

Visual Target Tracking *via* Online Reliability Evaluation and Feature Selection in the Framework of Correlation Filtering

Li Wei¹, Meng Ding^{2,*}, Yun-Feng Cao³ and Xu Zhang²

¹Jincheng College, Nanjing University of Aeronautics and Astronautics, Nanjing, China; ²School of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing, China; ³School of Astronautics, Nanjing University of Aeronautics and Astronautics, Nanjing, China

Abstract: Background: Although correlation filtering is one of the most successful visual tracking frameworks, it is prone to drift caused by several factors such as occlusion, deformation and rotation.

Objective: In order to improve the performance of correlation filter-based trackers, this paper proposes a visual tracking method *via* online reliability evaluation and feature selection.

Methods: The main contribution of this paper is to introduce three schemes in the framework of correlation filtering. Firstly, we present an online reliability evaluation to assess the current tracking result by using the method of adaptive threshold segmentation of response map. Secondly, the proposed tracker updates the regression model of correlation filter according to the assessment result. Thirdly, when the tracking result based on a handcrafted feature is not reliable enough, we propose a feature selection scheme that autonomously replaces a handcrafted feature used in the traditional correlation filter-based trackers with a deep convolutional feature that can re-capture the target by its powerful discriminant ability.

Results: On OTB-2013 datasets, the Precision rate and Success rate of the proposed tracking algorithm can reach 84.8% and 62.5%, respectively. Moreover, the tracking speed of proposed algorithm is 19 frame per second.

Conclusion: The quantitative and qualitative experimental results both demonstrate that the proposed algorithm performed favorably against nine state-of-the-art algorithms.

Keywords: Visual tracking, correlation filtering, reliability assessment, feature selection, appearance model, generative trackers.

1. INTRODUCTION

Visual target tracking is a promising research direction in computer vision, with its wide range of applications, *e.g.*, intelligent surveillance, robot and unmanned aerial vehicle. Generally, the goal of visual tracking tasks is to predict the locations of a target in the following frames of an image sequence according to the initial location of the target labelled by a bounding box in the first frame.

Existing tracking algorithms can be categorized as either discriminative or generative. The former firstly extracts the target feature, and then uses a classifier achieved from online learning to distinguish the target from backgrounds, such as Compressive Tracking (CT) [1], online Multiple Instance Learning (MIL) [2], and Tracking-Learning-Detection

(TLD) [3]. The latter represents a target by constructing an appearance model, and then searches for the region with maximum likelihood or minimum error in the following frames as a result [4], such as incremental visual tracking (IVT) [5], L1 Tracker Using Accelerated Proximal Gradient Approach (L1APG) [6], and distribution fields for tracking (DFT) [7]. Generally speaking, the discriminative trackers that make the most of background information is more competitive compared with generative trackers. Therefore, the discriminative trackers represented by correlation filtering have generally become mainstream in the past few years. Recently, correlation filter-based trackers have achieved outstanding performances in different benchmarks [8]. Furthermore, related researches show that replacing handcrafted features (*e.g.* histogram of oriented gradient, HOG) with features extracted from a pre-training Convolution Neural Network (CNN) can further enhance the performance of correlation filter-based methods [9-11].

*Address correspondence to this author at the School of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing, China; E-mail: nuaa_dm@hotmail.com



Reviewing the current research status of correlation filter-based trackers, we have the following observations:

(1) A robust tracker needs an effective approach to evaluate the current tracking result. This assessment method can avoid the incorrect tracking result caused by several interference factors containing target occlusion or out of the view to contaminate the update of the regression model and further lead to the error accumulates continuously.

(2) To deal with the stability-plasticity dilemma [12], the correlation filter-based trackers need a reasonable updating scheme of the regression model. Generally, the scheme of continuous model updating used in most correlation filter-based trackers can keep accurately tracking the target even if the appearance of the target changes frequently. However, in the case of occlusion, this scheme may lead to several tracking results without the target contaminates the regression model. On the contrary, trackers without model updating are also prone to drift since the regression model does not learn the variation of target appearance. Therefore, it is necessary to adjust the adaptively update rate of the regression model according to the assessment of the current tracking result.

(3) With the use of deep convolutional features, tracking algorithms become more time-consuming. Although deep convolutional features perform better in complex scenes since their inner semantic information has strong discrimination and strong invariability [11], it is not necessary to use the deep feature with the high computational load when handcrafted features can work well. Therefore, it is crucial to integrate the performance advantages of deep features and the speed advantages of traditional features.

In this paper, we propose a visual tracking method using online reliability evaluation and feature selection. The main contributions are summarized as follows. Firstly, we propose an effective method to evaluate the current tracking result by analyzing the corresponding response map. Secondly, the proposed tracker presents an update scheme of regression models according to the result of the online assessment. Finally, we propose a feature selection method that integrates the performance advantages of deep features and the speed advantages of traditional features. Experimental results demonstrate that the proposed tracking algorithm performs favorably against the other nine state-of-the-art trackers.

The remainder of this paper is organized as follows. Section 2 discusses the related work. Section 3 and 4 present details of the proposed algorithm and related experimental results, respectively. Conclusions are presented in Section 5.

2. RELATED WORKS

In this section, we will discuss several methods that are closely related to our work, containing tracking with correlation filter, tracking with convolution network and tracking with multiple classifiers.

2.1. Tracking with Correlation Filter

As we all known, the correlation filter-based trackers provided a competitive performance for the past one decade. In summary, there are two major advantages of correlation filter-based trackers. Firstly, in the framework of correlation

filtering, the correlation operation in spatial domain can be transferred into element-wise multiplication with a high computational efficiency in Fourier domain by using Fast Fourier Transform (FFT). Secondly, correlation filter makes extensive use of limited training data by implicitly including all shifted versions of the given samples. Since these two advantages can solve the problems containing the lack of training samples and high real-time requirement, the framework of correlation filtering is especially suitable for tracking tasks.

In 2010, Bolme *et al.* proposed an initial version of the correlation filter, a Minimum Output Sum of Squared Error (MOSSE) filter that can operate at speeds of up to hundreds of Frames Per Second (FPS) [13]. Henriques *et al.* improved the MOSSE by learning a kernelized least-squares classifier (CSK) of the target appearance [8]. The CSK builds on intensity features and is further improved by using HOG features in the Kernelized Correlation Filter (KCF) [14]. Danelljan *et al.* used the Principal Component Analysis (PCA) to reduce the dimensionality of the color attribute (CN) feature [15]. For the problem of scale adaptation, Danelljan *et al.* proposed a Discriminative Scale-Space Tracker (DSST) that establishes a translation detection model and a scale detection model, respectively [16]. Li *et al.* proposed a scale adaptive kernel correlation filter tracker with feature integration (SAMF) that combines HOG and CN features and uses a multiscale detection method to detect the position and scale of the target simultaneously [17]. However, these trackers only use the traditional handcrafted features and update the regression model in each frame. Therefore, these trackers are prone to drift caused by weak discriminant ability and the training samples with errors.

