IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 4, 2019, accepted March 18, 2019, date of publication March 28, 2019, date of current version May 28, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2907967

Cascaded Iterative Training Model and Parallel Multi-Classifiers for Visual Object Tracking

DONGZHU FENG[1], FEIFEI GAO[1], XIN WANG[2], GUANGMIN WANG[1], AND HAO DAI[1]
[1]School of Aerospace Science and Technology, Xidian University, Xi'an 710071, China
[2]School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China

Corresponding author: Dongzhu Feng (dongzhufengnet@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61873210, Grant 61503292, and Grant 61701383.

ABSTRACT Because of the superiority in distinguishing the target from the background by learning mechanism, discriminative classifier attracts a great deal of attention in target detection and tracking systems. Generally, a discriminative classifier is learned from the initial image patch. However, the other algorithms, using only one image frame inhibit the performance of the classifier. A novel training strategy of cascade iteration, which is based on the multiple image frames, is used to initialize the discriminative classifier for the purpose of weakening the initialization limitation and improving the algorithms' adaptability and robustness. Furthermore, to cope with the changes in natural images, online learning is usually adopted to enhance the performance of classifiers, but it easily leads to tracking failures, especially in the case of full occlusion. In order to attenuate the influence of full occlusion, a multi-classifiers parallel tracking algorithm is proposed by designing the multi-classifiers and inducing extra constraints. The experiments are performed on the visual tracking benchmark, and the proposed algorithm is more robust while the target is fully occluded. The simulation results show that the proposed algorithm exhibits better performance compared to the state-of-the-art trackers, especially with KCF, TLD, and STRUCK.

INDEX TERMS Visual tracking, cascade iteration, multiple classifiers, parallel tracking, correlation filters.

I. INTRODUCTION

Visual object tracking, tasked with locating a target in subsequent frames automatically (given only its initial state), is an utmost research direction in computer vision [1]. It is widely applied to activity analysis [2], intelligent video surveillance [3], and human–computer interaction [4]. Although remarkable progress has been made in the past decade [5]–[9], it is still a huge challenge for a robust tracking due to illumination variation, occlusion or background clutter, etc. [10].

As a core component of trackers, existing appearance models are usually categorized as either generative or discriminative. Generative models are ordinarily formulated to search for a region that best matches a learned target appearance [21]. Recent efforts in this domain include subspace learning [11], [12], sparse representation [13]–[16], matrix decomposition [17], [18], graph regularization [19], etc. By contrast, discriminative trackers treat tracking as a binary classification problem that distinguishes the target from the background [22]. Various methods are developed based on this tracking model, such as multiple instance learning [5], boosting [20], support vector machine [23], compressive sensing [24], correlation filter [25], [29]–[31], etc.

Owing to the excellent performance of the discriminant classifier, tracking-by-detection has become the most popular framework for visual tracking. Kalal et al. [26] figured out the long-term tracking by decomposing tracking into three sub-tasks: tracking, learning, and detection, and derived the well-known TLD algorithm. Hare et al. [27] proposed Structured Output Tracking with Kernels (STRUCK) algorithm where the output space was defined as the translations of the target relative to the previous frame. Tracking accuracy is considerably improved in several benchmarks. Nevertheless, STRUCK suffers from a high computational complexity of its complex optimization and its training samples are still not dense enough.

Recently, correlation filter based trackers have been received great attentions for its real-time tracking [28]. Bolme et al. [29] proposed a correlation filter tracker via

The associate editor coordinating the review of this manuscript and approving it for publication was Tao Zhou.

learning the minimum output sum of squared error (MOSSE). Attributing to the high computational efficiency of correlation filters, MOSSE achieves a significant improvement in tracking speed. Henriques et al. [30] exploited a circulant structure of shifted image patches in a kernel space and proposed the circulant structure of tracking-by-detection with kernels (CSK) method for tracking. Subsequently, Henriques et al. [31] extended the CSK to the kernel correlation filters (KCF) algorithm with the multi-channel HOG feature [32], which enabled further extension for high dimensional features while remaining the real-time capability, to improve the accuracy and robustness of tracker.

However, the aforementioned work can't be well adapted to dynamic backgrounds. By analyzing the KCF, we empirically find that only one frame of the image is used to initialize classifier, which leads to lack of target interpreting capability. Although additional information of the target is slowly incorporated in later updating, the classifier is unable to adapt to the rapid variation of the target and the robustness is poor. Furthermore, the major defect of KCF is the updating strategy, which directly fusions the weight of the new classifier and the target template with the old, can lightly lose the original information of the target when the target is fully occluded. As a result, the tracking fails at the last.

A. CONTRIBUTIONS

In this paper, we tackle these issues in recent Discriminative Correlation Filter trackers, while restoring their hallmark real-time capabilities. The main contributions of the proposed algorithm can be summarized as follows:

The first contribution is that we propose an adaptive learning method for classifier initialization, which is an iterative classifier based on the cascade model. Multiple image frames, which contain the original information and the transformation information of the target, are utilized to initialize classifiers to make up the shortcomings that the classifier can't adapt to the target variance when it is initialized with a single sample.

The second contribution is the parallel tracking using multi-classifiers, that can prevent the loss of the original information during the variations of the target or background. The strategy, using three classifiers, which independently estimates the location of the target, can not only retain a variety of information of target, but also make the classifier perfectly adapted to the various variations.

The third contribution is to induce extra constraints to minimize the pseudo target information and maximize the performance of classifiers. The response of the classifier will quickly increase after a rapid decrease, and the classifier will be trained with pseudo target information when the target is occluded. Therefore, it's necessary to formulate an effective measure to minimize the pseudo target information and maximize the performance of classifiers.

The fourth contribution is the extensive evaluation of the state-of-the-art algorithms on benchmark data sets, where the proposed algorithm achieves robust tracking performance. Therefore, a significant improvement over state-of-the-art is derived based on the experimenting on more challenging datasets.

The rest of the paper is organized as follows. Section II reviews the work related to the kernelized correlation filters tracking. Section III introduces the implementation of multi-classifiers parallel tracking algorithm in detail. Section IV then performs a number of comparative experiments on visual tracking benchmark [33]. Section V finishes this paper with contributions and suggestions for future researches.

II. BASELINE APPROACH: THE KCF TRACKER

We base our approach on the kernelized correlation filters (KCF) [31], which achieves the highest speed and precision among the recent top-performing trackers. The key of KCF tracker is that the skill of using cyclic matrix structure to enhance the discriminative ability of track-by-detector scheme, which makes the algorithm more efficient. The overall tracking procedure can be summarized into Algorithm 1. In the following, we briefly review the key idea of KCF tracker.

In KCF, the tracking problem is treated as the classification problem. The classifier is trained with a image patch x of size $M \times N$ that is centered around the target. The patch is twice larger than the size of the target. The tracker considers all cyclic shift $x_i, i \in \{0, \ldots, M-1\} \times \{0, \ldots, N-1\}$ as the training examples for the classifier. The matching score $y_i \in [0, 1]$ is generated by a Gaussian function, and the classifier $f(x) = \langle \phi(x_i), \omega \rangle$ is trained by minimizing the ridge regression error:

$$\varepsilon = \arg\min_{\omega} \sum_{i=1}^{m} |y_i - \langle \phi(x_i), \omega \rangle|^2 + \lambda \|\omega\|^2 \tag{1}$$

where $\phi(x)$ is the mapping to a Hilbert space induced by the kernel κ, and $\lambda \geq 0$ is a regularization parameter that controls the complexity of the model and avoids overfitting. The larger $\lambda \geq 0$, the sparser the parameters.

Defining kernel function $\kappa(x, x') = \phi(x)^T \phi(x')$, and expressing the solution ω as a linear combination of inputs: $\omega = \sum_i \alpha_i \phi(x_i)$, the discriminant function of kernel function space is expressed as

$$f(z) = w^T z = \sum_{i}^{n} \alpha_i \kappa(z, x_i) \tag{2}$$

The closed-form solution to the classifier in kernel space is given by [34]

$$\alpha = (K + \lambda I)^{-1} y \tag{3}$$

where K is the kernel matrix with elements $K_{ij} = \kappa(x_i, x_j)$, I is the identity matrix, y represents a vector whose elements are y_i, and α is the vector of coefficients α_i.

