A NOVEL CORRELATION FILTER BASED ON KERNEL REGRESSION FOR VISUAL TRACKING

Jianyu Shen[1], Min Jiang[1,*], Jun Kong[1,2], Hongtao Huo[3]
and Xiaofeng Wang[1]

[1]Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education)
Jiangnan University
No. 1800, Lihu Avenue, Wuxi 214122, P. R. China
*Corresponding author: minjiang@jiangnan.edu.cn

[2]College of Electrical Engineering
Xinjiang University
No. 14, Shengli Road, Urumqi 830046, P. R. China

[3]Department of Information Security Engineering
People's Public Security University of China
No. 1, Muxidi Nanli, Xicheng Dist., Beijing 100038, P. R. China

Received July 2016; accepted October 2016

Abstract. *Visual tracking is a challenging problem in computer vision. Recently, correlation filter-based trackers (CFTs) have achieved excellent performance in different competitions and shown great robustness to challenging situations during the tracking process. The core component of the most trackers is a discriminative classifier and its task is to distinguish the target from the surrounding environment. Most of the methods train the classifier with translated and scaled sample patches. However, these samples are riddled with redundancies because there are lots of overlapping pixels in it. Therefore, these samples restrict the speed and stability of the processing of regression problem. To solve this problem, an analytic model for datasets of thousands of translated patches is adopted and a data circulant matrix consists of these patches. According to some properties of circulant matrix with the discrete Fourier transform, the non-linear regression can be solved with the kernel trick in a fast way. In addition, a separate filter is learned for scale estimation during the tracking process. Our method performs well in different challenging situations and the speed of our tracker can arrive to hundreds of frames per second. The results of extensive experiments on benchmark datasets prove the effectiveness of our method.*

Keywords: Visual tracking, Circulant matrix, Discrete Fourier transform, Kernel methods, Ridge regression, Correlation filters

1. **Introduction.** Visual tracking plays an important role in many research fields such as video monitoring, automobile navigation and activity recognition. Recently, discriminative learning methods have been widely adopted and made a great breakthrough. The goal of this method is to learn a classifier to separate the target object from the background. The classifier can be evaluated exhaustively at many locations, in order to detect it in subsequent frames. Generally, we pay more attention to the positive samples which characterize the object of interest for the classifier. However, the negative samples are also very important to discriminative methods because these samples contain useful background information which can be better used to train the classifier. A crucial challenging factor is that unlimited amount of negative samples can be obtained from an image. Due to taking account of the time-sensitive nature of tracking, current tracking methods usually choose only a few negative samples in each frame [1-5]. Although this sampling

method is understandable, limited negative samples are the main factor inhibiting performance in tracking. In order to solve this problem, Henriques et al. [6] proposed kernelized correlation filters (KCF) that develop tools which are called circulant matrices, to analytically incorporate thousands of samples at different relative translations, without iterating over them explicitly. They made it possible because they discovered that some learning algorithms actually become easier with more samples, if we use a specific model for translations. Although the tracking method proposed by Henriques et al. performs well, this method cannot solve the problem of scale variations. In order to overcome this problem and exploit the circulant matrices better, we propose our method which not only exploits the advantage of the circulant matrices but also solves the problem of scale variation that KCF cannot deal with. Our method is tested on the benchmark dataset and we compare the results of experiments with the top ten algorithms in visual tracking.

The paper is organized as follows. Section 2 introduces the method of sampling and the property of circulant matrix which consists of samples. Section 3 introduces the nonlinear regression in object tracking and the effective method to deal with it. Section 4 presents the detection model. Section 5 presents the scale estimate method. In Section 6, we compare the results of the proposed method with the state-of-the-art methods on benchmark. Section 7 concludes the paper.

2. **Sampling and Circulant Matrices.**

In this section, we introduce the method of sampling and the relationship between samples and circulant matrix. Firstly, consider an $n \times 1$ vector which represents a patch with the object of interest denoted x. It will be called the base vector (a positive sample) and several virtual samples which are called the negative samples are obtained by translating the base sample. We can model one-dimensional translations of the base vector by a cyclic shift operator, which is the permutation matrix

$$P = \begin{pmatrix} 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}. \tag{1}$$

The product $Px = [x_n, x_1, \ldots, x_{n-1}]^T$ shifts x by one element. We can achieve a large translation by using the matrix $P^u x$. When u is a negative value, it means shifting in the reverse direction. The full set of shifted signals is obtained with

$$\{P^u x | u = 0, \ldots, n-1\}. \tag{2}$$

In order to compute a regression with shifted samples, we use the set of Equation (2) as the row of a data matrix X:

$$X = C(x) = \begin{pmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_n & x_1 & x_2 & \cdots & x_{n-1} \\ x_{n-1} & x_n & x_1 & \cdots & x_{n-2} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ x_2 & x_3 & \cdots & x_4 & x_1 \end{pmatrix}. \tag{3}$$

