IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 11, 2021, accepted February 12, 2021, date of publication February 16, 2021, date of current version February 24, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3059642

Multi-Feature Fusion Target Re-Location Tracking Based on Correlation Filters

QINGZHONG SHU[1,2], HUICHENG LAI[1,2], LIEJUN WANG[1,2], AND ZHENHONG JIA[1]
[1]College of Information Science and Engineering, Xinjiang University, Ürümqi 830046, China
[2]Key Laboratory of Signal Detection and Processing, Xinjiang University, Ürümqi 830046, China

Corresponding author: Huicheng Lai (lai@xju.edu.cn)

This work was supported by the National Science Foundation of China under Grant U1803261 and Grant U1903213.

ABSTRACT Target tracking has been a research hotspot in computer vision, and the correlation filtered target tracking algorithm has the benefits of low computational complexity and fast speed. Still, the tracking effect is not good when dealing with complicated circumstances. This paper proposes a multi-feature fusion target repositioning tracking algorithm for the target tracking problem in complex environments. First, a multi-feature weighted fusion algorithm is presented. Since each feature has different advantages in different environments, we combine HOG, CN, ULBP, and image edge features and use the weighted coefficient method to adaptively fuse each feature component. Second, to address the target occlusion problem, an occlusion judgment mechanism is introduced, and the target is re-located by fusion weighted filtering. Third, the scale pool is established, and the scale filter is trained by the classification search method. Finally, an adaptive model update strategy is proposed. We conduct comparison experiments with current mainstream algorithms on the publicly available datasets OTB-2015, VOT2018, UAV123, and TColor-128, respectively, and the experimental results show that our proposed algorithm is more robust in complex scenarios.

INDEX TERMS Target tracking, multi-feature fusion, target repositioning, model update.

I. INTRODUCTION

Studies of tracking algorithms based on literature [1]–[4] have shown that target tracking has always held an important place in the field of computer vision. The tracking process estimates the tracking target's position in the continuous video image sequence and determines its motion direction and trajectory information. However, there are often various complex factors in video scenes, such as changes in target's scale and shape, changes in light intensity, and the target is being obscured, etc., which lead to significant challenges in the practical application of the target tracking.

In the past few years, a breakthrough in target tracking has been achieved, mainly due to the introduction of filtering related to the communications domain into target tracking. Based on correlation filtering, some tracking algorithms have also been developed, which can reach hundreds of frames per second and can be widely used in real-time tracking systems.

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

In 2010, Bolme *et al.* applied the correlation filter to target tracking for the first time and proposed the Minimum Output Sum of Squared Error (MOSSE) correlation filter for target tracking, which converts the time-domain computation to the frequency domain computation and can achieve breakneck tracking speed [5]. 2012, based on the MOSSE algorithm, Henriques *et al.* introduced the concept of cyclic matrix and proposed the Cyclic Structural Kernel (CSK) algorithm, which solved the problem of sample redundancy caused by sparse sampling in the traditional algorithm [6]. Since then, cyclic matrix and kernel techniques have shone in the field of target tracking of correlation filtering. In 2014, Henriques *et al.* proposed Kernelized Correlation Filtering (KCF) algorithm using HOG features instead of grayscale features used in the original CSK algorithm to convert single-channel features to multiple channels [7]. Still, the KCF algorithm is less robust in the face of occlusion and scale variation [8]. In the same year, Danelljan *et al.* proposed the CN algorithm by replacing grayscale features with color name features based on the CSK algorithm. Simultaneously, to improve the algorithm's running speed, they used PCA dimensionality reduction to reduce the 11-dimensional

features to 2-dimensional ones and achieved better tracking results. However, the tracking robustness is low for cases where the target color is similar to the background color [9]. For enabling the tracker to adapt to changes in the tracking scene, in 2020, Yuan *et al.* proposed the TRBACF [10] algorithm based on BACF [11] to enhance the tracking's robustness and accuracy. To address the problem of scale variation during tracking, Danelljan *et al.* 2014 used a scale pooling strategy to estimate the target scale and proposed a scale pyramid-based prediction model, the Discriminative Scale Space Tracking (DSST) algorithm [12]. However, the DSST algorithm's scale estimation is inaccurate when the target scale is highly variable, and the algorithm requires high accuracy of the locator, resulting in low generalizability. For the tracking failure brought by a single feature, in 2014, Li *et al.* proposed fusing Gray features, HOG features, and CN features in the SAMF algorithm from the feature fusion aspect [13]. However, due to its comprehensive search strategy, the SAMF algorithm is computationally inefficient. In 2016 Bertinetto *et al.* proposed the Staple algorithm based on the DSST algorithm, which combines HOG features with global color histograms [14]. However, the algorithm is poorly useful for tracking complex environments such as occlusion. In 2019, Yuan *et al.* fused HOG, CN, and Gray features to propose the MFFT algorithm, which achieved a good result [15].

To improve the algorithm's tracking robustness when it encounters complex environments during tracking, we propose a multi-feature fusion target repositioning tracking algorithm. The contribution of this algorithm is described below:

A. Fusing HOG, CN, ULBP, and EDGE features to obtain new features for various complex environments.
B. Using fusion-weighted filtering for target re-localization when the target is occluded.
C. Constructing a scale pool to predict the target scale using classification search.
D. Propose an adaptive model update strategy.

II. KERNELIZED CORRELATION FILTER

A Correlation filtering algorithm is mainly used to find a linear regression equation $f(x_i) = \omega^T x_i$ through the training sample set to calculate the weight coefficient ω to minimize the error between the result obtained by linear regression and the sample's real value. We use the sum of the squares of errors as the loss function, and the form of can be solved as follows:

$$\min_{\omega} \sum_{i} (f(x_i) - y_i) + \lambda \|\omega\|^2, \tag{1}$$

where x_i is the training sample, y_i is the label to which the sample corresponds, and λ is the regularization coefficient to prevent overfitting accessions in the training process. By taking the bias derivative of equation (1) we get the general solution as:

$$\omega = \left(X^T X + \lambda I\right)^{-1} X^T y, \tag{2}$$

where X is a matrix of training samples x_i, and each row of the matrix represents one sample x_i; I represents a matrix of units with the same dimension as X, and y is the label corresponding to the training sample x_i.