In this case, the analytical tool, namely circulant matrices, can be used to avoid the ''curse of kernelization''. Since K is

TABLE 1. The KCF Algorithm.

Algorithm 1 :
Inputs:
x : training image patch;
y : regression target;
z : test image patch;
p_0 : initial target position;
Outputs:
p_i : detected target position;
Training stage:
(1) Compute the Gaussian kernel correlation of x with itself k^{xx} using (5);
(2) Compute coefficient $\hat{\alpha}$ using (4);
Position Detection:
(3) Compute the response $f(z)$ using (6);
(4) Find the target position p_i by maximizing $f(z)$;
Model Online Update:
(5) Update the template using the learning rate using (7).

a circulant matrix, $\hat{\alpha}$ can be simplified as follow by diagonalizing K [35],

$$\hat{\alpha} = \frac{\hat{y}}{\hat{k}^{xx} + \lambda} \tag{4}$$

where $\hat{k}^{xx}$ is defined as kernel correlation in [31]. $\hat{}$ denotes the Discrete Fourier Transform (DFT).

In the KCF tracker, we adopt the Gaussian kernel

$$\mathcal{K}(x, x') = \exp\left(-\frac{1}{\alpha^2}\|x - x'\|^2\right)$$

which can be applied to the circulant matrix trick as below:

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

As the algorithm only requires dot-product and FT/IDFT, the computational cost is in $O(n \log n)$ time.

During the tracking stage, the circulant matrix trick can also be applied to detection to speed up the whole process. A patch z is cropped out in the subsequent frame. The matching score of z can be evaluated via

$$\hat{f}(z) = \hat{k}^{xz} \odot \hat{\alpha} \tag{6}$$

The position of the target in the next frame is estimated by (6), that is, the position of the maximum response is the target position.

Intuitively, for a single input z, calculating $f(z)$ for all samples can be regarded as a temporal (spatial) filter, and $f(z)$ is a linear combination of k^{xz} weighted by the learned coefficient. Since this is a filtering operation, it can be formulated more efficiently in the Fourier domain.

The classifier model $\hat{\alpha}$ and the target appearance model x_i are thus updated with a learning rate μ frame by frame as

$$\begin{aligned} \alpha_i &= (1-\mu) \times \alpha_{i-1} + \mu \times \alpha_i^{'} \\ x_i &= (1-\mu) \times x_{i-1} + \mu \times x_i^{'} \end{aligned} \tag{7}$$

III. OUR APPROACH

In this section, we first propose an adaptive learning method for classifier initialization, which improves the target interpreting capability and the tracking performance. Then we present a novel design of multi-classifiers parallel tracking. Furthermore, we induce extra judgment constraints for classifier to enhance the robustness of the tracker. Finally, we introduce the multi-channel HOG feature, that can simplify the recently-proposed multi-channel correlation filters.

A. CASCADED ITERATIVE TRAINING MODEL

The adaptive learning initialization method, which is based on cascaded iterative classifier model, utilizes multiple image frames to initialize classifiers. In the first iteration, a weak classifier is derived by fusing classifiers with equal-weighted, which are trained with each sample. Then, the obtained weak classifier is inputted with multiple initialization frames, and the hypothesis probability of the target is adaptively adjusted according to the response of weak classifier. Finally, a strong classifier is composed of weak classifiers with appropriate weights. The specific process is illustrated as follows.

With training N basic classifiers α_n' with N initialization frames x_n,$n = 1, \ldots, N$, the corresponding weight vectors of the weak classifier are defined as

$$W_m = \{w_{m1}, w_{m2}, \ldots, w_{mN}\}, \quad m = 1, \ldots, M \tag{8}$$

In (8), when $m = 1$, $w_{1n} = 1/N$, $n \in [1, N]$. m denotes the number of iterations. A weak classifier is trained at each iteration. The weak classifier is expressed as

$$\alpha_m = \sum_{n=1}^{N} w_{mn} \cdot \alpha_n' \tag{9}$$

The weak classifier is evaluated on initialization frame x_n to estimate target position P'_{mn}. The estimated position is compared with the actual target position P_n, and the detection error is given by

$$e_m = \sum_{n=1}^{N} w_{mn} \cdot I(P'_{mn} \neq P_n) \tag{10}$$

In (10), the value of $I\left(P'_{mn} \neq P_n\right)$ is 1 when $P'_{mn} \neq P_n$, otherwise the value of $I\left(P'_{mn} \neq P_n\right)$ is zero. ($P'_{mn} \neq P_n$ is considered if the predicted target position P'_{mn} is within a distance threshold of the actual target position P_n). The detection error e_m denotes the sum of weights of basic classifiers while the false detection emerges. Then the weight of the weak classifier obtained in the m^{th} iteration is given by

$$\beta_m = \frac{1}{2}\log\left(\frac{1 - e_m}{e_m}\right) \tag{11}$$

In (11), when $e_m \leq 1/2$, $\beta_m \geq 0$, and β_m decreases with e_m increasing. It means that the weak classifiers with more false detection have weaker effects in the strong classifier.

Adjust the weight vector W_{m+1} in the next iteration,

$$w_{(m+1)n} = \frac{w_{mn}}{Q_m} exp\left(-\beta_m \cdot S(P'_{mn} = P_n)\right) \tag{12}$$

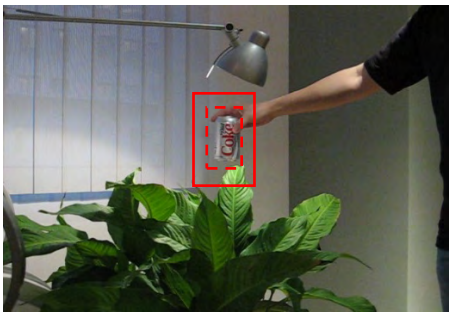

FIGURE 1. Original image frame.

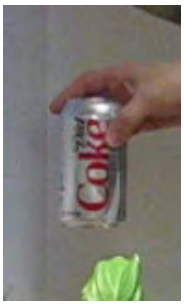

FIGURE 2. Target template.

where $S(P'_{mn} = P_n)$ represents that its value is 1 when $P'_{mn} = P_n$, otherwise its value is zero. Q_m is an normalization factor, which can be calculated by

$$Q_m = \sum_{n=1}^{N} \left(w_{mn} \cdot \exp\left(-\beta_m \cdot S\left(P'_{mn} \neq P_n \right) \right) \right) \tag{13}$$

The new weight vector W_{m+1}, obtained by (12), will distribute a higher weight to the basic classifier which is trained with the samples of error detection. Accordingly, in next iteration, it will be highlighted in the corresponding weak classifier.

The basic classifier is fused again with the new weight vector to model a new weak classifier. After M iterations, M weak classifiers are obtained. Finally, combining the M weak classifiers with the weight β, a strong classifier can be derived as.

$$\alpha = \sum_{m=1}^{M} \beta_m \cdot \alpha_m \tag{14}$$

In (14), the strong classifier α, which is initialized with multiple image frames, is able to accommodate to the change of target.

B. INITIALIZATION OF THE CLASSIFIER

In the previous section, the classifier initialization model and its calculation method are given in detail. In the coming sections, we will show the initialization of the classifier from two aspects: preprocessing and training the taking of a "Coke" video is used as an example.

1) PREPROCESSING

The ensemble samples are replaced with the cyclic shifts of the base sample in the sampling process. Taking into account the integrity of the target after cyclic shifts, the tracking region is generally centered as the position of estimated target, and doubles the size of the target to provide some context and additional negative samples.

As shown in Fig. 1, the two-dimensional image f is a frame of the video, and the beverage bottle is the target to be tracked.

(a) (b)

FIGURE 3. Results of windowed function. (a) Original image. (b) Result of windowed function.

In Fig. 1, the dotted rectangle represents the position and size of the target, while the solid rectangle denotes the target template. The tailored target template x_1 is shown in Fig. 2.