X is a circulant matrix and it has several interesting properties [7,8]. The most useful properties are that all circulant matrices can be diagonalized by the discrete Fourier transform (DFT), regardless of the generating vector x [7]. This can be written as

$$X = F diag\left(\hat{x}\right) F^H, \tag{4}$$

where F is a constant matrix which is known as the DFT matrix and it computes the DFT of any input vector as $\mathcal{F}(z) = nFz$. $\hat{x}$ denotes the DFT of the generating vector, $\hat{x} = \mathcal{F}(x)$. F^H is the Hermitian transpose.

3. **Kernel Trick and Non-Linear Regression.** The most important part of tracking process is how to solve the regression. The regression of the tracking-by-detection is often non-linear. Here, we exploit the kernel trick to solve this problem because the most useful quality of non-linear regression functions $f(z)$ with the kernel trick is that the optimization problem is still linear, albeit in a different set of variables (the dual space). Although evaluating $f(z)$ typically increases the complexity with the number of samples, we can overcome this drawback with the circulant matrix.

Firstly, we briefly introduce the kernel trick and define relevant notation. The inputs of a linear problem map to a non-linear feature-space $\varphi(x)$ with the kernel trick and it consists of:

(1) Expressing the solution w as a linear combination of the samples:

$$w = \sum_i \alpha_i \varphi(x_i). \tag{5}$$

The variables under optimization are α, in place of w. This alternative representation α is in the dual space and w is in the primal space;

(2) We can write the algorithm in terms of dot-products $\varphi^T(x)\varphi(x') = k(x, x')$, which are computed through the kernel function k. The goal of training is to find a function $f(z) = w^T z$ and z is the test samples. We use α in the dual space instead of w in the primal space. Therefore, the function $f(z)$ can be written as:

$$f(z) = w^T z = \sum_{i=1}^{n} \alpha_i k(z, x_i). \tag{6}$$

Unfortunately, the complexity of the regression function grows with the number of samples. However, this limitation can be overcome by circulant data.

Secondly, we introduce the efficient kernel regression by the circulant data. The solution to the kernelized version of ridge regression can be obtained from [8]

$$\alpha = (K + \lambda I)^{-1} y, \tag{7}$$

where K is the kernel matrix with elements $K_{i,j} = k(x_i, x_j)$, I is the identity matrix and vector y has elements y_i. The solution w is implicitly represented by the vector α, whose elements are the coefficients α_i in the dual space. If K is circulant for datasets of cyclic shifts, Equation (7) can be reduced to

$$\hat{\alpha} = \frac{\hat{y}}{\hat{k}^{xx} + \lambda}, \tag{8}$$

where k^{xx} is the first row of the kernel matrix $K = C\left(k^{xx}\right)$, and again a hat denotes the DFT of a vector. According to [6], we know radial basis function kernels satisfy Theorem 3.1. In this paper, we choose Gaussian kernel function because it is one of the radial basis function kernels. The Gaussian kernel function is expressed as

$$k(x_1, x_2) = \exp\left(-\frac{1}{\sigma^2}\|x_1 - x_2\|^2\right). \tag{9}$$

Theorem 3.1. *Given circulant data (x), the corresponding kernel matrix K is circulant if the kernel function satisfies $k(x, x') = k(Mx, Mx')$, and M is any permutation matrix [6].*

4. **Detection.** We want to evaluate the regression function $f(z)$ for one image patch independently. In order to detect the object of interest, we wish to evaluate $f(z)$ on several image locations, for example, for several candidate patches. These patches can be modeled by cyclic shifts.

K^z is denoted as the kernel matrix between all training samples and all candidate patches. It is easy to verify that this kernel matrix satisfies Theorem 3.1 and it is circulant for Gaussian kernel because the samples and patches are cyclic shifts of base sample x and base z respectively. Each element of K^z can be obtained by $k\left(P^{i-1}z, P^{j-1}x\right)$. As same as Section 3, we only need the first row to define the kernel matrix:

$$K^z = C(k^{xz}), \tag{10}$$

where k^{xz} is the kernel correlation of x and z. According to Equation (6), the regression function for all candidate patches can be computed with

$$f(z) = (K^z)^T\alpha, \tag{11}$$

where $f(z)$ is a vector which contains the detection response for all cyclic shifts of z. To compute Equation (11) efficiently, we diagonalize it to get

$$\hat{f}(z) = \hat{k}^{xz} \cdot \hat{\alpha}. \tag{12}$$

Each $f(z)$ is a linear combination of the neighboring kernel values from $k^x z$ and it is weighed by the learned coefficients. Intuitively, evaluating $f(z)$ at all locations can be seen as a spatial filtering operation over the kernel values k^{xz}. Therefore, it can be computed efficiently in the Fourier domain.

