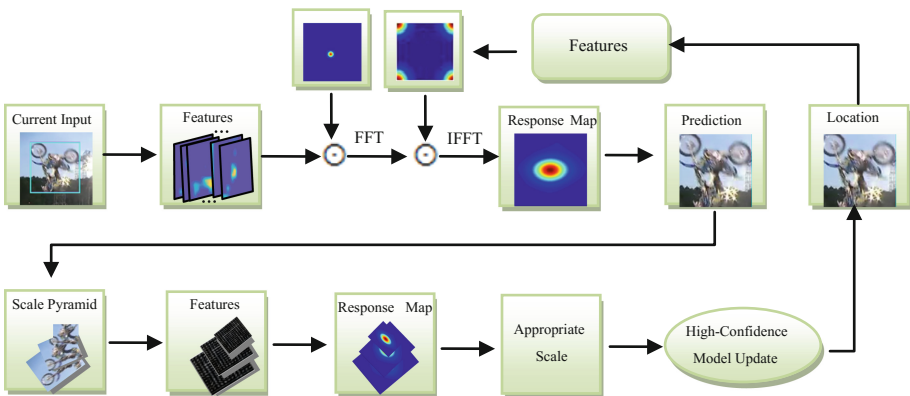


Fig. 2. Framework of proposed algorithm

2 Related Work

CNN Based Trackers. Visual representations play a very important role in object tracking. Numbers of hand-crafted features used to represent the target appearance such as Histogram of Oriented Gradient (HOG) and Color Names (CN) achieve great success. Since 2013, deep-learning methods spur in the field of visual tracking and exceed hand-crafted methods gradually. Wang *et al.* [14] propose a deep learning tracker (DLT) using a multi-layer auto-encoder network for the first time and solve the problem of insufficient training data through the idea of “offline pre-training and online fine tuning”. Hong *et al.* [15] learn target-specific saliency map using a pre-trained CNN. On the other hand, Wang *et al.* [16] use feature maps for target tracking from a two-layer neural network, whose earlier and last hierarchical features are complementary in semantic and spatial information. Held *et al.* [17] make full use of labeled videos and images to train a completely offline universal target tracker and achieve pleasant speed of 100 frames per second, while the precision is notoriously ineffective. Nam *et al.* [3] design the shallow “shared layers + domain-specific layers” framework for the acquisition of target representation and classification respectively, recommending with the introduction of hard negative mining and bounding box regression approaches. Therefore, they historically obtain the high accuracy, but regretfully only 1 frame per second in speed.

Correlation Filters Based Trackers. Correlation filters for visual tracking have attracted considerable attention due to the high computational efficiency with fast Fourier transforms (FFT). Bolme *et al.* [18] learn a Minimum Output Sum of Squared Error filter over luminance channel for fast visual tracking. Henriques *et al.* [19] propose CSK algorithm based on correlation filter by introducing kernel methods and employing ridge regression, but the simplicity of gray features for learning and training makes it lower accuracy. Subsequently they put forward the Kernelized Correlation Filters (KCF) [20], extending the input features from single channel to multiple channels namely HOG, but there is no ideal effect when faced with challenges of multi-scale and fast motion. Xiong *et al.* [21] propose a kernelized correlation filters tracking based on adaptive feature fusion, which combines global CN features and local HOG features, solving the problem of tracking failure caused by simple feature due to deformation and illumination. Danelljan *et al.* [22] figure out the fast scale estimation problem by learning separate filters for translation and scale estimation. Ma *et al.* [11] adaptively learn correlation filters on three convolutional layers to encode the target appearance and hierarchically infer the maximum response of each layer to locate targets. Wang *et al.* [23] propose to transfer the features of image classification to visual tracking domain via convolutional channel reductions, which significantly increases the tracking speed to real-time performances. Chi *et al.* [9] integrate the hierarchical feature maps in different layers with an edge detector, and update it with stochastic and periodic methods. Wang *et al.* [24] make full use of the strong discriminative ability of structured SVM and advantage of correlation filter in speed, combining with multimodal target detection and high-confidence update strategy to

improve the speed and accuracy effectively. Danelljan *et al.* [1] introduce a factorized convolution operator to reduce dimensions of features and propose a compact generative model to better the diversity of training samples, which effectively prevents the samples from being contaminated by backgrounds and wrong targets.