We have difficulty finding a plane for the tracked target to separate the background from the target, so we need to map the sample into the high dimensional space by nonlinear mapping $x_i \to \varphi(x_i)$, which makes it linearly separable. The weight vector at this point is expressed as follows:

$$W = \sum_{i} \alpha_i \varphi(x_i), \tag{3}$$

From equations (2), (3) the expression for a can be obtained as follows:

$$\alpha = (k + \lambda I)^{-1} y, \tag{4}$$

where k denotes the kernel correlation coefficient between samples, expressed as follows:

$$k(x_i, x_j) = \varphi(x_i)^T \varphi(x_j), \tag{5}$$

simultaneous Fourier transformations for both sides of equation (4) are as follows:

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

where $\hat{k}^{xx}$ denotes the Fourier transform of the kernel matrix $K = \langle \varphi(x) \varphi(x) \rangle$.

Since the algorithm's training sample is obtained by the cyclic shift of the initial target sample, the kernel correlation matrix between the training samples is a cyclic matrix. For the test sample, its corresponding response output is as follows:

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

In equation (7), $\hat{k}^{xz}$ represents the Fourier transform of the nuclear matrix $K = \langle \varphi(x) \varphi(z) \rangle$.The coordinates corresponding to the maximum value obtained by taking the inverse Fourier transform of equation (7) are the target's predicted positions.

III. RELATED WORK

A. MULTI-FEATURE FUSION METHODS

The selection and extraction of features significantly impact target tracking results, whereas traditional target tracking algorithms use a single feature. HOG features consist of histograms that compute and count the gradient directions of local regions. Since HOG features run on the image's local grid cells and capture the target contours, they can better accommodate interference from geometric distortion and background color similarity. However, they are insensitive to occlusion and less robust to motion blur [16]. CN features use a probabilistic mapping method to transform the image from the original 3-dimensional RGB space to an 11-dimensional color feature space to take full advantage of the target's color features. The feature is robust to motion blur and light intensity variation. Still, it poorly adapts to similar background colors [17], and literature [18] shows that fusing HOG features

with CN features gives better tracking results. The traditional LBP feature is an algorithm used to describe an image's local texture features, reflecting the texture changes around the image pixels [19], [20]. It has the advantage of being insensitive to image rotation and illumination changes, but it is computationally complex. ULBP feature [21] improves the LBP feature, which uses Uniform Pattern to downscale the LBP operator's pattern types, reducing the computational complexity without losing any information. The Edge features are generally found in areas of the image where the brightness changes drastically. They are advantageous in separating the background and can adapt well to changes in light intensity [22].

In this paper, to improve the algorithm's robustness for tracking in complex environments, a multi-feature fusion algorithm is proposed, which is weighted by calculating the response values of HOG, CN, ULBP, and EDGE features separately.

Firstly, four features of the image are extracted for training, and according to Eq. (7), the training formula is as follows:

$$\hat{f}\left(z_{\text{feature}}\right) = \hat{\alpha}_{\text{feature}} \odot \hat{k}^{xz}_{\text{feature}}, \tag{8}$$

According to the training of Eq. (8), the maximum output response values of the four position-correlated filters can be obtained, which can be expressed as follows:

$$\Phi_{\text{feature}} = \arg\max F^{-1}\left(\hat{f}\left(z_{\text{feature}}\right)\right), \tag{9}$$

where Φ_{feature} denotes the maximum output response of each feature, and the corresponding coordinates are the predicted target locations.

From Eq. (4), it can be concluded that the distance between the training sample and the actual location of the target is related to the size of the filter response value; the closer the distance, the larger the response value, and the farther the distance, the smaller the response value. So we can exploit the difference between the maximum response values of different filters for feature fusion, expressed as follows:

$$\beta_{\text{feature}} = \frac{\max\left(f\left(Z_{\text{feature}}\right)\right)}{f\left(Z\right)}, \tag{10}$$

where β_{feature} denotes each feature's weighting coefficients, respectively, and $f\left(Z\right)$ denotes the sum of the four feature response values. According to Eq. (10), the formula for the target's final predicted position could be obtained as follows:

$$\Phi = \beta_{\text{feature}}\Phi_{\text{feature}}, \tag{11}$$

B. SCALE ESTIMATION METHODS

The scale change affects the accuracy of the tracking algorithm. In the tracking process, the tracking box fails to fully include the target when the target becomes large, resulting in losing part of the target information. When the target becomes small, the tracking box contains the target and other objects other than the target, resulting in an increase of interference information, which will lead to the failure of tracking. Inspired by literature [23], we propose to create a scale pool using a categorical search method to obtain the scale adaptiveness of the algorithm through scale estimation.

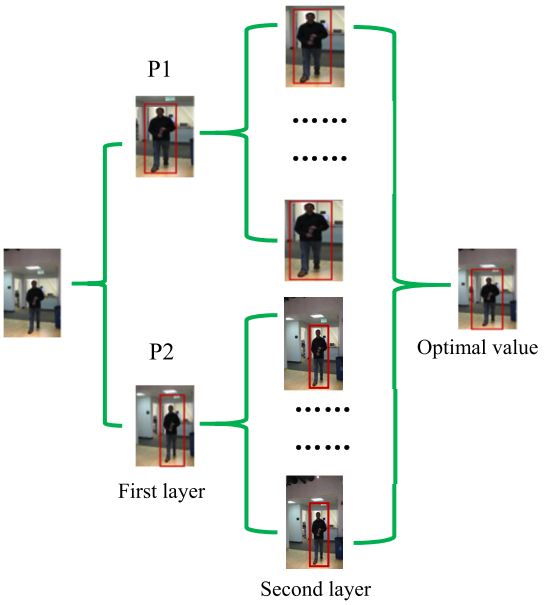

FIGURE 1. Schematic of the scale classification search method.

Using Eq. (11) to calculate the central coordinates of the target position, scale based on the central coordinates, as shown in Fig. 1.

We use the first layer to determine whether the target scale is amplifying or shrinking. If the P1 response value is greater than the P2 response value, then the current target scale is amplifying, and if the P1 response value is less than the P2 response value, then the current target scale is shrinking. If the target scale is amplified, the scale change is calculated through the P1 branch; if the target scale is shrunk, the scale change is calculated through the P2 branch. We set 12 scale filters in the scale pool.

