LETTER

Twofold Correlation Filtering for Tracking Integration*

Wei WANG[†,††a)], *Student Member*, Weiguang LI[†,††], Zhaoming CHEN[†], *and* Mingquan SHI[†b)], *Nonmembers*

SUMMARY In general, effective integrating the advantages of different trackers can achieve unified performance promotion. In this work, we study the integration of multiple correlation filter (CF) trackers; propose a novel but simple tracking integration method that combines different trackers in filter level. Due to the variety of their correlation filter and features, there is no comparability between different CF tracking results for tracking integration. To tackle this, we propose twofold CF to unify these various response maps so that the results of different tracking algorithms can be compared, so as to boost the tracking performance like ensemble learning. Experiment of two CF methods integration on the data sets OTB demonstrates that the proposed method is effective and promising.

key words: *object tracking, correlation filter, end-to-end represent learning, complementary features, trackers integration*

1. Introduction

Recent survey of trackers shows that single tracking method may be not effective enough to tackle complicated background and fast variations of target appearance [1], [2]. Many improvements like multiple tracking algorithm integrating and multiple features fusing have been proposed for the CF based tracking method. As the use of features integration, Staple [3] proposes complementary method that is inherently robust to both color changes and deformations. Ma [4] *et al*. explores the hierarchies of different CNN layers and interpret them as a nonlinear counterpart of an image pyramid representation for tracking. Danelljan [5] *et al*. utilize the fusing of hierarchical multiple features with the features dimension reduction to improve the performance and speed. In [6] the authors propose a novel visual tracking sampler that can robustly handle in challenging scenarios. STRCF [7] handles boundary effects with less loss in efficiency and achieves remarkable performance by incorporating both temporal and spatial regularization. DRT [8] jointly models the discrimination and reliability information to reduce the tracking model degradation caused by the unexpected salient regions on the feature map. LCT [9] trains an online random fern classifier to re-detect objects in case of Correlation tracking failure. A multiple tracker [2] combines KCF and TLD based on complementary measure with strategic model updates, which takes advantages of both and outperforms them.

Despite these above improvement, there are also some drawbacks need to be mentioned. On one hand, simple fusing of multiple features may not really integrate the advantage of both individual features. On the other hand, complex tracking method integration suffers from relatively expensive computing costs, which make a hurdle to improve the ensemble based visual tracking. In this work, we simplify the complex tracking fusing task; propose a novel but simple twofold CF method to dig out the strengths of different tracking methods and complementary features. Our method makes a unified way in filter level not only for different features fusing but also for different tracking methods integrating. Moreover, to balance the tracking speed of fusing complementary features and various tracking method, we choose two typical trackers with equivalent tracking speed, named CFnet [10] and Staple [3]. Our twofold CF method integrates the two CF trackers with complementary features and various tracking tricks, which run at 41 fps and outperform both the algorithms. Noted that the proposed method is flexible and promising for the integration of other tracking method.

The contributions of this paper can be summarized as follows:

- Selection of two state-of-the-art tracking methods with complementary performance in handling tracking challenges.
- A simple but efficient twofold CF method is proposed for the integrating of different CF trackers, which unifies incompatible CF response maps to adapt relevant better tracking result.

2. The Proposed Method

2.1 Correlation Filter Tracking

The CF model is trained through dense sampling from an image patch, which uses Fast Fourier Transform (FFT) to improve the computational efficiency. The appearance of a target object is given by using a filter w trained on an image patch x of M × N pixels. The correlation filter w can be learned by minimizing the ridge regression loss:

$$w^* = argmin \sum_{m,n} ||w \cdot \varphi(x(m,n)) - \mathrm{y(m, n)}||^2 + \lambda ||w||^2 \quad (1)$$

Manuscript received May 9, 2018.
Manuscript revised June 26, 2018.
Manuscript publicized July 10, 2018.

[†] The authors are with Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, China.

[††] The author is with University of Chinese Academy of Sciences, China.

