LETTER

Accurate Scale Adaptive and Real-Time Visual Tracking with Correlation Filters

Jiatian PI[†,††a)], *Member*, Shaohua ZENG[†,††], Qing ZUO[†], *and* Yan WEI[†,††], *Nonmembers*

SUMMARY Visual tracking has been studied for several decades but continues to draw significant attention because of its critical role in many applications. This letter handles the problem of fixed template size in Kernelized Correlation Filter (KCF) tracker with no significant decrease in the speed. Extensive experiments are performed on the new OTB dataset.

key words: *correlation filters, kernel methods, scale estimation, visual tracking*

1. Introduction

Online visual object tracking is one of the most fundamental tasks in the field of computer vision and is related to a wide range of real-time vision applications, such as smart surveillance systems, autonomous driving, intelligent traffic control, and human-computer-interfaces. Although great progress has been made in the past decade, it remains a challenging problem due to baffling factors, such as illumination variations, background clutter and shape deformation.

Recent benchmark [1], [2] studies show that the top-performance trackers are usually deep-learning based trackers. However, in the pursuit of ever increasing tracking performance, their characteristic speed and real-time capability have gradually faded. Except for those complicated trackers, recently proposed correlation filter (CF) based trackers [3]–[6] also have achieved appealing performance despite their great simplicity and superior speed. Those trackers train a discriminative filter, where convolution output can indicate the likeness between candidate and target. Because the element-wise operation in Fourier domain is equal to the convolution operation in time domain (spatial domain in tracking), they evaluate the cyclically shifted candidates very efficiently. However, Minimum Output Sum of Squared Error (MOSSE) tracker [3], Circulant Structure Kernels (CSK) tracker [5], and Kernelized Correlation Filter (KCF) tracker [6], are limited to only estimating the target position with the fixed size. Discriminative Scale Space Tracker (DSST) [4] has proposed an efficient method for estimating the target scale by training a classifier on a scale pyramid, which is the best tracker in the competition [7]. However, there is still room for improvement in translation estimation in the DSST. Recently, Discrimination Reliability Tracker (DRT) [13] has proposed a novel CF-based optimization problem to jointly model the discrimination and reliability information. Spatial Temporal Regularized Correlation Filters (STRCF) tracker [14] introduces the temporal regularization to spatially regularized discriminative CF by online Passive-Agressive (PA) algorithm, and achieves superior performance.

Motivated by the lasts developments on the fast DSST [8], we incorporate the proposed scale estimation approach in the fast DSST tracker [8] into the KCF without much computational overhead. The key contributions of this work can be summarized as follows. Firstly, we extend the KCF tracker with the capability of handling scale changes, which obtains an impressive performance in accuracy. Secondly, we verify that the applied scale estimated approach is generic and can be incorporated into the KCF tracker framework. Finally, we perform extensive experiments on the new OTB dataset [1], and show that the proposed tracker achieved a very appealing performance both in accuracy and robustness against the state-of-the-art trackers.

2. The Proposed Tracker

2.1 Translation Estimation with KCF

Recently, the tracking system based on the Kernelized Correlation Filter (KCF) achieves favorable performance with high speed. In that work, Henriques et al. [6] demonstrate that it is possible to analytically model natural image translations, which shows that the resulting data and kernel matrices become circulant under some conditions. The diagonalization by the Discrete Fourier Transform (DFT) provides a general blueprint for creating fast algorithms that deal with translations. By considering correlation filters as classifiers, the goal of training is to find a function $f(\mathbf{z}) = \mathbf{w}^T\mathbf{z}$ that minimizes the squared error over samples x_i and their regression targets y_i according to:

$$\min_{\mathbf{w}} \sum_i (f(x_i) - y_i)^2 + \lambda\|\mathbf{w}\|^2, \qquad (1)$$

where $\mathbf{w}$ denotes the parameters, and λ is the regularization parameter to prevent over fitting. The Ridge Regression has the close-form solution according to:

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}, \qquad (2)$$

Manuscript received June 3, 2018.
Manuscript revised July 11, 2018.
Manuscript publicized July 27, 2018.