2.2. Tracking with Convolution Network

In the tracking tasks, feature representation is a very considerable issue. Diversiform handcrafted features have been employed for target representation, such as HOG [18] and histogram of color. Recently, for the problem of object classification and recognition, CNN has shown strong advantages with good generalization and migration ability. Zhang *et al.* constructed a method of robust feature representation for target tracking by using a two-layer convolutional network without training [19]. Wang *et al.* proposed a multiscale sparse networks-based tracker under the particle filter framework without the offline pre-training [20]. In the framework of the KCF, Ma *et al.* proposed a tracking algorithm that achieves the final response by weighted combining the outputs of three different convolution layers [21]. Danelljan *et al.* used the deep feature from the outputs of a single convolutional layer of a pre-training CNN to replace the HOG feature [11].

Although the deep convolutional feature from a pre-training CNN can significantly improve the tracking performance in the same framework, the computational speed of these methods is especially slow. For example, the tracker in [21] using handcrafted features has a tracking speed of about 10 FPS. Conversely, the computational speed of the tracker in the study [11] using deep features has dropped to less than 1 FPS. Consequently, we propose a feature selection scheme that utilizes a handcrafted feature

when the tracking result is highly reliable, switches to the deep convolutional feature to track the target once the tracking result based on the handcrafted feature has a low confidence. This selection scheme comprehensively considers the balance between robustness and real-time performance. By using this scheme, the tracking speed of the proposed method can reach 19 FPS.

2.3. Tracking with Multiple Classifiers

Several discriminative trackers use multiple classifiers to improve the robustness of tracking. The TLD proposed by Kalal *et al.* consists of three parts: tracking module, detection module and learning module [3]. This method not only combines a tracker and a detector, but also adds an improved online learning mechanism, which makes the overall performance of target tracking more stable and effective. Zhang *et al.* proposed a multi-expert restoration scheme to select the best expert ensemble based on the minimum entropy criterion for preventing improper model updating [22]. Zhong *et al.* proposed a robust appearance model that exploits both holistic templates and local representations [23]. The update scheme of this method considers both the latest observations and the original template, thereby enabling the tracker to deal with appearance change effectively and alleviate the drift problem. Inspired by these trackers with multiple classifiers, the proposed algorithm trains two different classifiers (regression models) by respectively using the handcrafted feature and deep convolutional feature, and then selects the different classifiers by analyzing the response map during the tracking process.

3. METHODOLOGY

The proposed algorithm contains the following stages: Firstly, the tracker estimates the target position and scale by using a handcrafted feature in the framework of correlation filtering. Secondly, the tracker evaluates the reliability of the current tracking result by analyzing the correlation response map. Finally, according to the assessment result, the tracker determines whether to update the regression model and replace the handcrafted feature with a deep convolutional feature.

3.1. Tracking Framework

The baseline of the proposed tracker is closely related to the DSST tracker. The tracker firstly extracts the feature map of the target region according to the location and size of the target in the first frame. The feature map f is composed of d -channel features, where d is the number of channels. f^l denotes a feature channel off, where $l \in \{1, \dots, d\}$. By minimizing the L-2 error ε between the correlation response and the desired output g , the tracker computes the correlation filters h composed of different filters h^l corresponding to each feature channel,

$$\varepsilon = \left\| g - \sum_{l=1}^d h^l \star f^l \right\|^2 + \lambda \sum_{l=1}^d \|h^l\|^2 \quad (1)$$

where \star is the correlation operation. The desired correlation output g is a Gaussian-shaped label matrix, and λ is a

regularization coefficient. As a linear least square problem, Eq. (1) can be solved in the Fourier domain by using the Parseval's theorem. Eq. (2) gives the optimal solution of the filter h .

$$\mathbf{H}^l = \frac{\overline{\mathbf{G}}\mathbf{F}^l}{\sum_{k=1}^d \overline{\mathbf{F}^k}\mathbf{F}^k + \lambda}, \quad l=1, \dots, d \quad (2)$$

here, the Discrete Fourier Transform (DFT) of the corresponding variables is denoted by its capital letters. The bar $\overline{}$ represents the complex conjugate of the corresponding variable. Since the multiplication and division of Eq. (2) are both point wise operations, the computation in the Fourier domain significantly reduced compared with in the spatial domain. In order to learn a robust correlation filter, the filter must be updated during the tracking process. The framework of correlation filtering updates the numerator \mathbf{A}^l and denominator \mathbf{B}^l of the filter \mathbf{H}^l by using the tracking result of the t -th frame.

$$\mathbf{A}_t^l = (1-\eta)\mathbf{A}_{t-1}^l + \eta\overline{\mathbf{G}}\mathbf{F}_t^l, \quad l=1, \dots, d \quad (3)$$

$$\mathbf{B}_t = (1-\eta)\mathbf{B}_{t-1} + \eta \sum_{k=1}^d \overline{\mathbf{F}_t^k}\mathbf{F}_t^k \quad (4)$$

where the parameter η is a learning rate.

Based on the numerator \mathbf{A}_{t-1}^l and denominator \mathbf{B}_{t-1}^l of the filter \mathbf{H}_{t-1}^l , and the feature map of candidate region \mathbf{z}_t of current frame, the DFT of the response of the correlation filter \mathbf{H}_{t-1}^l is calculated by:

$$\mathbf{Y}_t = \frac{\sum_{l=1}^d \overline{\mathbf{A}_{t-1}^l}\mathbf{z}_t^l}{\mathbf{B}_{t-1} + \lambda} \quad (5)$$

The location of the target in the current frame can be estimated based on the maximum of the response \mathbf{y}_t . In order to solve the problem of scale variation during the tracking process, we use a one-dimensional scale correlation filter to estimate the scale of target. To obtain the scale training samples $f_{t,scale}$, we extract features using image patches with different size centered around the target. Suppose that $P \times Q$ is the target size in the current frame and N is the number of the scale in the scale set S ,

$$S = \{a^n \mid n = [-\frac{N-1}{2}, [-\frac{N-3}{2}, \dots, [\frac{N-1}{2}]]\} \quad (6)$$

Here, S is the scale coefficient set of different scales Eq. (6). For each scale a^n of S , we take the estimated location as the center and $a^n P \times a^n Q$ as the size, extracting the image patches \mathbf{I}_n . Moreover, we use a 1-dimensional Gauss function to generate the desired output label. The tracker learns and updates the scale model and detect the detection samples based on Eq. (2-5).

