PAPER

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

Jiatian PI[†a)], Keli HU[††b)], Yuzhang GU[†], Lei QU[†], Fengrong LI[†], *Nonmembers*, Xiaolin ZHANG[†], *Member*, *and* Yunlong ZHAN[†], *Student Member*

SUMMARY Visual tracking has been studied for several decades but continues to draw significant attention because of its critical role in many applications. Recent years have seen greater interest in the use of correlation filters in visual tracking systems, owing to their extremely compelling results in different competitions and benchmarks. However, there is still a need to improve the overall tracking capability to counter various tracking issues, including large scale variation, occlusion, and deformation. This paper presents an appealing tracker with robust scale estimation, which can handle the problem of fixed template size in Kernelized Correlation Filter (KCF) tracker with no significant decrease in the speed. We apply the discriminative correlation filter for scale estimation as an independent part after finding the optimal translation based on the KCF tracker. Compared to an exhaustive scale space search scheme, our approach provides improved performance while being computationally efficient. In order to reveal the effectiveness of our approach, we use benchmark sequences annotated with 11 attributes to evaluate how well the tracker handles different attributes. Numerous experiments demonstrate that the proposed algorithm performs favorably against several state-of-the-art algorithms. Appealing results both in accuracy and robustness are also achieved on all 51 benchmark sequences, which proves the efficiency of our tracker.

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

1. Introduction

Visual object tracking is one of the most fundamental tasks in the field of computer vision and is related to a wide range of applications like surveillance and robotics. 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.

The process of visual tracking could be described as a dynamic state estimation problem, and the state information is usually the appearance representation. There exist two main approaches to handle visual tracking, namely generative and discriminative methods. The generative methods tackle the problem by searching for regions that are most similar to the target model. The models in these methods are either based on templates or subspace models. The discriminative approaches aim at differentiating the target from the background by posing tracking as a binary classification problem. To cope with natural image changes, the classifier is typically trained with translated and scaled sample patches. Unlike generative methods, discriminative approaches use both target and background information to find a decision boundary for differentiating the target object from the background. This is employed in tracking-by-detection methods, which have shown to provide excellent tracking performance.

Recent benchmark [1]–[3] studies show that the top-performance trackers are usually discriminative trackers or hybrid ones. Canonical examples of the tracking-by-detection paradigm include those based on Support Vector Machines (SVM) [4], Random Forest classifiers [5], or boosting variants [6], [7]. Zhang et al. [8] propose a projection to a fixed random basis, to train a Naive Bayes classifier, inspired by compressive sensing theory. Tracking-Learning-Detection (TLD) tracker [9] exploits a set of structural constraints with a sampling strategy using boosting classifier. Among more complicated trackers, recently proposed correlation filter based trackers [10]–[14] 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 [10], Circulant Structure Kernels (CSK) tracker [11], and KCF tracker [14], are limited to only estimating the target position with the fixed size. Danelljan et al. [12] extend the CSK tracker with color attributes to better represent the input data, which have shown to obtain superior performance due to their good balance between photometric invariance and discriminative power. Although those trackers [10], [12], [14] achieve very appealing performance in terms of accuracy and robustness, they have insufficient scale variation. In addition, Discriminative Scale Space Tracker (DSST) [11] 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 [3]. However, there is still room for improvement in translation estimation in the DSST.

In this paper, we incorporate the proposed scale estimation approach in the DSST into the KCF tracker with-

Manuscript received November 12, 2015.
Manuscript revised February 21, 2016.
Manuscript publicized April 7, 2016.

[†] The authors are with SIMIT, Shanghai, 200050 China.
[††] The author is with Shaoxing University, Shaoxing, 312000 China.
a) E-mail: pijiatian@126.com
b) E-mail: ancimoon@gmail.com

DOI: 10.1587/transinf.2015EDP7459

out much computational overhead. Compared to the traditional DSST and the KCF tracker, the improved algorithm is more robust and so can deal with more challenging scenarios. The key contributions of this work can be summarized as follows. Firstly, we extend the KCF tracker with the capability of handling scale changes. 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 51 sequences in the recent benchmark evaluation [1]. Experimental results show that the proposed tracker achieves outstanding performance both in accuracy and robustness against the state-of-the-art trackers.

The rest of this paper is organized as follows. Section 2 reviews related trackers based on correlation filter techniques. Section 3 introduces the proposed tracker. Section 4 presents experimental results on different sequences. Finally, the conclusion and our future work are summarized in Sect. 5.