5. Correlation Filters for Adaptive Scale Evaluation.

5.1. Discriminative correlation filters.

In this section, we briefly introduce the discriminative correlation filter formulation based on the minimum output sum of squared error (MOSSE) tracker [9]. Using a number of training samples f, these samples are labelled with the desired correlation outputs g from the filter. The optimal correlation filter can be obtained by minimizing the sum of squared errors:

$$\tau = \left\|\sum_{l=1}^{d} h^l * f^l - g\right\|^2 + \gamma \sum_{l=1}^{d} \|h^l\|^2, \tag{13}$$

where d denotes the dimension of feature and $*$ denotes circular correlation. Equation (13) can be computed to

$$H = \frac{\bar{G}F}{\sum_{k=1}^{d} \bar{F^k}F^k + \gamma}, \tag{14}$$

where all the capital letters denote the discrete Fourier transform (DFTs) of the corresponding functions.

5.2. Adaptive correlation filters.

We propose an adaptive scale evaluation method based on the correlation filters. Our method can adapt to the scale variation during the tracking process. By learning a separate 1-dimensional correlation filter which is also called scale filter to estimate the target scale in an image. We use variable patch sizes around the target to extract the features of the training samples f which will be used to update the scale filter. Denote $A \times B$ as the target size in the current frame and $S \times 1$ as the size of the scale filter. We extract an image patch j_n of size $a^nA \times a^nB$ around the target and $n \in \left\{\lfloor -\frac{s-1}{2} \rfloor, \ldots, \lfloor \frac{s-1}{2} \rfloor\right\}$, and a represents scale factor. We compute the correlation score y on a rectangular region by Equation (15)

$$y = \mathcal{F}^{-1}\frac{\sum_{k=1}^{d} Z^k \bar{M^k}}{N + \gamma}, \tag{15}$$

where M and N are the numerator and denominator of Equation (14) and we update them by Equation (16) and Equation (17). Z is the feature map at the predicted target location.

$$M_t = (1-\theta)M_{t-1} + \theta\bar{G}_tF_t, \tag{16}$$

$$N_t = (1 - \theta)N_{t-1} + \theta \sum_{k=1}^{d} \bar{F}^k F^k, \tag{17}$$

where t denotes the t-th frame and θ is a learning rate parameter. The algorithm of the tracking can be seen in Table 1. c and d in Table 1 are temporary variables for Gaussian kernel calculation.

Table 1. The framework of the algorithm

```
Algorithm
When the $i$-th frame arrives
Inputs:
  Target position $P_{i-1}$ and scale $S_{i-1}$
  $x$: training image patch from $P_{i-1}$
  $y$: regression target, Gaussian-shaped
  $z$: test image patch from $P_{i-1}$
Output:
  Estimated target position $P_i$ based on the response and scale $S_i$
The algorithm can be divided into translation estimate and scale estimate
1. Translation estimate
    Step 1: Extract training samples $x$ from $P_{i-1}$ and use Equation (8) to train the
    classifier
              Function $\alpha = \text{train}(x, y, \theta, \lambda)$
                        $k = \text{kernel\_correlation}(x, x, \theta)$
                        $\alpha = \mathit{fft2}(y)./(\mathit{fft2}(k) + \lambda)$
              End
    Step 2: Extract candidate samples $z$ from the $i$-th frame and use Equation (12)
    to obtain responses
              Function responses = $\text{detect}(\alpha, x, z, \theta)$
                        $k = \text{kernel\_correlation}(z, x, \theta)$
                        responses = $real(\mathit{ifft2}(\alpha. * \mathit{fft2}(k)))$
              End
              Function $k = \text{kernel\_correlation}(x_1, x_2, \theta)$
                        $c = \mathit{ifft2}(sum(conj(\mathit{fft2}(x_1)). * \mathit{fft2}(x_2), 3))$
                        $d = x_1' * x_1 + x_2' * x_2 - 2 * c$
                        $k = \exp\left(-\frac{1}{\sigma^2} * \frac{abs(d)}{numel(d)}\right)$
              End
    Step 3: Set $P_i$ to the target position that maximizes the responses
2. Scale estimate
    Step 1: Extract a scale sample $Z_{scale}$ from $P_i$ and $S_{i-1}$
    Step 2: Use Equation (15) to compute the scale correlation response
    Step 3: Set $S_i$ to the target scale that maximizes the response
```