[†] The authors are with College of Computer and Information Science, Chongqing Normal University, Chongqing, 401331 China.

[††] The authors are with Digital Agriculture Service Engineering Technology Research Center, Chongqing 401331, China.

a) E-mail: pijiatian@cqnu.edu.cn

DOI: 10.1587/transinf.2018EDL8116

Copyright © 2018 The Institute of Electronics, Information and Communication Engineers

where the data matrix $\mathbf{X}$ has one sample per row x_i and each element of $\mathbf{y}$ is a regression target y_i. $\mathbf{I}$ is an identity matrix.

To introduce the kernel functions for improving the performance, input data x can be mapped to a non-linear-feature space as $\varphi(x)$, and $\mathbf{w} = \sum_i \alpha_i \varphi(x_i)$. Then the solution to the kernelized version of Ridge Regression in the KCF tracker is given by:

$$\alpha = (\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{y}, \tag{3}$$

where $\mathbf{K}$ is the kernel matrix and α is the vector of coefficients α_i, that represents the solution. With the help of circulant matrix, all the translated samples around the target can be collected for training with no significant decrease in the speed. Given a base sample $\mathbf{x} = (x_0, \ldots, x_{n-1})$, all the cyclic shift visual samples are concatenated to form the circulant matrix $\mathbf{X} = C(\mathbf{x})$. Then the solution of α can be expressed as follows with the various interesting properties of circulant matrices

$$\hat{\alpha} = \frac{\hat{\mathbf{y}}}{\hat{\mathbf{k}} + \lambda}. \tag{4}$$

where $\hat{\alpha}$, $\hat{\mathbf{y}}$ and $\hat{\mathbf{k}}$ denote the DFT of α, $\mathbf{y}$ and $\mathbf{k}$, respectively. It has been proven that the kernel function of a circulant kernel matrix should be unitarily invariant [6]. Although dot-product, radial basis kernel and polynomial kernels functions are found to satisfy this condition, we apply the Gaussian kernel which can be expressed as follows:

$$\mathbf{k}^{\mathbf{x}\mathbf{x}'} = \exp(-\frac{1}{\sigma^2}(||\mathbf{x}||^2 + ||\mathbf{x}'||^2 - 2F^{-1}(\hat{\mathbf{x}}^* \odot \hat{\mathbf{x}}'))), \tag{5}$$

where $\hat{\mathbf{x}}$ denote the DFT of the base sample $\mathbf{x}$, and $\hat{\mathbf{x}}^*$ represents complex conjugation. In a new frame, the target can be detected by the trained parameter α and a maintained base sample $\mathbf{x}$. If the new sample is $\mathbf{z}$, a confidence map $\mathbf{y}_{trans}$ can be obtained by:

$$\mathbf{y}_{trans} = C(\mathbf{k}^{\mathbf{xz}})\alpha. \tag{6}$$

The position with a maximum value in $\mathbf{y}_{trans}$ can be predicted as new position of the target.

2.2 Scale Estimation with Fast DSST Filters

The Kernelized Correlation Filter (KCF) in Sect. 2.1 is used for estimating the translation, then we can find the accurate position of the target without scale change. To handle the challenging problem of scale change, we incorporate separate filters [8] for scale estimation. The discriminative correlation filter is closely related to the MOSSE filter [3], which produces stable correlation filters when trained on a small number of image windows. Firstly, the MOSSE filter need a set of training images f_i, as well as a set of training outputs g_i. Training is conducted in the Fourier domain to take advantage of the simple element-wise relationship between the input and the output. To find a filter that maps training inputs to the desired training outputs, MOSSE finds a filter h that minimizes the sum of squared error. The minimization problem takes the form according to:

$$\min_{\hat{h}^*} \sum_i |\hat{f}_i \odot \hat{h}^* - \hat{g}_i|^2, \tag{7}$$

where $\hat{f}_i$, $\hat{g}_i$ and the filter $\hat{h}$ are the Fourier transform of f_i, g_i and h, respectively. $\hat{h}_i^*$ represents complex conjugation. By solving for $\hat{h}^*$, a closed form expression for the MOSSE filter is found

$$\hat{h}^* = \frac{\sum_i \hat{g}_i \odot \hat{f}_i^*}{\sum_i \hat{f}_i \odot \hat{f}_i^*}. \tag{8}$$

where $\hat{f}_i^*$ represents complex conjugation.