2. Related Work on Correlation Filter-Based Trackers

Conventionally, correlation filters are designed to produce correlation peaks for each interested target in the scene while yielding low responses to background, which are usually used as detectors of expected patterns. Correlation filter-based trackers model the appearance of objects using filters trained on example images. The target is initially selected based on a small tracking window in the first frame and tracked by correlating the filter over a search window in the next frame. The location corresponding to the maximum value in the correlation output indicates the new position of the target. An online update is then performed based on that new location. More modern approaches such as Average of Synthetic Exact Filters (ASEF) [15] introduce a method of tuning filters for particular tasks. Although ASEF has shown to perform well in eye localization [15] and pedestrian detection [16], a large number of samples are required for training, which makes it too slow for online visual tracking. David S. Bolme et al. [10] propose the MOSSE filter, which produces ASEF-like filters from fewer training images. Based on the basic framework of the MOSSE filter, numerous improvements have been made later. The CSK tracker [13] improves the MOSSE filter by introducing kernel methods. The KCF tracker [14] extends the CSK tracker by making use of the circulant structure within training samples, which achieves high speed. Moreover, the KCF tracker enhances the conventional correlation filters with kernel trick and supports multi-channel features, while the scale problem remains unresolved. By further handling the scale changes, the Scale Adaptive Multiple Features (SAMF) tracker [17] and the DSST [11] have achieved state-of-art results. They have beaten all other attended trackers in terms of accuracy in the competition [3]. The SAMF tracker, as an extended version of the KCF tracker, handles scale changes by sampling with several predefined scale perturbations. The correlation filter is then applied to those samples individually to find out the best scale and target position. Moreover, powerful features including Histogram of Gradient (HOG) and color-naming are integrated together to further boost the overall tracking performance. The DSST proposes an efficient method for estimating the target scale by training a classifier on a scale pyramid, which allows to independently estimate the target scale after the optimal translation is found. Compared to an exhaustive scale space search scheme, the DSST provides improved performance while being computationally efficient by learning discriminative correlation filters for estimating translation and scale independently. With more correlation filter-based trackers [18]–[21] developed recently, correlation filter-based tracking model has proven its great strengths in efficiency and robustness, and has considerably accelerated the development of visual object tracking.

3. The Proposed Tracker

In this section, we give the details of our proposed tracker. In order to incorporate the scale estimation approach in the DSST into the KCF tracker, we decompose the task into translation and scale estimation of objects. Section 3.1 presents the translation estimation based on the KCF tracker for its competitive performance and efficiency. In Sect. 3.2, the scale estimation is carried out by learning a discriminative correlation filter applied in the DSST. At last, Sect. 3.3 provides a brief outline of our proposed tracker and discusses the differences in our proposed tracker compared to the KCF tracker and DSST.

3.1 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. [14] 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:

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

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:

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

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 [14]. 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:

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

where $\hat{\mathbf{x}}$ denotes 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.

3.2 Scale Estimation with Discriminative Correlation Filter

The Kernelized Correlation Filter (KCF) in Sect. 3.1 is used for estimating the translation, then we can find the accurate position of the target without scale change. According to the DSST [11], the proposed discriminative correlation filter for scale estimation is generic and can be incorporated into any tracking framework. In addition, this discriminative correlation filter is closely related to the MOSSE filter [10], which produces stable correlation filters when trained on a small number of image windows. Consequently, the discriminative correlation filter is an efficient and ideal approach for robust scale estimation. After finding the accurate position with the KCF tracker, we apply the discriminative correlation filter for scale estimation. Firstly, the MOSSE filter needs 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 obtain a robust approximation, Danelljan et al. [11] update the numerator A_t^l and denominator B_t^l of the correlation filter $\hat{h}_t^l$ at time step t separately as follows:

$$\begin{aligned} A_t^l &= (1-\eta)A_{t-1}^l + \eta \hat{g}_t^* \hat{f}_t^l \\ B_t^l &= (1-\eta)B_{t-1}^l + \eta \textstyle\sum_{k=1}^d \hat{f}_t^{k*} \hat{f}_t^k, \end{aligned} \tag{10}$$

where η is a learning rate parameter. Considering a three-dimensional scale space correlation filter, the filter size is fixed to $M \times N \times S$, where M and N are the height and the width of the filter, respectively. S is the number of different scales. After finding the accurate position by the conventional KCF tracker, we extract a three-dimensional scale sample Z with the size $M \times N \times S$. The correlation scores $\mathbf{y}_{scale}$ between scale filter and sample are then computed as follows:

$$\mathbf{y}_{scale} = F^{-1}\left(\frac{\sum_{l=1}^S A^{l*} Z^l}{B + \lambda}\right). \tag{11}$$