The classification search method is used for scale estimation in this paper, with 12 scale comparisons per branch, requiring 14 scale operations per updated frame. This can reduce 19 unnecessary operations and improve the algorithm's speed compared to the 33 scale changes of the DSST algorithm. It can cover a broader range of scales and improve the algorithm's accuracy compared to the seven mesoscale changes of the SAMF algorithm.

C. OCCLUSION HANDLING METHODS

1) OCCLUSION DETECTION

To improve the target's tracking robustness during occlusion, we introduced an occlusion judgment mechanism inspired by the literature [24], [25], as shown in Figure 2.

As shown in Figure 2(c), if the tracking target is not occluded, the tracker's response graph will show a single peak. As shown in Figure 2(d), if the tracking target is occluded, the tracker's response graph will oscillate violently, and multiple peaks will appear in the response graph. According to Figure 2, by calculating the oscillation degree of the

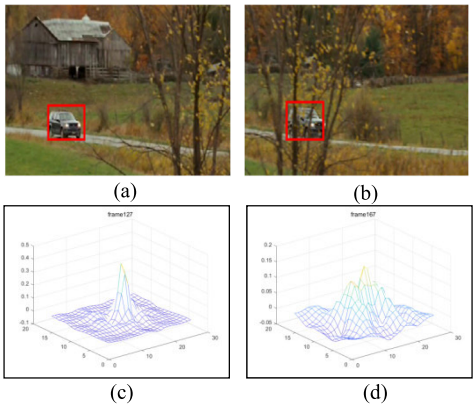

FIGURE 2. Tracker response peak graph.

crest, we can judge whether the target is blocked or not. When the difference between two adjacent frames of the video sequence is less than the threshold value, it means that the target is blocked. The specific calculation formula is as follows:

$$\Psi = \frac{|f_{\max} - f_{\min}|^2}{mean\left(\sum_{w,h}\left(f_{w,h} - f_{\min}\right)^2\right)}, \tag{12}$$

$$[\Psi(i) - \Psi(i-1)] - \mu \begin{cases} < 0 & OCC = 1, \\ \geq 0 & OCC = 0, \end{cases} \tag{13}$$

where $f_{\max}$ represents the maximum response value, $f_{\min}$ represents the minimum response value, $f_{w,h}$ represents the response value at position (w, h), $\Psi(i)$ and $\Psi(i-1)$ represent the peak oscillation degree of i frame and $i-1$ frame respectively, μ represents the occlusion threshold, OCC equals one means the target is occluded, OCC equals zero means the target is not occluded.

2) TARGET RETARGETING

When the target is judged to be occlusion, we introduce a weighted window filter to reposition the target. The prediction of the weighted window filter is usually divided into three stages. First, the weighted window filter is initialized, and the weights of the window filter are set by asymptotic memory p, expressed as follows:

$$p = (p_1, p_2, \ldots, p_{r-1}), \tag{14}$$

After starting tracking, the first r frames are collected to form the target position window d, expressed as follows:

$$d = (d_1, d_2, \ldots, d_{r-1}), \tag{15}$$

When the target is occluded in the t frame, the window d is used to obtain the coordinate difference d' of the adjacent frames, and the window filter predicts the target position offset.

$$d' = (d_2 - d_1, d_3 - d_2, \ldots, d_r - d_{r-1}), \tag{16}$$

$$\Delta d = d' \times p^T, \tag{17}$$

We can obtain the target position d_{t+1} in frame $t+1$ through the target position d_t in frame t and the predicted offset Δd. The specific calculation formula is as follows:

$$d_{t+1} = d_t + \Delta d, \tag{18}$$

In this paper, the target position is stored in the filter window, the data is analyzed to predict the next frame's target position. Compared with other methods that only use the previous frame's information to predict the next frame's target position, this method has better resistance to the large area and long-time blocking.

D. MODEL UPDATE METHODS

Traditional correlation filtering algorithms use fixed parameters to update the target model, which is prone to accumulate tracking bias and decrease tracking accuracy. To improve the accuracy of the algorithm, we introduce an adaptive dynamic update model method. The specific calculation formula is as follows:

$$\begin{cases} \alpha_t = (1 - \theta \cdot \eta) \cdot \alpha_{t-1} + \theta \cdot \eta \cdot \alpha_t, \\ x_t = (1 - \theta \cdot \eta) \cdot x_{t-1} + \theta \cdot \eta \cdot x_t, \end{cases} \tag{19}$$

where x_t and x_{t-1} are the target feature models for frame t and frame t-1, respectively, α_t and α_{t-1} are the coefficient matrices for frame t and frame t-1, and η is the learning coefficient.

$$\theta = \begin{cases} 1 & \Psi \geq TH1, \\ \left(\dfrac{\Psi - TH2}{TH1 - TH2}\right)^2 & TH2 < \Psi < TH1, \\ 0 & \Psi \leq TH2, \end{cases} \tag{20}$$

where θ is the model interpolation weight and TH1 and TH2 are the thresholds for Ψ, respectively.

When the value of Ψ is greater than or equal to TH1, the tracking result is completely reliable, and the interpolation weight θ can be set to 1.When the value of Ψ is less than or equal to TH2, the tracking result is wrong, and the interpolation weight θ can be set to 0 (the model is not updated).When the value of Ψ is between TH1 and TH2, we can dynamically adjust the weight coefficient according to the value of Ψ.

E. ALGORITHM FLOW CHART IN THIS PAPER

The specific flow of the algorithm in this paper is shown in Figure 3.

IV. RESULTS AND DISCUSSION

The proposed method is implemented in MATLAB2017b and runs at 24+ frames per second on a PC with an Intel Core-i7-7700 CPU (3.60 GHz) and 8 GB RAM.That is the whole algorithm running time is about 0.04 seconds. The initialization parameter λ is set to 0.001, η is set to 0.002, TH1, TH2 is set to 6 and 4, respectively, and the occlusion threshold μ is set to 20.