3.2. Online Assessment of Tracking Result

Most current correlation filter-based trackers update the model directly without considering the reliability of the cur-

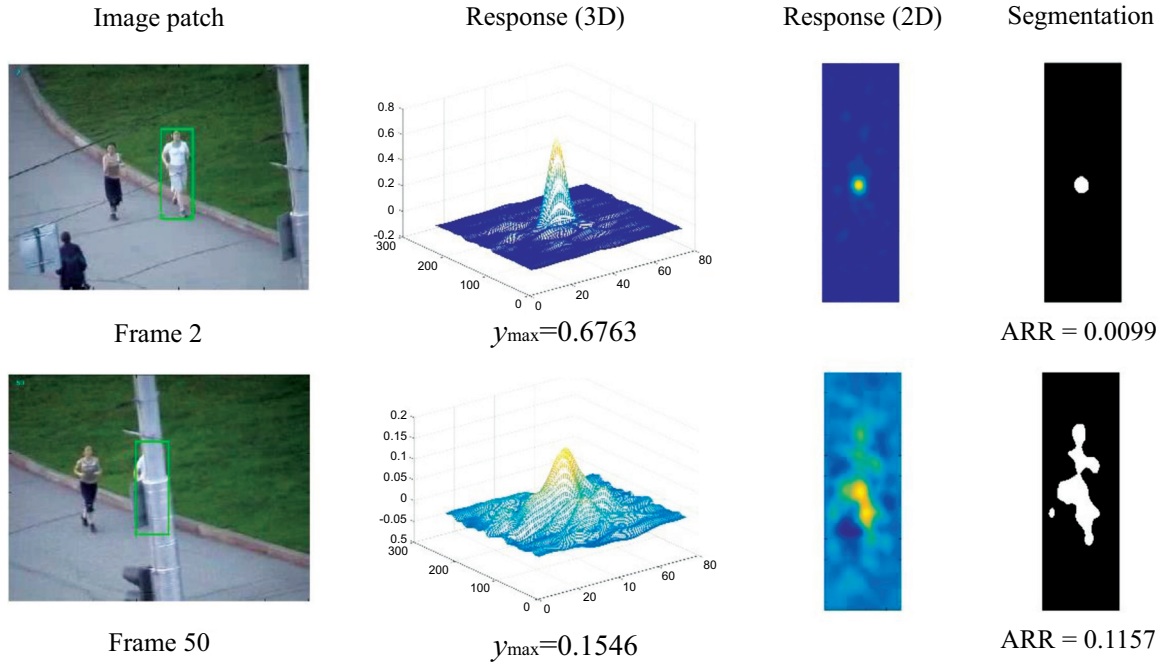


Fig. (1). Visualization of the calculation of two evaluation indexes. (A higher resolution / colour version of this figure is available in the electronic copy of the article).

rent tracking result. If the estimated position is inaccurate, updating the model continuously is likely to lead to tracking failure. To address the problem, the proposed algorithm presents a method that evaluates the confidence of the current tracking result based on the response of the filter. Based on the evaluation results, the tracker determines whether updating the model and whether using the deep convolutional feature.

Related researches show the maximum response and the shape of the response map can reflect the confidence of the tracking results in the framework of correlation filtering. The steeper the single peak is, the more reliable the tracking result will be. If there are obvious fluctuations in the response map, it means that the confidence of the tracking results is very low. Based on this conclusion, our tracker designs two evaluation indexes to represent the shape of response map:

(1) The maximum y_{max} in the response map \mathbf{y} , defined as in Eq. (7),

$$y_{max} = \max(\mathbf{y}) \quad (7)$$

Generally, the higher the maximum response is, the more reliable the tracking result is. Moreover, considering the maximum response cannot reflect the fluctuation of the response map, we use the second index called *ARR*.

(2) Area Ratio of Response (*ARR*), which is defined as Eq. (8),

$$ARR = \frac{\text{numel}(\text{find}(\text{otsu}(\mathbf{y})) == 1)}{\text{area}(\mathbf{y})} \quad (8)$$

here, $\text{otsu}(\mathbf{y})$ denotes the binary image of the response map obtained by *Otsu* method [24]. The function $\text{numel}(\text{find}(\cdot) == 1)$ means the number of pixels whose value is equal

to one in the binary image, and $\text{area}(\cdot)$ is the area of the binary image. As shown in Fig. (1), the index *ARR* can reflect the fluctuation of the response map. If the shape of the response map tends to a single sharp peak and smooth around, the value of *ARR* will be smaller. On the contrary, if the target is occluded or disappeared, the value of this index will increase significantly.

3.3. Regression Model Updating

For the regression model update, the proposed tracker implements a selective update strategy according to the current tracking result. The tracker updates the model continuously in the first T frames of a test sequence to retain stable historical information of the evaluation indexes. From $T+1$ t th frame, the tracker evaluates the confidence of the current tracking result. If the following two conditions are met simultaneously, the current tracking result will be trusted to have a high confidence and then the tracker updates the model in Eqs. (9,10).

$$y_{max,t} > \theta \text{mean}(y_{max,2}, \dots, y_{max,t-1}) \quad (9)$$

$$ARR_t < \delta \text{mean}(ARR_2, \dots, ARR_{t-1}) \quad (10)$$

Where θ and δ are two proportional coefficients. If these two indexes cannot satisfy the above conditions, the tracking result will be too unreliable to update the target model. Fig. (2a) shows that the fluctuation of the *ARR* and the maximum response plots are obvious when the target has serious occlusion at the 40th frame of the *coke* sequence. Furthermore, the tracker can update the regression model quickly in the case of the target with frequent appearance variation. For example, in the *Basketball* sequence shown as Fig. (2b), the appearance of the basketball player changes frequently, and the historical average of the *ARR* is increasing relatively. Thus,

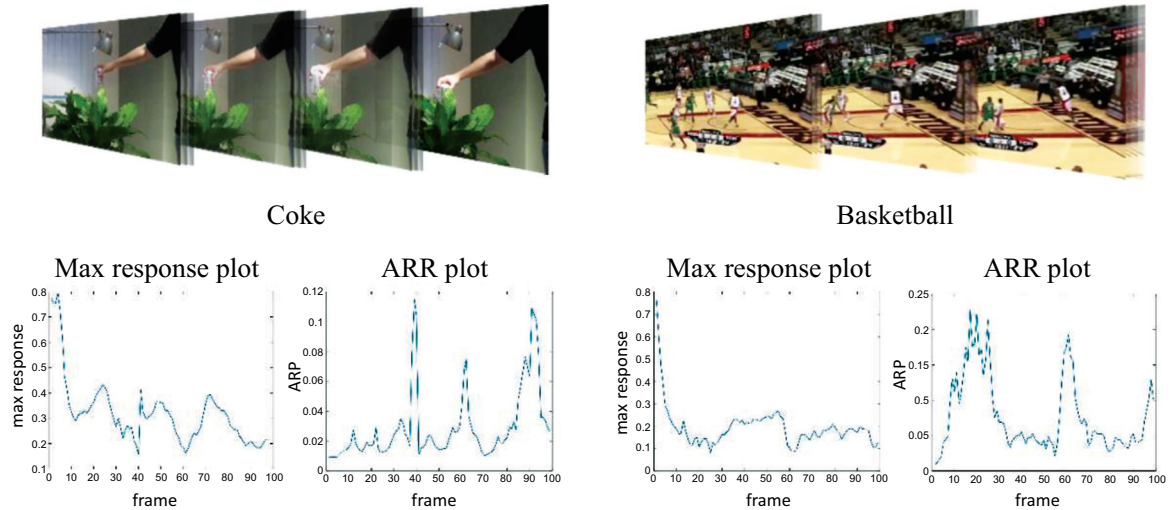


Fig. (2). Maximum response plot and ARR plot: (a) Sequence Coke. (b) Sequence Basketball. (A higher resolution / colour version of this figure is available in the electronic copy of the article).

the conditions for updating the model can be satisfied more easily with a high historical average.