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

3.3 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 Kernelized Correlation Filter (KCF) is only applied for translation estimation and the discriminative correlation filter cooperates on scale estimation. Unlike our tracker, the DSST [11] uses separate filters for translation and scale estimation, which are all based on discriminative correlation filters. In addition, we

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 $A_{t-1}^{scale}$ and $B_{t-1}^{scale}$.
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 $z_{scale}$, $A_{t-1}^{scale}$ and $B_{t-1}^{scale}$.
  • 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 $A_t^{scale}$ and $B_t^{scale}$.
5: Output:
  • Estimated target position $p_t$ and scale $s_t$.
  • Updated the translation model $X_t^{trans}$, $\alpha_t^{trans}$ and scale model $A_t^{scale}$, $B_t^{scale}$.
```

extract translation sample with fixed size to find the target position without considering the scale, whereas the DSST extracts translation sample according to 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.

4. Experiments

In this section, our proposed algorithm is evaluated with other 13 state-of-the-art methods on 51 challenging sequences. The compared trackers include other correlation filter-based trackers, such as SAMF tracker [17], DSST [11], CSK tracker [13] and KCF tracker [14]. Moreover, the top five trackers reported in the recent benchmark [1] are compared in the experiments, e.g., Structure tracker (Struck) [24], Sparsity-based Collaborative Model (SCM) tracker [25], TLD [9], Adaptive Structural Local Appearance (ASLA) tracker [22] and Context Tracker (CXT) [27]. The other compared trackers are L1 Accelerated Proximal Gradient (L1APG) tracker [23], Incremental Learning Tracker (IVT) [28], Distribution Fields Tracker (DFT) [29] and Compressive Tracking (CT) tracker [8]. In most case, we use the corresponding ground truth files, the compared code library in the benchmark [1]. However, the SAMF tracker, DSST and the KCF tracker are proposed after the benchmark. Thus, we utilize the source code released by these three trackers to test benchmark sequences. For each tracker, we use the default parameters which are tuned well by the authors to evaluate all sequences. The proposed algorithm runs at 70 frame per second (FPS) with a matlab implementation on an Intel Xeon(R) E5-2650 2 core 2.00 GHz CPU with 64 GB RAM without any optimizing.

4.1 Experiment Setup and Evaluation Criteria

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 0.25. In addition, we use Principal Component Analysis Histogram of Gradient (PCA-HOG) [30] for target representation. The cell size of HOG is 4×4 and the orientation bin number of HOG is 9. In order to get fair experimental results, all the parameters are kept constant for the following experiments.