In the DSST, the MOSSE filter has been extended to multi-dimensional features. Assuming the feature dimension number $l \in \{1, 2, \ldots, d\}$, the solution for the optimal correlation filter $\hat{h}$, which consists of one filter $\hat{h}^l$ per feature, is obtained in the DSST as follows:

$$\hat{h}^l = \frac{\hat{g}^* \odot \hat{f}^l}{\sum_{k=1}^{d} \hat{f}^k \odot \hat{f}^{k*} + \lambda}, \tag{9}$$

where λ is the regularization parameter to prevent over fitting, and $\hat{g}^*$ represents complex conjugation. To reduce the computational cost of the scale estimation with separate filters, we use Principal Component Analysis (PCA) to reduce the feature dimensionality. The projection matrix R_t is $\tilde{d} \times d$, where $\tilde{d}$ is the dimensionality of the compressed feature representation. We obtain R_t by minimizing the reconstruction error of the target template u_t as similar to the fast DSST [8]. Then the compressed numerator $\tilde{A}_t^l$ and denominator $\tilde{B}_t$ of the filter is updated as follows:

$$\begin{aligned} &\tilde{A}_t^l = \eta \hat{g}^* \tilde{U}_t^l \\ &\tilde{B}_t = (1-\eta)\tilde{B}_{t-1} + \eta \textstyle\sum_{k=1}^{\tilde{d}} \tilde{F}_t^{k*} \tilde{F}_t^k, \end{aligned} \tag{10}$$

where $\tilde{U}_t = F\{R_t u_t\}$, $u_t = (1-\eta)u_{t-1} + \eta f_t$, $\tilde{F}_t = F\{R_t f_t\}$, $F\{\}$ represents Discrete Fourier Transform.The correlation scores $\mathbf{y}_{scale}$ are then computed as follows:

$$\mathbf{y}_{scale} = F^{-1}\{\frac{\sum_{l=1}^{\tilde{d}} \tilde{A}^{l*} \tilde{Z}^l}{\tilde{B} + \lambda}\}. \tag{11}$$

The scale with a maximum value in $\mathbf{y}_{scale}$ can be predicted as the new scale of the target.

2.3 Tracking Algorithm

The main steps of our tracker are presented in Algorithm 1 (see Table 1). We use two independent correlation filters for translation and scale estimation. The KCF is only applied for translation estimation and the discriminative correlation filter cooperates on scale estimation. Unlike our tracker, the DSST [8] uses separate filters for translation and scale estimation, which are all based on discriminative correlation filters. In addition, we extract translation sample with fixed size to find the target position without considering the scale, whereas the DSST extracts translation sample according to