3.4. Deep Convolutional Feature

The proposed algorithm integrates the performance advantage of deep features and the speed advantage of handcrafted features. When tracking in a simple scene without complex interference, handcrafted feature, *e.g.*, HOG, is good enough to estimate the target position accurately and efficiently. Inversely, when tracking a target in complex scenes, the deep feature is more discriminative to track the target. Therefore, we construct a feature selection scheme to combine these two features. The handcrafted feature is the chief one and a deep convolutional feature is an auxiliary one. When the tracking result is unreliable that means there is at least an index not satisfied the conditions of Eq. (9-10), the tracker uses the deep feature from the outputs of convolution neural network, such as AlexNet [25] and VGG-Net [26] network to estimate the target location. Furthermore, we evaluate the tracking result by the response map of deep features simultaneously. If the *ARR* of the deep feature response satisfies the tracking result is adopted and the regression model based on handcraft feature is updated, where μ is a threshold of peak area ratio. Otherwise, the tracking result obtained by the deep feature model is not adopted. This situation means that the target may be under the interference of occlusion or other challenges, and the tracker has to continue to use the results of handcrafted feature-based filter. Moreover, considering the advantage of semantic discriminative information and real-time performance, we only learn and update the deep feature-based regression model only in the first k frames of the video sequence in Eq. (11).

$$ARR_{deep} < \mu \quad (11)$$

We use the outputs of the last convolution layer that include more semantic information as the deep feature. However, the spatial resolution of the feature map gradually decreases with the increase in the number of convolution layers. This low resolution cannot ensure the location of the

target is accurate enough. In order to solve this problem, we expand the size of the feature by bilinear interpolation to locate the target more accurately [8].

3.5. Algorithm Implementation

According to the above discussion, the proposed tracking method is summarized as follows:

Algorithm 1. The proposed tracking algorithm

Input: initial target bounding box x_0

Output: estimated object state $x_t = (\hat{x}_t, \hat{y}_t, \hat{s}_t)$, handcraft feature-based model H_h and deep feature model H_d

1. Repeat
 2. Crop out the searching window in frame t according to $(\hat{x}_{t-1}, \hat{y}_{t-1})$ and extract the handcraft features;
 - // Translation estimation
 3. Compute the correlation response map \mathbf{y} using H_h and Eq. 5 to estimate the new position (x_t, y_t) , using Eq. 7 and Eq. 8 compute y_{max} and *ARR*;
 - //Re-detection
 4. if the tracking result is not reliable
 5. Crop out the searching window in frame t according to $(\hat{x}_{t-1}, \hat{y}_{t-1})$ and extract the deep features;
 6. Compute the correlation response map \mathbf{y} using H_d and Eq. 5 to estimate the new position (x_{t_cm}, y_{t_cm}) , using Eq. 8 compute *ARR*;
 7. if the tracking result is reliable
 8. $(x_t, y_t) = (x_{t_cm}, y_{t_cm})$;
 9. End
 10. End
-

```

// Scale estimation
11. Estimate the optimal scale  $\hat{s}_t$ ;
// Model update
12. if the first frame
13. Using Eq. 4 learning model  $H_h$  and  $H_d$ ;
14. Else if the tracking result is not reliable
15. Update the handcrafted model  $H_h$ ;
16. End
17. Until End of video sequences

```

4. EXPERIMENTAL RESULTS AND DISCUSSION

This section contains three parts. Firstly, we show the implementation details of our method and introduce the evaluation criteria. Secondly, we demonstrate the validity of the proposed through comparative experiments. Finally, we provide both quantitative and qualitative comparisons with state-of-the-art trackers.

4.1. Experimental Setup and Evaluation Criteria

In this paper, PCA-HOG [27] is employed as the handcraft feature of image representation. In order to achieve the dense representation, the cell size of HOG is set to 1×1 . In order to enhance the HOG feature, the image gray value is added to the HOG feature map as a feature channel. The feature extracted by VGG-16 [26] network trained on ImageNet [28] is used as the deep feature, which can be obtained from the Matconvnet toolkit [29]. We first remove the full connection layer in the network and then use the last convolutional layers to extract the features of the target. Secondly, the image patch for deep feature extraction is expanded to 224×224 by bilinear interpolation. After subtracting the mean parameters of the network training, the size of the deep feature map used in our method is 14×14 . Finally, in order to locate accurately, the feature is interpolated back to the size of the search bounding $P \times Q$ when the initial size of the target is less than 3000 pixels. If the initial target region is larger than 3000 pixels, the size of the feature is interpolated to the size of $(P/2) \times (Q/2)$.

In scale estimation, the proposed tracker also uses the PCA-HOG. Firstly, the image patches with different sizes are adjusted to the fixed size that is equal to the initial size of the target, and the cell with 4×4 pixels is used to extract the features. If the initial size of the target is less than 512 pixels, we reduce the size of image to 512 pixels.

The proposed algorithm is implemented in MATLAB 2016a and all the evaluation algorithms run on a 3.40GHz PC with 16GB RAM. The specific parameters in this paper are set as follows: The regularization parameter of Eq. (1) is set to $\lambda=10^{-2}$, and the learning rate is set to $\eta=0.025$. The size of the search window is set to 2 times the target size for handcraft feature-based model as 2.5 times for a deep feature-based model. The standard deviation of the desired Gaussian function output is set to 1/16 of the target size for handcraft feature-based model as 1/5 of the target size for

deep feature-based model. The number of frames to update the deep feature-based model is set to $k=3$. For scale estimation, the number of scale-space is set to $N=33$, scale coefficient is set to $\alpha=1.02$ and the standard deviation of the desired Gaussian function output is set to $\sqrt{N}/4$. The related parameters of the tracking result assessment mechanism are set to $\theta=0.4$, $\delta=3$, $\mu=0.2$, respectively. For fairness, when testing different video sequences, the parameters of the tracker are fixed.

We evaluate the proposed method on a large benchmark dataset OTB-2013 [30], that contains 50 videos with comparisons to state-of-the-art methods. All video sequences are annotated with 11 attributes, including Scale Variation (SV), Illumination Variation (IV), occlusion (OCC), deformation (DEF), motion blur (MB), Fast Motion (FM), in-plane rotation (IPR), Out-Plane Rotation (OPR), Out of View (OV), Background Clutter (BC), and Low Resolution (LR). The test method is tracking the target by frame by frame after initializing the initial frame in the sequence. The contrast algorithms used in the experiment are set according to the open-source code in the database.

Three important criteria from [30] are used for quantitative performance evaluation.

(1) Precision Rate (PR) shows the percentage of frames whose estimated location is within the given threshold distance (20 pixels generally) of the ground truth.

(2) Success Rate (SR), which is defined as the percentage of frames where the bounding box overlap surpasses a threshold (50% generally). Here, the bounding box overlap between the tracking result r_t and the ground-truth r_g is defined as in Eq. (12):

$$\text{overlap} = \frac{r_t \cap r_g}{r_t \cup r_g} \quad (12)$$

(3) Area under the curve (AUC) is defined as the area under the curve of success rate plot.