6. **Experiments.**

6.1. **Details and parameters.** λ in Equation (8) and γ in Equation (15) are fixed to be 0.0001 and 0.01 respectively based on empirical result. Learning rate parameters θ in Equation (16) and Equation (17) are all selected by experimental validation on a number of sequences. We find when the learning rate parameter θ is all set to 0.025, the performance of tracking is robust. We choose sequences containing serious scale variation to adjust the number of scales S and scale factor a during the experiments. We find that when $S = 33$ and $a = 1.02$, the results of the experiments achieve the best. We use different σ of Gaussian between 0.1 and 0.9 during the tracking process and we find when σ is between 0.2 and 0.4, the results are stable; therefore, we choose $\sigma = 0.2$.

6.2. **Dataset.** Here, we present a comprehensive evaluation of the proposed method. Results are evaluated on recent benchmark datasets which include 26 state-of-the-art methods from [10]: CPF, LOT, IVT, ASLA, SCM, L1APG, MTT, VTD, VTS, LSK, ORIA, DFT, KMS, SMS, VR-V, Frag, OAB, SemiT, BSBT, MIL, CT, TLD, Struck, CSK, CXT and KCF.

6.3. **Robustness evaluation.** Videos in the benchmark dataset are annotated with attributes, which describe the challenges that tracking algorithms will face in each sequence, for example, illumination changes, deformation and occlusions. These attributes are used for analyzing and characterizing the behavior of trackers in such a large dataset without

