Kernelized correlation filter tracking with scale adaptive filter and feature integration

Shan Jiang
University of Chinese Academy of Science
Institute of Automation, CAS
Beijing, China
Email: jiangshan2017@ia.ac.cn

Shuxiao Li
Institute of Automation, CAS
Beijing, China
Email: shuxiao.li@ia.ac.cn

Chengfei Zhu
Institute of Automation, CAS
Beijing, China
Email: chengfei.zhu@ia.ac.cn

***Abstract*—Recently, kernelized correlation filter (KCF) has been a popular tracker for high accuracy and robustness with high speed. However, KCF tracks objects with a fixed size template without scale estimation, causing tracking failure during target scale changes because of learning background or local appearance of the target. In this paper, we incorporate a separate scale filter into KCF tracker with feature integration. Experiments have shown that our tracker outperforms KCF and other scale adaptive trackers on distance and overlap precision while attaining relatively high speed.**

***Index Terms*—computer vision, visual tracking, correaltion filter, scale, feature**

I. Introduction

Visual tracking is one of the most important problems in computer vision, for its extensive applications ranging from video surveillance, human-computer interaction and medical imaging. The problem is to estimate the state(e.g., position and scale) of the target object given the initial state in the first frame. It is a model-free problem for the tracking is performed without using any explicit appearance or shape model. Therefore, the problem is challenging due to motion blur, occlusion, deformation and scale change in the video.

Recently, correlation filter based trackers have achieved excellent performance, showing accurate tracking with real-time speed. In 2010, Bolme *et al.* [1] first introduce correlation filter into visual tracking, setting up a discriminative model with Minimum Output Sum of Squared Error (MOSSE) filter. In 2012, Henriques *et al.* [2] exploit the circulant property of sample data matrix, proposing CSK based on MOSSE. In 2015, Henriques *et al.* proposed KCF [3], incorporating HoG feature and kernelized multi-channel correlation into the tracker, greatly improving the tracker performance.

Despite high speed and robustness of KCF, it fails to estimate the scale variation of the target with a fixed scale template, resulting in tracking failure when the target scale changes because of the target appearance change in the sample window. To tackle this problem, Li *et al.* [4] proposed SAMF, performing KCF tracker on multiple scales with multiple features integrated, showing high accuracy but low tracking speed. Danelljan *et al.* [5] use a separate scale filter to estimate scale variation based on MOSSE, namely DSST, achieving high accuracy and efficiency simultaneously. Huang *et al.* [6] incorporate detection proposal search into tracking framework based on KCF, enabling the tracker scale and aspect ratio adaptability. In 2017, Danelljan *et al.* [7] incorporate PCA dimensionality reduction into feature extraction step, expanding tracking speed and accuracy.

In this paper, we propose a robust scale adaptive tracking algorithm based on KCF. The main contributions include:

(1)we use a separate correlation filter to estimate scale change, which has high accuracy and efficiency than that of SAMF.

(2)we integrate HoG feature of 4×4 cell size, raw grayscale and color naming features, and employ interpolation to improve the tracking accuracy, resulting in both higher accuracy and efficiency than DSST(with HoG feature of 1×1 cell size + raw grayscale.

(3)Experimental results on OTB-100 have shown that the proposed method achieves better robustness and is about or more than 2X faster than SAMF and DSST.

II. Method

In this section, we first review the kernelized correlation filter tracker, then introduce improvement we make in our approach, including scale adaptive filter and feature selection.