Table 1 Main steps of our algorithm

```
Algorithm 1: Proposed tracking algorithm: iteration at time t
1: Inputs:
  • A bounding box with previous target position $p_{t-1}$ and scale $s_{t-1}$ in Image $I_t$.
  • Training sample feature $X_{t-1}^{trans}$ and parameter $\alpha_{t-1}^{trans}$ for translation model.
  • Training scale model $\tilde{A}_{t-1}^{scale}$ and $\tilde{B}_{t-1}^{scale}$.
2: Translation estimation:
  • Extract a translation sample $z_{trans}$ with fixed size at $p_{t-1}$ in $I_t$.
  • Compute the translation response $y_{trans}$ using $z_{trans}$, $X_{t-1}^{trans}$ and $\alpha_{t-1}^{trans}$.
  • Set $p_t$ to the target position that maximizes the response $y_{trans}$.
3: Scale estimation:
  • Extract a scale sample $z_{scale}$ with scale $s_{t-1}$ at $p_t$ in $I_t$.
  • Compute the scale response $y_{scale}$ using $z_{scale}$, $\tilde{A}_{t-1}^{scale}$ and $\tilde{B}_{t-1}^{scale}$.
  • Set $s_t$ to the target scale that maximizes the response $y_{scale}$.
4: Model update:
  • Extract sample feature with fixed size at $p_t$ in $I_t$ to update $X_t^{trans}$ and $\alpha_t^{trans}$.
  • Extract sample feature with scale $s_t$ at $p_t$ in $I_t$ to update $\tilde{A}_t^{scale}$ and $\tilde{B}_t^{scale}$.
5: Output:
  • Estimated target position $p_t$ and scale $s_t$.
  • Updated the translation model $X_t^{trans}$, $\alpha_t^{trans}$ and scale model $\tilde{A}_t^{scale}$, $\tilde{B}_t^{scale}$.
```

the previous scale. Thus, we really separate the translation and scale estimation in a way. Furthermore, the major difference between the KCF tracker and our tracker is that the KCF tracker is unable to deal with the challenge of scale change.

The main reasons that our algorithm performs favorably can be attributed to three factors. Firstly, both the KCF tracker and DSST have already achieved very appealing performance both in accuracy and robustness against the state-of-the-art trackers. Secondly, we apply the KCF tracker for translation estimation independently, which obtains an accurate position of the target. In addition, we take advantage of the discriminative correlation filter in the DSST for scale estimation specially. Thirdly, we combine the strengths of the KCF tracker and DSST to improve the performance. Consequently, the improved algorithm is more accurate and robust.

3. Experiments

In this section, our proposed algorithm is evaluated with other 7 state-of-the-art methods on the new OTB dataset [1]. These methods are DSST [8], CSK [5], KCF [6], Adaptive Structural Local Appearance (ASLA) [9], Incremental Learning Tracker (IVT) [10], Distribution Fields Tracker (DFT) [11] and Compressive Tracking (CT) [12]. For each tracker, the default parameters with the source code are used in all evaluations. We select 50 difficult and representative ones in the OTB dataset for analysis. The proposed algorithm runs at 110 frame per second (FPS) with a matlab implementation on an Intel Core(TM) i5-4590 3.00 GHz CPU with 4 GB RAM without any optimizing.

In our experiments, we use a Gaussian function to initialize the desired translation and scale filter output, respectively. The regularization parameter is set to 10^{-4}, the learning rate is set to 0.02. The bandwidth of the Gaussian kernel $\sigma = 0.5$, spatial bandwidth for the desired translation filter output is $\sqrt{mn}/10$ for a $m \times n$ target, and the standard for the desired scale filter output is $1/16$ times the number of scales $S = 33$. We use the Principal Component Analysis Histogram of Gradient (PCA-HOG) for target representation. In order to get fair experimental results, all the parameters are kept constant for the following experiments. To compare the performance of different trackers, the precision plot and the success plot with temporal robustness evaluation (TRE), as well as in the benchmark [1], are used to rank the algorithms.

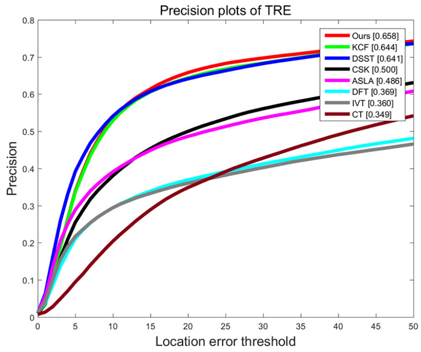

Fig. 1 Precision plots over all 50 sequences. The results at error threshold of 20 are used to ranking as shown in the top right corner.