3 Correlation Filters

A correlation filter based algorithm learns a discriminative classifier and estimates the translation of the target by searching the maximum value of correlation response map in the search window. Here, we denote \mathbf{x} as the feature vector of size $M \times N \times D$, where M , N and D indicate the width, height and the number of channels, respectively. Algorithms based on correlation filters use cyclic offset to generate numbers of training samples $x_{m,n} = \{0, 1, \dots, M-1\} \times \{0, 1, \dots, N-1\}$, where m, n indicate shifted position of the samples in the directions of width and height. The core problem of correlation filters is to minimize the square error of the regression function $f(x) = w_t \cdot x$, that is to solve the following problem:

$$w^* = \arg \min \sum_{m,n} \|w \cdot x_{m,n} - y(m,n)\|^2 + \lambda \|w\|_2^2, \quad (1)$$

where w_t is classifier parameter of frame t , w^* is classifier parameter when the error is minimized, w is classifier parameter, \cdot is the inner product which is induced by a linear kernel in the Hilbert space, y is Gaussian labeled function of training samples and λ is a regularization parameter. According to [19], we obtain the closed-loop solution quickly in the Fourier domain by sampling the circulant matrix with shifting so can get the classifier parameters of data's filter on the d -th channel:

$$W^d = \frac{Y \odot \bar{X}_d}{\sum_{d=1}^D X_d \odot \bar{X}_d + \lambda}, \quad (2)$$

where Y is the Fourier transformation of the Gaussian labeled function y , the bar indicates complex conjugation and d is the dimension. The operator \odot means Hadamard product.

Given a new image patch, we note z_d as the convolutional feature. Therefore the response for its Fourier transformation Z_d and the classifier parameter W^d can be computed by

$$f = \mathbb{F}^{-1} \left(\sum_{d=1}^D W^d \odot \bar{Z}_d \right) \quad (3)$$

The operator \mathbb{F}^{-1} denotes the inverse FFT. And therefore the target location can be estimated by searching for the maximum value of the correlation response map f .

4 Robust and Real-Time Visual Tracking Based on Single-Layer Convolutional Features and Accurate Scale Estimation

4.1 Single-Layer Convolutional Features and Bandwidth Adjustment Strategy

According to [11], the last convolutional layer encode the semantic information and such representations are robust to significant appearance variations; in contrast, earlier layers provide precise localization but are less invariant to appearance changes. So it encodes the object appearance with features extracted from multiple layers (C3-4, C4-4 and C5-4). But redundant features and amounts of computation make the tracking speed rather poor, which is a big trouble for practical application. Therefore, we propose to decrease to a single layer to speed up.

Along with the VGG-19 forward propagation, the semantic discrimination between objects from different categories is strengthened, as well as a gradual reduction of spatial resolution for precise localization. While in visual tracking task, we need features extracted not only possess abundant semantic information to better adapt to appearance variations, but also retain spatial information so as to localize targets. Thus, compared with C3-4 which has better resolution while poor semantic information and C5-4 which is in verse, we take layers before or after C4-4 into account, namely C4-3, C4-4, C5-1, C5-2, C5-3 (more semantic information for appearance variations).

The VGG-19 network is trained by large-scale classification databases. But the difference between classification and tracking lies in the former regarding the similar objects as a category, while the other sorting out representations in all angles and directions of an object from other objects. Therefore, there exist serious interferences from backgrounds when applying the network to tracking. So we take the distribution of training samples into account to increase their diversity to better the discriminative ability for interferences.

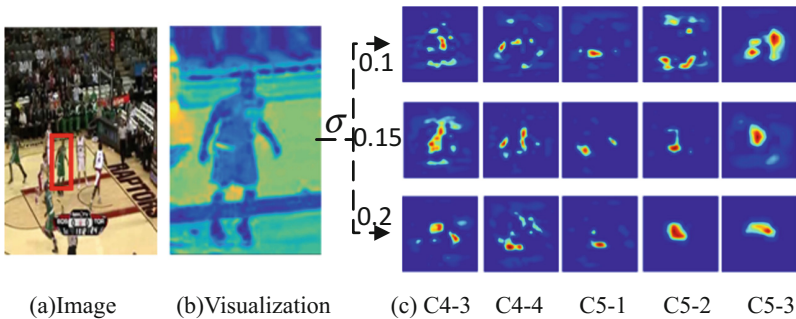


Fig. 3. Visualization of convolutional layers' features with different σ . (a) Image from *Basketball* sequence and ground truth foreground mask. (b) Visualization of the input image patch. (c) Feature map extracted from layers C4-3, C4-4, C5-1, C5-2, C5-3 with different bandwidths of the Gaussian labeled function of training samples.

