Received: 26 October 2021 | Revised: 21 December 2021 | Accepted: 10 February 2022

IET Cyber-Systems and Robotics

DOI: 10.1049/csy2.12042

ORIGINAL RESEARCH

Long-time target tracking algorithm based on re-detection multi-feature fusion

Junsuo Qu[1] | Chenxue Tang[1] | Yuan Zhang[2] | Kai Zhou[2] | Abolfazl Razi[3]

[1]School of Automation, Xi'an Key Laboratory of Advanced Control and Intelligent Process, Xi'an University of Posts and Telecommunications, Xi'an, Shaanxi, China

[2]School of Communication and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an, Shaanxi, China

[3]School of Informatics, Computing and Cyber Systems, Northern Arizona University, Flagstaff, AZ, USA

Correspondence

Junsuo Qu, School of Automation, Xi'an Key Laboratory of Advanced Control and Intelligent Process, Xi'an University of Posts and Telecommunications, Xi'an, Shaanxi, 710121, China.
Email: qujunsuo@xupt.edu.cn

Funding information

International Cooperation and Exchange Program of Shaanxi Province, Grant/Award Number: 2022KW-04; Natural Science Foundation of Shaanxi Province, Grant/Award Number: 2018JM6120; Xi'an Science and Technology Plan Project, Grant/Award Number: 21XJZZ0072; Major Science and Technology Projects of Xian Yang City, Grant/Award Number: 2017k01-25-12

Abstract

This paper considers the problem of long-term target tracking in complex scenes when tracking failures are unavoidable due to illumination change, target deformation, scale change, motion blur, and other factors. More specifically, a target tracking algorithm, called re-detection multi-feature fusion, is proposed based on the fusion of scale-adaptive kernel correlation filtering and re-detection. The target tracking algorithm trains three kernel correlation filters based on the histogram of oriented gradients, colour name, and local binary pattern features and then obtains the fusion weight of response graphs corresponding to different features based on average peak correlation energy criterion and uses weighted average to complete the position estimation of the tracked target. In order to deal with the problem that the target is occluded and disappears in the tracking process, a random fern classifier is trained to perform re-detection when the target is occluded. After comparing the OTB-50 target tracking dataset, the experimental results show that the proposed tracker can track the target well in the occlusion attribute video sequence in the OTB-100 test dataset and has a certain improvement in tracking accuracy and success rate compared with the traditional correlation filter tracker.

KEYWORDS

machine learning, pedestrian identification, robustness, visual surveillance, visual tracking

1 | INTRODUCTION

Target tracking is one of the important branches in the field of computer vision, with a wide range of applications in robot navigation, video surveillance, and human–computer interaction [1–3]. In the past decades, it has been the centre of attention of many scholars. Video-based target tracking is usually achieved by predicting the size and position of the target of interest in each frame by the tracking algorithm after the size and position of the target are given artificially in the first frame of a video.

In the past few decades, various algorithms proposed by many scholars have made great success in the field of target tracking. However, most of the current algorithms can only track the target in a single scene, which cannot meet the requirements of robust tracking in complex scenes. The research focuses on the field of target tracking is still to solve the target deformation (DEF) and illumination changes encountered in the tracking process as well as the scale change and target occlusion (OCC).