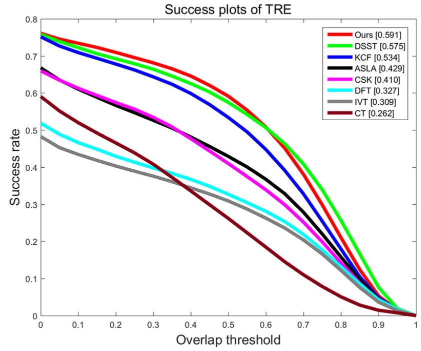

Fig. 2 Success plots over all 50 sequences. The AUC scores of each plot are used to ranking as shown in the top right corner.

Figures 1 and 2 show the ranking scores on the precision plot and success plot. Experimental results show that our tracker achieves 59.1% on the AUC score, which is 5.7% improvement over the KCF and 1.6% improvement over the DSST. In addition to high accuracy, our tracker runs efficiently at an average speed of 110.0 FPS, which is more than 1.2 times faster than the DSST. Although the speed of the KCF tracker is 252.0 FPS on average and is faster than ours, it is not able to handle scale changes. The speeds of the other trackers are demonstrated in Table 2.

There are many factors affect the experimental results when evaluating tracking algorithms. For better analysis of our tracker, we use the sequences annotated with the other 11 attributes in the benchmark [1] to evaluate how

Table 2 The AUC scores of success plots in 11 attributes. The best result is highlighted in red color and the second result is highlighted in blue color. The speeds of different trackers are shown in the last line.

Attris	Ours	KCF	DSST	CSK	ASLA	DFT	IVT	CT
FM	58.4	49.1	50.9	35.1	28.6	28.1	20.5	18.4
MB	63.5	54.0	55.4	38.5	28.7	30.7	20.9	17.5
DEF	52.8	48.6	53.8	35.8	37.5	34.2	26.1	29.4
IPR	57.8	51.8	53.7	40.0	39.9	34.1	27.8	28.2
OCC	52.3	47.2	51.6	33.7	43.2	32.6	33.2	26.7
OPR	55.8	50.9	51.7	35.6	43.9	32.8	29.7	29.3
OV	44.3	39.4	42.8	26.5	32.8	29.1	27.8	25.8
IV	60.8	57.2	61.7	42.0	49.0	34.2	31.5	30.2
BC	62.6	60.9	61.9	44.5	47.9	35.7	31.0	32.4
LR	45.4	43.5	53.9	37.1	48.8	25.8	35.5	22.2
SV	54.1	47.9	54.2	37.1	44.3	28.8	31.3	23.6
FPS	110.0	252.0	91.7	210.6	5.3	7.2	21.7	27.8

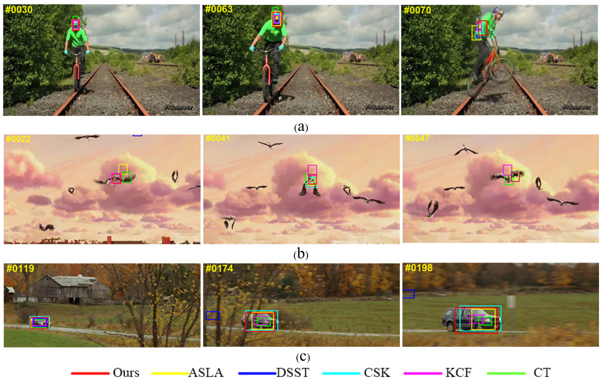

Fig. 3 Performance on (a) 'biker', (b) 'bird' and (c) 'carScale' sequences by 6 trackers.

well the tracker handles different attributes. The name of the attributes are listed as follows: fast motion (FM), scale variation (SV), motion blur (MB), deformation (DEF), in-plane rotation (IPR), occlusion (OCC), out-of-plane rotation (OPR), out-of-view (OV), illumination variation (IV), background clutter (BC) and low resolution (LR). The AUC score of success plots in each attribute are demonstrated in Table 2. According to the experimental result, the proposed algorithm is close to the best performance to 7 of the 11 attributes. So the discriminative correlation filter can be indeed incorporated into the KCF tracker framework to improve the scale estimation. Our tracker performs more favorable than DSST because we apply the KCF tracker to find the optimal translation before scale estimation, which is more accurate than the DSST and can improve the scale estimation. The intuitive illustration is shown clearly in Fig. 3. However, if the scale of the target is changed abruptly and frequently, our tracker performs unfavorably as shown in the first row of Fig. 3. Because the scale change is estimated after the translation estimation, which performs inaccurately when the fast move and scale change happen at the same time.