4.2. Effectiveness of the Proposed Tracking Method

In order to demonstrate the effectiveness of the proposed tracking algorithm using Deep Convolutional Feature and Discriminant Mechanism (DFDM), we compare it with the other two trackers, the tracker (DFDM-u) by only use the discriminant model updating mechanism without deep feature, and the tracker (Baseline) neither a deep feature nor a discriminant model updating method. We report the results on the 50-benchmark sequences using the success rate in Fig. 3, where the legend contains the AUC score for each tracker. In Fig. (3a), the tracking performances of DFDM-u and Baseline are almost the same because DFDM-u only improves the performance in case of occlusion but is less adaptive to the variation of the target appearance. However, as shown in Fig. (3b), in the sequence of occlusion attribute, the performance of the DFDM-u tracker is better than the Baseline tracker due to selective model updating. By comparison, the proposed tracker significantly outperforms the other two trackers due to the use of deep convolutional features and model updating mechanism together.

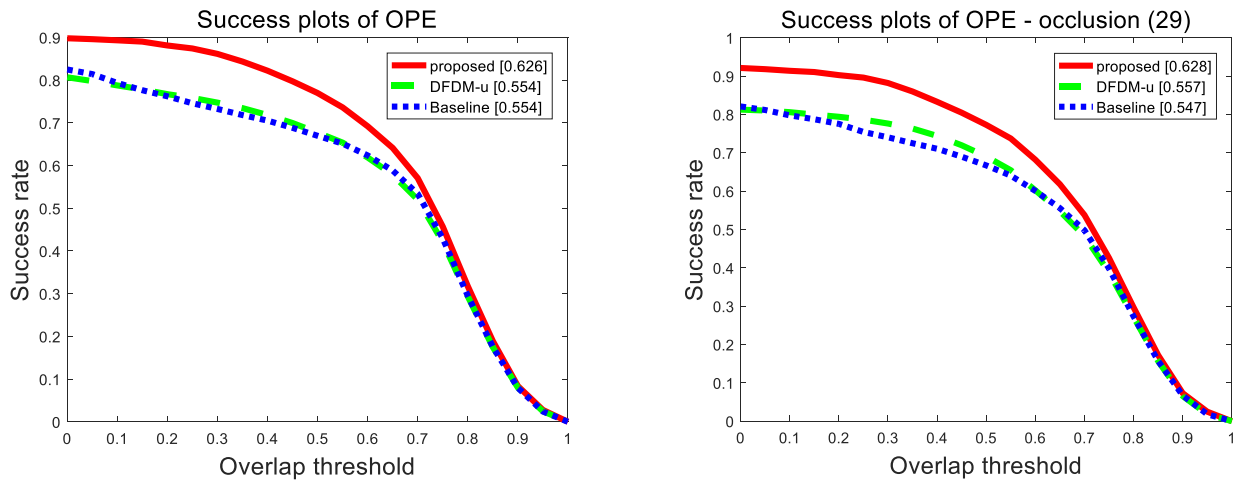


Fig. (3). The success plots on OTB-2013. (a) Overall performance comparison (b) Comparison facing occlusion. (A higher resolution / colour version of this figure is available in the electronic copy of the article).

Table 1. quantitative comparison with state-of-the-art trackers.

-	KCF	DSST	SAMF	Staple	SCM	TLD	MEEM	CNT	SiamFC_3s	Proposed
PR(%)	74	74	78.5	79.3	64.9	60.8	<u>83.0</u>	72.3	80.9	84.6
AUC(%)	51.4	55.4	57.9	60	49.9	43.7	56.6	54.5	<u>60.8</u>	62.6
Speed(FPS)	172	24	7	80	0.5	21	10	5	<u>86(GPU)</u>	19

4.3. Comparison with the State-of-the-art Trackers

4.3.1. Quantitative Comparison

We evaluate the proposed algorithm with comparisons to nine state-of-the-art trackers. These trackers can be divided into three typical categories of tracking algorithms: (1) tracking with correlation filter (Staple [31], SAMF [17], DSST [16], KCF [14]); (2) tracking with multiple online classifiers (TLD [3], SCM [23], MEEM [22]); (3) tracking with convolution neural network (CNT [19], SiamFC [32]). We report the results in one-pass evaluation (OPE) using the precision rate and success rate, as shown in Fig. (4), where the legend contains the AUC score for each tracker. Moreover, Table 1 demonstrates the quantitative comparisons of precision rate when the given threshold distance equals to 20 pixels, AUC score and tracking speed. The first and second highest values are highlighted by bold and underline in Table 1. The results of Fig. (4) and Table 1 show that our approach performs well against the other nine methods. Specifically, the proposed tracker performs well for the precision rate with 84.6%, which is approximate 1.6% higher than the tracker ranked second. Moreover, the proposed tracker also achieves the best success rate of 62.6% among all the trackers.

Table 1 shows that our algorithm performs favorably against state-of-the-art methods in precision rate and success rate. Our tracker integrates the deep features and traditional features at a speed of 19 FPS. The main computational load is the extraction of deep convolution features.

Furthermore, the video sequences in the OTB-2013 dataset are annotated with 11 attributes [30], which are used to

describe the different challenges in the tracking problem. We also evaluate the performance of trackers under different challenges. We report the success rate plot for 11 challenging attributes and show them in Fig. (5). As shown in Fig. (5), the proposed method is optimal in illumination variation (AUC=61.9%), out-of-plane rotation (AUC=61.6%), occlusion (AUC=62.8%), motion blur (AUC=59.1%), in-plane rotation (AUC=59.4%), and low resolution (AUC=50%).

4.3.2. Qualitative Evaluation

The tracking results of Fig. (6) show that because of using the robust feature representation HOG, the KCF tracker can perform well in handling motion blur and illumination variation (*Fleetface* and *Singer2*). However, it drifts when target objects undergo heavy occlusions (*Lemming*, *Jogging-2*, *Coke* and *Tiger2*). Additionally, the KCF tracker fails to handle fast motion (*Jumping*) and background clutter (*Shaking*), due to the limited search scope and low discriminative ability of HOG in a cluttered background. Compared with the KCF, the performance of the DFDM proposed in this paper is improved from Fig. (6). Staple complements the HOG features and color histogram to enhance the robustness of the target. However, it does not perform well in the case of tracking failure (*Lemming*, *Jogging-2*, *Singer2* and *Jumping*). In contrast, when the target is partially occluded (*Jogging-2*, *Tiger2* and *David3*), completely occluded (*Coke* and *Soccer*) and the target is blocked for a long time (*Lemming*), the proposed algorithm can track the target with a robust performance by using deep convolutional feature and selection mechanism. Although the TLD is able to re-detect target objects in the case of tracking failure, it updates its detector

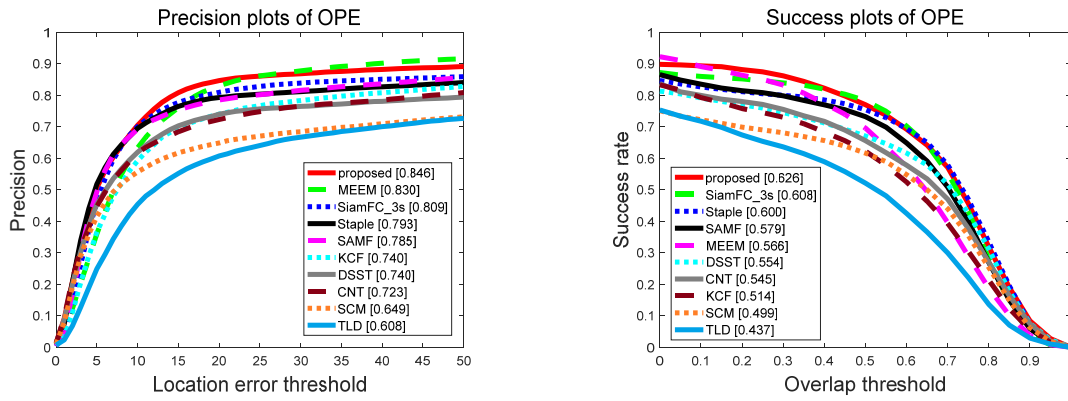


Fig. (4). Overall performance comparison on OTB-2013. (a) Precision plot (b) Success plot. (A higher resolution / colour version of this figure is available in the electronic copy of the article).