To demonstrate the effectiveness of the algorithm in this paper, we conducted comparative experiments with other

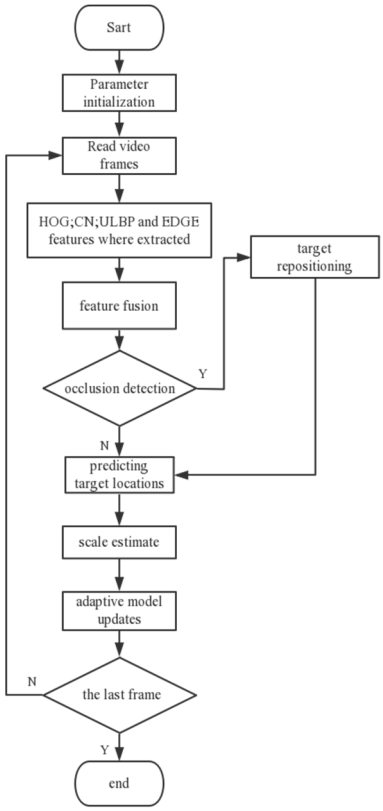

FIGURE 3. Algorithm flow chart in this paper.

mainstream algorithms on the publicly available data sets OTB-2015 [26], VOT2018 [27], UAV123 [28], and TColor-128 [29] to analyze the overall performance and the performance in different complex environments, respectively.

A. COMPARATIVE EXPERIMENT ON OTB

To verify our proposed algorithm's validity, we compare it with the KCF, DSST, Staple, SAMF, DeepSRDCF and DeepSTRCF algorithms on the data set OTB-2015, respectively.

∗ The KCF algorithm uses only HOG features and has no scale adaptation.

∗ The DSST algorithm uses only HOG features and has scale adaptation,

∗ The Staple algorithm combines HOG and global color histogram features and has scale adaptation.

∗ The SAMF algorithm combines Gary, HOG, and CN features and has scale adaptation.

∗ The DeepSRDCF and DeepSTRCF algorithm combines depth features and has scale adaptation.

1) EVALUATION METRICS

The OTB data set is the most widely used data set in target tracking and mainly consists of two versions, OTB-2013 and OTB-2015. The evaluation criteria use two metrics: precision and success rate. Accuracy is the percentage of the number of frames with CLE (Center Location Error) less than a certain threshold to the video sequence's total number of frames.

$$CLE = \sqrt{(x_P - x_T)^2 + (y_P - y_T)^2}, \tag{21}$$

where CLE is the Euclidean distance between the center position coordinate (x_P, y_P) of the tracked target and the actual target center position coordinate (x_T, y_T).

The success rate evaluation indicator is the percentage of frames where the OS between the tracked target area R_P and R_T the real target area is more significant than a particular threshold value over the total number of frames in the video sequence, where the OS is expressed as follows:

$$OS = \frac{|R_P \cap R_T|}{|R_P \cup R_T|}, \tag{22}$$

where R_P denotes the tracking region of the current frame, R_T denotes the standard target region, $\cap$ denotes the intersection of the two regions, $\cup$ denotes the union of the two regions.

2) QUALITATIVE ANALYSIS

We selected six subsets (in the order of Jogging1, Bird1, Soccer, Singer1, Bolt, and Tiger2) from the OTB-2015 data for comparison experiments. These six subsets represent multiple complex scenes of short-time target occlusion, long-time target occlusion, background clutters, scale variation, fast motion, and illumination variation. Figures 4, 5, and 8 show that multiple targets appear in the occlusion, complex background, and fast motion tracking scene, and the algorithm proposed in this paper can maintain excellent robustness. The results of the experiments are as follows:

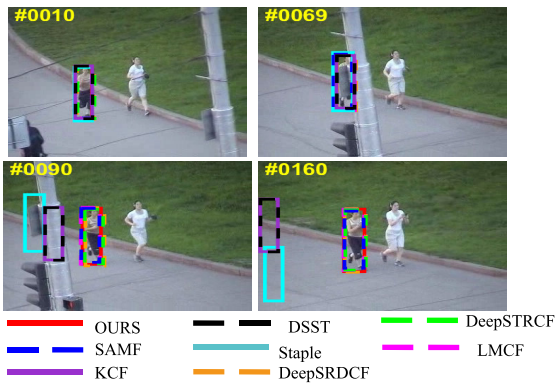

FIGURE 4. The tracking result of eight trackers at the Jogging1.

a: SHORT-TIME TARGET OCCLUSION

As shown in Figure 4, the target encounters occlusion at frame 69, and the tracking of Staple, DSST, and KCF algorithms Fail. The algorithm in this paper can always track the target stably.

b: LONG-TIME TARGET OCCLUSION

As shown in Figure 5, the target occludes for a long time in 126-133 frames. Compared with other algorithms. The tracking accuracy of the algorithm proposed in this paper is the highest.

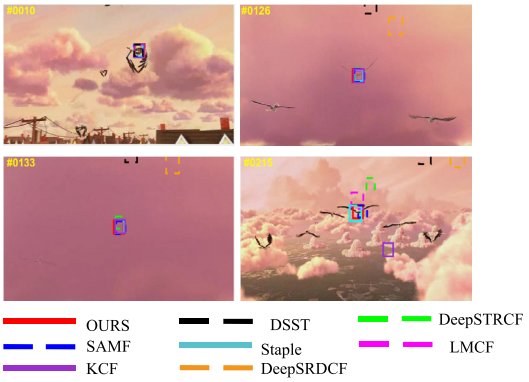

FIGURE 5. The tracking result of eight trackers at the Bird1.

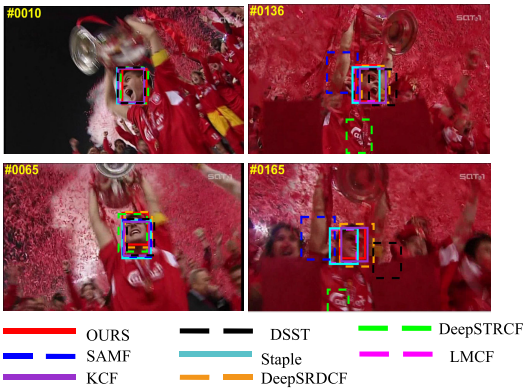

FIGURE 6. The tracking result of eight trackers at the Soccer.