[*] This work was supported by National Nature Science Foundation of China (Grant No: 61605205).

a) E-mail: wangwei1@cigit.ac.cn
b) E-mail: shimq@cigit.ac.cn

DOI: 10.1587/transinf.2018EDL8100

Where λ is a regularization parameter for reducing overfitting, $\cdot$ is the inner product symbol; φ is the mapping to a kernel space. The ideal response $y \in R^{\wedge(M\times N)}$ is given by the following equation:

$$y(\mathrm{m},\mathrm{n}) = e^{-\frac{(m-M/2)^2+(n-N/2)^2}{2\sigma^2}} \tag{2}$$

where σ is the kernel width. All the circular shifts of x(m,n), where $(\mathrm{m},\mathrm{n})\in\{0,1,\cdots,\mathrm{M}-1\}\times\{0,1,\cdots,\mathrm{N}-1\}$, are generated as training samples with Gaussian function label $\mathrm{y}(\mathrm{m},\mathrm{n})$. Using FFT to compute this problem, this objective function can be identically expressed as $w = \sum_{m,n}\alpha(m,n)\varphi(m,n)$, the coefficient a is defined as:

$$\mathrm{a} = F^{-1}\left(\frac{F(y)}{F(\varphi(x)\cdot\varphi(x))+\lambda}\right) \tag{3}$$

where F denotes the discrete Fourier operator. In the tracking process, patch z with the same size as x is cropped from the new frame image. The response map is calculated by:

$$\hat{f} = F^{-1}(F(\mathrm{a})\odot F(\varphi(z)\cdot\varphi(\hat{x}))) \tag{4}$$

Where $\hat{f}$ means $\hat{f}$ = FFT(x) as well as other symbols. The target can be located by searching for the position of maximum value of the correlation response map.

2.2 Overview of Staple

Staple merges two CF model respectively using template-based feature histogram features as complementary learners. Staple propose a linear combination of template and histogram scores function:

$$f(x) = \gamma_{tmpl}f_{tmpl}(x) + \gamma_{hist}f_{hist}(x) \tag{5}$$

where the subscript *tmpl* and *hist* denote the variables of the template learner the histogram learner respectively. These two response map are proposed by learning two independent rigid-regression models with complementary features:

$$\begin{aligned} h_t &= argmin_h\left\{L_{tmpl}(h;X_t)+\frac{1}{2}\lambda_{tmpl}\|h\|^2\right\} \\ \beta_t &= argmin_\beta\left\{L_{hist}(\beta;X_t)+\frac{1}{2}\lambda_{hist}\|\beta\|^2\right\} \end{aligned} \tag{6}$$

where h and β indicates the parameters of the correlation filter and the color model respectively. The function L(•) represents a L2 loss.

For the target translation and various scales, Staple search only in a region around the previous location during search as well as for training.

2.3 Overview of CFnet

CFnet integrates the CF learner as a differentiable layer in a deep CNN network and trains lightweight CNN features to achieve state-of-the-art tracking performance. Such integrating makes errors be propagated through the CF back to the CNN features. There has closed-form expression for the derivative of the Correlation Filter which can integrate the correlation filters in an end-to-end network.

For the CF learner integration, CFnet use a fully-convolutional Siamese framework to learn the feature maps of search patch $\varphi(\mathrm{z})$ with a Correlation Filter block, which has formalized process:

$$\begin{aligned} & L(\theta) = \left\|R(\theta)-\tilde{R}\right\|^2+\gamma\|\theta\|^2 \\ & \qquad s.t. \quad R(\theta) = \sum_{l=1}^{d}\varphi(z,\theta)*f \\ & f = F^{-1}(F(\mathrm{a})\odot F(\varphi(z)\cdot\varphi(\hat{x}))) \end{aligned} \tag{7}$$

where is the ideal response generated around the real target location by Gaussian distribution. The back-propagation of loss about the image pair $\varphi(\mathrm{x})$ and $\varphi(\mathrm{z})$ are formulated as:

$$\begin{aligned} \frac{\partial L}{\partial\varphi(x)} &= F^{-1}\left(\frac{\partial L}{\partial(\hat{\varphi}(x))^*}+\left(\frac{\partial L}{\partial\hat{\varphi}(x)}\right)^*\right) \\ \frac{\partial L}{\partial\varphi(z)} &= F^{-1}\left(\frac{\partial L}{\partial(\hat{\varphi}(z))^*}\right) \end{aligned} \tag{8}$$