Since the Fourier transform is periodic, all calculations are performed in the frequency domain, and the image boundary is not sufficiently considered [36]. For aperiodic images, the high-frequency noise is generated at the image boundaries. As the standard of correlation filters, the input patches are weighted by a cosine or sine window. The processing of a cropped $n \times n$ image x^{raw} is given by

$$x_{ij} = (x_{ij}^{raw} - 0.5) \sin\left(\frac{\pi i}{n}\right) \sin\left(\frac{\pi j}{n}\right) \quad \forall i, j = 0, \ldots, n-1 \tag{15}$$

After processed by Eq. (15), the pixels near the boundary are given 0 weights, which can smoothly remove discontinuities. The results are shown in Fig. 3.

Fig. 3 (a) is a gray target template, and Fig. 3 (b) is the result of windowed function. It can be seen that the windowed function eliminates most of the background interference in the target template and only retains the target, which makes base sample undistorted during cyclic shifts, and improves the accuracy of classifiers.

2) TRAINING OF THE CLASSIFIER

As previously described, the training samples are composed of the shift of a base sample, so we must specify a regression target for each one in y. The regression target y simply follows

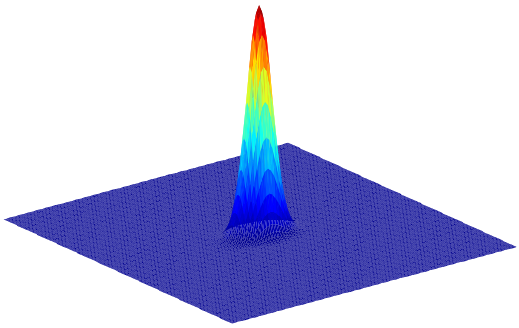

FIGURE 4. Expected output.

a two-dimensional Gaussian function which takes a value of 1 for a centered target position (i', j') and smoothly decays to 0 as the distance increases. Assuming that the spatial bandwidth of the Gaussian function is s, the smaller s, the steeper the Gauss function. That means that the maximum value of the classification results is more prominent. However, if s increases, the samples closer to the target will get higher scores. The expected output of the classifier y is given by

$$y_{ij} = \exp\left(-\left((i-i')^2 + (j-j')^2\right)/s^2\right) \quad \forall i, j = 0, \ldots, n-1 \quad (16)$$

According to the target template in Fig.3 (a), the expected output is shown in Fig. 4.

The peak of the Gaussian function in Fig.4 is the position of the target. Gaussian targets are smoother than binary labels, and have the benefit of reducing ringing artifacts in the Fourierdomain, which will receive a more accurate position estimate than binary labels.

The continuous 15 frames of Coke are used to initialize the classifier. The target template is extracted from each image frame. As formulated in (4), α'_n that represents 15 basic classifiers is trained by the kernel circulant matrices (sample set) and expected output (sample label), where $n = 1, 2, \cdots, 15$. Here, the initial weights of these basic classifiers are $w_{1n} = 1/15$. Then, with the initial weight, these basic classifiers are fused to derive a weak classifier α_1. The calculation flow of the weak classifier is shown in Fig. 5.

Next, the weak classifier is utilized to estimate the position of the target in the above 15 initialization frames. According to the actual position of the target, the detection error can be calculated. Then we can calculate and update weights vector of weak classifiers by the detection error. The whole process is shown in Fig. 6.

A weak classifier can be obtained at each iteration process (shown in Fig. 5 and Fig. 6). By iterating 10 times, we can derive 10 weak classifiers and their assigned weights. Then these weak classifiers are fused to obtain a strong classifier α (Eq. (14)). The strong classifier α contains the original and transformation information of the target, so the adaptability and anti-interference of the algorithm is improved. In addition, it does not require any prior knowledge about the target in the process of training the classifier [37], [38].

C. DESIGN OF MULTI-CLASSIFIERS

When the target is occluded, the target template will be gradually updated to the pseudo target, which is utilized to train the classifier. It will contribute to a loss of the original information. Then, the pseudo target will be tracked due to the trained classifier, and the real target will be lost. In order to tackle this problem and combining with the temporal context information, we propose a novel multi-classifiers parallel tracking method on the bases of the kernelized correlation filter. On account of the importance of previous video information, three classifiers are designed and trained to adapt to the dynamic variations and occlusion in the process of target detection and tracking.

a) The first classifier α_1, a benchmark classifier, is used to detect the occurrence of the object with initial state, which is never updated in the further detection and tracking.

b) The second classifier α_2 is configured to update the target template when the target attitude changes or non-rigid deformation occurs, and is updated when the responses of α_1 and α_2 are greater than that of the third classifier. The two classifiers contain the original information and transformation information of the target, which can ensure the comprehensiveness of the target information even if the target is occluded.

c) The third classifier α_3 is used to update the target template (pseudo target template) with occlusion. Before the target disappears, it ensures that the tracker will not lose the position under the circumstances of the target is occluded. In addition, it also can prevent α_1 and α_2 from updating with the pseudo template. When the occluded target reappears, the first and second classifiers still can successfully detect the real target.

The above three classifiers are initialized with the strong classifier as described in section III-A. Then the target location is estimated independently. A robust classifier can not only retain multiple information, but also adapt to various variations of the target.

D. MODEL ONLINE UPDATE SCHEME

To obtain a robust performance, extra constraints that are different from prior work in KCF is induced in the update stage to further enhance the robustness of the proposed algorithm.

Since target changes slowly during the movement, the difference of two adjacent frames is relatively small. The output response of the classifier decreases when the target or background varies. However, there are great shortcomings on the updating strategy of directly weighted fusion. When the target is occluded, the target template will be gradually updated to pseudo-target and the classifier will be trained with pseudo target information. Then the response will quickly increase after a rapid decrease. At this moment, the target template already contains more pseudo target information, and the classifier treats the pseudo target as a real target that is to be

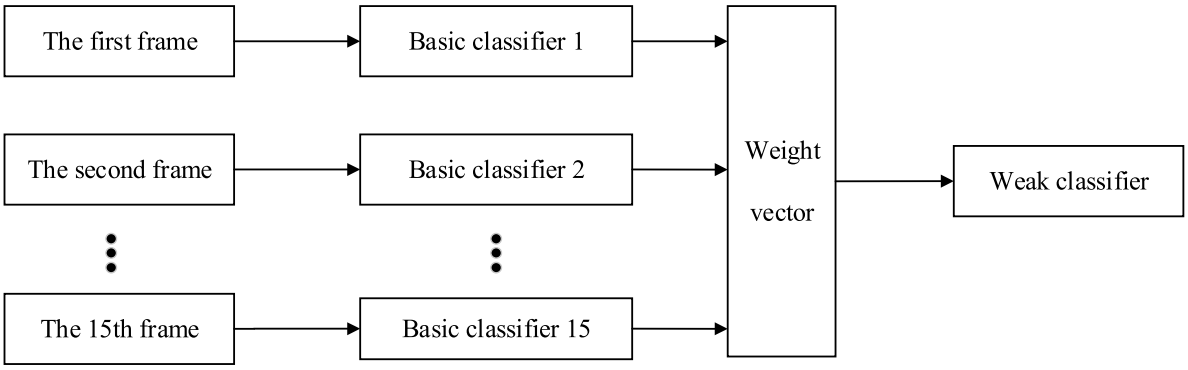

FIGURE 5. Iteration diagram of weak classifier.

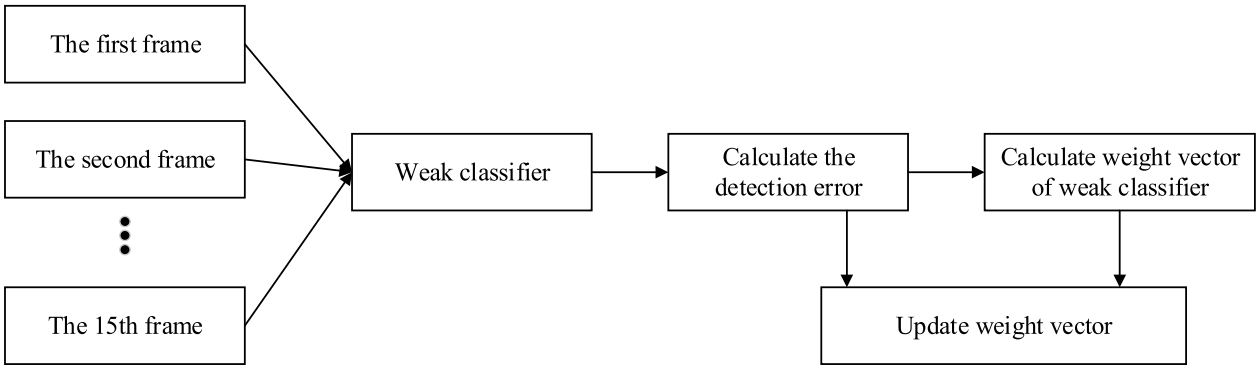

FIGURE 6. Schematic diagram of weight vector calculation.

tracked. In order to minimize the pseudo target information, we induce extra constraints for the updating of classifiers and target template. That is,

$$f_{\min(1,2,3)} > T \tag{17}$$

In (17), $f_{\min(1,2,3)}$ denotes the minimum output response of the three classifiers. T is a threshold that distinguishes between slow and large variations in the target. When the response is greater than the threshold, it indicates that the changes in the target between adjacent frames are small. The classifier and the template are updated when the output responses f_1, f_2, f_3 are all greater than the threshold T, which can prevent classifiers from integrating non-target information.