In recent years, due to the fast-computing characteristics of correlation filtering in the frequency domain, the computing speed of the target tracking algorithm has reached hundreds of frames per second, which makes the correlation filtering target tracking algorithm more popular than before. A good correlation filter can produce a correlation peak at the target position and a low response at the background position. The minimum output sum of squared error (MOSSE) algorithm proposed by Bolme [4] initiated the field of correlation-filtering target tracking. Henriques improved the MOSSE algorithm and proposed a circular structure with kernels (CSK) algorithm [5]. The CSK algorithm introduced a regularisation term after the utilised minimisation function to prevent over fitting of the obtained filter. In addition, this algorithm generated a large number of samples by cyclic displacement of the image matrix as input for the training classifier. Afterwards, Henriques proposed the kernelised correlation filters (KCF) algorithm [6] to improve the CSK algorithm, by using the histogram of oriented gradients (HOG) feature to replace the single-channel grey feature; they also used the circulant matrix around the target area as the positive and negative samples when training the detector and used the circulant matrix Fourier space diagonalisation to simplify the matrix calculation so that the algorithm can meet the real-time requirements. Daneljan et al. proposed the colour name (CN) algorithm [7] to improve the ability of tracking algorithm in dealing with target DEF based on colour information and then proposed a discriminative scale space tracker (DSST) algorithm to train a separate scale filter to estimate the target scale in each frame [8]. The scale adaptive kernel correlation filter (SAMF) algorithm [9] introduces scale estimation based on the KCF algorithm and combines HOG and CN features to describe the target from different angles to tackle the issues of target tracking in complex scenes. However, the algorithm does not have a redetection module to combat the tracking failure when the target encounters OCC. In Ref. [10], an improved method based on the image block re-detection is proposed based on the SAMF algorithm. When the tracking accuracy of the target is reduced due to OCC, the re-detection module is initiated to recover the tracking of the target. This method is suitable for the situation where the target is occluded for a short time. An OCC detection mechanism based on the KCF algorithm is proposed in Ref. [11]. When the algorithm concludes that the target is occluded, Kalman filtering is used to predict the target position to solve the problem of target relocation after OCC. However, this method is applicable only to linear systems. In Ref. [12], the target position is estimated by feature fusion based on the peak to side lobe ratio of the grey feature and local binary pattern (LBP) feature response. The idea of applying the features obtained by the deep convolution neural network to correlation filtering target tracking is proposed in Ref. [13]. However, due to the high computational cost of convolution feature extraction, it is difficult to meet the real-time requirements of target tracking.

In view of the ideas and shortcomings of the above algorithms, the main contributions of this paper are as follows: (1) under the framework of KCF algorithm, a multi-feature fusion scale adaptive algorithm integrating HOG [4], CN [14], and LBP [15] is proposed. (2) For the problem of complex scenes in the tracking process, multi-feature fusion is carried out based on the average peak correlation energy (APCE) criterion [16]. In the tracking task, a scale filter is trained to estimate the scale of the target. (3) A re-detector is trained online based on the random fern algorithm [17]. When tracking fails due to severe OCC of the target, we call the redetector to reposition the target.

2 | PRINCIPLE OF KERNEL CORRELATION FILTER

The purpose of target tracking based on correlation filtering is to find a classifier function $f(x, w) = (w, x)$ based on training samples and minimise its loss. w represents the parameter of the classifier. By taking the sum of squared errors as the loss function, the solution of w can be written as Equation (1):

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

Here, x_i and y_i represent the training samples and their corresponding labels, respectively, and λ represents the regularisation coefficient to prevent the classifier from overfitting. By calculating the partial derivative of the above equation and making it equal to zero, we can get the solution of Equation (1) as

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

In the above formula, $X = [x_1, x_2, \ldots x_n]^T$ represents the sample matrix, x_i represents a sample, I represents the identity matrix, and y_i represents the tag value of each training sample in $y = [y_1, y_2, \ldots y_n]^T$. Since the training samples based on kernel correlation filter are obtained by the cyclic shift of the target samples, using the Fourier transform property of cyclic matrix, Equation (2) can be converted into Equation (3):

$$\widehat{w} = \frac{\widehat{x}^* \odot \widehat{y}}{\widehat{x}^* \odot \widehat{x} + \lambda} \tag{3}$$

where $\odot$ represents matrix dot multiplication, $\widehat{x}$, $\widehat{y}$ and $\widehat{w}$ represent the discrete Fourier transform of x, y and w, respectively; $\widehat{x}^*$ represents the complex conjugate of $\widehat{x}$. Then kernel correlation filter maps the input sample x to the high-dimensional feature space through the kernel function processing method, so the classifier parameter w can be expressed as $w = \sum_{i=1}^{n} \alpha_i \phi(x_i)$ in the dual space, which transforms the problem of solving w into solving α in the frequency domain. Therefore, the problem can be expressed as Equation (4)

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

where $\wedge$ represents the Fourier transform, $\widehat{k}^{xx}$ represents the discrete Fourier transform of kernel matrix $K = \langle \phi(x), \phi(x) \rangle$ and has Equation (5)

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

In the phase of testing sample response, the tracker considers the candidate sample z by cropping an $M \times N$ pixel image segment from the position of the target in the previous frame to calculate the response graph Equation (6)

$$f(z) = F^{-1}\left(\widehat{k}^{xz} \odot \widehat{\alpha}\right) \tag{6}$$

where k^{xz} represents the correlation between the training sample x and the test sample z, F^{-1} represents the inverse Fourier transform, and $f(z)$ represents the response value of the test sample z. We regard the position of the test sample with the largest response value as the position of the target in the new frame.