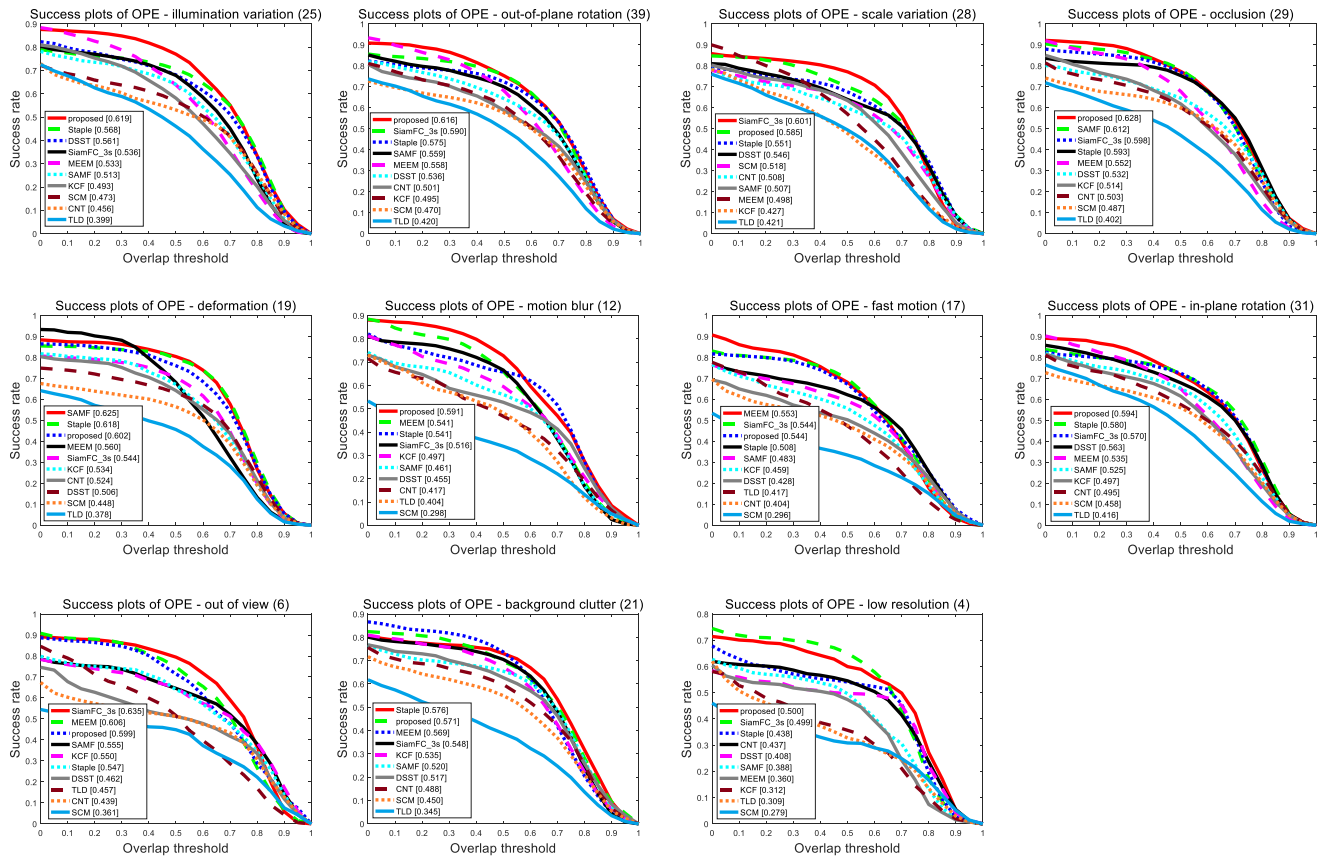


Fig. (5). Success plots over eight tracking challenges of 11 attributes. (A higher resolution / colour version of this figure is available in the electronic copy of the article).

frame-by-frame leading to drifting (Trellis and Jump) and false re-detection (*Jogging-2* and *David3*). In contrast, the proposed algorithm can achieve high accuracy for these test sequences.

In sum, the proposed tracker DFDM performs well in estimating both the scales and positions of target objects on these challenging sequences, which can be attributed to three reasons. Firstly, the proposed tracker analyzes the current

tracking state according to the response map and it is effective to judge the abnormal state such as occlusion and deformation. Secondly, the mechanism of the regression model update effectively alleviates the drifting problem caused by occlusion (*Lemming*, *Jogging-2* and *Coke*) and out of view (*Tiger2*). Thirdly, the feature selection mechanism effectively re-captures the target when facing the out-plane rotation (*Fleeface* and *Soccer*).