As mentioned in previous section, the first classifier retains the original information. It is utilized to estimate the position of target with initial state and never be updated in the subsequent detection and tracking. It can prevent the second and third classifiers from losing the original information caused by updating.

The second classifier is updated when the output responses of the classifiers satisfy the condition,

$$\begin{cases} f_{\min(1,2)} \geq f_3 \\ f_{\min(1,2)} > T. \end{cases} \tag{18}$$

where $f_{\min(1,2)}$ represents the minimum value of the responses of both the first classifier and the second classifier. It means that there is no occlusion or partial occlusion in the moving progress. The information of changes of the target is used to update the second classifier and the target template.

The other case is that the third classifier is updated when the responses of the classifiers meet the condition,

$$\begin{cases} T > f_{\max(1,2)} \\ f_3 \geq f_{\max(1,2)}. \end{cases} \tag{19}$$

In this case, the target changes greatly with the movement, such as occlusion or larger deformation. The information of changes of the target is used to update the third classifier and the target template, so as to ensure that the search region of the classifier stays near the position where the target greatly changes. Under this circumstance, the correct detection is performed.

E. MULTI-CHANNEL HOG FEATURES

It can be Noticed that all kernels' correlation are based on either dot-products or norms of the arguments, so we can easily extend the single-channel to multi-channel and replace the natural gray pixels of the original image with more dimension features. If the description of the image contains multiple channels, it allows us to sum the result for each channel in

the Fourier domain. The kernel correlation filter uses multi-channel HOG features to get a better tracking performance.

Assume that a vector x concatenates the individual vectors for C channels(e.g. 31 gradient orientation bins for a HOG variant [26]),

$$x = [x_1, x_2, \ldots, x_c] \tag{20}$$

In (20), x_i denotes a single-channel feature descriptor, and C channels are combined together to describe a pair of images. According to the response, we can estimate the position of the target because multi-channel filters [39] allow multiple inputs simultaneously, but only one output.

By linearity of the DFT, a dot-product can be computed by simply summing the individual dot-products for each channel. We can apply this reasoning to the linear kernel $\mathcal{K}(x, x') = x^T x'$, obtaining the multi-channel analogue of (21),

$$k^{xx'} = \mathcal{F}^{-1}\left(\sum_C \hat{x}_C^* \odot \hat{x}'_C\right) \tag{21}$$

As a concrete example, the multi-channel extension from the previous section simply yields

$$k^{xx'} = \exp\left(-\frac{1}{\sigma^2}\left(\|x\|^2 + \|x'\|^2 - 2\mathcal{F}^{-1}\left(\sum_C \hat{x}_C^* \odot \hat{x}'_C\right)\right)\right) \tag{22}$$

IV. EXPERIMENTS

We present an outline of our method in Algorithm 2 and more implementation details are discussed as follows.

A. EXPERIMENTAL SETUPS

1) IMPLEMENTATION

In this section, our algorithm is implemented in MATLAB. The experiments are performed on a PC, equipped with Intel (R) Core i5-3470 running at 3.2GHZ and 8G memory. Then we compare the proposed method with several the state-of-the-art visual tracking methods and all the tracking results are using the reported results to ensure a fair comparison.

2) DATASET

Comparative experiments are performed on standard visual tracking dataset: OTB-2013, which contains many sequences annotated with ground truth bounding boxes. It comprises a large variety of environments, including illumination variation(IV), occlusion(OCC), scale variation(SV), in-plane rotation(IPR), out-plane rotation(OPR), background clutters(BC), deformation(DEF), fast motion(FM), etc. These attributes represent the challenges in visual tracking, so this dataset has been frequently used to evaluate the general performance of visual trackers.

3) PARAMETERS SETTINGS

Our tracker requires few parameters whose values are given and fixed for all videos. We set the regularization parameters

TABLE 2. Multi-classifiers Parallel Tracking Algorithm.

Algorithm 2:

Inputs:

x_n : training image patch in n^{th} initialization frame;

y : regression target;

z : test image patch;

p_0 : initial target position;

Outputs:

p_i : detected target position;

Training stage:

1) Train N basic classifiers α_n' with N initialization frames

a Compute the Gaussian kernel correlation of x_n with itself $k^{x_n x_n}$ using (5);

b Compute coefficient $\hat{\alpha}'_n$ using (4);

2) Fuse these basic classifiers with the weight vector w_{mn} to obtain a weak classifier α_m using (9)

3) Evaluate α_m on initialization frames x_n to update the weight vector w_{mn} and the weight β_m of the weak classifier α_m using (11)-(13).

4) Repeat the step of 2) and 3) m times.

5) Fuse weak classifiers to obtain a strong classifier α using (14).

Position Detection:

6) Compute the response $f(z)$ using (6);

7) Find the target position p_i by maximizing $f(z)$;

Model Online Update:

8) Update the template using the learning rate using (7), according to the (17)-(19).

λ to 10^{-4}. Spatial bandwidth $s = 0.1$, is used to generate the Gaussian function labels. The adaptation rate is set to 0.02 and the feature bandwidth is 0.5. We set extra area surrounding the target to 2 for providing some context and additional negative samples. The threshold T is set to 0.25 for all sequences except, the threshold of ''Basketball'' is set to 0.4.

4) EVALUATION METHODOLOGY

On the one hand, for the quantitative analysis, all trackers are evaluated by the metric of frame per second (FPS). The FPS plotted indicates the numbers of frames which are processed per second. On the other hand, for the qualitative analysis, all trackers are evaluated by the metric of precision. For precision plot, one frame is treated as a successful tracked frame once the Euclidean distance between the center of predicated bounding box and corresponding ground-truth box is lower than a specified threshold [40]. A higher precision at low thresholds demonstrates the tracker is more accurate. Precision at threshold 10 pixels is chosen as the representative precision value.

B. QUANTITATIVE ANALYSIS

In this section, we provide a comprehensive comparison of our algorithm with other 6 state-of-the-art trackers,

TABLE 3. Running speed on video sequences.

	Doll	MountainBike	Walking	Freeman4	Girl	Coke	Sylvester
Proposed	**187**	**124**	**185**	**804**	**283**	**119**	**143**
KCF	183	117	180	757	269	117	139
STRUCK	10	13	13	9	11	9	8
TLD	27	46	33	52	37	46	31
MOSSE	**1796**	**1240**	**1931**	**5200**	**2790**	**1728**	**1506**
CT	100	70	39	128	154	52	128
WMIL	101	67	42	130	180	53	132

TABLE 4. Representative precision value.

	Doll	MountainBike	Walking	Freeman4	Girl	Coke	Sylvester
Proposed	**0.944**	**0.882**	**0.998**	**0.972**	**0.998**	**0.746**	**0.802**
KCF	0.924	0.807	0.998	0.816	0.85	0.512	0.752
STRUCK	0.669	0.329	0.728	0.194	**1**	0.32	**0.824**
TLD	0.848	0.419	0.369	0.587	0.8	0.399	0.722
MOSSE	0.005	0.088	0.027	0.025	0.02	0.034	0.012
CT	0.07	0.026	0.005	0.01	0.084	0.065	0.071
WMIL	0.073	0.031	0.005	0.067	0.024	0.048	0.078

namely KCF [31], TLD [26], STRUCK [27], MOSSE [29], WMIL [41] and CT [7]. The running speed demonstrating the computational efficiency of our framework is shown in Table 3, where the processing time is from reading the first frame of video to the end of tracking the last frame (excluding video display and error calculation).