A. KCF tracker

The KCF tracker uses ridge regression as its discriminate model, then exploits the property of the circulant matrix to augment negative samples and reduce computation, achieving high performance and high efficiency. [3] uses the circulant shift of the base sample to augment the data samples around the base sample. Suppose the base sample is one-dimensional data $\mathbf{x} = [x_1, x_2, ..., x_{n-1}, x_n]$, then a cyclic shift of $\mathbf{x}$ is $P\mathbf{x} = [x_n, x_1, x_2, ..., x_{n-1}]$. Concatenating all cyclic shifts $\{P^u\mathbf{x}|u = 1, ..., n-1\}$ with $\mathbf{x}$ by rows, we get the data matrix as below:

$$X = C(\mathbf{x}) = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_n & x_1 & x_2 & \cdots & x_{n-1} \\ x_{n-1} & x_n & x_1 & \cdots & x_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_2 & x_3 & x_4 & \cdots & x_1 \end{bmatrix} \quad (1)$$

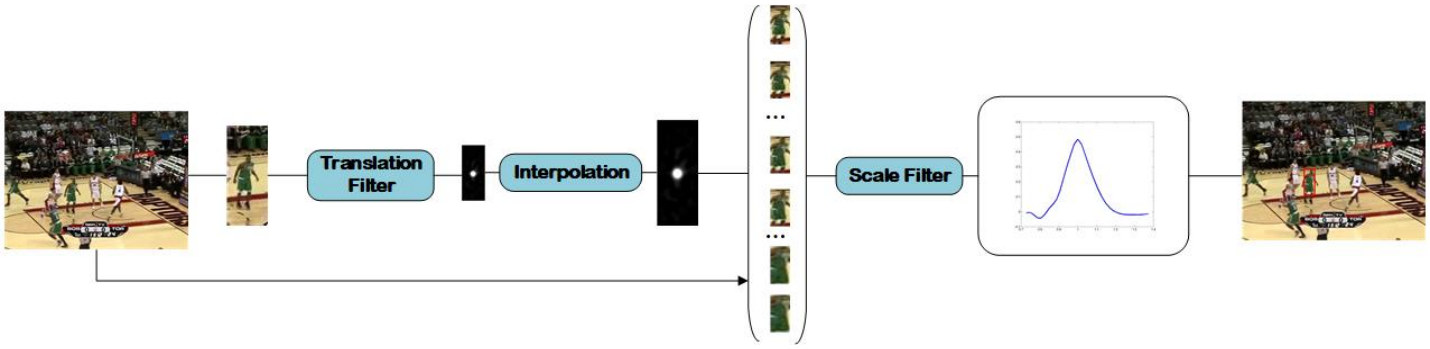

Fig. 1. Flow chart of the tracking process. The peak of response map is shifted to the center for better visualizaion.

All circulant matrices can be diagonalized by discrete Fourier transform(DFT) as following. [8]

$$X = F^H diag(\hat{\mathbf{x}}) F \tag{2}$$

where F is the DFT matrix, and F^H is the Hermitian transpose of F. $\hat{\mathbf{x}} = F\mathbf{x}$ is the Fourier domain representation of $\mathbf{x}$.

The goal of ridge regression is finding a function $f(\mathbf{z}) = \mathbf{w^T z}$ that minimize the following objective function.

$$\min_{\mathbf{w}} \sum_i (f(\mathbf{x}_i) - y_i)^2 + \lambda ||\mathbf{w}||^2 \tag{3}$$

This regression has the closed-form solution [9]

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X} + \lambda \mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} \tag{4}$$

Substituted by (2), the Fourier transform of $\mathbf{w}$ can be solved as [2]

$$\hat{\mathbf{w}}^* = \frac{\hat{\mathbf{x}}^* \odot \hat{\mathbf{y}}}{\hat{\mathbf{x}}^* \odot \hat{\mathbf{x}} + \lambda} \tag{5}$$

where $\hat{\mathbf{x}}^*$ denotes the complex-conjugate of $\hat{\mathbf{x}}$, and multiplication and division are all element-wise.

Then Henriques *et al.* [3] introduces kernel trick, which maps the sample $\mathbf{x}$ to non-linear feature space $\phi(\mathbf{x})$, decomposing solution $\mathbf{w}$ as the weighted sum of samples $\mathbf{w} = \sum_i \alpha_i \phi(\mathbf{x_i})$. The dot-products are turned into kernelized dot-product $k(\mathbf{x}, \mathbf{x}^{'}) = \phi^T(\mathbf{x})\phi(\mathbf{x}^{'})$. Thus the optimization variable turns into the coefficients α.