We use two metrics to evaluate the performance. The first one is the precision plot which is based on the location error. The other one is the success plot which is based on the overlap rate. The location error is defined as the average Euclidean distance between the center locations of the tracked targets and the manually labeled ground truths. The precision plot shows the percentage of the frames whose tracking location is within the given threshold distance of the ground truth. To compare the performance of different trackers, the results at error threshold of 20, as well as in the benchmark [1], are used to ranking in the precision plot. Another evaluation metric is the overlap rate of the bounding box. Given the tracked bounding box r_t and the ground truth bounding box r_a, the overlap rate is defined as $S = |\frac{r_t \bigcap r_a}{r_t \bigcup r_a}|$, where $\bigcap$ and $\bigcup$ represent the intersection and union of two regions, respectively, and $|\cdot|$ denotes the number of corresponding pixels. The success plot shows the ratios of successful frames while the overlap thresholds vary from 0 to 1. We use the area under the curve (AUC) of each success plot to rank the tracking algorithms. To analyze the robustness to initialization, each sequence is partitioned into 20 segments and each tracker is performed on around 310,000 frames. This evaluation metric is referred as temporal robustness evaluation (TRE) in the benchmark [1].

4.2 Robust Scale Estimation

We use the 28 sequences [1] annotated with "scale variation" to evaluate the scale adaptability of our proposed algorithm. Precision plots and success plots with temporal robustness

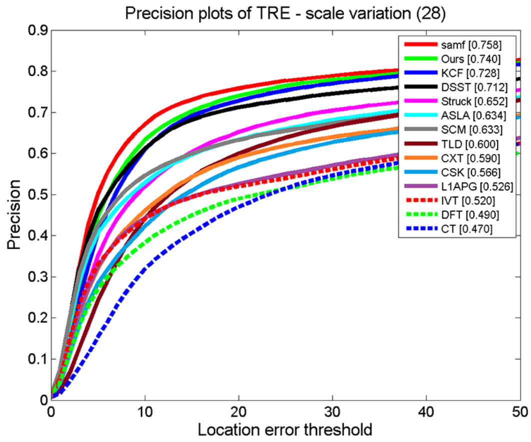


Fig. 1 Precision plots over all 28 sequences annotated with scale variation. The results at error threshold of 20 are used to ranking as shown in the top right corner.

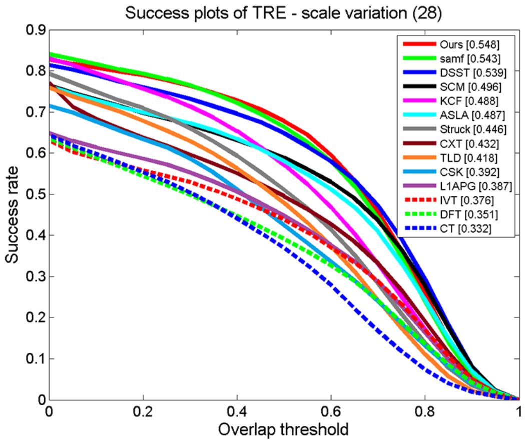
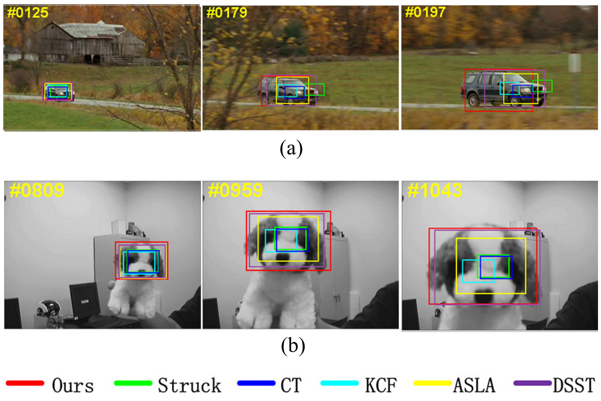


Fig. 2 Success plots over all 28 sequences annotated with scale variation. The AUC scores of each plot are used to ranking as shown in the top right corner.



(a)

(b)

Fig. 3 Performance on (a) ‘carScale’ and (b) ‘dog1’ sequences by 6 trackers. The sequences include scale variation at a different level. Our algorithm results are marked with red line as shown in the up panel.

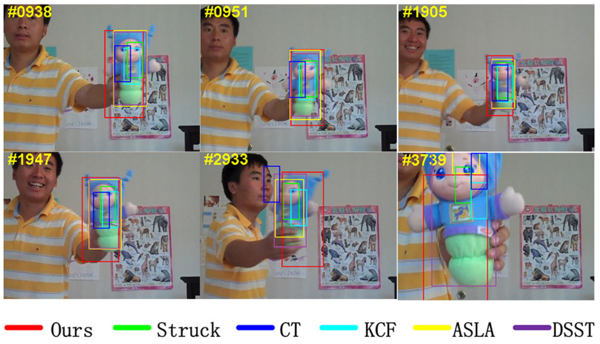


Fig. 4 Performance on the ‘doll’ sequence by 6 trackers. The sequence includes abrupt scale variation. Our tracker performs inaccurately as shown in the second row.

evaluation (TRE) are shown in Fig. 1 and Fig. 2, which show our tracker is superior compared to other trackers. Due to the success plot represents the overlap score between the tracked bounding box and the ground truth bounding box, the scale adaptability of trackers can be presented excellently. Experimental results show that our tracker achieves 54.8% on the AUC score, which is 6% improvement over the KCF tracker. So the discriminative correlation filter can be indeed incorporated into the KCF tracker framework to improve the scale estimation. Furthermore, experimental results show that our tracker is 0.9% improvement over the DSST. 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 Fig. 4. 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.

4.3 Experiments with Sequence Attributes