In addition, we also record the representative precision value is shown in Table 4. By combining Table 3 with table 4, we can synthetically evaluate the tracking performance.

It can be seen that our approach improves the real-time capability and precision significantly. Although the running speed is only second to MOSSE, it is worth to mention that the tracking precision of our algorithm surpasses MOSSE by a relatively large margin. Similarly, precision of STRUCK is slightly higher than our algorithm on ''Girl'' and ''Sylvester'', but the tracking speed of our algorithm is about 20 times faster than that of STRUCK.

C. QUALITATIVE ANALYSIS

In order to visualize the tracking performance of the proposed algorithm in various interferences, we provide tracking results of some representative frames, and draw the precision curve to directly reflect the tracking performance.

1) ROTATION SCENE TESTING

In order to validate the tracking effect under the condition of target rotation, the ''MountainBike'' video is selected as the test sequence, where person and vehicle are the target to be tracked. The tracking results are shown in Fig. 7.

Although the target rotates during the movement, it can be seen that our algorithm does not deviate and lose the target in the whole process, and the tracking effect is robust. The tracking precision curve is shown in Fig. 8 that shows that the accuracy of the proposed algorithm is higher than other algorithms.

2) PARTIAL OCCLUSION SCENE TESTING

The ''Walking'' sequence is selected for the testing to verify the tracking effect when the target is partially occluded. The pedestrian is the target to be tracked. The tracking results are shown in Fig. 9.

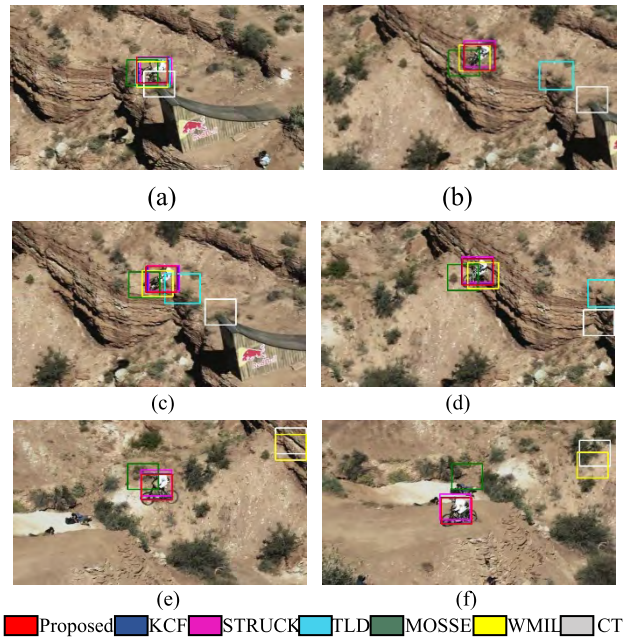

FIGURE 7. Results of rotation scene testing – MountainBike. (a) the 45th frame. (b) the 70th frame. (c) the 95th frame. (d) the 118th frame. (e) the 175th frame. (f) the 213th frame.

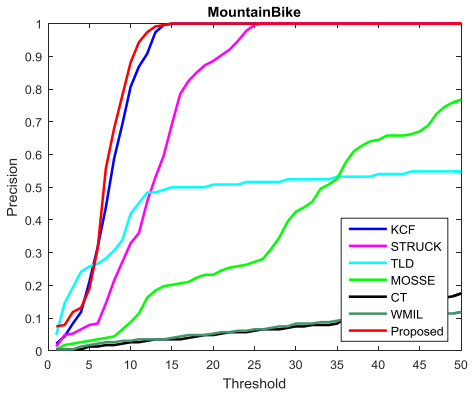

FIGURE 8. Tracking precision curve-MountainBike.

In Fig. 9, we can conclude that although the target is partially occluded by a street lamp (the 88th frame) during the movement, the proposed algorithm still performs well. Tracking precision curve is shown in Fig. 10. We can discover that the precision of the algorithm is higher than other algorithms.

3) ROTATION & PARTIAL OCCLUSION SCENE TESTING

In order to confirm the tracking of the target in the presence of both rotation and blocking, Girl video is selected as the test data. Fig. 11 shows the tracking results.

As shown in Fig. 11, the girl's face is the target, and the boy is the shelter. The tracking box does not deviate from the target in the whole tracking process, even though the target rotates and is occluded (the 427th frame). When the target is occluded, the classifier and target template stop updating when the output response of the classifier decreases to a

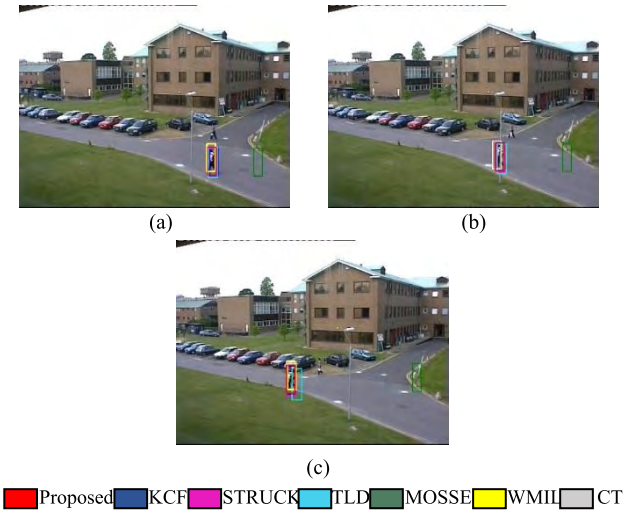

FIGURE 9. Results of partial occlusion scene testing –Walking. (a) the 60th frame. (b) the 88th frame. (c) the 180th frame.

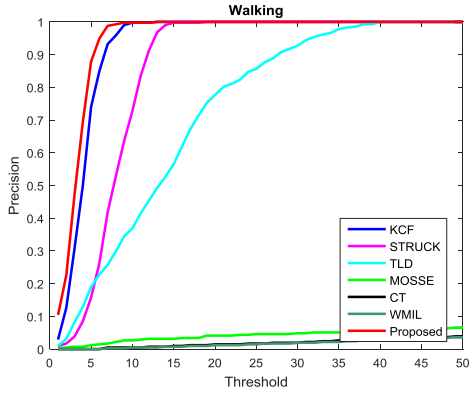

FIGURE 10. Tracking precision curve –Walking.

certain threshold. So the classifier still retains the information of the target, and the target is still detected successfully when the target reappears.

The tracking precision curve is shown in Fig. 12. It can be observed that the precision of the proposed algorithm is higher than other algorithms and is only second to the STRUCK algorithm. While the precision of the proposed algorithm is slightly different from that of STRUCK (Table 1), the running speed of our algorithm is 10 ∼ 20 times faster than that of the STRUCK algorithm (obtainable from Table 4).

4) ILLUMINATION VARIATIONS & ROTATION & PARTIAL OCCLUSION SCENE TESTING

Based on the previous experiments, the tracking results on ''Doll'', which includes illumination variations, target rotation and occlusion interference, are shown in Fig. 13. The tracking precision curve is shown in Fig. 14.

We can see that the tracking box of the algorithm can track the target accurately in the whole process (Fig. 13) and the precision of our approach is higher than other algorithms (Fig. 14).