As proved in [3], circulant trick can also be used in the cases of most commonly used kernels, such as Gaussian kernel, polynomial kernel and linear kernel. With the kernel correlation $\mathbf{k^{xx}}$ of base sample $\mathbf{x}$, the solution α can be obtained as

$$\hat{\alpha} = \frac{\hat{\mathbf{y}}}{\hat{\mathbf{k}}^{\mathbf{xx}} + \lambda} \tag{6}$$

where kernel correlation $\mathbf{k^{xx^{'}}}$ is the vector with elements $k_i^{\mathbf{xx}^{'}} = k(\mathbf{x}^{'}, P^{i-1}\mathbf{x}) = \phi^T(\mathbf{x}^{'})\phi(P^{i-1}\mathbf{x})$.

In detection process, the circulant matrix property is also substituted to simplify the computation. The patch $\mathbf{z}$ is collected at the current position as the base sample, then regression function $f(\mathbf{z})$ can be simply computed in Fourier domain by

$$\hat{f}(\mathbf{z}) = \hat{\mathbf{k}}^{\mathbf{xz}} \odot \hat{\alpha} \tag{7}$$

Then $\hat{f}(\mathbf{z})$ is transformed back into spatial space, the movement of the maximum response is considered as the translation of the target respect to the previous frame. Then the position is updated. At the new position, we collect new patch $\mathbf{x}^{'}$ and learn the new coefficients $\alpha^{'}$ by (6), then perform the following model update scheme.

$$\hat{\alpha}_{new} = (1-\eta)\hat{\alpha}_{pre} + \eta\alpha^{'} \tag{8}$$

$$\mathbf{x}_{new} = (1-\eta)\mathbf{x}_{pre} + \eta\mathbf{x}^{'} \tag{9}$$

where η denotes interpolation factor, or learning rate.

For most kernels, the computation of kernelized correlation is based on the dot-products of samples. Computing a multi-channel dot-product can be simply performed by summing the individual dot-products of each channel. Due to linearity of DFT, the computation of kernelized correlation can be easily expanded to multi-channel case. In this paper, we use multi-channel Gaussian correlation, computed as below,

$$\mathbf{k^{xx^{'}}} = \exp(-\frac{1}{\sigma^2}(||\mathbf{x}||^2 + ||\mathbf{x}^{'}||^2 - 2F^{-1}(\sum_c \hat{\mathbf{x}}_c^* \odot \hat{\mathbf{x}}_c^{'}))) \tag{10}$$

B. Feature selection

One of the advantages that KCF have over CSK [2] is the employment of Histogram of Gradient(HoG) features [10]. HoG decriptor describes the gradient feature in a cell, extracting global shape and local texture information from the image. This feature is very effective in application and is one of the most popular features in computer vision. In our method, we use the HoG feature of 31 gradient orientation bins.

However, KCF tracker fails to take advantage of color information of the target. To incorporate color information into the tracker feature, we append color-naming and grayscale pixel feature to the feature channels, as presented in [4]. Color-nameing, or color attributes, is linguistic color labels assigned by human to describe the color. This is a color space which is more similar to human visual sense than RGB space, which has obtained fantastic results in object detection, object recognition and action recognition. According to [11], we map RGB color to a 10 dimensional feature vector, with each value representing the probability of the color to be assigned to the corresponding label, providing visual perception of the

target color. Raw grayscale pixels are normalized to [0, 1] and minus 0.5. HoG, color-naming and grayscale features are concatenated to be complementary with each other. Note that the HoG feature size does not consist with color-naming and grayscale features, the image patch needs to be resized to be aligned with HoG feature before the extraction of color-naming and grayscale features.