For each shifted sample, there exists a Gaussian labeled function

$$y(m, n) = \exp\left(-\frac{(m - M/2)^2 + (n - N/2)^2}{2\sigma^2}\right),$$

where σ (generally set to 0.1) is the Gaussian kernel bandwidth, determining the pixels' classification. The larger bandwidth is, the more diversely the sample distributes, which makes the classification of pixels more prominent (target or background) and is of benefit to tracker. Figure 3 shows the relationship between the bandwidth σ and the layers in feature extraction of the input image. When increase the bandwidth σ of Gaussian labeled function to change the degree of concentration of the target and backgrounds, the diversity of training samples will be changed and match the required need of different layers. And thus we increase the value of σ with the interval 0.05 and find the layer C5-2 with bandwidth $\sigma = 0.2$ performs excellently. Therefore, we only extract features for tracking task from a single layer.

4.2 High-Confidence Model Update

Most existed trackers update at each frame without considering whether the detection is accurate or not. The ideal response map should have only one sharp peak and be smooth in all other areas when the detected target is extremely matched to the correct target as shown on the right in Fig. 4. However, the unimodal detection will regard the highest peak as the target leading to false detection especially faced with interference of similar object as shown in the middle. To guarantee the robustness, we exploit the high-confidence model update [24] to tackle the challenging problems of occlusion and interference of similar object. We define the average peak-to-correlation energy (APCE) measure, which indicates the fluctuated degree of response maps and the confidence level of the detected target, as

$$APCE = \frac{|f_{\max} - f_{\min}|^2}{\text{mean}(\sum_{w,h} (f_{w,h} - f_{\min})^2)}, \quad (4)$$

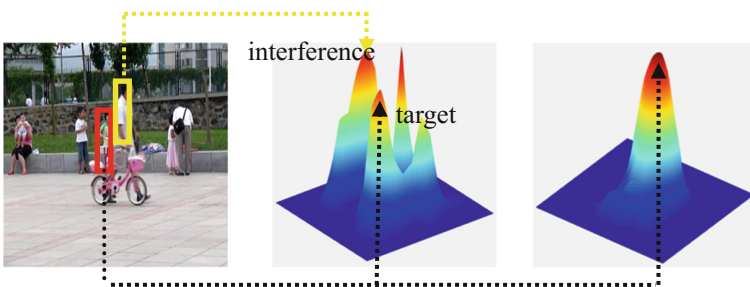


Fig. 4. Illustration for interference of similar object in sequence *Girl2*. The red bounding box indicates the correct location of target while the yellow is interference. Apparently, the response of the target is weaker. (Color figure online)

where f_{\max} , f_{\min} and $f_{w,h}$ denote the maximum, minimum response score of the response map and the w -th row h -th column elements of f .

When there are occlusion, interference and target missing, APCE will significantly decrease. While when f_{\max} and APCE are both greater than their respective historical average values with certain ratios ρ_1 , ρ_2 , the tracking result in the current frame is considered high confidence and then the proposed tracker will be updated online using a moving average:

$$\begin{aligned} A_t^d &= (1 - \eta)A_{t-1}^d + \eta Y \odot \bar{X}_t^d, \\ B_t^d &= (1 - \eta)B_{t-1}^d + \eta \sum_{i=1}^D X_t \odot \bar{X}_t^d, \\ W_t^d &= \frac{A_t^d}{B_t^d + \lambda}, \end{aligned} \quad (5)$$

where η is the learning rate and W_t^d is the correlation filter of t -th frame and d -th dimension of the features.