4. Conclusion

In this paper, we propose a robust tracking algorithm which combines the method of discriminative correlation filters (DCF) with the Kernelized Correlation Filter (KCF) tracker. First, we extract translation sample with fixed size to find the initial target position without considering the scale, which separate the translation and scale estimation. After finding the initial position with the KCF, we apply the DCF for scale estimation. Our tracker handles the problem of fixed template size in KCF tracker without much decrease in the speed. Finally, experiments on benchmark sequences demonstrated that the proposed algorithm performs favorably in terms of accuracy and robustness.

Acknowledgments

This work is supported by Fundamental Research and Frontier Exploration Project in Chongqing under Grant No. cstc2018jcyjA1684.

References

[1] Y. Wu, J. Lim, and M.-H. Yang, "Object Tracking Benchmark," IEEE Trans. Pattern Anal. Mach. Intell., vol.36, no.9, pp.1834–1848, Jan. 2015.

[2] M. Kristan, A. Leonarids, J. Matas, et al., "The Visual Object Tracking VOT2017 Challenge Results," Proc. IEEE ICCV Workshops, Venice, Italy, pp.1949–1972, Oct. 2017.

[3] D.S. Bolme, J.R. Beveridge, B.A. Draper, and Y.M. Liu, "Visual Object Tracking using Adaptive Correlation Filters," Proc. IEEE CVPR, San Francisco, CA, USA, pp.2544–2550, June 2010.

[4] M. Danelljan, G. Häger, F.S. Khan, and M. Felsberg, "Accurate Scale Estimation for Robust Visual Tracking," Proc. British Machine Vision Conference 2014, pp.65.1–65.11, 2014.

[5] J.F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the Circulant Structure of Tracking-by-Detection with Kernels," Proc. European Conf. on Computer Vision, Firenze, Italy, vol.7575, pp.702–715, Oct. 2012.

[6] J.F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed Tracking with Kernelized Correlation Filters," IEEE Trans. Pattern Anal. Mach. Intell., vol.37, no.3, pp.583–596, March 2015.

[7] M. Kristan, R. Pflugfelder, A. Leonarids, et al., "The visual object tracking VOT2014 challenge results," Proc. European Conf. on Computer Vision, pp.191–217, Sept. 2014.

[8] M. Danelljan, G. Häger, F.S. Khan, and M. Felsberg, "Discriminative Scale Space Tracking," IEEE Trans. Pattern Anal. Mach. Intell., vol.39, no.8, pp.1561–1575, Aug. 2017.

[9] X. Jia, H. Lu, and M.-H. Yang, "Visual Tracking via Adaptive Structural Local Sparse Appearance Model," Proc. IEEE CVPR, Providence, Rhode, USA, pp.1822–1829, June 2012.

[10] D.A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental Learning for Robust Visual Tracking," Int. J. Comput. Vision., vol.77, no.1-3, pp.125–141, May 2008.

[11] L. Sevilla-Lara and E. Learned-Miller, "Distribution Fields for Tracking," 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp.1910–1917, 2012.

[12] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time Compressive Tracking," Proc. European Conf. on Computer Vision, Firenze, Italy, pp.864–877, Oct. 2012.

[13] C. Sun, D. Wang, H. Lu, and M.-H. Yang, "Correlation tracking via joint discrimination and reliability learning," CVPR, Salt Lake City, Utah., USA, pp.489–497, June 2018.

[14] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, "Learning spatial-temporal regularized correlation filters for visual tracking," CVPR, Salt Lake City, Utah., USA, pp.4904–4913, June 2018.