The correlation filters can be formulated as a CF layer in network by derived this back-propagation. The back-propagation in correlation filter layer still can be computed in Fourier frequency domain so that the CNN connected a CF layers apply the offline training on large-scale datasets. Since this ultra-lightweight CNN feature has approximately the same number of parameters as hand-crafted features, we use the two kinds of trackers to improve the whole tracking performance.

2.4 Twofold CF Method for Tracking Method Integration

Single tracker may have not enough power to handle all the tracking challenges and effective tracking methods integration may significantly improve the tracking performance [2]. Different from previous works [5], [9] which directly fuse CNN, hog and color name features as single model to improve the tracking performance, we propose a simple hybrid method that works on filter level to integrate the advantage of multiple features and various tracking method. Our proposed method considers the results of both two trackers and selects the best one by measuring the max response map value that depends on correlation with the stored object model sample patches. The proposed tracking architecture consists of two base trackers, and is simple but effective to integrate the advantage of the two trackers.

The key idea of our method is the twofold CF process, which takes the result of one tracker as the previous frame of another tracker. For the integration of the two trackers, our method has two parallel streams and obtains the final results by integrating the response map of these two streams. Firstly, each base tracker is working on its own line and makes their individual response map. Subsequently, we get the twofold response map based on the result of CF1 by us-

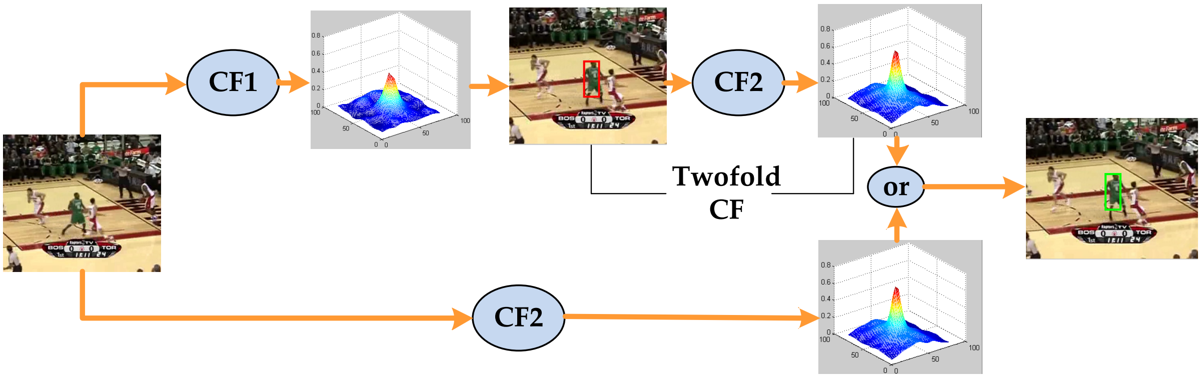

Fig. 1 Flow chart of the proposed method for one frame progress: (the final result is obtained by choosing the response map with higher value.)