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 10 attributes in the benchmark [1] to evaluate how well the tracker handles different attributes. The name of the attributes are listed as follows: fast motion (FM), 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 10 attributes. Moreover, the results at error threshold of 20 in precision plots are presented in Table 3. Our tracker achieves second-best performance to 8 of the 10 attributes. The intuitive illustration is shown clearly in Fig. 5. Because the SAMF tracker combines the HOG feature and color-naming to represent the target, the tracking performance is superior to our method. However,

Table 2 The AUC scores of success plots in 10 attributes. The best result is highlighted in bold type and the second result is highlighted in red color.

Attris	**Ours**	**KCF**	**DSST**	**SAMF**	**TLD**	**Struck**	**SCM**	**CSK**	**ASLA**	**CXT**	**IVT**	**DFT**	**CT**	**L1APG**
FM	48.6	45.5	45.1	**51.6**	39.2	46.4	28.2	33.0	28.5	37.5	22.8	34.9	29.9	31.9
MB	51.4	49.2	48.4	**54.3**	38.8	48.5	29.0	33.6	29.6	38.5	22.5	37.4	28.0	30.1
DEF	59.4	57.0	57.1	**65.7**	42.5	50.0	51.5	44.0	46.6	39.4	38.7	47.2	42.4	41.2
IPR	55.3	51.9	54.8	**56.5**	40.6	47.3	45.3	42.6	45.1	45.3	35.2	40.0	32.8	40.5
OCC	57.4	54.6	56.0	**61.5**	42.6	46.2	50.2	42.0	44.4	41.0	37.7	41.1	36.3	40.2
OPR	56.1	53.0	54.3	**58.7**	42.5	47.7	48.0	43.0	46.5	44.4	37.4	41.6	34.9	40.4
OV	52.9	53.8	50.0	**58.4**	43.4	41.7	34.4	32.8	32.5	40.3	26.9	30.7	32.9	30.2
IV	55.4	52.7	56.5	**59.1**	40.2	48.6	47.5	43.5	46.8	40.3	35.3	38.5	35.6	37.3
BC	57.2	56.4	53.1	**57.3**	37.2	47.8	46.9	43.2	44.5	37.4	34.1	41.7	34.0	39.8
LR	44.0	38.3	44.2	43.9	29.9	**45.6**	30.4	36.7	27.8	31.3	26.3	28.7	19.5	36.0

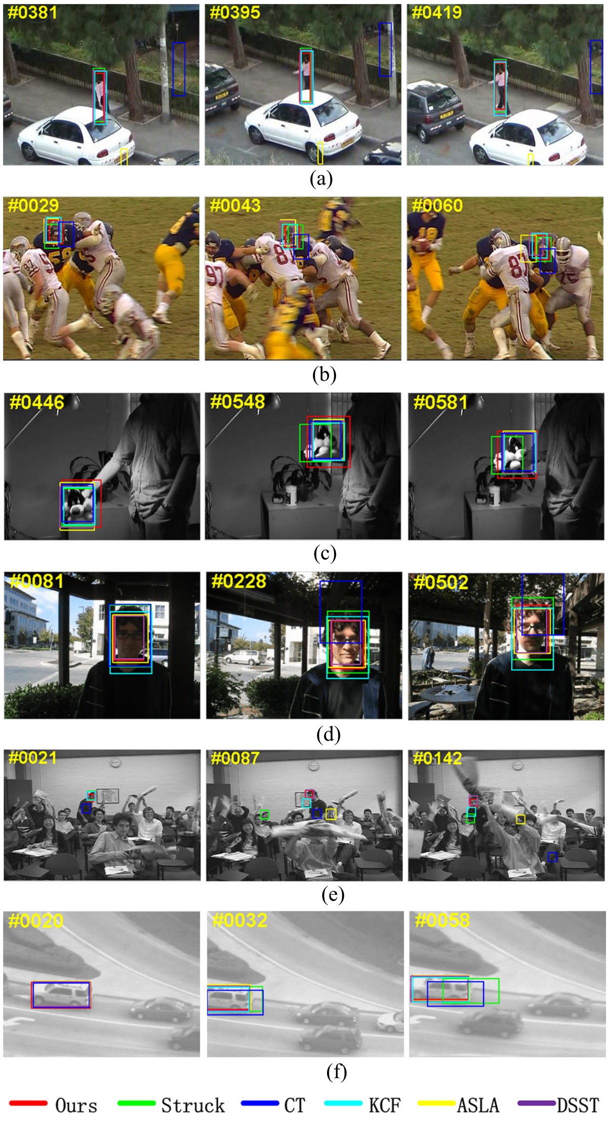


Fig. 5 Performance on 10 attributes by 6 trackers. We select 6 sequences to illustrate the performance. The different attributes including in 6 sequences are listed as follows: (a) ‘DEF’+‘OCC’; (b) ‘FM’+‘BC’; (c) ‘IPR’+‘OPR’; (d) ‘IV’; (e) ‘LR’+‘OCC’; (f) ‘OV’+MB.

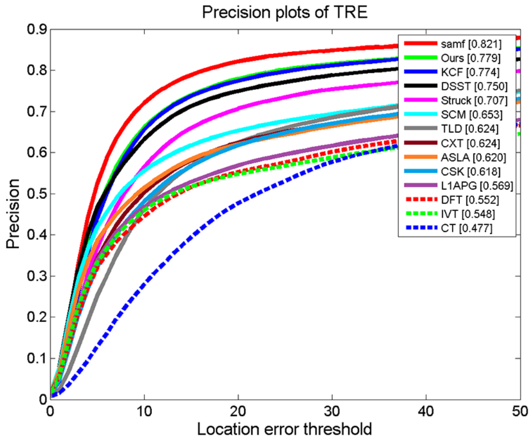


Fig. 6 Precision plots over all 51 benchmark sequences. Results at error threshold of 20 are used to ranking as shown in the top right corner.

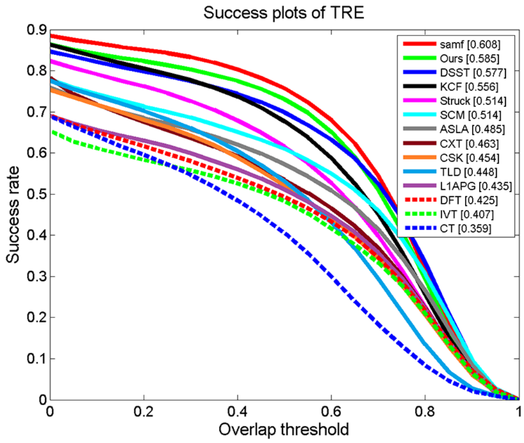


Fig. 7 Success plots over all 51 benchmark sequences. AUC scores of each plot are used to ranking as shown in the top right corner.

the speed of the SAMF tracker is 14.0 FPS on average due to the complicated feature representation. Our tracker uses only the HOG feature and is more than 5 times faster than the SAMF tracker.

4.4 Experiments on the Whole Dataset

To further evaluate the robustness and efficiency of our tracker, we set up a comparison on the whole dataset. Results are shown in Fig. 6 and Fig. 7, where our tracker outperforms the other trackers except the SAMF tracker. In addition to high accuracy, our tracker runs efficiently at an av-

Table 3 The results at error threshold of 20 of precision plots in 10 attributes. The best result is highlighted in bold type and the second result is highlighted in red color.