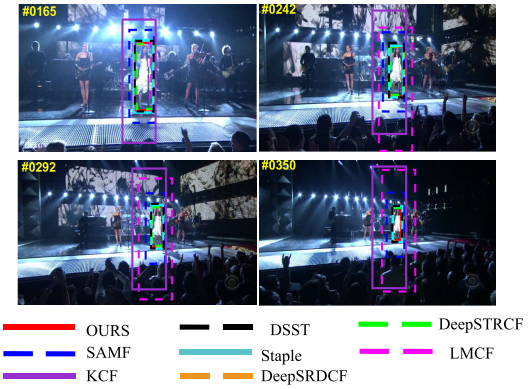

FIGURE 7. The tracking result of eight trackers at the Singer1.

c: BACKGROUND CLUTTERS

As shown in Figure 6, Objects similar to the target appear in the background from frame 10 to frame 165. The algorithm in this paper has strong robustness.

d: SCALE CHANGING

As shown in Figure 7, the target scale changes significantly from 165 to 350 frames, and this paper's algorithm always has better tracking accuracy than other algorithms.

e: TARGET FAST MOVING

As shown in Figure 8, when the target moves fast, this paper's algorithm can always track the target accurately and better than other algorithms.

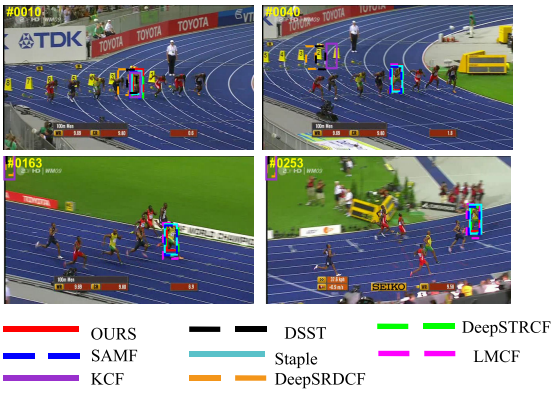

FIGURE 8. The tracking result of eight trackers at the Bolt.

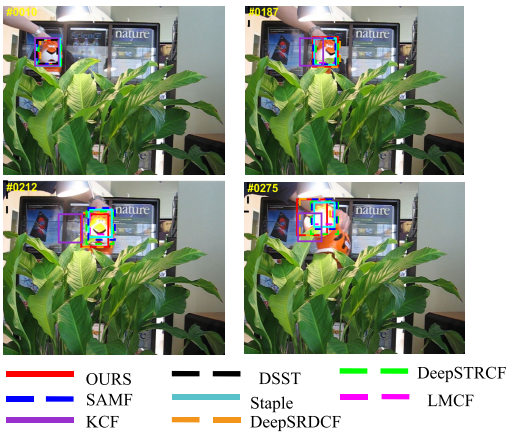

FIGURE 9. The tracking result of eight trackers at the Tiger2.

TABLE 1. Average performance of algorithms on the OTB-2015 data set.

Tracker	Precision (Threshold)	Success rate (AUC)	Speed (FPS)
OURS	77.4%	62.4%	27.27
DSST	64.3%	46.1%	18.14
KCF	62.2%	41.5%	**162.87**
SAMF	73.0%	50.0%	20.85
Staple	75.7%	53.0%	33.94
LMCF	71.8%	50.2%	78.21
DeepSTRCF	**88.1%**	**67.5%**	4.26
DeepSRDCF	78.9%	59.8%	0.32

f: ILLUMINATION VARIATION

As shown in Figure 9, the light changes significantly at 212 frames, and the algorithm in this paper can still track the target accurately.

g: QUANTITATIVE COMPARISON

As shown in Table1, the average performance metrics obtained for all algorithms tested on OTB-2015.

The bold character indicates that the performance of the current tracker ranks first in the comparison process. As can be seen from Table1, the accuracy of the algorithm proposed

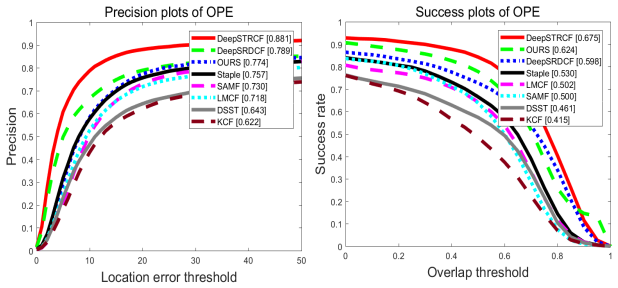

FIGURE 10. Precision and success rate plots of different algorithms on the OTB-2015 data set.

in this paper ranks third, second only to DeepSTRCF [30] and DeepSRDCF [31] algorithm using the deep learning method. And the success rate ranks second, second only to DeepSTRCF. The algorithm proposed in this paper is tens of times faster than DeepSTRCF and DeepSRDCF algorithm in terms of speed, achieving real-time performance, as shown in Fig 10 is the precision and success rate curve of different algorithms on the OTB-2015 data set.

B. COMPARATIVE EXPERIMENT ON VOT2018

We use the VOT2018 data set, which contains 60 test videos, to objectively compare and analyze this algorithm's performance with other algorithms. Compared with the OTB data set, tracking is more complicated. On the VOT2018 dataset, we compare it with KCF, DSST, Staple, ECO [32], LADCF [33], and UPDT [34], in which ECO, LADCF, and UPDT all introduce depth features.

1) EVALUATION METRICS

The VOT data set is evaluated differently from the OTB data set. The OTB data set focuses on the algorithm's long-term tracking ability and is initialized only once during testing. Simultaneously, the VOT selects sequences that are more difficult to track, re-initializes them after each tracking failure, and continues to count the overlap rate after re-initializing the frame.

The performance of the algorithm in this paper is evaluated using A-R (Accuracy Robustness), Failures, EAO (Expected Average Overlap), and EFO (Equivalent Filter Operations). A-R is the abbreviation of accuracy robustness, in which accuracy evaluates the overlap rate between the predicted result and the actual state of the tracker in each frame. In contrast, robustness considers each sequence's average failure times, and VOT2018 calculates the corresponding average value by using the result of the tracker running 15 times on the sequence. Failures represent tracking failure statistics. When the overlap is below the threshold, the algorithm's tracking is considered to have failed; EAO represents the expected average overlap rate. The higher the value, the more accurate the tracker is. The specific statistical method is to

∗ Intercept short clips in the test video

∗ Perform a one-time tracking using an uninitialized method

TABLE 2. Average performance of algorithms on the VOT2018 data set.