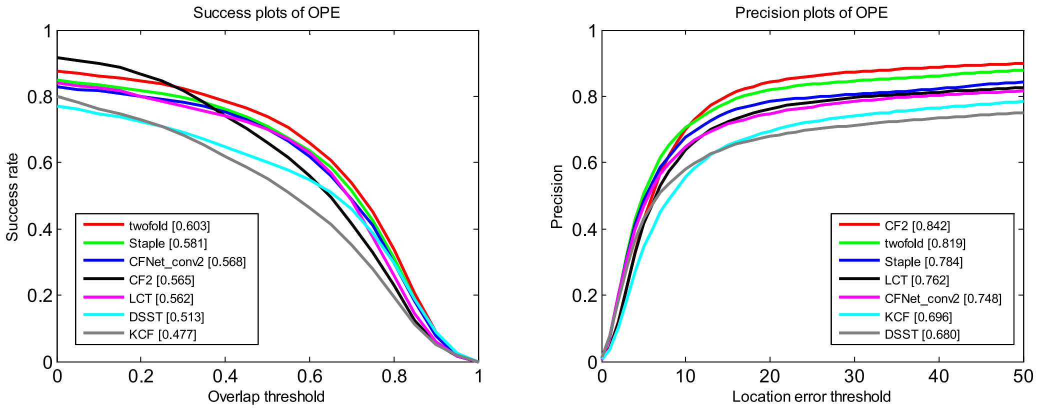

Fig. 2 The precision and success plots of OPE on the OTB2015 benchmark.

ing the filter of CF2. As shown in Fig. 1, two unified response maps are acquired for the tracking results comparison. Notice that CF1 and CF2 is complementary each other and can be exchanged.

Denote $w1$, $w2$ as the Correlation filter parameters of CF1 and CF2 respectively, the twofold CF process can be defined by rewriting (1):

$$w^* = argmin \sum_{m,n} ||w2 \cdot \left(w1 \cdot \emptyset(x(m,n))\right) - \mathrm{y(m,n)}||^2 + \lambda||w||^2 \quad (9)$$

Where the whole filter $w = w1 \cdot w2$ can be obtained by solving (1).

In general, CF tracker locates the target by finding the max response map value and updates the discriminative model with dense samples generated around the prior results [1]. Thus, we compare the two response map maximum values and choose the better as the final tracking result by:

$$max(R) = max(max(R_{CF1 \cdot CF2}), max(R_{CF2}) \quad (10)$$

Where R denotes the CF response map and can be obtained by (4). By this comparison, optimal tracking results can be found. For the model updating of CF1 and CF2, they use their individual updating strategy respectively. The followed experiment will demonstrate the integrating performance.

3. Experimental Results

3.1 Data Sets and Evaluation Metrics

We evaluate our proposed method on the datasets OTB2015 [11], which has large number (100 sequences) of different sequences and categorizes these sequences with 11 attributes. OTB provides three evaluation metrics: one pass evaluation (OPE), time robustness evaluation (TRE) and spatial robustness evaluation (SRE). Each metrics use Precision plot and Success plot to evaluate the tracker performance. Precision plot is the center location error, which computes the average Euclidean distance between the canter locations of the tracked targets and the manually labeled ground-truth positions of all the frames. Another measure for evaluating trackers is the area under curve (AUC) of success plot, which is the average of the success rates according to the sampled overlap thresholds. Given a tracked bounding box $\mathrm{K_t}$ and the ground-truth bounding extent $\mathrm{K_0}$ of a target object, the overlap score is defined as:

Table 1 The AUC score of OPE success plots of the compared trackers on OTB 2015: 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). Red: best Blue: second best.

	overall	IV	OPR	SV	OCC	DEF	MB	FM	IPR	OV	BC	LR
twofold	60.3	60.5	58.3	58.5	58.4	55.9	57.4	57.7	58.4	57.4	60.1	55.7
Staple	58.1	59.8	53.4	56.3	56.8	56.4	54.6	53.7	55.2	58.1	57.4	54.6
CFNet	56.8	54.4	54.2	55.9	57.1	54.2	54.8	55.9	56.8	57.1	55.9	56.1
CF2	56.5	54.9	54	54.1	55.3	53.9	58.5	57	56.6	51.4	58.5	52.8
LCT	56.2	56.6	53.8	53.8	53.7	51.7	53.3	53.4	55.7	55.2	55	48.8
DSST	51.2	55.8	47	51.2	49.1	49.1	46.9	44.7	50.2	48.6	52.3	47
KCF	47.7	47.9	45.3	42.4	46.7	43.9	45.9	45.9	46.9	49.3	49.8	39.1

$$\varphi_K = \left| \frac{K_0 \cap K_t}{K_0 \cup K_t} \right| \tag{11}$$

where $\cap$ and $\cup$ are regional intersection and union operation, while $|*|$ is definite for the number of pixels in the frame.