4.3 Accurate Scale Prediction

To better accommodate useful features of the target in different scales, the accurate scale estimation on a scale pyramid [22] is adopted. In visual tracking scenarios, the scale difference between two frames is typically smaller compared to the translation filter. Therefore, we first apply the translation filter W^d given a new frame. According to the scale pyramid which is constructed the size of the target at its estimated scale, each image patch is zoomed into the appropriate scale. Let $w \times h$ denote the target size in the current frame and S be the size of the scale filter. For each $i \in \{-\frac{S-1}{2}, \dots, \frac{S-1}{2}\}$, we extract an image patch J_i of size $s_d w \times s_d h$, where $s_d > 1$ denotes the scale factor between feature layers, centered around the target position predicted by the translation filter. Afterwards, the scale filter W_s is applied at the new target location. An example x computed by extracting features using variable patch size centered around the target is extracted from this location. By maximizing the correlation output (4) between W_s and x , we obtain the scale difference. That is

$$s = \arg \max(\max(f_1), \max(f_2), \dots, \max(f_i)), \quad (6)$$

where f_i is response map of scale filter. In addition, to obtain a robust approximation, (5) is used to update the scale filter with the new sample x .

5 Experiments

We implement our algorithm in Matlab R2015b underlying Ubuntu 16.04 system, and utilize the MatConvNet toolbox in this work. Our implementation runs at 29.4 frames per second on a computer with an Intel I5-4590K 4.00 GHz CPU, 8 GB RAM, and a GeForce GTX1070 GPU card. All the following experiments are carried out with the fixed parameters: the tradeoff parameter is set to $\lambda = 0.0001$; the learning rate is set to

$\eta = 0.01$; the Gaussian kernel bandwidth for translation filter is 0.2, while $S = 33$ number of scales with a scale factor of $s_d = 1.02$ with kernel bandwidth 0.1. We set value of ρ_1, ρ_2 in high-confidence model update 0.3 and 0.6 respectively.

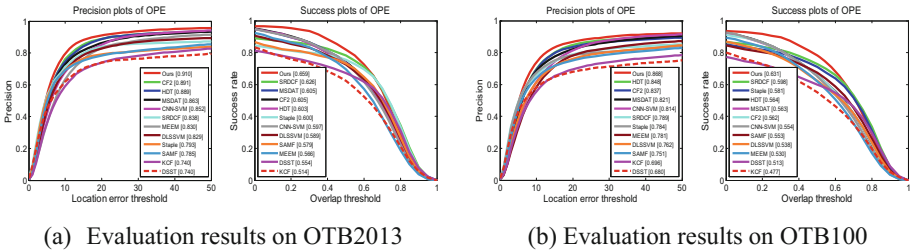


Fig. 5. Average precision plots and success plots over 50 and the entire 100 benchmark sequences

We compare our algorithm with 11 recent state-of-the-art trackers: MEEM [25], DLSSVM [26], KCF [20], SRDCF [27], SAMF [28], Staple [29], DSST [22], CF2 [11], MSDAT [23], CNN-SVM [15], HDT [12]. Among them, MEEM is developed based regression and multiple tracker, DLSSVM is structured SVM based method, KCF, SRDCF, SAMF, Staple, DSST are CF based methods, these above are designed with conventional hand-crafted features, while CF2, MSDAT, CNN-SVM, HDT are based on CNN features.

Comparison with State-of-the-Art Trackers

To fully assess our method, we use one-pass evaluation (OPE) metric on a large object tracking benchmark dataset OTB100 which contains 100 image sequences. For completeness, we also report the results on the benchmark OTB2013 [5], which is a subset of benchmark OTB100 [6].

To verify the contribution of each component in our algorithm, we implement and evaluate three additional variations of our tracking algorithm on OTB100—Ours with the Gaussian bandwidth $\sigma = 0.1$ ($\sigma = 0.1$); Ours without APCE model update strategy (noapce) and Ours without APCE model update strategy and scale estimation (noapcescale). The performance of all the variations are not as good as our full algorithm (Ours) and each component in our tracking algorithm is helpful to improve performance. The detailed results are illustrated in Fig. 6.