Tracker	A-R	Failures	EAO	EFO
OUR	0.5080	11.9009	0.3194	15.4127
ECO	0.4978	13.5112	0.3077	0.8806
LADCF	0.5337	**7.4410**	**0.4016**	0.1116
UPDT	**0.5507**	8.2848	0.3919	0.0845
DSST	0.4005	63.0723	0.0976	12.7101
KCF	0.4721	30.1225	0.1780	**30.3514**
Staple	0.5405	19.8836	0.2733	16.497

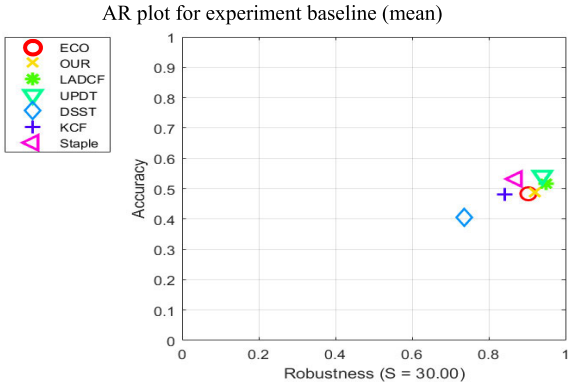

FIGURE 11. Accuracy robustness plots of different algorithms on the VOT2018 data set.

∗ Calculate the average overlap rate of the algorithm on the clips

∗ And finally, calculate the expectation value of the average overlap rate for several different lengths.

EFO is an evaluation of the speed; the larger the value, the faster the tracker.

2) QUANTITATIVE COMPARISON

As shown in Table2, the average performance metrics obtained for all algorithms tested on VOT2018.

The bold font indicates that the current tracker ranks first in comparison with the benchmark algorithm. Table 2 shows that the A-R of the algorithm proposed in this paper ranks the fourth, and the third in failures and EAO indexes, which is only worse than LADCF and UPDT algorithm. In EFO indexes, the algorithm proposed in this paper ranks third, which is tens of times faster than the algorithm using depth features. Therefore, the comprehensive performance of the proposed algorithm is the best. Figure 11 shows the accuracy robustness comparison of different trackers in VOT2018.

As shown in Figure 11, the algorithm proposed in this paper ranks fourth in accuracy among all the test algorithms, which is worse than LADCF, UPDT, and Stable algorithm. The Third is robustness, which is only worse than the LADCF and UPDT algorithm. Figures 12 and 13 show

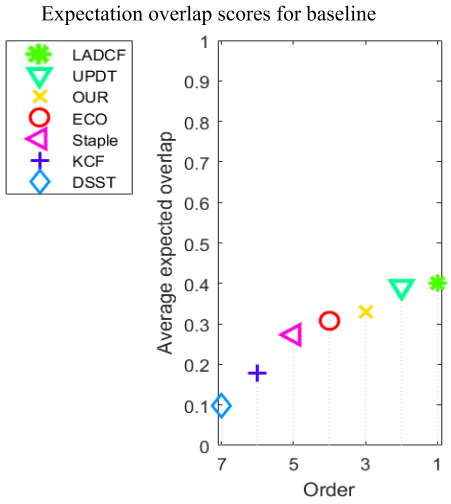

FIGURE 12. Average overlap expectation scores graph.

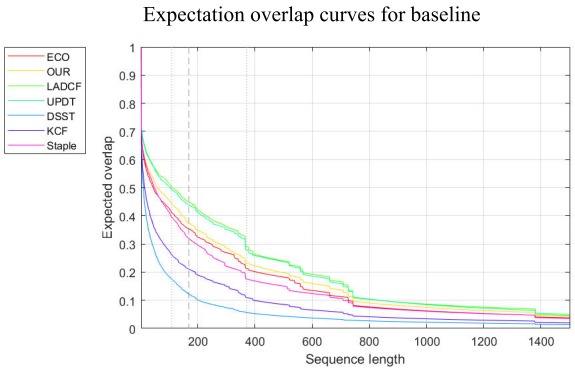

FIGURE 13. Average overlap expectation curve comparison plot.

TABLE 3. Average performance of algorithms on the UAV123 data set.

Tracker	Precision (Threshold)	Success rate (AUC)	Speed (FPS)
OURS	70.7%	50.9%	23.71
SiamRCNN	**83.4%**	**64.9%**	4.35
ECO	74.1%	52.2%	1.46
SRDCF	67.6%	46.4%	5.23
UDT	67.3%	48.0%	67.28
MEEM	62.7%	39.2%	7.71
SAMF	59.2%	39.6%	19.32
MUSTER	59.1%	39.1%	1.24
DSST	58.6%	35.6%	16.15
KCF	52.3%	33.1%	**146.23**

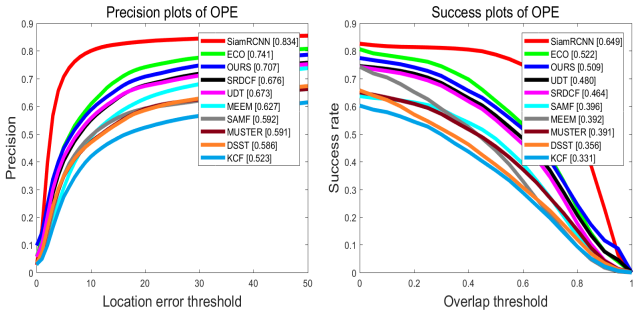

FIGURE 14. Precision and success rate plots of different algorithms on the UAV123 data set.

the ranking of each tracker's expected overlap performance and the change of expected overlap with sequence length, respectively.

As shown in Figure 12, the algorithm proposed in this paper has the third-highest average expected overlap rate score compared to the other algorithms. Figure 13 shows that the expected average overlap rate of all the compared algorithms gradually decreases as the tracking sequence's length increases. Because the longer the sequence length, the more errors are accumulated, resulting in a worse average performance of the algorithm in longer sequences. The average overlap rate of the algorithms proposed in this paper decreases faster than the LADCF and UPDT algorithms and performs better than the other algorithms.

C. COMPARATIVE EXPERIMENT ON UAV123

1) DATA SET INTRODUCE AND EVALUATE INDEX

The UAV123 dataset consists mainly of 91 UAV videos, several of which are long and split into three or four shorter segments, used several times. Hence, there are 123 video sequences in total, and this dataset is characterized by a clean background and more variations in viewpoint. The UAV123 dataset uses the same evaluation metrics as the OTB, which also uses the two metrics of precision and success rate to evaluate the algorithm performance.