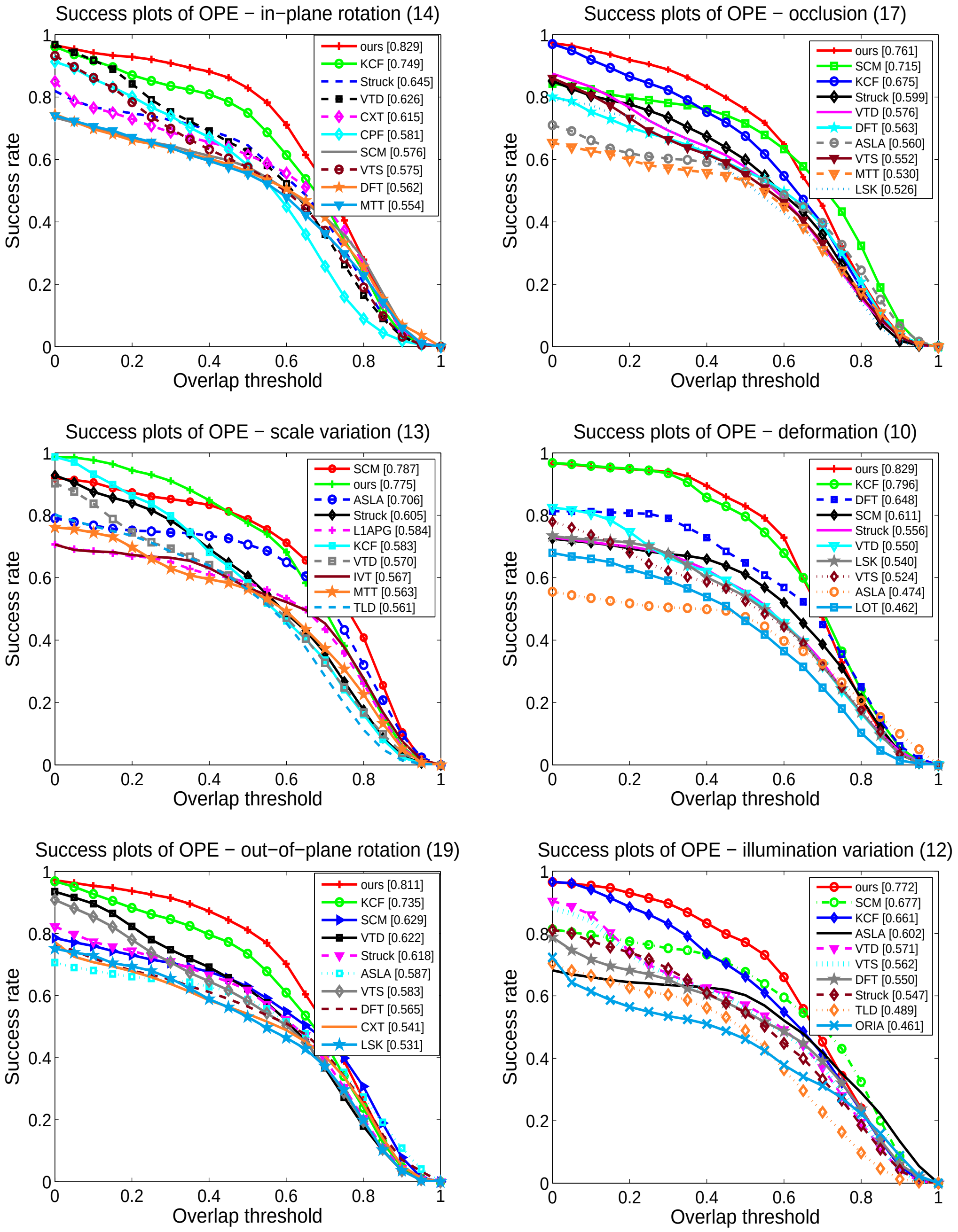

Figure 1. The performance score of different challenging factors of each tracker

having to diagnose each individual video. We show results for 6 attributes in Figure 1: illumination variation, out-of-plane rotation, deformation, occlusion, in-plane rotation and scale variation. Only the top 10 trackers for each attribute are displayed for clarity. Our method performs well compared to other methods in these scenarios. Several reasons can account for the performance of our method. Firstly, we adopt enough negative and positive samples to train the discriminative classifier during the tracking process. Secondly, kernel trick is introduced to handle the problem of non-linear regression. Therefore, the classifier is powerful to locate the target object. In addition, when the location of the target is found by the classifier, we extract features around the location to learn another scale filter which is used to deal with the serious scale variation. Through the cooperation between the discriminative classifier and scale filter, our tracking method can overcome different challenging situation during the tracking process. From Figure 1, we can find that although the KCF tracker also performs well, the template of the KCF is fixed and cannot deal with serious scale variation.

7. **Conclusion.** In this paper, we propose a robust tracking algorithm which exploits the strength of circulant matrices and kernel trick with discrete Fourier transform to deal with the regression effectively and lock on the tracked target quickly. After the target position is established, we learn a 1-dimensional discriminative correlation filter to estimate the target scale around the target position. Since all of the calculation can be translated into the Fourier domain, our tracking method can arrive to hundreds of FPS and achieve real-time object tracking. The results of numerous experiments on various challenging videos demonstrate that the proposed tracker performs favorably against several state-of-the-art algorithms. In our future work, we will focus on other kernel function and features to improve the tracking performance. Furthermore, we will extend our method for real-time tracking of multiple objects.

Acknowledgment. This work is partially supported by the National Natural Science Foundation of China (61201429, 61362030), the Project Funded by China Postdoctoral Science Foundation (2015M581720), Technology Research Project of the Ministry of Public Security of China (2014JSYJB007).

REFERENCES

[1] K. Zhang, L. Zhang and M. H. Yang, Real-time compressive tracking, *European Conference on Computer Vision*, vol.7574, no.1, pp.864-877, 2012.

[2] Z. Kalal, K. Mikolajczyk and J. Matas, Tracking-learning-detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.34, no.7, pp.1409-1422, 2012.

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

[4] A. Saffari, C. Leistner, J. Santner, M. Godec and H. Bischof, On-line random forests, *International Conference on Computer Vision Workshops*, pp.1393-1400, 2009.

[5] S. Hare, A. Saffari and P. H. S. Torr, Structured output tracking with kernels, *International Conference on Computer Vision*, vol.23, no.5, pp.263-270, 2011.

[6] J. F. Henriques, C. Rui, P. Martins and J. Batista, High-speed tracking with kernelized correlation filters, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.37, no.3, pp.583-596, 2015.

[7] R. M. Gray, Toeplitz and circulant matrices: A review, *Foundations and Trends in Communications and Information Theory*, vol.2, no.3, pp.155-239, 2006.

[8] I. Kra and S. R. Simanca, On circulant matrices, *Notices of the American Mathematical Society*, vol.59, no.3, pp.368-377, 2004.

[9] D. S. Bolme, J. R. Beveridge, B. A. Draper and Y. M. Lui, Visual object tracking using adaptive correlation filters, *IEEE Conference on Computer Vision and Pattern Recognition*, vol.119, no.5, pp.2544-2550, 2010.

[10] Y. Wu, J. Lim and M. H. Yang, Online object tracking: A benchmark, *IEEE Conference on Computer Vision and Pattern Recognition*, vol.9, no.4, pp.2411-2418, 2013.