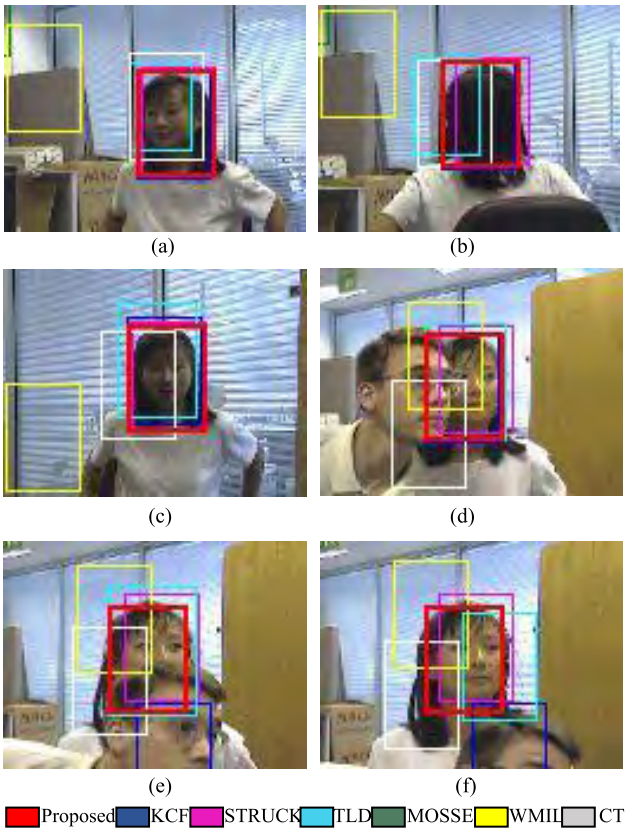

FIGURE 11. Results of rotation & partial occlusion scene testing–Girl. (a) the80th frame. (b) the105th frame. (c) the155th frame. (d) the427th frame. (e) the441th frame. (f) the447th frame.

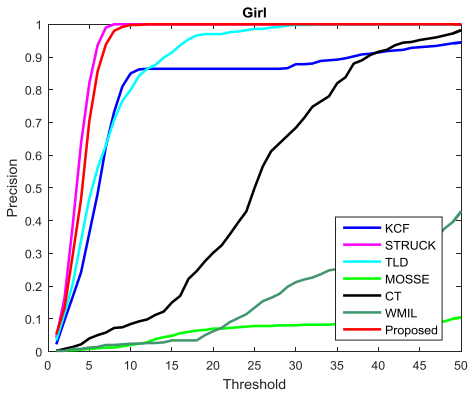

FIGURE 12. Tracking precision curve –Girl.

5) ILLUMINATION VARIATIONS & FULL OCCLUSION SCENE TESTING

The occlusion is divided into partial occlusion and full occlusion. The experiment chooses Coke video, where the canned beverage is the target to be tracked to carry out illumination variations and occlusion test. The result is displayed in Fig. 15.

During the process of movement, the target is occluded in a large area (the 39th frame), and reappears (the 45th frame) with illumination variations. Despite these interferences, our

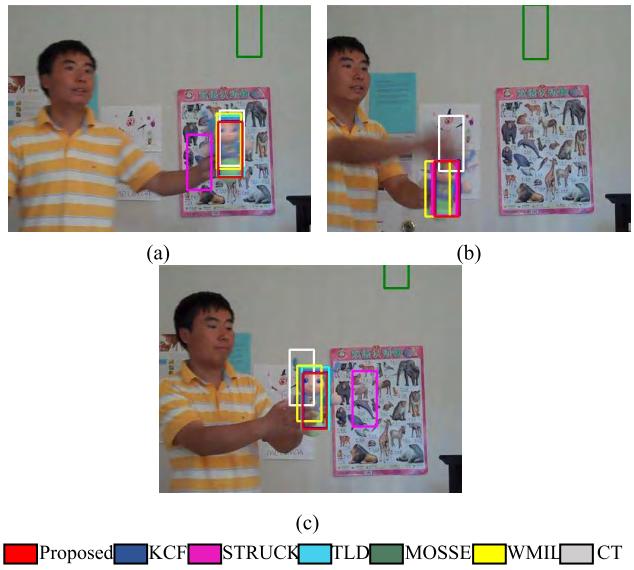

FIGURE 13. Results of illumination variations & rotation & partial occlusion scene testing –Doll. (a) the 2341th frame. (b) the 2399th frame. (c) the 2555th frame.

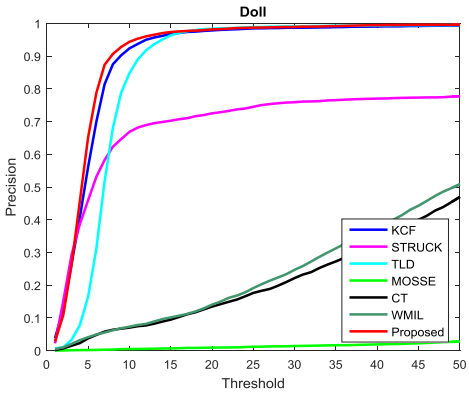

FIGURE 14. Tracking precision curve -Doll.

tracker can still accurately estimate the position of the target. The target is fully occluded (the 258th frame) and lasts about 10 frames. In this case, the tracking box of the proposed algorithm stays at the vanishing position of the target, when the tracker searches the target. When the target reappears (the 268th frames), the tracker can still detect the real target and track it successfully.

In this paper, the algorithm adopts multiple classifiers to estimate the target position independently, avoiding losing the information of the real target caused by using pseudo target information to train and update the classifier when the target is fully occluded. Therefore, the proposed algorithm has good adaptability and robustness to full occlusion interference. The tracking precision curve is presented in Fig. 16. It can be seen that our method is more accurate than other algorithms.

6) ATTITUDE VARIATIONS SCENE TESTING

In order to validate the effectiveness of our algorithm to attitude variations, "Sylvester" video is selected as the test

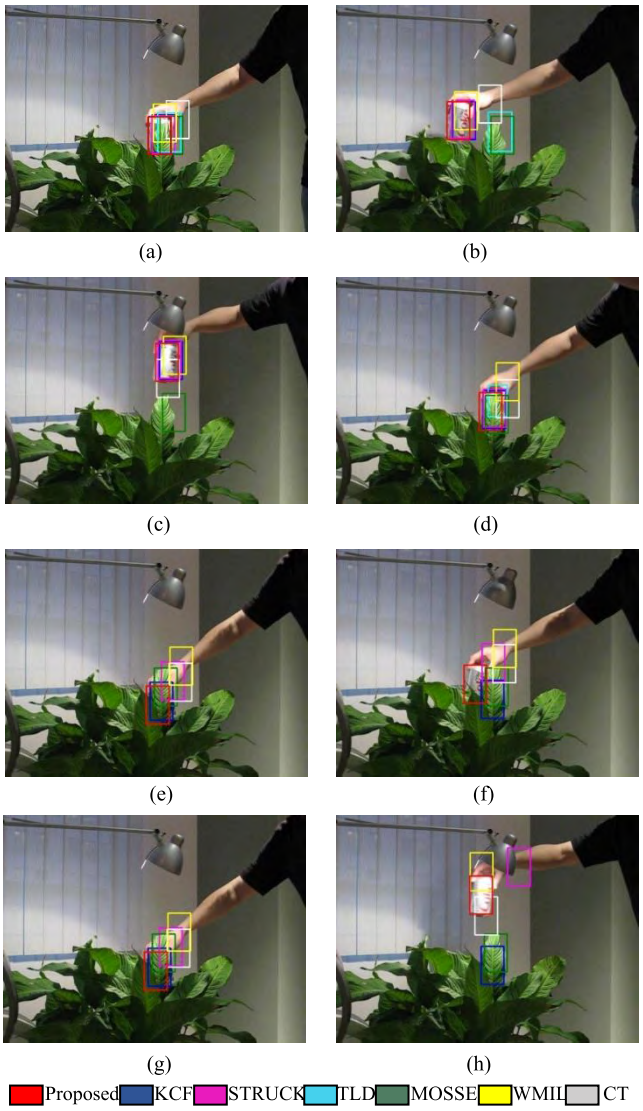

FIGURE 15. Results of illumination variations & fully occlusion scene testing -Coke. (a) the 39th frame. (b)the 45th frame. (c) the 175th frame. (d) the 254th frame. (e) the 258th frame. (f) the 268th frame. (g) the 275th frame. (h) the 288th frame.

sequence. Cat doll to be tracked has attitude variations due to various rotations and pitching. The results are shown in Fig. 17.

As is shown in Fig. 17, when it is vertically downward (the 1140th frame), the target attitude varies greatly which leads to texture changes, so that the extracted features are fully different from the original target. Eventually many algorithms failed to detect the target. Our algorithm uses multiple image frames to initialize the classifier, which makes the classifier more expressive to the target, and has stronger adaptability to the attitude variations. The tracking precision curve is shown in Fig. 18. It can be seen that the tracking precision of the proposed algorithm is slightly lower than that of the STRUCK. However, the tracking speed of the proposed algorithm is considerably faster than that of STRUCK.