C. Scale Filter

Inspired by [5], we use a separate one-dimensional scale filter to incorporate scale estimation to the tracker, reducing the scale search space. In visual tracking, the scale variation is much less frequent than translation. Therefore, the KCF tracker is first applied to detect translation, then the scale filter is applied at the new target location.

The filter learnt by extracting samples of different scales on the center of the target. Let $M \times N$ denote the target size in the current frame and S be the size of the scale filter. For each $n \in \{\lfloor -\frac{S-1}{2} \rfloor, ... \lfloor \frac{S-1}{2} \rfloor\}$, the scale samples are obtained by extracting an image patch I_n of $a^n M \times a^n N$ centered around the target. Then the patches are resized to the same base sample size and extracted a d-dimensional column feature descriptor. In our algorithm, the feature descriptor of scale samples is HoG descriptor, which is resized to one-dimensional vector. S feature descriptors concatenated by columns forms the $d-$channel scale filter training sample $\mathbf{x}_s \in d \times S$.

With feature extracted, a one-dimensional MOSSE-like filter is trained to estimate scale variation. The training process is to minimize the following cost function

$$h = \underset{h}{\operatorname{argmin}} || \sum_{l=1}^{d} h^l \star x_s^l - y||^2 + \lambda \sum_{l=1}^{d} ||h^l||^2 \tag{11}$$

In equation (11), h^l is the $l-$th channel of the d-channel filter to be trained, x_s^l is the $l-$th channel of the extracted training sample, y is the desired output . The parameter λ is the regularization coefficient. Eq (11) has a solution in Fourier domain as

$$H^l = \frac{\bar{Y} F^l}{\sum_{k=1}^{d} \bar{F^k} F^k + \lambda} \tag{12}$$

where H^l is the $l-$th channel of the Fourier transform of the filter h, F^l is the $l-$th channel of the training sample $\mathbf{x}_s$, and Y is the Fourier transform of the desired output y. $\bar{X}$ denotes the complex-conjugate of X. Here, we denote the numerator and denominator by A and B.

$$\begin{aligned} A &= \bar{Y} F^l \\ B &= \sum_{k=1}^{d} \bar{F^k} F^k \end{aligned} \tag{13}$$

In the tracking process, the numerator and denominator of the filter H are updated separately.

$$A_{new} = (1 - \eta) A_{pre} + \eta A \tag{14}$$

$$B_{new} = (1 - \eta) B_{pre} + \eta B \tag{15}$$

Sequence	Frames	Challenges
Basketball	725	IV, OCC, DEF, OPR, BC
Biker	142	SV, OCC, MB, FM, OPR, OV, LR
Bird1	408	DEF, FM, OV
Bird2	99	OCC, DEF, FM, IPR, OPR
BlurBody	334	SV, DEF, MB, FM, IPR
BlurCar1	742	MB, FM
BlurCar2	585	SV, MB, FM
BlurCar3	359	MB, FM
...		

TABLE I
TEST SEQUENCES IN OUR EXPERIMENT

The scale estimation is performed by maximizing the output score y with the scale sample $\mathbf{z}_s$ extracted at the new location.

$$y = F^{-1}\{\frac{\sum_{l=1}^{d} \bar{A^l} Z^l}{B + \lambda}\} \tag{16}$$

where Z^l is the $l-$th channel of the Fourier transform of $\mathbf{z}_s$, and F^{-1} denotes inverse Fourier transform.

By finding the maximum of y, we obtain the scale difference with the previous frame as a^r, where r is the index of the maximum.

D. Details

As stated in [3], the input patches are multiplied by a cosine window, in order to remove discontinuity on image boundaries caused by the cyclic assumption. The desired output label y is a Gaussian function with the peak at the top-left corner.

For the cell size of HoG descriptor and scale variation of the target, the translation estimated from the response should be multiplied with the HoG cell size and scale factor to obtain the target translation. To estimate the translation more precisely, we interpolate the response to the real sample patch size. The interpolated response $\hat{\mathbf{y}}$ is obtained by zero-padding the high frequencies to make its size identical to the feature interpolation grid. This technique is called sub-grid interpolation in [7]. The translation is estimated directly from the inverse DFT of interpolated response. Figure 1 summarizes the tracking process.