3 | LONG-TIME TARGET TRACKING BASED ON MULTI-FEATURE FUSION

Robust representation of the target is an essential part of the target tracking task. Target tracking based on a single feature is not suitable for all scenarios [18]. For example, the HOG feature can reflect the contour of human body, so it can yield excellent performance in pedestrian detection, but the gradient descriptor is sensitive to noise points; the LBP feature can describe the local feature information of the image, so it can effectively deal with the motion blur (MB) caused by the rapid movement of the target, but the operator is sensitive to the direction information; the CN feature can effectively describe the colour information of the target, so it has a preferable effect on the DEF and partial OCC of the target, but it is easy to be affected by the change of illumination. Therefore, we use the fusion of multiple feature domains along with the target re-detection for target tracking in this paper. More specifically, the robust model representation of the target is established by multi-feature fusion; when a target is occluded, the re-detector is activated to prevent target loss.

3.1 | Multi-feature fusion based on average peak correlation energy

The target tracking based on correlation filtering infers the location of the target through the response graph. According to Equation (6), we set the location with the largest response value as the location of the target. It is well known that the ideal response graph should have only one sharp peak while the rest is flat, which indicates that the target located by the tracker matches the actual target very well, and the tracking effect is superior at this time. Conversely, if the rest of the response graph is not flat enough except the peak value, it indicates that the tracking quality is degraded due to the complex background or the OCC of the target.

Here, we propose to evaluate the tracking performance of the tracker based on the APCE criterion [19]. The APCE value of the response graph reflects the flatness of the response graph. As shown in Figure 1a,b, when the target is correctly tracked, the maximum value and the APCE value of the response graph are 0.46 and 37.71, respectively, and there is only one sharp peak in the response graph and the rest of the response graph is flat. When the tracking quality of the target is degraded due to OCC, as shown in Figure 1c,d, the maximum and APCE values of the response graph are 0.31 and 17.41, respectively. At this time, there is a peak value in the response graph, but the rest of the response graph is not flat and includes considerable fluctuations.

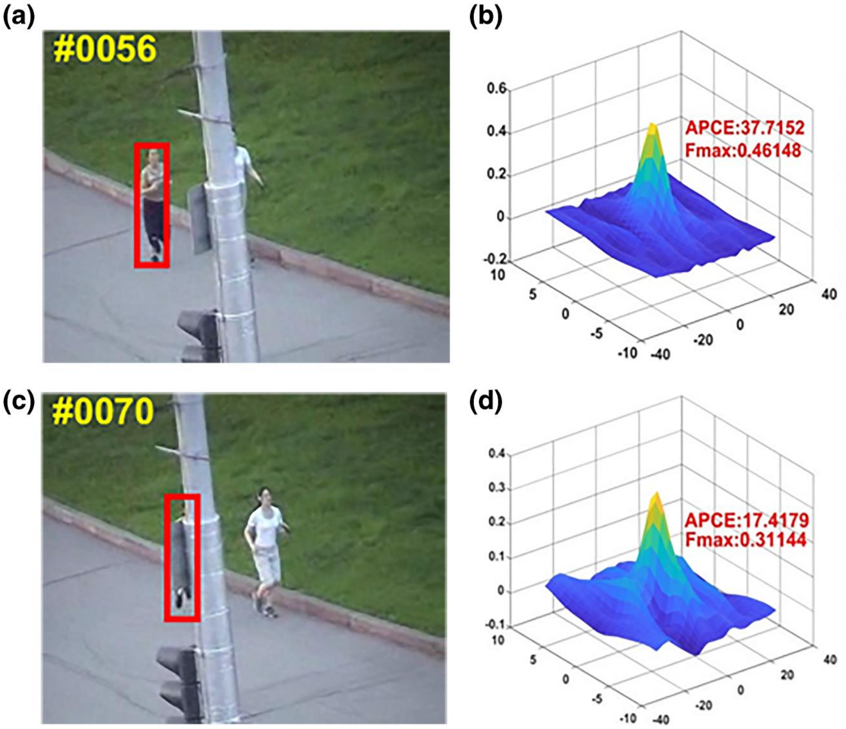

FIGURE 1 Diagram of average peak correlation energy (APCE) value and maximum response under different tracking quality: (**a**) no Occlusion (OCC), (b) response graph without OCC, (c) it's covered and (d) response graph with OCC