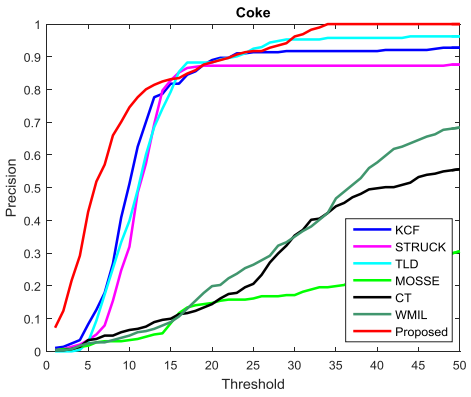

FIGURE 16. Tracking precision curve -Coke.

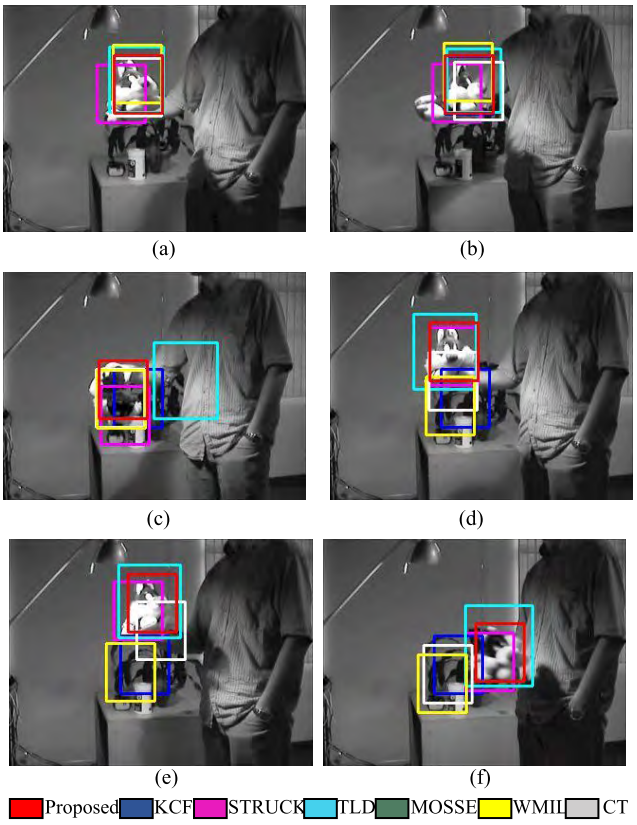

FIGURE 17. Results of attitude variations scene testing -Sylvester. (a) the 962th frame. (b) the 1017th frame. (c) the 1140th frame. (d) the 1175th frame. (e) the 1195th frame. (f) the 1280th frame.

7) BACKGROUND CLUTTERS SCENE TESTING

"Freeman4" video is selected for experimental testing to verify the anti-interference and stability of the proposed algorithm, the test results is provided in Fig. 19.

From Fig. 19, we can see that the face is the target that is in a chaotic environment with a variety of interference factors, such as constantly waving arms, newspapers and books and other shielding. When the target is occluded (the 61th frame) and reappears (the 63th frame), other trackers drifts except our approach. In this case, the proposed algorithm adopts a strategy that the third classifier is updated with the pseudo

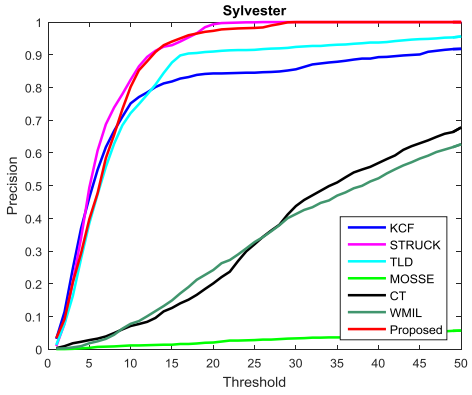

FIGURE 18. Tracking precision curve -Sylvester.

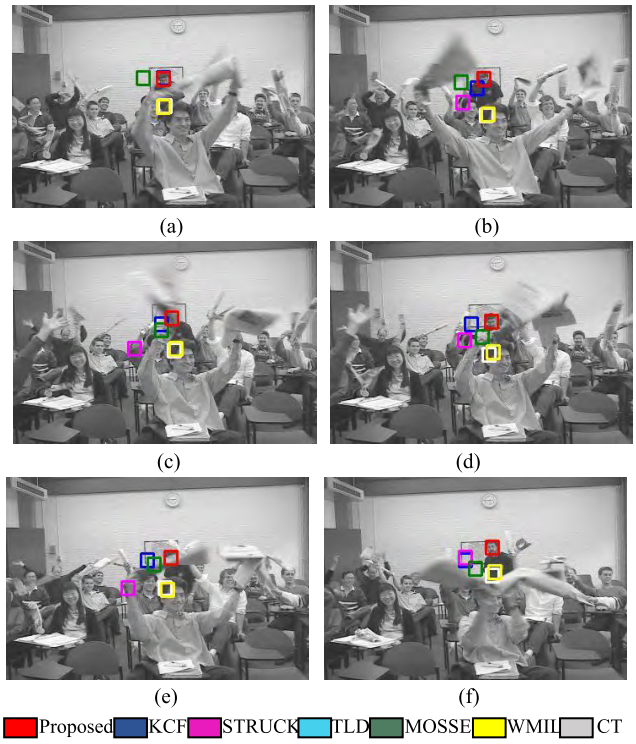

FIGURE 19. Results of background clutter scene testing -Freeman4. (a) the 48th frame. (b) the 52th frame. (c) the 61th frame. (d) the 63th frame. (e) the 68th frame. (f) the 78th frame.

target, and the first and second classifier are not updated, so the rectangle box stays near the vanishing position of the target. When the target appears again, it is detected accurately by the first and second classifier, so the algorithm can still track the target accurately even if the target is occluded many times. The tracking precision curve of the algorithm on the Freeman4 is shown in Fig. 20, which shows that the precision of the proposed algorithm is much higher than other algorithms.

In summary, we can conclude that the proposed algorithm has strong adaptability and anti-interference to illumination variations, occlusion and background clutter, etc. So the anti-interference and adaptability of our algorithm is stronger than other algorithms.

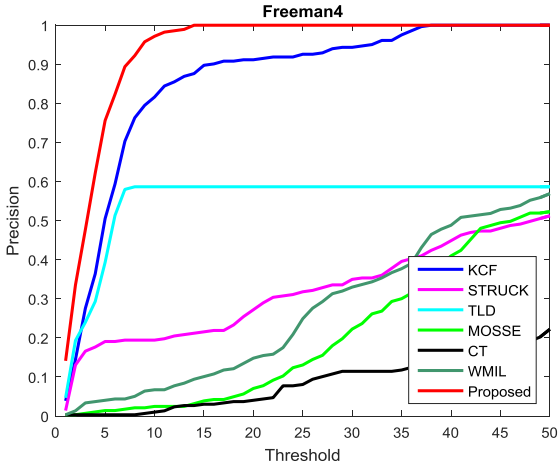

FIGURE 20. Tracking precision curve -Freeman4.

V. CONCLUSIONS AND FUTURE WORK

A. CONCLUSIONS

In this paper, we studies the problem of visual object tracking, in which the target or background varies frequently. We propose a novel design of parallel tracking by multi-classifiers. Three classifiers are used to estimate the position of the target independently. Furthermore, for initialization issues, we present an adaptive learning method for classifier initialization. The method is an iterative classifier based on cascaded iterative. Multiple image frames, which contain the original information and transformation information of the target, are utilized to initialize the discriminative classifiers for the purpose of improving adaptability and robustness. In addition, we also induce extra constraints for the classifiers updating to minimize the pseudo target information and maximize the performance of classifiers. And an extensive set of experiments is also performed. The results of the experiments clearly demonstrate the superiority of our algorithm to other state-of-the-art tracking methods. It is worth to emphasize that our approach not only performs superiorly, but also runs at a very fast speed that is extremely appropriate for real-time applications.