III. EVALUATION

To comprehensively evaluate the efficacy of proposed tracker, we conduct experiments on the visual tracking benchmark [12], namely OTB-100, which includes 100 video sequences with various challenges. These challenges include illumination variation(IV), scale variation(SV), occlusion(OCC), deformation(DEF), motion blur(MB), fast motion(FM), in-plane rotation(IPR), out-of-plane rotation(OPR), out-of-view(OV), background clutters(BC), low-resolution(LR). Table I lists some of test sequences in our experiment with challenging attributes in visual tracking. We first compare feature integrated and sub-gird interpolated KCF tracker with the original KCF tracker to validate the effect of feature integration and sub-grid interpolation. Then we compare our tracker with KCF, SAMF, DSST tracker, which have shown excellent performance in literature.

Fig. 2. In sequence Tiger2(left), Panda(middle) and Girl(right), standard KCF fails to track the object because of temporary occlusion and deformation while KCF with CN and grayscale feature integrated can handle these occasions and successfully track the target.

A. Experiment Setup

We implement the proposed tracker with Matlab and C++ mex. C++ in only employed to resize image samples with OpenCV library. All experiments are conducted on an Intel i7-7200U CPU(2.50 GHz) computer with 8 GB memory. The σ in translation y label is $\sqrt{mn}/10$, where m and n are height and width of the target. The σ used in scale filter is 0.25. The sample of translation filter is 2.5 times the size of target size, with a padding of 1.5. The cell of HoG feature is 4×4. The learning rate η for translation filter is set to 0.01, and 0.025 for scale filter. The regularization factor λ is 10^{-4}. The scale filter has $S = 33$ scale size with scale factor of $a = 1.02$.

B. Experiment criteria

In the experiment, two criteria are employed, one is center distance precision(DP) and the other is overlap precision(OP). These precisions are popular performance measures in tracker evaluation. We can consider a frame to be successfully tracked if the predicted target location error with the ground truth is within a specified threshold. Precision curves are plotted showing the proportion of successfully tracked frames with a range of different thresholds. Distance precision is the proportion of the frames with the distance between predicted target and ground truth under the specified threshold, which is usually 20 pixels. Overlap precision is the proportion of the frames with the overlap between the target over the specified threshold, which is usually 0.5. Overlap between the predicted target and ground truth is defined as following.

$$Overlap = \frac{|B_T \bigcap B_G|}{|B_T \bigcup B_G|} \tag{17}$$

where B_T and B_G are respectively the tracking bounding box and ground truth.

C. Experiment 1: Comparison between KCF and our improved KCF

To evaluate the performance gain of color feature integration and response map interpolation, we test the standard and our improved KCF on OTB-100 dataset. Learning rate of improved KCF is set to 0.15 for more descriptive feature of the target. We plot mean distance precision over the 100 video sequences on Figure 3, which shows that our improved KCF tracker tracks object more robustly and more accurately than standard KCF tracker, for the utilization of color feature and more precise estimation of the target movement. As is shown in Figure 2, the improved tracker performs more robustly compared with standard KCF tracker.

