Online Visual Tracking via Correlation Filter with Convolutional Networks

Zheng Li†, Jianfei Yang†, Juan Zha†, Chang-Dong Wang and Weishi Zheng
School of Data and Computer Science, Sun Yat-sen University, China
hsqmlz@foxmail.com, jianfei_mars@hotmail.com, zhaj0804@foxmail.com, changdongwang@hotmail.com, zhwshi@mail.sysu.edu.cn
†These authors contributed equally to this work.

***Abstract*—Robust online visual tracking is a challenging task of the computer vision due to its violent variation within the video sequences. To approach these issues, deep networks have been applied in order to improve accuracy and correlation filter based trackers perform excellent efficiency and adaptation to scale. In this paper, we present a novel method with convolutional networks and correlation filter. A simple two-layer convolutional network is constructed to learn robust representations, which encode the inner geometric layout and local structural information, and the tracking framework resorts to learning discriminative correlation filters based on them. For our method satisfies both veracity and efficiency by finding a compromise between two theories, it performs favorably against several state-of-the-art methods with 50 public challenging videos.**

***Index Terms*—Visual tracking, convolutional network, correlation filter**

I. Introduction

Visual object tracking is a popular problem in computer vision, referring to estimating the trajectory of a target in video sequences based on its initial location. Despite dramatic advances in recent years, the problem still encounters some difficulties in face of huge circumstance variation considerations such as partial occlusion, fast motion, illumination variation, background clutter, deformation and scale variations. Apart from these, another constraint affecting the current trackers is limited prior knowledge about the target. Thus, more robust representative technique and tracking framework remain to be innovated.

Current tracking algorithms are sorted into either generative or discriminative methods. Generative methods [1]–[3] use a set of vectors from a subspace to stand for the target and search for the most similar region as the prediction position. In comparison, discriminative approaches make more use of the difference between the target and background, therefore regarding tracking as a classification problem. This category of tracking is termed tracking-by-detection and often employs machine learning techniques to train the classification model. For example, Babenko *et al.* [2] developed an online multiple instance learning consisting of a discriminative model by putting ambiguous positive and negative samples into bags. In [4], an online structured output support vector machine is proposed to derive promising results. Another advanced approach namely CSK tracker [5] learns a kernelized least-squares classifier for tracking. To address the strengths and weaknesses of these trackers, [6] showed a comprehensive evaluation of most approaches, whose datasets and assessment means are widely accepted.

Recently, deep networks have attracted much attention with admirable results in many challenging tasks in computer vision for they can learn features from raw data with fewer parameters. Nevertheless, less attention has been paid to incorporate deep networks to tracking problem. The reason proves to be lack of sufficient training data and huge computational complexity. To deal with these issues, Li *et al.* [7] used a convolutional neural network with multiple images as inputs, which is equivalent to extending the initial conditions. Another scenario turns to auxiliary data for offline training the deep networks, acquiring a pre-trained model for visual tracking [8]. These methods strive to gain a robust feature extractor offline with more initial conditions, which seems not practical in fact, but miss the idea of similar local structural and inner geometric layout information among the targets in successive frames. They may be the most reliable and easily-obtained raw data to discriminative the target from background, which will constitute our learning approach to construct the appearance model.

Among the sophisticated trackers in the benchmark, correlation filter based tracking-by-detection methods have achieved top performance. Bolme *et al.* [9] firstly proposed a tracker based on correlation filter, named Minimum Output Sum of Squared Error (MOSSE), drawing classical signal processing theory in. Their method trains the filter directly in the Fourier domain with an image patch as well as several circulant virtual ones with the purpose of increase efficiency. [10] then utilized the circulant structure generated by a base sample to propose a kernelized correlation filter (KCF). Correlation filter based methods are also tailored for the solution of scale variation [3]. Undoubtedly, correlation filter has shown its robustness on many tracking issues and we will extend the correlation filter based tracker by right of deep networks.

By synthesizing these ideas and analysis, we propose a tracking method based on discriminative correlation filters using expressive image representation which is learned by an online two-layer convolutional networks. We exploit the local structure and inner geometric layout information, denoted by local feature map, as a robust representation, and then employ a different pooling process to make the feature map global and reprensentitive. This feature map representation is applied to finding an optimal correlation filter, which are used for the estimation of position, scale and translation. In summary, our contribution are as follows:

978-1-5090-5316-2/16/$31.00

- We construct a correlation filter based tracker with online convolutional networks to find a reasonable compromise between the accuracy and efficiency.
- Our tracker enhances the translation estimation by boosting the deep feature map.
- The experiment on public dataset in benchmark [6] demonstrates that our method gets competitive results against most recent state-of-the-art trackers.