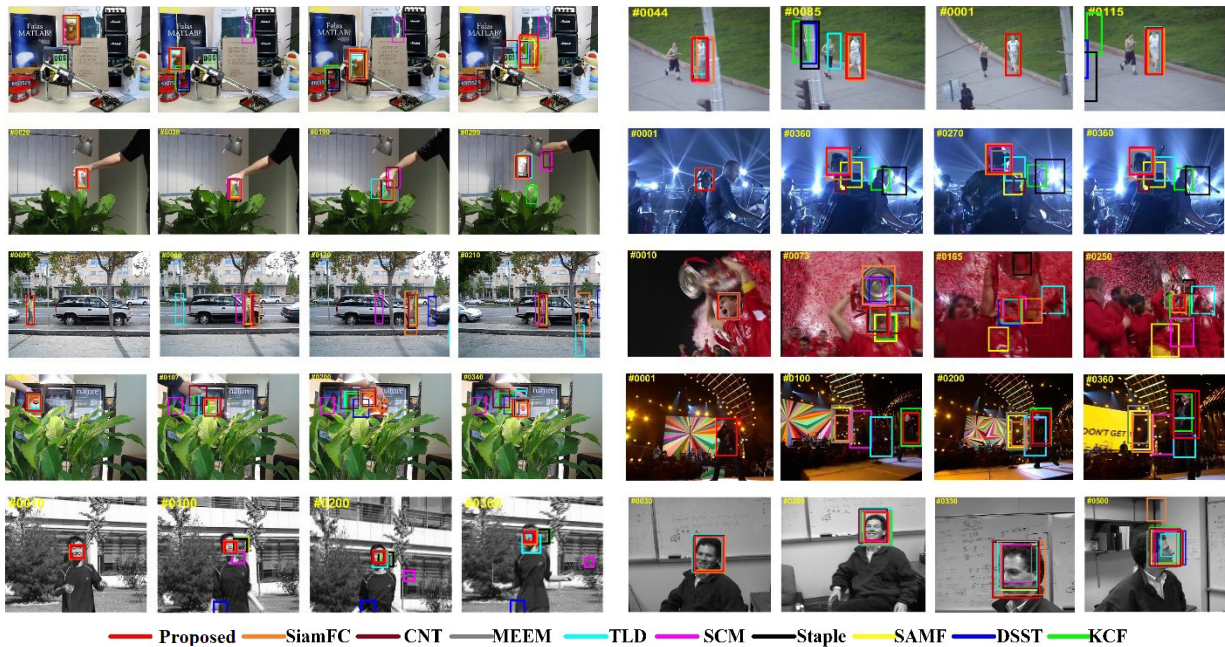


Fig. (6) Tracking results of all the ten algorithms on ten challenging sequences. (from left to right and top to down are *Lemming*, *Jogging-2*, *Coke*, *Shaking*, *David3*, *Soccer*, *Tiger2*, *Singer2*, *Jumping* and *Fleetface*.) (A higher resolution / colour version of this figure is available in the electronic copy of the article).

CONCLUSION

In this paper, we propose an effective tracking algorithm in the framework of correlation filtering. The proposed tracker can evaluate the current tracking result effectively by analyzing the response map. Based on the result of online assessment, the tracker determines whether to update the regression model and to use the deep convolutional feature. The quantitative and qualitative experimental results both demonstrate that the proposed algorithm performed favorably against nine state-of-the-art algorithms. Specifically, on OTB-2013 datasets, the Precision rate and Success rate of the proposed tracking algorithm can reach 84.8% and 62.5%, respectively. Moreover, the tracking speed of the proposed algorithm is 19 frames per second.

The main limitation of the proposed method is that we judge whether to update the regression model of the correlation filter by comparing the relationship between two indexes (y_{max} and *ARR*) and their mean values. In order to improve the robustness and autonomy of the proposed tracker, one of the main tasks in the future is to design a new intelligent algorithm to evaluate the reliability of the response map from the outputs of the correlation filter. Moreover, the proposed algorithm will be further explored to meet a wide range of engineering application needs, particularly for safety-critical situations such as autonomous driving.

LIST OF ABBREVIATIONS

- CT = Compressive Tracking
- MIL = Online Multiple Instance Learning
- TLD = Tracking-Learning-Detection
- IVT = Incremental Visual Tracking
- L1APG = L1 tracker using Accelerated Approach Proximal Gradient

- HOG = Histogram of Oriented Gradient
- DFT = Distribution Fields for Tracking
- CNN = Convolution Neural Network
- FFT = Fast Fourier Transform
- FPS = frames Per Second
- KCF = Kerneilized Correlation Filter
- MOSSE = Minimum Output Sum of Squared Error
- PCA = Principal Component Analysis (PCA)
- CN = The Color Attribute
- SAMF = Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration
- DSST = Discriminative Scale Space Tracker
- ARR = Area Ratio of Response
- PR = Precision Rate
- SR = Success Rate
- AUC = Area Under the Curve

CONSENT FOR PUBLICATION

Not applicable.

AVAILABILITY OF DATA AND MATERIALS

Not applicable.

FUNDING

This work was co-supported by the National Natural Science Foundation of China (No.61673211, No.U1633105, No.61203170), Aeronautical Science Foundation of China (No.20155152041, No. 20170752008), and Natural Science

Foundation of the Jiangsu Higher Education Institutions of China (No.18KJB590002).

CONFLICT OF INTEREST

The authors declare no conflict of interest, financial or otherwise.

ACKNOWLEDGEMENTS


Declared none.