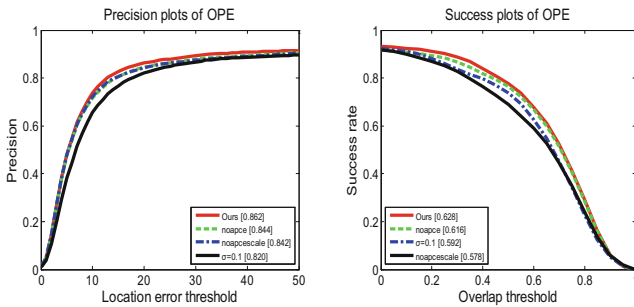


Fig. 6. Precision and success plots on OTB100 for the internal comparisons

Quantitative Evaluation. We evaluate the proposed algorithm with comparisons to 11 state-of-the-art trackers. Figure 5 illustrates the precision plots and success plots under OPE metric. Obviously, the proposed algorithm performs favorably against the state-of-the-art methods. Moreover, we present the quantitative comparisons of average distance precision rate (DPR), average overlap success rate (OSR) and average center location error (CLE) on two benchmarks [3, 4] in Table 1. The first, second and third best values are highlighted in color. Among the trackers, ours achieves the best results and obtains the lower CLE of 21.3 pixels over 100 video sequences compared to the baseline CF2 with 22.8 pixels.

Table 1. Comparisons of average DPR, OSR and CLE with state-of-the-art trackers on OTB2013 and OTB100.

		Proposed	Deep trackers				Correlation-filters based trackers					Others	
		[Ours]	CF2	MSDAT	CNN-SVM	HDT	KCF	SRDCF	SAMF	DSST	Staple	MEEM	DLSSVM
OTB2013	DPR (%)	91.0	89.1	86.3	85.2	88.9	74.0	83.8	78.5	74.0	79.3	83.0	82.9
	OSR (%)	81.9	74.0	74.1	73.4	73.7	62.3	78.1	73.2	67.0	75.4	69.6	72.4
	CLE (pixels)	11.8	15.7	14.6	17.9	15.9	35.5	35.2	30.1	41.2	30.6	20.9	24.1
OTB100	DPR (%)	86.8	83.7	82.1	81.4	84.8	69.6	78.9	75.1	68.0	78.4	78.1	76.2
	OSR (%)	78.1	65.5	65.5	65.1	65.7	55.1	72.8	67.4	60.1	70.9	62.2	62.4
	CLE (pixels)	21.3	22.8	20.5	21.8	20.1	45.0	38.6	36.5	50.4	31.5	27.8	32.9

Attribute-Based Evaluation. To thoroughly evaluate the robustness of the proposed algorithm in various scenes, we summarize the performances based on OTB100 dataset, where all videos are annotated with 11 different attributes, namely: illumination variation (IV), out-of-plane rotation (OPR), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-view (OV), background cluttered (BC) and low resolution (LR). For clarity, we report the results in Table 2. Our tracking algorithm achieves the best performances under 9 out of 11 attributes in terms of DPR and obtains 10 out of 11 when it comes to OSR but doesn't perform well in handling fast motion and low resolution, which can be explained that features from a single layer can't contain rich spatial details from earlier layers and semantics from last layer simultaneously. Overall, compared with other state-of-the-art tracking algorithms, ours can better locate the target object.

Qualitative Evaluation. We present some tracking results of the top performing tracking methods in Fig. 7: DSST [21], Staple [29], CF2 [11], MSDAT [23], KCF [20] and the proposed algorithm on 12 challenging sequences. KCF learns a kernelized correlation filter over HOG features. It doesn't perform well in deformations (*Couple*, *Girl2*, *Skiing*, *Bolt2*), motion blur and fast motion (*BlurCar2*). DSST performs well in sequences with scale variations (*Shaking*, *Lemming*), but fails when there are in-plane rotation (*Diving*, *MotorRolling*) and background clusters (*DragonBaby*, *Freeman4*, *Couple* and *Bolt2*) occur. Staple combines a correlation filter (using HOG features) with a global color histogram and thus achieves excellent performance to challenging

Table 2. Average DPR and OSR of our tracker and other top five trackers on different attributes. The first and second highest values are highlighted by bold and underline.