II. Learning Representation via Convolutional Networks

This section will introduce a hierarchical representation architecture for a given target template. The learning process is implemented by a efficient convolutional network including simple and complex layers. The simple layer extracts the local selective features from a bank of filters convolving the input image at each position while the complex one combines the selective features to obtain a global representation, which is robust to appearance variations by means of a approximately sparse vector.

A. Preprocessing

The intensity values are chosen to represent each input image which has been resized to a canonical size of $n \times n$ pixels, denoted as $\mathbf{I} \in \mathbb{R}^{n \times n}$. Then we utilize a sliding window of size $w \times w$ to densely sample a set of overlapping local image patches $\mathscr{P} = \{\mathbf{P_1}, ..., \mathbf{P_u}\}$ centered at each pixel location, where $\mathbf{P_i}$ is the $i-$th image patch and $u = (n - w + 1) \times (n - w + 1)$. To normalize the patches, each patch is preprocessed by deducting the mean of local brightness and l_2 normalization of contrast.

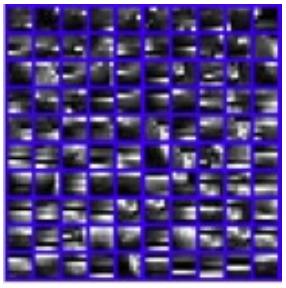


Fig. 1: Representative filters are selected by k-means algorithm.

B. Simple Layer

The simple layer aims to directly extract a representation distinguishing the target from the background from the preprocessed patch $\mathscr{P}$. A bank of patches $\mathscr{F}^o = \{\mathbf{F_1^o}, ..., \mathbf{F_v^o}\} \subset \mathscr{P}$ is selected through k-means algorithm in the first frame as fixed filters, which extracts selective features to form simple cell feature map as shown in Figure 1. Given the i-th filter $\mathbf{F_i^o} \in \mathbb{R}^{w \times w}$, we can obtain a feature map of the input image $\mathbf{I}$ by calculating its response referring to $\mathbf{M_i^o} = \mathbf{F_i^o} \otimes \mathbf{I}$, where feature map is $\mathbf{M_i^o} \in \mathbb{R}^{(\mathbf{n-w+1}) \times (\mathbf{n-w+1})}$ and $\otimes$ is the convolution operator. The fixed filter $\mathbf{F_i^o}$ is localized and selective in order to extract local structural features such as corners and edges, and show similar geometric layout, therefore preserving the global geometric layout information. Moreover, the fixed filters encode the fixed local templates which are stable and reliable visual information in the first frame so they address the drifting problem to some extent.

Then the representation of the background context is presented in a similar way. We choose m background samples surrounding the object and adopt the k-means algorithm to select a bank of filters $\mathscr{F}_i^b = \left\{\mathbf{F_{i,1}^b}, ..., \mathbf{F_{i,v}^b}\right\} \subset \mathscr{P}$ from the i-th background sample. The background context filter set can be summarized by average pooling method, denoted as $\mathscr{F}^b = \left\{\mathbf{F_1^b}, ..., \mathbf{F_v^b}\right\} \subset \mathscr{P}$ where $\mathbf{F_i^b} = \frac{1}{m}\sum_{j=1}^{m} \mathbf{F_{j,i}^b}$. Thus, the i-th background feature map of the input image $\mathbf{I}$ is defined as $\mathbf{M_i^b} = \mathbf{F_i^b} \otimes \mathbf{I}$. To discriminatively connect the target with the background, we define the simple cell feature map as

$$\mathbf{M_i} = \mathbf{M_i^o} - \mathbf{M_i^b} = \left(\mathbf{F_i^o} - \mathbf{F_i^b}\right) \otimes \mathbf{I}, \quad i - 1, ..., v \tag{1}$$

C. Complex Layer

To enhance the robustness of the representation $\mathbf{M_i}$, we manage to construct a complex layer with analogy to the pooling layers in the CNNs. By introducing a 3D tensor $\mathbf{C} \in \mathbb{R}^{(n-w+1) \times (n-w+1) \times v}$ as a complex cell feature, we converge v simple cell feature maps built with the filter set $\mathscr{F} = \mathscr{F}^o \cup \mathscr{F}^b$. The complex cell feature map performs scale-invariant to adapt to different scales and shift-variant to overcome drifting problem.