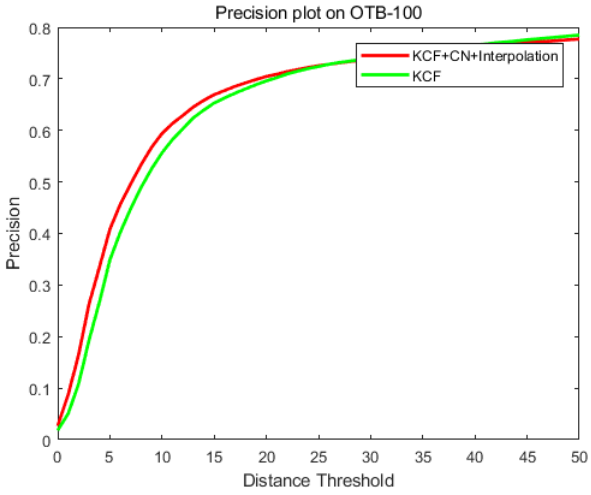

Fig. 3. Distance precision plot of KCF and improved KCF

D. Experiment 2: Comparison with other trackers

We test proposed tracker on OTB-100 to compare with KCF, SAMF, DSST. Table II summarizes the key properties of the trackers for comparison. As is shown in Table III, our tracker has the superior distance and overlap precision in the trackers for comparison while attaining an FPS of 91.67, which is faster than scale-adaptive filters for comparison. This shows our tracker can perform with high accuracy while attaining relatively high speed. Our tracker runs faster than DSST because our tracker uses a HoG feature of 4×4 cell

Fig. 4. In sequence CarScale, our tracker and DSST correctly estimate the scale variation of the target while SAMF underestimate the car scale variation

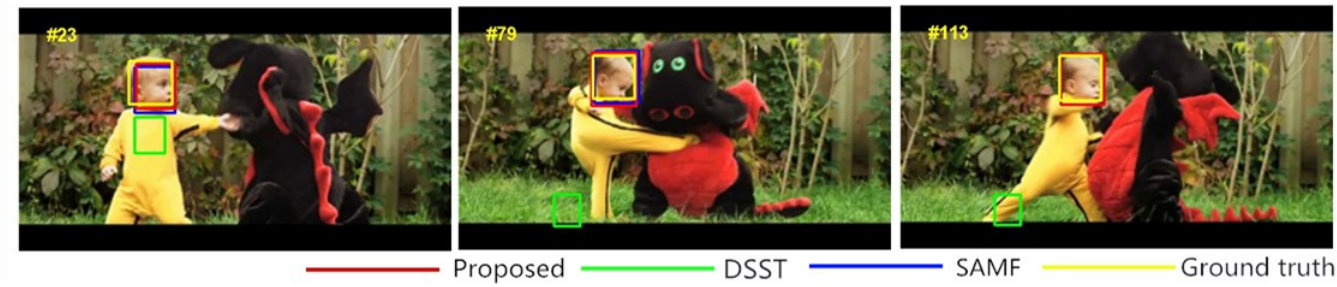

Fig. 5. In sequence Dragonbaby, DSST fails to track the target while SAMF and our tracker successfully tracks the target because of the integration of color-naming feature

Tracker	Feature	Scale Adaptive
Proposed	HoG+Grayscale+CN	Yes
KCF	HoG	No
SAMF	HoG+Grayscale+CN	Yes
DSST	HoG+Grayscale	Yes

TABLE II
TRACKERS FOR COMPARISON

Tracker	DP, 20px	OP, 0.5	Mean FPS
Proposed	**0.7610**	**0.5782**	**91.6774**
KCF	0.6960	0.4544	354.1412
SAMF	0.7394	0.5807	21.5490
DSST	0.6887	0.5368	46.3543