Attris	Ours	KCF	DSST	SAMF	TLD	Struck	SCM	CSK	ASLA	CXT	IVT	DFT	CT	L1APG
FM	59.8	57.9	52.3	**64.6**	48.7	58.0	30.2	37.9	30.5	47.9	24.8	41.9	33.6	37.1
MB	64.7	62.7	58.7	**68.3**	49.1	61.7	32.3	39.0	33.2	52.0	26.4	46.8	33.2	35.2
DEF	76.7	75.7	71.9	**85.1**	57.1	65.5	63.5	56.9	57.1	51.5	50.7	58.7	53.2	51.3
IPR	73.8	72.8	72.1	**77.3**	56.9	65.0	58.1	57.4	58.2	61.7	47.9	52.9	42.6	54.0
OCC	76.2	75.8	72.5	**82.8**	57.9	63.1	63.3	56.9	56.0	53.7	50.5	52.8	46.2	52.2
OPR	75.3	74.9	71.5	**80.4**	59.7	66.0	61.7	59.1	60.5	60.1	51.3	55.0	45.8	53.9
OV	63.3	64.3	57.7	**70.4**	48.5	48.4	37.1	33.6	33.9	46.8	28.2	33.7	31.1	32.2
IV	72.6	72.9	72.4	**79.3**	54.3	64.3	58.5	57.5	58.4	53.7	46.6	49.1	44.1	47.4
BC	77.1	**77.6**	69.3	76.4	48.8	62.2	60.0	57.3	57.5	51.5	46.3	52.1	43.0	50.6
LR	58.8	50.2	56.3	57.4	37.6	**62.8**	35.0	48.2	32.5	40.8	32.1	35.1	23.4	44.1

erage speed of 70.0 FPS. As a comparison with scale adaptive correlation filter trackers, our tracker is more than 2.4 times faster than the DSST and 5 times faster than the SAMF tracker. Although the speed of the KCF tracker is 260.0 FPS on average and is faster than ours, it is not able to handle scale changes.

5. Conclusion

In this paper, we propose a robust tracking algorithm which combines the method of discriminative correlation filters with the Kernelized Correlation Filter (KCF) tracker. Our tracker handles the problem of fixed template size in KCF tracker without much decrease in the speed. Experiments on benchmark sequences demonstrated that the proposed algorithm performs favorably in terms of accuracy and robustness. Recently, N. Wang et al. [26] propose that the feature extractor is the most important part of a tracker and the observation model is not that important if the features are good enough. Considering that our tracker uses only the HOG feature, we plan to incorporate more robust features into our tracker in the future.

Acknowledgments

This work is partially sponsored by Shanghai Sailing Program (15YF1414500) and Public Welfare Technology Research Project in Zhejiang Province (2016C31082).

References

[1] Y. Wu, J. Lim, and M.-H. Yang, “Online Object Tracking: A Benchmark,” Proc. IEEE CVPR, Portland, Oregon, pp.2411–2418, June 2013.

[2] A. Smeulders, D. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, “Visual Tracking: An Experimental Survey,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.36, no.7, pp.1442–1468, July 2013.

[3] M. Kristan, R. Pflugfelder, A. Leonarids, and et al., “The Visual Object Tracking VOT2014 Challenge Results,” Proc. European Conf. on Computer Vision, Zurich, Switzerland, pp.191–217, Sept. 2014.

[4] S. Avidan, “Support Vector Tracking,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.26, no.8, pp.1064–1072, Aug. 2004.

[5] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, “Online Random Forests,” Proc. IEEE ICCV Workshops, Kyoto, Japan, pp.1393–1400, Sept. 2009.

[6] B. Babenko, M. Yang, and S. Belongie, “Robust Object Tracking with Online Multiple Instance Learning,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.33, no.8, pp.1619–1632, Aug. 2011.