2) QUANTITATIVE COMPARISON

To demonstrate the effectiveness of the algorithms proposed in this paper, we compared them with the KCF, DSST, MUSTER [35], SAMF, MEEM [36], UDT [37], SRDCF [38], ECO, and SiamRCNN [39]algorithms on the UAV123 dataset, respectively. Table 3 shows the average performance metrics obtained for all algorithms tested on UAV123. The bold character indicates that the current tracker ranks first compared with other algorithms.

As shown in Table 3 and Figure 14, the algorithm proposed in this paper ranks third in precision and success rate, and it's only inferior to SiamRCNN and ECO algorithms. In terms of speed, the algorithm proposed in this paper ranks third, only inferior to KCF and UDT algorithms. In terms of overall performance metrics, the algorithm proposed in this paper has the best overall performance.

TABLE 4. Average performance of algorithms on the TColor-128 data set.

Tracker	Precision (Threshold)	Success rate (AUC)	Speed (FPS)
OURS	71.2%	52.0%	25.42
DSST	53.5%	37.6%	21.47
SAMF	63.3%	46.2%	23.74
SITUP	63.9%	47.1%	31.35
MUSTER	64.1%	47.2%	2.63
UDT	65.8%	50.4%	**69.32**
BACF	66.0%	49.6%	30.13
Staple	66.8%	50.9%	29.31
SRDCF	69.6%	51.5.%	7.15
ECO	**79.2%**	**59.9%**	2.28

D. COMPARATIVE EXPERIMENT ON TColor-128

1) DATA SET INTRODUCE AND EVALUATE INDEX

TColor-128 is a benchmark dataset dedicated to color vision tracking, consisting of 50 color sequences frequently tested in previous studies and 78 color sequences collected from the Internet. These 128 sequences have many challenging factors, such as complete target occlusion, high illumination variation, high target distortion, and low resolution. The dataset uses the same evaluation method as the OTB dataset, which also uses two evaluation metrics, precision and success rate, to evaluate the algorithm's performance.

E. QUANTITATIVE COMPARISON

To verify the effectiveness of the algorithm proposed in this paper, we also test and verify our proposed tracker on the TColor-128 benchmark against 9 state-of-the-art trackers, including DSST, SAMF, SITUP [40], MUSTER, UDT, BACF, Staple, SRDCF, and ECO.

As can be seen from Figure 15 and Table 4, compared with other trackers, the algorithm proposed in this paper ranks second in the precision and success rate indicators, which is only worse than the ECO algorithm using the deep learning method. In terms of speed, the algorithm proposed in this paper is tens of times faster than the ECO algorithm and has an excellent comprehensive performance.

F. MULTIPLE TARGETS IN COMPLEX SCENES

To verify the proposed algorithm's tracking effect in complex scenes with multiple targets, we selected several such scenes on the OTB-2015 dataset. The tracking effects are shown in Figures 4,5,6,8 and 16.

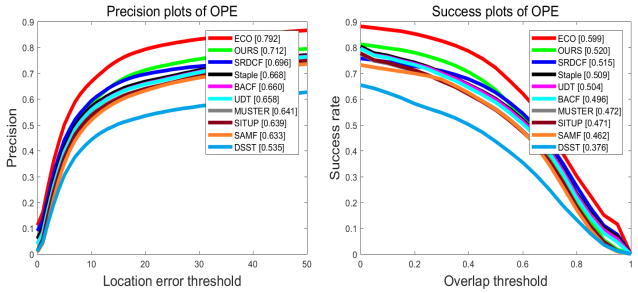

FIGURE 15. Precision and success rate plots of different algorithms on the TCOLOR-128 data set.

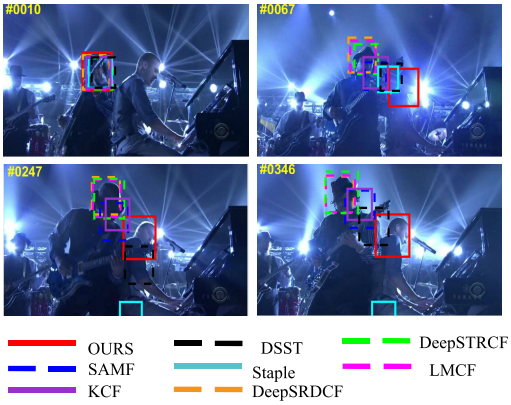

FIGURE 16. The tracking result of eight trackers at the Shaking.

From Figs. 4, 5, 6, 8, and 16, it can be seen that the algorithms proposed in this paper track well in scenes with multiple targets in occlusion and fast motion. But the algorithms proposed in this paper fail to track when multiple targets are present in scenes with complex factors such as illumination, in-plane rotation, and high target similarity. The main reason for the tracking failure is that the conventional features are not as good as the in-depth features.

V. CONCLUSION

This paper proposes a multi-feature fusion target repositioning tracking algorithm to address the problem of low robustness of correlation filter tracking in complex environments.

1. The HOG, CN, ULBP, and edge features under different conditions are fully utilized to obtain new features by weighted fusion of the calculated response values.

2. A scale pool is constructed to estimate the target scale by classification search to improve the algorithm's computational speed.

3. A blocking judgment mechanism is introduced, while a weighted window filter is used to reposition the target

4. Finally, an adaptive update method is used to update the model.

To verify the proposed algorithm's effectiveness, we have done comparison experiments with the state-of-the-art algorithms on OTB-2015, VOT-2018, UAV-123, and Tcolor-

128 datasets, respectively. The experimental results show that the proposed algorithm improves the tracking accuracy and success rate while ensuring the algorithm's real-time performance, which has good comprehensive performance and high practical value. However, due to the shortcomings of traditional features, this paper's method is not very effective in some scenes. The next work will explore the combination of correlation filtering with depth features for target tracking.

REFERENCES

[1] D. Yuan, X. Lu, D. Li, Y. Liang, and X. Zhang, "Particle filter re-detection for visual tracking via correlation filters," *Multimedia Tools Appl.*, vol. 78, no. 11, pp. 14277–14301, Jun. 2019.

[2] T. Liu, G. Wang, and Q. Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4902–4912.