The APCE criterion is calculated as

$$APCE = \frac{|F_{max} - F_{min}|^2}{mean\left(\sum_{w,h}\left(F_{w,h} - F_{min}\right)^2\right)} \tag{7}$$

where F_{max} and F_{min} represent the maximum and minimum values in the response graph, respectively, and $F_{w,h}$ represents the response value at the position of column h in row w of the response graph.

After getting the corresponding response graphs of HOG, CN and LBP, we can get the final response graph by weighted averaging the corresponding response graphs of each feature according to the APCE criterion and select the position with the largest response value in the final response graph as our estimated target in the current frame position. The formula is

$$res = \frac{\left(A_{hog}res_{hog} + A_{cn}res_{cn} + A_{lbp}res_{lbp}\right)}{\lambda} \tag{8}$$

where $\lambda = A_{hog} + A_{cn} + A_{lbp}$. A_{hog}, A_{cn}, A_{lbp}represent the APCE value of the response graph corresponding to each feature, res_{hog}, res_{cn}, res_{lbp}, respectively represent the response graph corresponding to each feature, and the response graph is calculated according to Equation (6).

3.2 | Abstract scale adaptive strategy

The training process of the scale filter is as follows: at the beginning of the tracking task, an initial tracking box is calibrated around the target under tracking, and then training samples with different scales are generated around the centre of the target as the input of the scale filter. Then, S image blocks with different scales (here we have $S = 33$) are extracted. The scale size of training samples follows formula $a^n M \times a^n N$, where, $n \in \left\{-\frac{S-1}{2}, \ldots \frac{S-1}{2}\right\}$ and $a = 1.02$ represent the scale factor. We extract HOG features from the training samples of different scales obtained in this way.

Let$f$ represent the feature vector extracted around the target under we are tracking, and there exist d training samples at different scales,$l \in \{1, \ldots d\}$. Then, the feature matrix composed of training samples at different scales can be represented by $\left\{f^1, \ldots f^d\right\}$, and the calculation formula of scale filter h^l is

$$\min_h \left\| \sum_{l=1}^{d} h^l * f^l - g \right\|^2 + \lambda \sum_{l=1}^{d} \left\|h^l\right\|^2 \tag{9}$$

where g is the expected output of each training sample, and $\lambda \geq 0$ is the regularisation parameter. Our goal is to find an optimal filter h to minimise the above formula. The solution of the above equation is given in Equation (10):

$$H^l = \frac{\overline{G}F^l}{\sum_{k=1}^{d} \overline{F}^k F^k + \lambda} \tag{10}$$

Here, H^l represents the scale filter we require. Let A_t^l and B_t^l represent the numerator and denominator of H^l, respectively, then the update mode of the scale filter can be found as follows:

$$A_t^l = (1 - \eta)A_{t-1}^l \ + \ \eta\overline{G}_t F_t^l \tag{11}$$

$$B_t^l = (1 - \eta)B_{t-1}^l + \eta \sum_{k=1}^{d} \overline{F}_t^k F_t^k \tag{12}$$

where, η is the update rate of the scaling filter, which is set to 0.025 in this work.

After training the scale filter, in a new frame, we first obtain the centre translation position of the target through the translation filter based on multi-feature fusion. Then, we obtain multiple test samples under different scales by taking the position as the centre. We use the scale filter to calculate the response of each test sample and take the size of the sample with the largest response as the actual size of the target in the current frame. The calculation formula is as Equation (13):

$$y = F^{-1}\left(\frac{\sum_{l=1}^{d} \overline{A}^l Z^l}{B + \lambda}\right) \tag{13}$$

Here, Z is the test sample of different sizes, F^{-1} is the inverse Fourier transform, and y is the scale filter's response output of the test sample. The size with the largest response output can be taken.