[7] H. Grabner, C. Leistner, and H. Bischof, “Semi-supervised OnLine Boosting for Robust Tracking,” Proc. European Conf. on Computer Vision, Marseille, France, pp.234–247, Oct. 2008.

[8] 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.

[9] Z. kalal, K. Mikolajczyk, J. Matas, “Tracking-Learing-Detection,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.34, no.7, pp.1409–1422, July 2012.

[10] 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.

[11] M. Danelljan, H. Gustav, F.S. Khan, and M. Felsberg, “Accurate Scale Estimation for Robust Visual Tracking,” Proc. British Machine Vision Conf., Nottingham, England, pp.65.1–65.11, Sept. 2014.

[12] M. Danelljan, F.S. Khan, M. Felsberg, and J.V. Weijer, “Adaptive Color Attributes for Real-time Visual Tracking,” Proc. IEEE CVPR, Columbus, Ohio, USA, pp.1090–1097, June 2014.

[13] 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, pp.702–715, Oct. 2012.

[14] J.F. Henriques, R. Caseiro, P. Martins, and J. Batista, “High-speed Tracking with Kernelized Correlation Filters,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.37, no.3, pp.583–596, March 2015.

[15] D.S. Bolme, B.A. Draper, and J.R. Beveridge, “Average of synthetic exact filters,” Proc. IEEE CVPR, Miami, Florida, USA, pp.2105–2112, June 2009.

[16] D.S. Bolme, Y.M. Lui, B.A. Draper, and J.R. Beveridge, “Simple realtime human detection using a single correlation filter,” in Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop, Miami, Florida, USA, pp.1–8, Dec. 2009.

[17] Y. Li and J. Zhu, “A scale adaptive kernel correlation filter tracker with feature integration,” Proc. European Conf. on Computer Vision, Zurich, Switzerland, pp.254–265, Sept. 2014.

[18] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, “Multistore tracker (muster): A cognitive psychology inspired approach to object tracking,” Proc. IEEE CVPR, Boston, Massachusetts, USA, pp.749–758, June 2015.

[19] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, “Long-term correlation tracking,” Proc. IEEE CVPR, Boston, Massachusetts, USA, pp.5388–5396, June 2015.

[20] T. Liu, G. Wang, and Q. Yang, “Real-time part-based visual tracking via adaptive correlation filters,” Proc. IEEE CVPR, Boston, Massachusetts, USA, pp.4902–4912, June 2015.

[21] Y. Li, J. Zhu, and S.C. Hoi, “Reliable patch trackers: Robust visual

tracking by exploiting reliable patches," Proc. IEEE CVPR, Boston, Massachusetts, USA, pp.353–361, June 2015.

[22] 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.

[23] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real Time Robust L1 Tracker Using Accelerated Proximal Gradient Approach," Proc. IEEE CVPR, Providence, Rhode, USA, pp.1830–1837, June 2012.

[24] S. Hare, A. Saffari, and P.H.S. Torr, "Struck: Structured Output Tracking with Kernels," Proc. IEEE ICCV Workshops, Barcelona, Spain, pp.263–270, Nov. 2011.

[25] W. Zhong, H. Lu, and M.-H. Yang, "Robust Object Tracking via Sparsity-based Collaborative Model," Proc. IEEE CVPR, Providence, Rhode, USA, pp.1838–1845, June 2012.

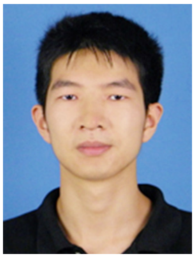