B. LIMITATIONS AND FUTURE WORK

Although our approach has yielded satisfied results, there are still some aspects that need to be addressed to get more reliable and general system based on the proposed algorithm. For instance, the implementation of our algorithm only uses HOG features, but other feature descriptors can be combined to improve the tracking performance in the future research. In addition, the rapid movement of the target can easily result in tracking failure, but this issue is not covered in this paper. The target scale variations are not taken into account either, and the tracking precision is seriously affected when a target changes in the scale. Furthermore, the tracker does not perform well when the position changes greatly due to occlusion. In that case, it will cause the target to detach from the search area. Eventually, the target is lost.

However, our algorithm has no error correction mechanism of target loss yet, which needs to be designed in the subsequent research.

ACKNOWLEDGMENTS

The authors would like to thank the reviewers for their useful comments and suggestions on their manuscript.

REFERENCES

[1] X. Zhang, X. Shi, W. Hu, X. Li, and S. Maybank, "Visual tracking via dynamic tensor analysis with mean update," *Neurocomputing*, vol. 74, no. 17, pp. 3277–3285, Oct. 2011.
[2] S. Zhang *et al.*, "Action recognition based on overcomplete independent components analysis," *Inf. Sci.*, vol. 281, pp. 635–647, Oct. 2014.
[3] I. Haritaoglu, D. Harwood, and L. S. Davis, "W^{4}S: A real-time system for detecting and tracking people in 2 1/2D," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, May 1998, pp. 877–892.
[4] M. Kim, S. Kumar, V. Pavlovic, and H Rowley, "Face tracking and recognition with visual constraints in real-world videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[5] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.
[6] H. Lu, X. Jia, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* Jun. 2012, pp. 1822–1829.
[7] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer-Verlag, Oct. 2012, pp. 864–877.
[8] D. C. Chen, M. Zhu, and W. Gao, "Real-time object tracking via online weighted multiple instance learning," *Opt. Precis. Eng.*, vol. 22, no. 6, pp. 1661–1667, 2014.
[9] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1090–1097.
[10] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, Jul. 2014.
[11] Y. Xie, W. Zhang, Y. Qu, and Y. Zhang, "Discriminative subspace learning with sparse representation view-based model for robust visual tracking," *Pattern Recognit.*, vol. 47, no. 3, pp. 1383–1394, Mar. 2014.
[12] Y. Sui, S. Zhan, and L. Zhang, "Robust visual tracking via sparsity-induced subspace learning," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4686–4700, Dec. 2015.
[13] T. Zhang, B. Ghanem, and S. Liu, "Robust visual tracking via structured multi-task sparse learning," *Int. J. Comput. Vis.*, vol. 101, no. 2, pp. 367–383, Jan. 2013.
[14] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2259–2272, Nov. 2011.
[15] H. Zhang, Y. Wang, L. Luo, X. Lu, and M. Zhang, "SIFT flow for abrupt motion tracking via adaptive samples selection with sparse representation," *Neurocomputing*, vol. 249, pp. 253–265, Aug. 2017.
[16] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2042–2049.
[17] Y. Wu, B. Shen, and H. Ling, "Visual tracking via online nonnegative matrix factorization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 3, pp. 374–383, Mar. 2014.
[18] H. Zhang, S. Hu, X. Zhang, and L. Luo, "Visual tracking via constrained incremental non-negative matrix factorization," *IEEE Signal Process. Lett.*, vol. 22, no. 9, pp. 1350–1353, Sep. 2015.
[19] T. Zhou, H. Bhaskar, F. Liu, and J. Yang, "Graph regularized and locality-constrained coding for robust visual tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 10, pp. 2153–2164, Oct. 2017.
[20] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proc. Brit. Mach. Vis. Conf.*, Edinburgh, U.K., Sep. 2006, pp. 1–6.
[21] J. Yan, X. Chen, D. Deng, and Q. Zhua, "Structured partial least squares based appearance model for visual tracking," *Neurocomputing*, vol. 144, pp. 581–595, Nov. 2014.
[22] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, "Recent advances and trends in visual tracking: A review," *Neurocomputing*, vol. 74, no. 18, pp. 3823–3831, Nov. 2011.
[23] S. Avidan, "Support vector tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 1064–1072, Aug. 2004.
[24] K. Zhang, L. Zhang, and M.-H. Yang, "Fast compressive tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 2002–2015, Oct. 2014.
[25] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf.*, Sep. 2014, pp. 1–5.
[26] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.
[27] S. Hare, A. Saffari, and P. H. S. Torr, "STRUCK: Structured output tracking with kernels," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2012, pp. 263–270.
[28] B. Bai *et al.*, "Kernel correlation filters for visual tracking with adaptive fusion of heterogeneous cues," *Neurocomputing*, vol. 286, pp. 109–120, Apr. 2018.
[29] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2544–2550.
[30] J. F. Henriques, C. Rui, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science. Oct. 2012, pp. 702–715.
[31] 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, Mar. 2014.
[32] 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, Sep. 2010.
[33] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2411–2418.
[34] R. Rifkin, G. Yeo, and T. Poggio, "Regularized least-squares classification," *Nato Sci. Ser. Sub III, Comput. Syst. Sci.*, vol. 190, pp. 131–154, May 2003.
[35] R. M. Gray, "Toeplitz and circulant matrices: A review," *Found. Trends Commun. Inf.*, vol. 2, no. 3, pp. 155–239, Jan. 2006.
[36] R. C. Gonzalez and P. Wint, *Digital Image Processing*, vol. 28, no. 4. Upper Saddle River, NJ, USA: Prentice-Hall, 2001, pp. 484–486.
[37] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.
[38] X. Z. Wen, W. Fang, and Y. H. Zheng, "An algorithm based on Haar-like features and improved AdaBoost classifier for vehicle recognition," *Acta Electron. Sinica*, vol. 39, no. 5, pp. 1121–1126, May 2011.
[39] H. K. Galoogahi, T. Sim, and S. Lucey, "Multi-channel correlation filters," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3072–3079.
[40] C. Jiang, J. Xiao, Y. Xie, T. Tillo, and K. Huang, "Siamese network ensemble for visual tracking," *Neurocomputing*, vol. 275, pp. 2892–2903, Jan. 2018.
[41] K. Zhang and H. Song, "Real-time visual tracking via online weighted multiple instance learning," *Pattern Recognit.*, vol. 46, no. 1, pp. 397–411, Jan. 2013.

DONGZHU FENG was born in Shanxi, China, in 1977. She received the Ph.D. degree from Northwestern Polytechnical University, Shaanxi, in 2006.

She became an Associate Professor with Xidian University, Shaanxi, China, in 2009. She is currently an Associate Professor. Her research interests mainly include machine learning, object tracking, and visual navigation. In recent years, she has hosted and participated in many National Natural Science Funds. At the same time, a number of papers have been published in some famous journals and periodicals, such as the *Aerospace Science and Technology*.

FEIFEI GAO was born in Henan, China, in 1993. She received the B.E. degree from Henan Agricultural University, Henan, China, in 2016. She is currently pursuing the M.Sc. degree with Xidian University, Shaanxi, China. Her research interests include machine learning and visual object tracking.

XIN WANG was born in Shanxi, China, in 1977. He received the M.S. and Ph.D. degrees from Northwestern Polytechnical University, China, in 2003 and 2007, respectively.

He is currently an Associate Professor with Northwestern Polytechnical University, China. His current research is focused on design and simulation of control systems, optimization, and flight dynamics.

GUANGMIN WANG was born in Sichuan, China, in 1985. He received the B.E. degree in information and communication engineering and the Ph.D. degree in control science and engineering from Xi'an Jiaotong University, Xi'an, China, in 2008 and 2014, respectively. He became an Assistant Professor with Xidian University, Shaanxi, China, in 2015. His research interests are in the areas of adaptive filtering, statistical and array signal processing, and pattern recognition.

HAO DAI was born in Shaanxi, China, in 1985. He received the B.S. degree from the Department of Mathematics, Xidian University, Xi'an, China, in 2008, and the M.S. and Ph.D. degrees from the School of Electrical Engineering, Xi'an Jiaotong University, Xi'an, in 2011 and 2014, respectively.

He is currently an Assistant Professor with the School of Aerospace Science and Technology, Xidian University, Xi'an. His current research interests include complex networks, machine learning, and adaptive control.