3.3 | Construction of re-detector

In this paper, a re-detector is trained to deal with the problem of the disappearance of target due to OCC, which eventually leads to the tracking failure [6]. In the training phase, the positive samples are generated by selecting 10 image frames closest to the target. Then, we use data augmentation by applying scaling, rotation, and affine transformation to each frame to obtain more positive samples. The image frames far away from the target are selected as negative samples.

Let the training sample set be $X = \{x_1, x_2, \ldots x_K\}$, and the samples are divided into positive and negative samples, representing the target and non-target, respectively, which are represented by class labels $C = \{c_1, c_2\}$. F_j stands for the j th fern tree, where $j = 1, 2, \ldots 10$. The training process of random fern cluster classifier is as follows:

Step 1: Input positive or negative samples x_j as training samples into fern tree F_j. Each layer of the fern tree corresponds to the 2-bit LBP feature at the corresponding position of training samples, that is, $f_d = \{00, 01, 10, 11\}$, where $d = 1, 2, \ldots 8$ corresponds to layer d of the fern tree.

Step 2: The eigenvalues of all layers of fern tree F_j are counted to get the eigenvalue $B_j = f_8 LBP$, where $B_j \in \{0, 1, 2, \ldots 4^8\}$.

Step 3: For the sample c_i, whose sample category is x_i, we can get the eigenvalues of the sample in the last layer of the fern tree, and we can get the histogram of the eigenvalues corresponding to a single fern by counting the eigenvalues of all training samples. The abscissa is the eigenvalue, and the ordinate represents the number of times that the eigenvalue appears in the training process.

Step 4: The Step 3 completes the histogram statistics of the eigenvalues of a single fern [20]. Steps 1–3 is executed 10 times to complete the histogram statistics of the eigenvalues of the training samples.

If S is the test sample and the sample category is represented by C, the process of classifying the test sample to determine whether or not it is a candidate sample is as follows:

Step 1: Input the sample into the random fern classifier k_j to get the 2-bit-lbp eigenvalue k_j of the test sample.

Step 2: Find the maximum likelihood estimation of k_j, using the following equation:

$$p_j^{c_t} = p\left(C = c_t \middle| k_j\right) = \frac{N_j^{k_j c_t}}{N_j^{c_t}}, t = 1, 2 \tag{14}$$

Here, $N_j^{c_t}$ represents the sum of the histogram frequencies corresponding to the jth fern tree of category c_t, and $N_j^{k_j c_t}$ represents the occurrence frequency of the characteristic value k_j in the histogram.

Step 3: If the final output meets $\frac{1}{n}\sum_{j=1}^{n} p\left(C = c_t \middle| k_j\right) > 0.6$, for $n = 10$, then the test sample belongs to class c_t.

Through the above three steps, we can classify the test samples, but after executing the random fern classifier, we will still get multiple candidate samples. Therefore, we need to select the best sample from these candidate samples as our re-detection target. Here, we calculate the correlation between the candidate samples and the sample set based on a K-Nearest Neighbor classifier, that is, the Euclidean distance between the selected and the candidate samples. Then, according to the category of these five training samples, we decide whether or not the candidate samples are the targets we want to detect.

3.4 | Algorithm flow

The algorithm flow is as follows:

Input: the initial position $P_0 = (x_0, y_0, s_0)$ of the target, where (x_0, y_0) represents the centre position coordinate of the target and s_0 represents the target scale.

Output: estimated target position $P_t = (x_t, y_t, s_t)$.

- For $t = 1$ to T (T is the total count of frames of the input video frames).

Location estimation:

- Extract an image block in the 2nd frame. The centre of the image block is the location of the target in the previous frame, that is, (x_{t-1}, y_{t-1}), and the length and width of the image block are 2.5 times of the length and width of the target in the previous frame. The HOG, CN and LBP features of the image block and its virtual samples generated by translation are extracted, respectively [21].
- Calculate response matrices f_{hog}, f_{cn}, f_{lbp} for corresponding to each feature map according to Equation (6) based on the corresponding filter, where the response matrices f_{hog}, f_{cn}, f_{lbp} are corresponding to the response graphs res_{hog}, res_{cn}, res_{lbp}.
- Assign the weight of each response graph according to Equation (8) based on the APCE criterion to get the final response graph. The position of the maximum response value in the final response graph is the estimated target position (x_t, y_t).