[26] N. Wang, J. Shi, D.-Y. Yeung, and J. Jia, "Understanding and Diagnosing Visual Tracking Systems," 2015 IEEE International Conference on Computer Vision (ICCV), pp.3101–3109, 2015.

[27] T.B. Dinh, N. Vo, and G. Medioni, "Context Tracker: Exploring Supporters and Distracters in Unconstrained Environments," In CVPR, Colorado Springs, CO, USA, pp.1177–1184, June 2011.

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

[29] L. Sevilla-Lara and E. Learned-Miller, "Distribution Fields for Tracking," In CVPR, Providence, Rhode, USA, pp.1910–1917, June 2012.

[30] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.32, no.9, pp.1627–1645, Sept. 2010.



Jiatian Pi received the B.S. degree in Communication Engineering from Chongqing University, Chongqing, China, in 2012. He is currently working towards the Ph.D. degree in Information and Communication Engineering in Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, Chain. His current research includes computer vision, target tracking, machine learning and image processing.

Keli Hu received the B.S. degree in Communication Engineering from Hangzhou Dianzi University, Hangzhou, China, in 2009 and the Ph.D. degree in Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, Chain, in 2014. From May 2014 to the present, he is currently working in the Department of Computer Science and Engineering Shaoxing University, Zhejiang, China. His research interests include artifical intelligence, pattern recognition, computer vision and image processing.

Yuzhang Gu is an associate professor of Shanghai institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, China. He received his Ph.D. degree in Tokyo Institute of Technology. His research focuses on computer vision, target tracking, object detection and 3D video.

Lei Qu received the B.S. degree in Communication Engineering from South China University of Technology, Guangzhou, China, in 2012. He is currently working towards the Ph.D. degree in Information and Communication Engineering in Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, China. His current research includes object detection and recognition, driver assistance and related research.

Fengrong Li is an assistant professor in Shanghai institute of Microsystem and Information Technology (SIMIT), Chinese Academy of Sciences, Shanghai, China. She received her Ph.D. degree in SIMIT. Currently, her research main focuses on the cyber physical system, including collaborative information processing, communication protocols, and information security technology.

Xiaolin Zhang is a professor of Shanghai institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, China. He received his Ph.D. degree in Yokohama National University. His research focuses on visual physiology, robotics, image processing and computer vision.

Yunlong Zhan received the B.S. degree in Electronic Engineering from Shanghai Jiao Tong University, Shanghai, China, in 2011. He is currently working towards the Ph.D. degree in Information and Communication Engineering in Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, China. His current research includes stereo vision, pattern recognition and image processing.