TABLE III
MEAN DISTANCE AND OVERLAP PRECISION ON OTB-100

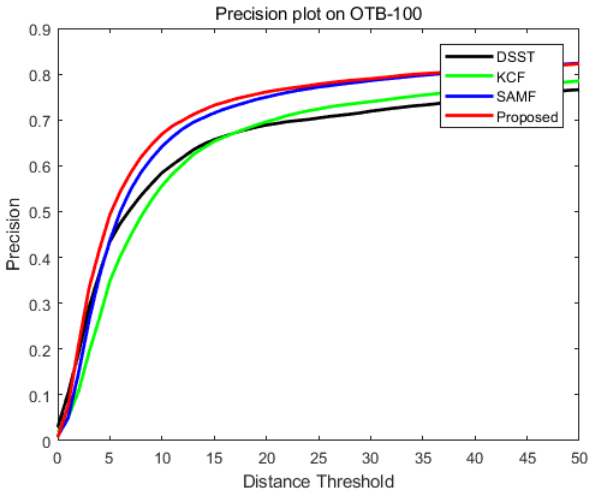

Fig. 6. Distance precision plot of trackers for comparison on OTB-100

size for translation filter while DSST uses a HoG feature of 1×1 cell size. For SAMF, which performs KCF tracker on multiple scales, a separate scale filter has a faster performance because of less computation. We plot mean distance and overlap precision over 100 video sequences on Figure 6 and 7, showing that our tracker has a better performance over DSST and SAMF. As is shown in Figure 4, a separate one-dimensional scale filter can better handle scale variation than performing detection on multiple scales as SAMF. On the other hand, we can see from Figure 5 that trackers can perform more robustly with color feature integrated for taking advantage of color information.

IV. CONCLUSION

This paper presents a scale adaptive tracker based on KCF with color-naming and grayscale feature integration. The experiments on visual tracking benchmark have shown that our tracker outperforms KCF and other scale adaptive trackers both on distance and overlap accuracy, while attaining relatively high speed despite additional computation on scale search. However, all of these trackers fails in cases of occlusion(e.g. Box, Girl2), deformation(e.g. Diving, Bolt2) and out-of-view(e.g. Bird2, Biker), and the value of learning rate largely influences the performance of the tracker, which indicates that the tracker model updating scheme should be investigated in future work.

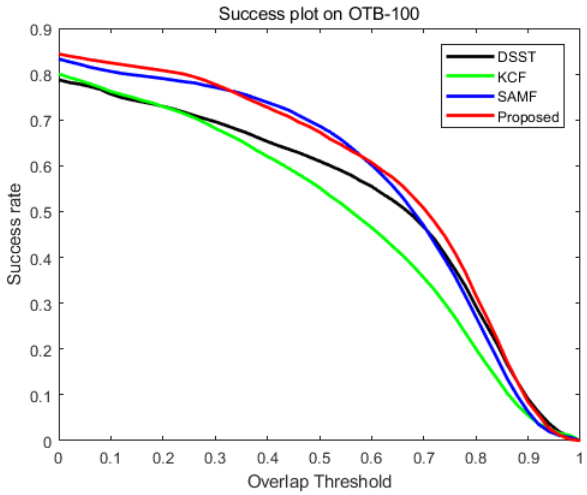

Fig. 7. Overlap precision plot of trackers for comparison on OTB-100

ACKNOWLEDGEMENT

This work is supported by the National Science Foundation of China (NSFC) with granting No. 61573350.

REFERENCES

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

[2] J. F. Henriques, C. Rui, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *European Conference on Computer Vision*, 2012, pp. 702–715.

[3] ——, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[4] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," vol. 8926, pp. 254–265, 2014.

[5] M. Danelljan, G. Hger, and F. S. Khan, "Accurate scale estimation for robust visual tracking," in *British Machine Vision Conference*, 2014, pp. 65.1–65.11.

[6] D. Huang, "Enable scale and aspect ratio adaptability in visual tracking with detection proposals," in *British Machine Vision Conference*, 2015.

[7] M. Danelljan, G. Hger, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 8, pp. 1561–1575, 2017.

[8] R. M. Gray, *Toeplitz And Circulant Matrices: A Review (Foundations and Trends(R) in Communications and Information Theory)*. Now Publishers Inc., 2006.

[9] R. Least-squares, R. Rifkin, G. Yeo, and T. Poggio, "Regularized least-squares classification," vol. 190, 06 2003.

[10] D. Forsyth, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[11] M. Danelljan, F. S. Khan, M. Felsberg, and J. V. D. Weijer, "Adaptive color attributes for real-time visual tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1090–1097.

[12] Y. Wu, J. Lim, and M. H. Yang, "Online object tracking: A benchmark," in *Computer Vision and Pattern Recognition*, 2013, pp. 2411–2418.