Scale estimation:

- Taking (x_t, y_t) as the centre, cut out the image blocks of different scales around the centre, and extract their HOG features. According to Equation (13), the image blocks of different scales are correlated with the scale filter, and the image block size with the largest response value is selected as the size s_t of the current frame target.

Determine the target location:

- Calculate the APCE value for each feature according to Equation (7). When A_{hog}, A_{cn}, A_{lbp} are all less than 0.45 times of the historical mean value, we conclude that the performance of the tracker is poor due to the OCC of the target. Therefore, the re-detector is activated to re-locate the target position $P'_t = \left(x'_t, y'_t, s'_t\right)$ and $P_t = P'_t$; otherwise, the re-detector is not activated, and P_t is the result of the position and scale estimations.

Model update:

- When A_{hog}, A_{cn}, A_{lbp} are greater than or equal to 0.45 times of the historical mean value, we conclude that the target position is more accurate. At this time, we use the results of the tracker to train the position filter and scale filter. Also, positive and negative samples are collected around the target as training samples of the re-detector. We update the re-detector every time we collect 200 training samples.

The pseudo code of the algorithm is as follows:

```
void main ()
{
   Read video;
   if (First frame?)
   {
      Generate virtual samples based on
      target location;
      Training position filter;
      Training scale filter;
    }
    else
    {
     Cut the test sample based on the target
     position of the previous frame;
     Extract the characteristics of hog, LBP
     and CN of test samples;
     Perform correlation operation with the
     position filter to obtain the response
     diagram;
     The estimated target position is
     obtained by fusing multiple features
     based on APCE;
     The estimated scale is obtained by
     correlation operation with the scale
     filter;
     if (Is the tracker reliable?)
     {
      Training filter based on estimated
      target position and scale;
      The training samples of the re-detector
      are collected around the target;
      if (Training sample reaches 200?)
         Update re-detector;
     }
     else
     {
      Activate the re-detector for target
      relocation;
      Target location and scale training filter
      based on relocation of detector;
     }
    }
}
```

The framework of this algorithm is given in Figure 2.

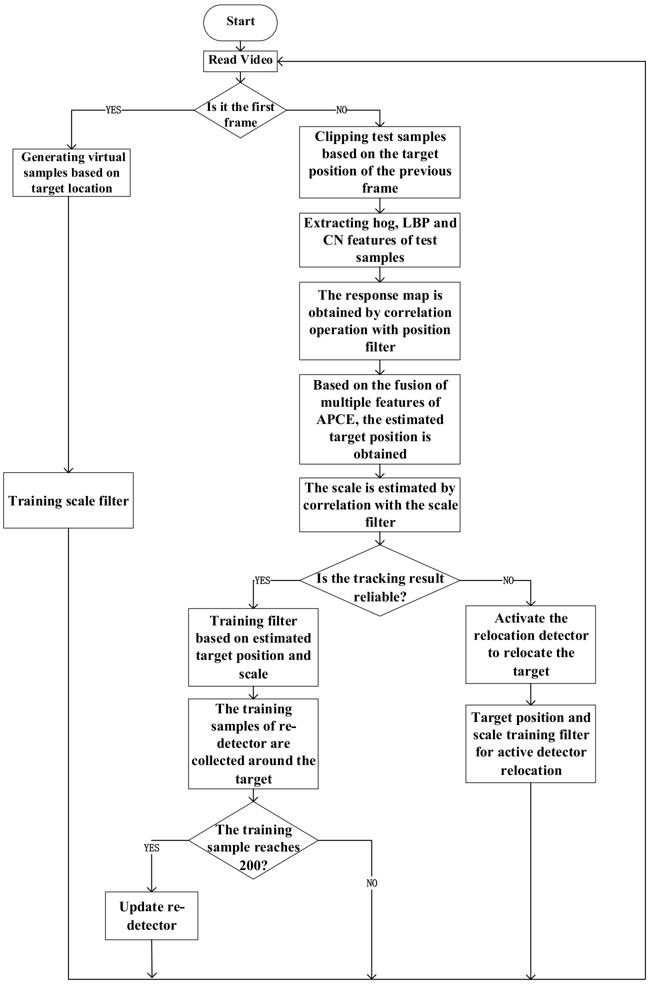
Start
Read Video
Is it the first frame
YES
NO
Generating virtual samples based on target location
Clipping test samples based on the target position of the previous frame
Extracting hog, LBP and CN features of test samples
The response map is obtained by correlation operation with position filter
Based on the fusion of multiple features of APCE, the estimated target position is obtained
Training scale filter
The scale is estimated by correlation with the scale filter
Is the tracking result reliable?
YES
NO
Training filter based on estimated target position and scale
Activate the relocation detector to relocate the target
The training samples of re-detector are collected around the target
Target position and scale training filter for active detector relocation
The training sample reaches 200?
YES
NO
Update re-detector