REFERENCES

- [1] K. Zhang, L. Zhang, and M.H. Yang, "Fast compressive tracking", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 2002-2015, 2014.
<http://dx.doi.org/10.1109/TPAMI.2014.2315808> PMID: 26352631
- [2] B. Babenko, M.H. Yang, and S. Belongie, "visual tracking with online multiple instance learning", In: *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, USA, 2009, pp. 983-990.
<http://dx.doi.org/10.1109/CVPR.2009.5206737>
- [3] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-Learning-Detection", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409-1422, 2012.
<http://dx.doi.org/10.1109/TPAMI.2011.239> PMID: 22156098
- [4] Y. Fang, C. Wang, and W. Yao, "On-road vehicle tracking using part-based particle filter", *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 12, pp. 4538-4552, 2019.
<http://dx.doi.org/10.1109/TITS.2018.2888500>
- [5] D.A. Ross, J. Lim, R.S. Lin, and M.H. Yang, "Incremental learning for robust visual tracking", *Int. J. Comput. Vis.*, vol. 77, no. 1-3, pp. 125-141, 2008.
<http://dx.doi.org/10.1007/s11263-007-0075-7>
- [6] C.L. Bao, Y. Wu, and H.B. Ling, "Real time robust L1 tracker using accelerated proximal gradient approach", In: *IEEE Conference on Computer Vision and Pattern Recognition*, Providence, USA, 2012, pp. 1830-1837.
- [7] L. Sevilla-Lara, and E. Learned-Miller, "Distribution fields for tracking", In: *IEEE Conference on Computer Vision and Pattern Recognition*, Providence, USA, 2012, pp. 1910-1917.
<http://dx.doi.org/10.1109/CVPR.2012.6247891>
- [8] Z. Ji, and K. Feng, Y. Qian, "Part-based visual tracking via structural support correlation filter", *J. Vis. Commun. Image Represent.*, vol. 64, no. 102602, 2019.
- [9] F. Mustansar, M. Arif, and J. Sajid, "Handcrafted and deep trackers: Recent visual object tracking approaches and trends", *ACM Comput. Surv.*, vol. 52, no. 2, pp. 1-44, 2019.
<http://dx.doi.org/10.1145/3309665>
- [10] P.X. Li, D. Wang, and L.J. Wang, "Deep visual tracking: Review and experimental comparison", *Pattern Recognit.*, vol. 76, pp. 323-338, 2018.
<http://dx.doi.org/10.1016/j.patcog.2017.11.007>
- [11] M. Danelljan, G. Häger, and F.S. Khan, "Convolutional features for correlation filter based visual tracking", In: *IEEE International Conference on Computer Vision Workshop (ICCVW)*, Santiago, Chile, 2015, pp. 621-629.
<http://dx.doi.org/10.1109/ICCVW.2015.84>
- [12] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 810-815, 2004.
<http://dx.doi.org/10.1109/TPAMI.2004.16> PMID: 18579941
- [13] D.S. Bolme, J.R. Beveridge, and B.A. Draper, "Visual object tracking using adaptive correlation filters", In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, USA, 2010, pp. 2544-2550.
<http://dx.doi.org/10.1109/CVPR.2010.5539960>
- [14] J.F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583-596, 2015.
<http://dx.doi.org/10.1109/TPAMI.2014.2345390> PMID: 26353263
- [15] M. Danelljan, F.S. Khan, and M. Felsberg, "adaptive color attributes for real-time visual tracking", In: *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, USA, 2014, pp.1090-1097.
<http://dx.doi.org/10.1109/CVPR.2014.143>
- [16] M. Danelljan, G. Hager, F.S. Khan, and M. Felsberg, "Discriminative scale space tracking", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1561-1575, 2017.
<http://dx.doi.org/10.1109/TPAMI.2016.2609928> PMID: 27654137
- [17] Y. Li, and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration", In: *European Conference on Computer Vision*, Zurich, Switzerland, 2014, pp. 254-265.
- [18] N. Dalal, and B. Triggs, "Histograms of oriented gradients for human detection", In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, USA, 2005pp. 886-893.
<http://dx.doi.org/10.1109/CVPR.2005.177>
- [19] K. Zhang, Q. Liu, Y. Wu, and M.-H. Yang, "Robust visual tracking via convolutional networks without training", *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1779-1792, 2016. PMID: 26890870
- [20] X. Wang, Z. Hou, and W. Yu, "Robust visual tracking via multiscale deep sparse networks", *Opt. Eng.*, vol. 56, no. 4, 2017.
<http://dx.doi.org/10.1117/1.OE.56.4.043107>
- [21] C. Ma, J.B. Huang, and X.K. Yang, "Hierarchical convolutional features for visual tracking", In: *IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, USA, 2015, pp. 3074-3082.
<http://dx.doi.org/10.1109/ICCV.2015.352>
- [22] J. Zhang, S. Ma, and S. Sclaroff, "Meem: Robust tracking via multiple experts using entropy minimization", In: *European Conference on Computer Vision*, Zurich, Switzerland, 2014, pp. 188-203.
http://dx.doi.org/10.1007/978-3-319-10599-4_13
- [23] W. Zhong, H. Lu, and M.H. Yang, "Robust object tracking via sparsity-based collaborative model", In: *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, Providence, USA, pp. 1838-1845.
<http://dx.doi.org/10.1109/CVPR.2012.6247882>
- [24] N. Otsu, "A threshold selection method from gray-level histograms", *IEEE Trans. Syst. Man Cybern.*, vol. 9, no. 1, pp. 62-66, 1979.
<http://dx.doi.org/10.1109/TSMC.1979.4310076>
- [25] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Image net classification with deep convolutional neural networks", *Commun. ACM*, vol. 60, no. 6, pp. 84-90, 2017.
<http://dx.doi.org/10.1145/3065386>
- [26] K. Simonyan, and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, 2014.
<https://arxiv.org/abs/1409.1556>
- [27] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627-1645, 2010.
<http://dx.doi.org/10.1109/TPAMI.2009.167> PMID: 20634557
- [28] J. Deng, W. Dong, and R. Socher, "ImageNet: A large-scale hierarchical image database", *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, USA, 2009, pp. 248-255.
<http://dx.doi.org/10.1109/CVPR.2009.5206848>
- [29] A. Vedaldi, and K. Lenc, "MatConvNet: Convolutional Neural Networks for MATLAB", In: *MM '15: Proceedings of the 23rd ACM international conference on Multimedia*, 2015 pp. 689-692.
<http://dx.doi.org/10.1145/2733373.2807412>
- [30] Y. Wu, J. Lim, and M.H. Yang, "Online object tracking: A benchmark", In: *IEEE Conference on Computer Vision and Pattern Recognition*, Portland, OR, USA, 2013, pp. 2411-2418.
<http://dx.doi.org/10.1109/CVPR.2013.312>
- [31] L. Bertinetto, J. Valmadre, and S. Golodetz, "Staple: complementary learners for real-time tracking", In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, USA, 2016, pp. 1401-1409.
<http://dx.doi.org/10.1109/CVPR.2016.156>
- [32] L. Bertinetto, J. Valmadre, and J.F. Henriques, "Fully-convolutional siamese networks for object tracking", In: *The European Conference on Computer Vision Workshops*, Amsterdam, The Netherlands, 2016, pp. 850-865



1. Visual target tracking via online reliability evaluation and feature selection in the framework of correlation filtering

Wei, Li¹; Ding, Meng² ; Cao, Yun-Feng³; Zhang, U² **Source:** Recent Advances in Electrical and Electronic Engineering, v 13, n 7, p 1068-1077, 2020; **ISSN:** 23520965, **E-ISSN:** 23520973; **DOI:** 10.2174/2352096513666200316151351; **Publisher:** Bentham Science Publishers

Author affiliation:

¹ Jincheng College, Nanjing University of Aeronautics and Astronautics, Nanjing, China

² School of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing, China

³ School of Astronautics, Nanjing University of Aeronautics and Astronautics, Nanjing, China

Abstract: Background: Although correlation filtering is one of the most successful visual tracking frameworks, it is prone to drift caused by several factors such as occlusion, deformation and rotation. Objective: In order to improve the performance of correlation filter-based trackers, this paper pro-poses a visual tracking method via online reliability evaluation and feature selection. Methods: The main contribution of this paper is to introduce three schemes in the framework of correlation filtering. Firstly, we present an online reliability evaluation to assess the current tracking result by using the method of adaptive threshold segmentation of response map. Secondly, the proposed tracker updates the regression model of correlation filter according to the assessment result. Thirdly, when the tracking result based on a handcrafted feature is not reliable enough, we propose a feature selection scheme that autonomously replaces a handcrafted feature used in the traditional correlation filter-based trackers with a deep convolutional feature that can re-capture the target by its powerful discriminant ability. Results: On OTB-2013 datasets, the Precision rate and Success rate of the proposed tracking algorithm can reach 84.8% and 62.5%, respectively. Moreover, the tracking speed of proposed algorithm is 19 frame per second. Conclusion: The quantitative and qualitative experimental results both demonstrate that the proposed algorithm performed favorably against nine state-of-the-art algorithms.

© 2020 Bentham Science Publishers. (32 refs.)

Main Heading: Target tracking **Controlled terms:** Feature extraction - Regression analysis - Reliability

Uncontrolled terms: Adaptive threshold segmentation - Correlation filtering - Correlation filters - Current tracking - Reliability Evaluation - State-of-the-art algorithms - Tracking algorithm - Visual target tracking

Classification Code: 922.2 Mathematical Statistics

Funding details: Number: 20170752008, Acronym: -, Sponsor: Aeronautical Science

Foundation of China; Number: 18KJB590002, Acronym: -, Sponsor: Natural Science Research of Jiangsu Higher Education Institutions of China; Number: 61203170, Acronym: NSFC, Sponsor: National Natural Science Foundation of China;

Funding text: This work was co-supported by the National Natural Science Foundation of China (No.61673211, No.U1633105, No.61203170), Aeronautical Science Foundation of China (No.20155152041, No. 20170752008), and Natural Science Foundation of the Jiangsu Higher Education Institutions of China (No.18KJB590002).

Database: Compendex

ELSEVIER [Terms and Conditions](#) [Privacy Policy](#)
Copyright © 2020 [Elsevier B.V.](#) All rights reserved.

 RELX™