[3] D. Yuan, W. Kang, and Z. He, "Robust visual tracking with correlation filters and metric learning," *Knowl.-Based Syst.*, vol. 195, May 2020, Art. no. 105697.

[4] D. Yuan, X. Chang, P.-Y. Huang, Q. Liu, and Z. He, "Self-supervised deep correlation tracking," *IEEE Trans. Image Process.*, vol. 30, pp. 976–985, 2021.

[5] D. 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.*, San Francisco, CA, USA, Jun. 2010, pp. 2544–2550.

[6] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 702–715.

[7] 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. 2015.

[8] F. Xu, H. Wang, Y. Song, and J. Liu, "A multi-scale kernel correlation filter tracker with feature integration and robust model updater," in *Proc. 29th Chin. Control Decis. Conf. (CCDC)*, May 2017, pp. 1934–1939.

[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] D. Yuan, X. Shu, and Z. He, "TRBACF: Learning temporal regularized correlation filters for high performance online visual object tracking," *J. Vis. Commun. Image Represent.*, vol. 72, Oct. 2020, Art. no. 102882.

[11] H. K. Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1135–1143.

[12] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf.*, vol. 65, 2014, pp. 1–11.

[13] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 254–265.

[14] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1401–1409.

[15] D. Yuan, X. Zhang, J. Liu, and D. Li, "A multiple feature fused model for visual object tracking via correlation filters," *Multimedia Tools Appl.*, vol. 78, no. 19, pp. 27271–27290, Oct. 2019.

[16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.

[17] Y. Xu, H. Li, Y. Li, J. Wang, W. Xu, Y. Zhang, and Z. Miao, "Combining color attributes for scale adaptive correlation tracking," in *Proc. 3rd Int. Conf. Inf. Sci. Control Eng. (ICISCE)*, Jul. 2016, pp. 267–271.

[18] X. Yang, H. Zhang, L. Yang, C. Yang, and P. X. Liu, "A joint multi-feature and scale-adaptive correlation filter tracker," *IEEE Access*, vol. 6, pp. 34246–34253, 2018.

[19] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[20] Z. Shu, G. Liu, and Z. Xie, "Real time target tracking scale adaptive based on LBP operator and nonlinear meanshift," in *Proc. Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discovery (CyberC)*, Oct. 2017, pp. 130–133.

[21] L. L. Thike and T. L. L. Thein, "Parking space detection using complemented-ULBP background subtraction," in *Proc. IEEE 8th Global Conf. Consum. Electron. (GCCE)*, Oct. 2019, pp. 894–896.

[22] W. Xiong, X. Nie, X. Zou, Z. Yang, and X. He, "Face illumination invariant feature extraction based on edge detection operator," in *Proc. IEEE Int. Conf. Imag. Syst. Techn. (IST)*, Oct. 2017, pp. 1–5.

[23] E. Gundogdu, H. Ozkan, and A. A. Alatan, "Extending correlation filter-based visual tracking by tree-structured ensemble and spatial windowing," *IEEE Trans. Image Process.*, vol. 26, no. 11, pp. 5270–5283, Nov. 2017.

[24] C. Ma and G. Yu, "An improved kernel correlation filter for occlusion target tracking," in *Proc. IEEE 4th Int. Conf. Image, Vis. Comput. (ICIVC)*, Jul. 2019, pp. 674–678.

[25] S. Li, J. Chu, G. Zhong, L. Leng, and J. Miao, "Robust visual tracking with occlusion judgment and re-detection," *IEEE Access*, vol. 8, pp. 122772–122781, 2020.

[26] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.

[27] M. Kristan *et al.*, "The sixth visual object tracking VOT2018 challenge results," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2018, pp. 3–53.

[28] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," *Far East J. Math. Sci.*, vol. 2, no. 2, pp. 445–461, 2016.

[29] P. Liang, E. Blasch, and H. Ling, "Encoding color information for visual tracking: Algorithms and benchmark," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5630–5644, Dec. 2015.

[30] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, "Learning spatial-temporal regularized correlation filters for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 4904–4913.

[31] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 58–66.

[32] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6931–6939.

[33] T. Xu, Z.-H. Feng, X.-J. Wu, and J. Kittler, "Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5596–5609, Nov. 2019.

[34] G. Bhat, J. Johnander, M. Danelljan, F. S. Khan, and M. Felsberg, "Unveiling the power of deep tracking," in *Proc. ECCV*, Sep. 2018, pp. 483–498.

[35] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "MUlti-store tracker (MUSTer): A cognitive psychology inspired approach to object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 749–758.

[36] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust tracking via multiple experts using entropy minimization," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 188–203.

[37] N. Wang, Y. Song, C. Ma, W. Zhou, W. Liu, and H. Li, "Unsupervised deep tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1308–1317.

[38] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4310–4318.

[39] P. Voigtlaender, J. Luiten, P. H. S. Torr, and B. Leibe, "Siam R-CNN: Visual tracking by re-detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 6577–6587.

[40] H. Ma, S. T. Acton, and Z. Lin, "SITUP: Scale invariant tracking using average peak-to-correlation energy," *IEEE Trans. Image Process.*, vol. 29, pp. 3546–3557, 2020.

QINGZHONG SHU received the B.S. degree from the Suzhou College, Suzhou, China, in 2016. He is currently pursuing the master's degree with Xinjiang University, Ürümqi, China. His research interests include target tracking and image recognition.

HUICHENG LAI received the B.E. and M.S. degrees from Xinjiang University, China, in 1986 and 1990, respectively. He is currently a Professor with Xinjiang University. His current research interests include image processing, image recognition, image enhancement, image restoration, and communications technology.

LIEJUN WANG received the Ph.D. degree from the School of Information and Communication Engineering, Xi'an Jiaotong University, in 2012. He is currently a Professor with the School of Information Science and Engineering, Xinjiang University. His research interests include wireless sensor networks, encryption algorithm, and image intelligent processing.

ZHENHONG JIA received the B.S. degree from Beijing Normal University, Beijing, China, in 1985, and the M.S. and Ph.D. degrees from Shanghai Jiao Tong University, Shanghai, China, in 1987 and 1995, respectively. He is currently a Professor with the Key Laboratory of Signal and Information Processing, Xinjiang University, China. His research interests include digital image processing, and photoelectric information detection and sensors.