D. Model Update

As the external circumstance may change, the robust representation $\mathbf{C}$ is supposed to be updated incrementally to portray a delicate model. A temporal low-pass filtering method is defined as

$$\mathbf{C_t} = (1 - \rho)\mathbf{C_{t-1}} + \rho\hat{\mathbf{C}}_{\mathbf{t-1}} \tag{2}$$

where $\mathbf{C_t}$ is the target template at frame t, $\hat{\mathbf{C}}_{\mathbf{t-1}}$ denotes the complex representation of the target at frame $t-1$ and ρ is a learning parameter. The online update process provides good accommodation to appearance changes and alleviation of the drifting problem. Through four steps, we lick into shape a robust representation $\hat{\mathbf{C}}_{\mathbf{t}}$ of the target template $\mathbf{I}$ at specific frame t and update it with a learning process.

III. Discriminative Correlation Filters for Multi-dimensional Feature Map

Now that we have achieved a robust representation for target template, the extensive discriminative correlation filter for multi-dimensional features can help to make the accurate estimation. We construct 1-dimensional filters for the scale, 2-dimensional filters for translation and 3-dimensional filters for scale-space localization of the tracking object. The method is generic for any dense feature representation and when incorporated with our convolutional network representations, the scale and drifting problems are fairly handled.

Our 3-dimensional feature map representation is the material for training the tailor-made correlation filter. Considering a rectangular patch f extracted from the feature map, we denote feature dimension number $l \in \{1, ..., d\}$ of f by f^l where $d = 3$ in our method. We aims to find an optimal correlation

filter h composed of one filter h^l per feature dimension. The objective is achieved by minimizing the cost function:

$$\varepsilon = \left\| \sum_{l=1}^{d} h^l \star f^l - g \right\|^2 + \delta \sum_{l=1}^{d} \left\| h^l \right\|^2 \tag{3}$$

where g is the correlation output decided by filter and training example and $\delta \geq 0$ determines the weight of regularization term. The equation 3 can be transformed by Parseval's identity and the solution at frame $t = 1$ is to choose:

$$H^l = \frac{\overline{G} F^l}{\sum_{k=1}^{d} \overline{F^k} F^k + \delta} \tag{4}$$

In the equation 4, capital letters denote the discrete Fourier transforms (DFTs) of the corresponding functions and the bar such as $\overline{F^k}$ refers to complex conjugation. The product $\overline{G} F^l$ is point-wise. Then the numerator A_t and denominator B_t of H_t are updated separately as:

$$A_t^l = (1 - \eta) A_{t-1}^l + \eta \overline{G_t} F_t^l \tag{5}$$

$$B_t = (1 - \eta) B_{t-1} + \eta \sum_{k=1}^{d} \overline{F_t^k} F_t^k \tag{6}$$

where η is learning parameter. The new target position is determined by maximizing the correlation scores y at a rectangular region z.

$$y = \mathscr{F}^{-1} \left\{ \frac{\sum_{l=1}^{d} \overline{A^l} Z^l}{B + \delta} \right\} \tag{7}$$

The approach of our tracker at some frame t is summarized in Algorithm 1.

Algorithm 1 Correlation Filter Tracking with Convolutional Network

1: **Input:** Image $\mathbf{I_t}$, a bank of fixed filters $\mathscr{F}$, previous position $\mathbf{p_{t-1}}$ and scale s_{t-1}, translation model $A_{t-1}^{tr}, B_{t-1}^{tr}$ and scale model $A_{t-1}^{sc}, B_{t-1}^{sc}$
2: **Translation estimation:** Extract a sample z_{tr} from $\mathbf{I_t}$ at $\mathbf{p_{t-1}}$ and s_{t-1}. Compute the translation correlation y_{tr} in 7. Set $\mathbf{p_t}$ to the position that maximizes y_{tr}
3: **Scale estimation:** Extract a sample z_{sc} from $\mathbf{I_t}$ at $\mathbf{p_t}$ and s_{t-1}. Compute the scale correlation y_{sc} in 7. Set $\mathbf{s_t}$ to the scale that maximizes y_{sc}
4: **Model Update:** Update appearance model in 2. Update the translation and scale model using 5 and 6 respectively.
5: **Output:** Estimated position $\mathbf{p_t}$ and scale s_t, translation model A_t^{tr}, B_t^{tr} and scale model A_t^{sc}, B_t^{sc}