Attributes	DPR (%) on eleven attributes						OSR (%) on eleven attributes					
	Ours	HDT	CF2	MSDAT	CNN-SVM	SRDCF	Ours	HDT	CF2	MSDAT	CNN-SVM	SRDCF
IV	86.7	82.0	81.7	<u>82.5</u>	79.5	79.2	78.2	60.8	61.6	63.5	61.5	74.7
OPR	84.7	80.5	<u>80.7</u>	79.7	79.8	74.2	74.2	62.7	62.9	63.6	64.9	66.4
SV	82.8	<u>80.8</u>	79.9	77.1	78.7	74.5	71.1	51.4	51.9	50.8	52.9	<u>66.7</u>
OCC	79.4	<u>77.4</u>	76.7	74.0	73.0	73.5	70.2	61.1	60.6	59.7	60.6	<u>68.4</u>
DEF	83.1	<u>82.1</u>	79.1	79.2	79.3	73.4	71.7	61.8	60.3	60.4	63.4	<u>66.7</u>
MB	81.5	78.9	<u>80.4</u>	76.1	75.1	76.7	77.7	68.9	69.8	65.9	71.5	<u>72.9</u>
FM	79.9	81.7	<u>81.5</u>	74.4	74.7	76.9	73.9	66.4	66.8	63.4	64.9	<u>71.7</u>
IPR	87.8	84.4	<u>85.4</u>	85.4	81.3	74.5	76.0	65.7	66.2	<u>67.6</u>	65.7	66.2
OV	71.5	66.3	<u>67.7</u>	62.7	65.0	59.7	61.6	54.7	54.0	<u>56.0</u>	<u>59.1</u>	55.8
BC	86.6	<u>84.4</u>	84.3	83.3	77.6	77.5	77.8	71.3	72.1	<u>72.5</u>	68.1	70.1
LR	83.3	<u>88.7</u>	84.7	85.0	92.5	76.5	<u>52.9</u>	35.4	32.7	35.9	29.3	66.8
Overall	86.8	<u>84.8</u>	83.7	82.1	81.4	78.9	77.6	65.7	65.5	65.5	65.1	<u>72.8</u>



Fig. 7. Qualitative evaluation of the proposed algorithm and other five state-of-the-art trackers on twelve challenging sequences (from left to right and top to bottom are *Sylvester*, *DragonBaby*, *BlurCar2*, *MotorRolling*, *Bolt2*, *Shaking*, *Couple*, *Diving*, *Skiing*, *Freeman4*, *Girl2*, *Lemming*)

situations exhibiting motion blur (*BlurCar2*, *DragonBaby*) but notoriously sensitive to deformation (*Diving*, *Girl2*, *MotorRolling*, *Skiing*) as hand-crafted features are not effective in accounting for large appearance changes. CF2 is the baseline of MSDAT and Ours. Both of the two use deep features to represent object appearance so that they could fully exploit the semantic and fine-grained information as we do and can deal with these cases to some degree. Nevertheless, they still fail when heavy occlusion happens with other situations such as deformation and fast motion (*BlurCar2*, *Girl2*). Compared with these trackers, our approach accurately estimates the target scale and translation despite the mentioned factors.

6 Conclusions

In this paper, we propose an object tracking method combining the CNNs features with correlation filter. Hence the proposed algorithm absorbs the powerful representation ability from convolutional features and speeds up by correlation filter algorithm significantly. The accurate scale prediction and high-confidence model update strategy are adopted to improve the precision. It is worth to emphasize that our proposed algorithm not only performs superiorly, but also runs at a speed of 29.4 which is sufficient for real-time applications.

Acknowledgments. This work is supported in part by National Key R&D Program of China, 2017YFC0821102, in part by North China University of Technology Students' Technological Activity.

References

1. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: ECO: efficient convolution operators for tracking. In: Computer Vision and Pattern Recognition, pp. 6931–6939 (2017)
2. Fan, H., Ling, H.: SANet: structure-aware network for visual tracking. In: CVPR Deep Vision Workshop, pp. 2217–2224 (2016)
3. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: Computer Vision and Pattern Recognition, Las Vegas, pp. 4293–4302 (2016)
4. Han, B., Sim, J., Adam, H.: BranchOut: regularization for online ensemble tracking with convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, pp. 521–530 (2017)
5. Wu, Y., Lim J, Yang M.: Online object tracking: a benchmark. In: Computer Vision and Pattern Recognition, Portland, pp. 2411–2418 (2013)
6. Wu, Y., Lim J, Yang M.: Object tracking benchmark. In: Computer Vision and Pattern Recognition, pp. 1834–1848 (2015)
7. Nam, H., Baek, M., Han, B.: Modeling and propagating CNNs in a tree structure for visual tracking. <http://arxiv.org/abs/1608.07242>
8. Song, Y., Ma, C., Gong, L., Zhang, J., Lau, R.W., Yang, M.: CREST: convolutional residual learning for visual tracking. In: IEEE International Conference on Computer Vision, pp. 2574–2583 (2017)
9. Chi, Z., Li, H., Lu, H.: Dual deep network for visual tracking. IEEE Trans. Image Process. **26**, 2005–2015 (2017)