3.2 Performance Comparison

Our tracker is compared with recent proposed trackers including CF2 [4], Staple [3], CFnet [10], LCT [9], DSST [12] and KCF [1] on the main metric OPE of OTB2015. Among these compared trackers, CF2 and CFnet use deep CNN features and the others use hand-crafted features. For the tracking integration in this implementing, CFnet and Staple are regarded as CF1 and CF2 respectively. As shown in Fig. 2 (a), our proposed tracker framework achieves top rank and the remarkable performance with a large margin in the success tracking plots. From Fig. 2 (b) we can see that the precision rate of our approach is just below CF2, which does not runs at real-time speed and has a comparably low success rate. Meanwhile, it can be point out that our tracking method achieves expected performance and outperforms other trackers which include deep feature based trackers and correlation filter based trackers. Specifically, our method achieves a success score 0.603 and a precision score 0.819. Compared with the baselines Staple and CFnet, the success and precision improve from {3.7%, 7%} to {4.1%, 9.4%}, respectively.

We also show success rate of OPE performances on each attribute in Table 1, which gives the OPE performances of the ranked trackers on the 11 challenge attributes. The proposed tracker lies in the best or the second best rank line among all of the trackers compared, and the performance of different attribute demonstrates that our method clearly outperform against other trackers on accurate and robust. This superiority benefits from the twofold CF and optimization mechanism of the proposed method.

4. Conclusions

In this paper, we proposed a simple yet robust tracking integration method formed by ingeniously combining the two state-of- the-art trackers, which have complementary performance in handling tracking challenges. The proposed method runs at filter lever and unifies incompatible response map to refine the tracking results. Our proposed tracker not only has superior performance, but also runs at a fast speed which is enough for real-time applications.

References

[1] 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, Aug. 2015.

[2] M.K. Rapuru, S. Kakanuru, P.M. Venugopal, D. Mishra, and G.R.K.S. Subrahmanyam, "Correlation-Based Tracker-Level Fusion for Robust Visual Tracking," IEEE Transactions on Image Processing, vol.26, no.10, pp.4832–4842, April 2017.

[3] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P.H.S. Torr, "Staple: Complementary Learners for Real-Time Tracking," IEEE Conference on Computer Vision and Pattern Recognition, vol.38, no.2, pp.1401–1409, 2016.

[4] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical Convolutional Features for Visual Tracking," IEEE International Conference on Computer Vision, pp.3074–3082, 2015.

[5] M. Danelljan, G. Bhat, F.S. Khan, and M. Felsberg, "ECO: efficient convolution operators for tracking," IEEE Conference on Computer Vision and Pattern Recognition, pp.6931–6939, 2017.

[6] J. Kwon and K.M. Lee, "Tracking by sampling and integrating multiple trackers," IEEE Trans. Pattern Analysis & Machine Intelligence, vol.36, no.7, pp.1428–1441, 2014.

[7] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, "Learning spatial-temporal regularized correlation filters for visual tracking," IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[8] C. Sun, D. Wang, H. Lu, and M.-H. Yang, "Correlation tracking via joint discrimination and reliability learning," IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[9] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term Correlation Tracking," IEEE Conference on Computer Vision and Pattern Recognition, pp.5388–5396, 2015.

[10] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P.H.S. Torr, "End-to-end representation learning for Correlation Filter based tracking," IEEE Conference on Computer Vision and Pattern Recognition, pp.5000–5008, 2017.

[11] Y. Wu, J. Lim, and M.-H. Yang, "Object Tracking Benchmark," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.37, no.9, pp.1834–1848, Sept. 2015.

[12] M. Danelljan, G. Häger, F.S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," British Machine Vision Conference, vol.65, pp.1–11, 2014.