IV. Experiment

A. Experimental Setup

We refer our method to *Convolutional Networks based Correlation Filter Tracker* (**CNCFT**). The proposed algorithm is implemented by MATLAB and runs 23 frames per second on a PC with Intel i7 (2.2GHz). The initial position of target is specified by the ground truth in the first frame. The size of canonical image and sliding window is set to $n = 32$ and $w = 6$ respectively. The number of filters v is 100 and the learning parameter is set to $\rho = 0.95$ with updating at every

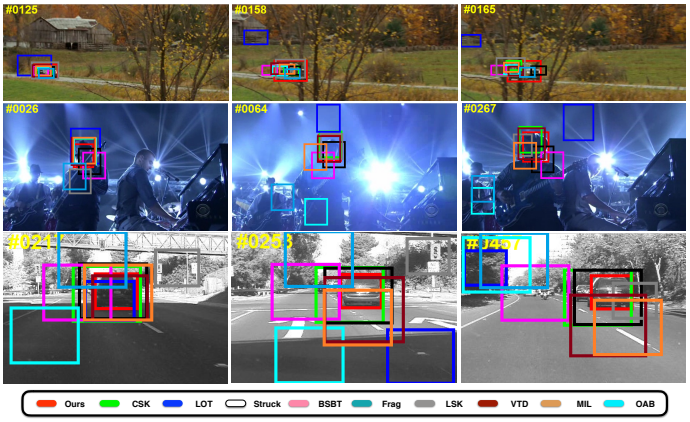
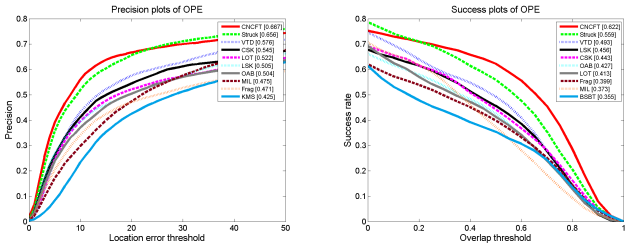


Fig. 2: Tracking results of qualitative comparisons.



(a) (b)

Fig. 3: Distance precision and overlap success plots over 50 video sequences from benchmark using one-pass evaluation (OPE). The legend contains the area-under-the-curve score for each tracker. Our proposed tracking methods CNCFT performs excellent against others.

frame. As for the correlation filter, we set the regularization parameter δ to 0.01. The standard deviations for the correlation output is set to 1/16 of the target size for the translation filter and 1.5 for the scale one. The filter size is initialized twice as the target size in the first frame. The learning rate in equation 5 is set to $\eta = 0.025$. We employ all the 50 video sequences in the benchmark [6] and make the quantitative and qualitative comparisons with 15 state-of-the-art trackers, top 10 of which are displayed in the legend. To validate the effects of our representation and correlation filter, we conduct attribute-based experiments on scale and deformation.

B. Quantitative and Qualitative Comparisons

The quantitative evaluation works by two metrics including precision and success plots, which offer the mean results over 50 sequences. The general metric namely distance precision (**DP**) refers to the number of frames in the sequence where the average Euclidean distance between the right and estimated center location of the target is smaller than a certain threshold. Based on it, the precision plots show the results of the distance precision under a certain range of thresholds. Another generic metric called overlap precision (**OP**) is the percentage of frames where the bounding box overlap excesses a ratio threshold. On the basis of it, the success plot shows the average overlap precision. In addition, the charts display the scores of

TABLE I: Comparisons with top 10 advanced trackers in [6] on the 50 video sequences from benchmark. Our method performs favorably against other methods in distance precision (DP), overlap success rate (OS), frame per second. The first and second highest values are highlighted by **red** and **blue** fonts.

	Ours	BSBT	Frag	KMS	LOT	LSK	MIL	MS	OAB	PD	SMS	Struck	VR	VTD	CT	CSK
DP(%)	66.7	39.6	47.1	42.5	52.2	50.5	47.5	28.4	50.4	39.0	42.2	65.6	33.9	57.6	40.6	54.5
OS(%)	62.2	35.5	39.9	33.1	41.3	45.6	37.3	19.4	42.7	33.5	20.6	55.9	30.1	49.3	0.341	44.3
Speed(FPS)	23.0	7.00	6.30	3.16	0.70	5.50	38.10	31.08	22.40	10.91	10.20	20.20	12.26	5.70	64.40	76.42

the area under the curve (AUC) over 10 different trackers. In the figure 3 and table I, our method outperforms other approaches including the best method in the benchmark [6] from an overview perspective. The precision and success plot respectively demonstrate the positional and scale robustness of our approach. Figure 2 compares the quality of these trackers and our approach gives accurate estimation over these sequences.