10. Fan, H., Ling, H.: Parallel tracking and verifying: a framework for real-time and high accuracy visual tracking. <http://arxiv.org/abs/1708.00153v1>
11. Ma, C., Huang, J., Yang, X.: Hierarchical convolutional features for visual tracking. In: *Computer Vision and Pattern Recognition*, Boston, pp. 3074–3082 (2015)
12. Qi, Y., Zhang, S., Qin, L., Yao, H., Huang, Q., Lim, J.: Hedged deep tracking. In: *Computer Vision and Pattern Recognition*, pp. 4303–4311 (2016)
13. Simonyan, K., Zisserman, A.: Very deep convolutional net works for large-scale image recognition. In: *International Conference on Learning Representations*, San Diego (2015)
14. Wang, N., Yeung, D. Y.: Learning a deep compact image representation for visual tracking. In: *International Conference on Neural Information Processing Systems*, pp. 809–817. Curran Associates Inc. (2013)
15. Hong, S., You, T., Kwak, S.: Online tracking by learning discriminative saliency map with convolutional neural network. In: *Computer Science*, pp. 597–606 (2015)
16. Wang, L., Ouyang, W., Wang, X.: Visual tracking with fully convolutional networks. In: *IEEE International Conference on Computer Vision*, Santiago, pp. 3119–3127 (2015)
17. Held, D., Thrun, S., Savarese, S.: Learning to track at 100 FPS with deep regression networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *ECCV 2016*. LNCS, vol. 9905, pp. 749–765. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_45
18. Bolme, D., Beveridge, J., Draper, B.: Visual object tracking using adaptive correlation filters. In: *Computer Vision and Pattern Recognition*, California, pp. 2544–2550 (2010)
19. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) *ECCV 2012*. LNCS, vol. 7575, pp. 702–715. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33765-9_50
20. Henriques, J.F., Rui, C., Martins, P.: High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**, 583–596 (2015)
21. Xiong, C., Zhao, L., Guo F.: Kernelized correlation filters tracking based on adaptive feature fusion. *J. Comput.-Aided Des. Comput. Graph.* 1068–1074 (2017). (in Chinese)
22. Danelljan, M., Häger, G., Khan, F.: Accurate scale estimation for robust visual tracking. In: *Proceedings of British Machine Vision Conference*, Nottingham, pp. 65.1–65.11 (2014)
23. Wang, X., Li, H., Li, Y.: Robust and real-time deep tracking via multi-scale domain adaptation. In: *IEEE International Conference on Multimedia and Expo*, Hong Kong, pp. 1338–1343 (2017)
24. Wang, M., Liu, Y., Huang, Z.: Large margin object tracking with circulant feature maps. In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, Hawaii, pp. 4800–4808 (2017)
25. Zhang, J., Ma, S., Sclaroff, S.: MEEM: robust tracking via multiple experts using entropy minimization. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) *ECCV 2014*. LNCS, vol. 8694, pp. 188–203. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10599-4_13
26. Ning, J., Yang, J., Jiang, S.: Object tracking via dual linear structured SVM and explicit feature map. In: *Computer Vision and Pattern Recognition*, Las Vegas, pp. 4266–4274 (2016)
27. Danelljan, M., Gustav, H., Fahad, S.: Learning spatially regularized correlation filters for visual tracking. In: *IEEE International Conference on Computer Vision*, Santiago, pp. 4310–4318 (2015)
28. Li, Y., Zhu, J.: A scale adaptive kernel correlation filter tracker with feature integration. In: Agapito, L., Bronstein, M.M., Rother, C. (eds.) *ECCV 2014*. LNCS, vol. 8926, pp. 254–265. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16181-5_18
29. Bertinetto, L., Valmadre, J., Golodetz, S.: Staple: complementary learners for real-time tracking. In: *Computer Vision and Pattern Recognition*, Las Vegas, pp. 1401–1409 (2016)