C. Specific Attribute-based Comparisons

As our method is designed by a robust representation to local structure and geometric layout as well as a multi-dimensional correlation filter with pretty adaptation to translation and scale, we allocate the attribute-based dataset from 50 sequences to conduct the specific quantitative attribute-based experiments. Figure 4 shows the results of three kinds of attribution including *deformation*, *occlusions* and *scale*. It is clear that our tracker is advance of others in all attribute-based comparisons. In the subfigures (a)(b) performing the results of *deformation* attribution, our tracker gets the AUC score of 0.526 and 0.505 in precision and success plots, outweighing the **Struck** and **VTD**, for our representation owns the characteristics of macroscopic layout and the updating process is efficacious. Subfigures (c)(d) demonstrate the excellent performance of scale variation of our approach. The success plots of our method makes a huge leap against others, reaching 0.624 amazingly while the second rank only has an AUC score of 0.471, which proves that the sophisticated correlation filter performs brilliant adaptation to scale.

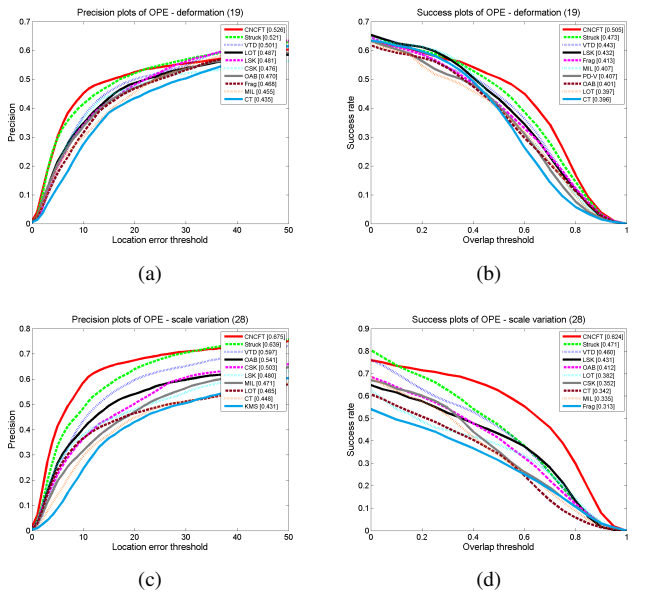


Fig. 4: Attribute-based result of distance precision and overlap success plots over 50 video sequences from benchmark using one-pass evaluation (OPE). Our tracker reaches the best in 2 attribute-metrics.

V. Conclusion

In this paper, we develop a robust visual tracking approach via improved correlation filter with representative feature extracted by convolutional networks. Our method starts with the construction of reliable representation and inherits the correlation filter for the tracking framework. We utilize a two-layer convolutional network to learn the feature map online. Then the feature map is intended for learning discriminative correlation filters for estimating translation and scale, which integrates the advantages of efficiency and accuracy. The experiments are conducted on 50 challenging sequences with different application scenes in the standard benchmark. The quantitative, qualitative and attribute-based evaluations validate the robustness and progressiveness of our tracker.

Acknowledgment

This work was supported by NSFC (61502543), Guangdong Natural Science Funds for Distinguished Young Scholar (2016A030306014).

References

[1] S. Avidan, "Ensemble tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 261–271, 2007.

[2] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1619–1632, 2011.

[3] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," *BMVC*, 2014.

[4] S. Hare, A. Saffari, and P. H. Torr, "Struck: Structured output tracking with kernels," in *ICCV*. IEEE, 2011, pp. 263–270.

[5] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *ECCV*, 2012, pp. 702–715.

[6] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *CVPR*, 2013, pp. 2411–2418.

[7] H. Li, Y. Li, and F. Porikli, *Robust Online Visual Tracking with a Single Convolutional Neural Network*. Springer, 2015.

[8] J. Fan, W. Xu, Y. Wu, and Y. Gong, "Human tracking using convolutional neural networks," *IEEE Transactions on Neural Networks*, vol. 21, no. 10, pp. 1610–1623, 2010.

[9] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *CVPR*, 2010, pp. 2544–2550.

[10] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.