FIGURE 2 Algorithm flow chart

4 | EXPERIMENTAL RESULTS AND ANALYSIS

This section is divided into three parts. First, the software and hardware environment used to verify the re-detection multi-feature fusion (RDMF) tracking algorithm RDMF proposed in this paper is introduced. Second, the test video dataset is introduced. Finally, the algorithm results of this paper are analysed and summarised from two aspects of qualitative evaluation and quantitative evaluation.

4.1 | Introduction of experimental software and hardware environment

The experimental conditions of RDMF algorithm are as follows:

TABLE 1 Video properties

Attribute classification	Property description
IV (illumination variation)	Light change
SV (scale variation)	Scale change
OCC(Occlusion)	Occlusion
DEF(Deformation)	Deformation
MB (motion blur)	Motion blur
FM (fast motion)	Fast motion
IPR (in-plane rotation)	Plane rotation
OPR (out-of-plane rotation)	Solid rotation
OV(Out-of-view)	The target is out of view of the camera
BC (background clutters)	The background of the target is complex
LR (low resolution)	Low resolution

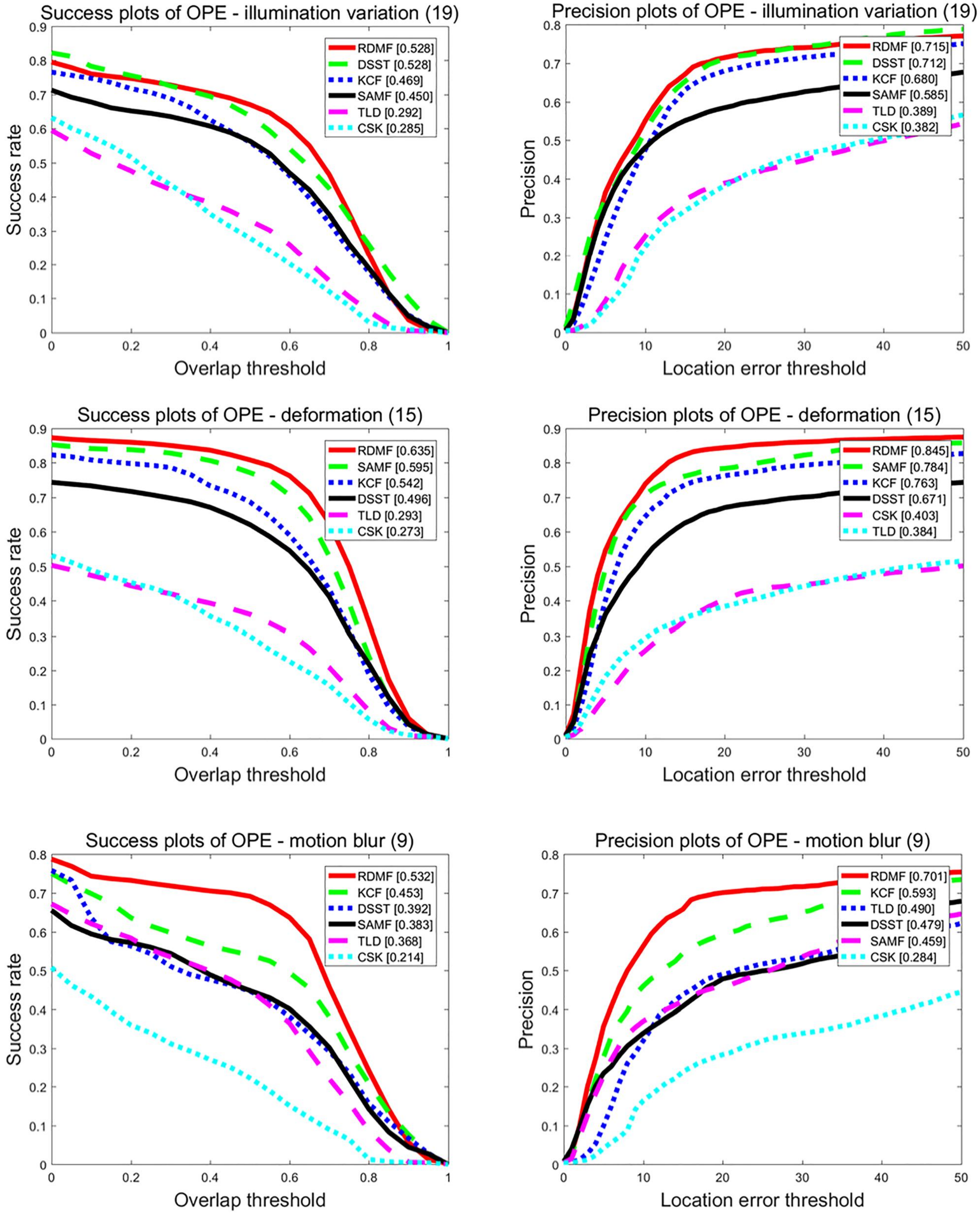

FIGURE 3 Comparision of illumination variation (IV), Deformation (DEF), motion blur (MB) and one pass evaluation (OPE)

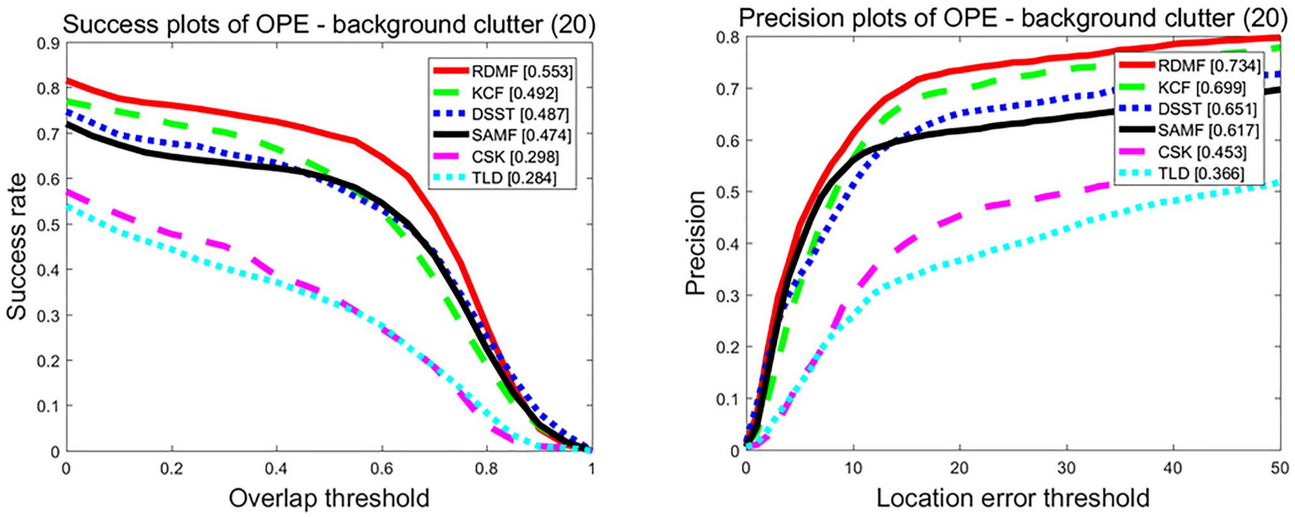

FIGURE 4 Comparision of the proposed re-detection multi-feature fusion (RDMF) algorithm with commonly used target tracking methods in terms of dealing with complex backgrounds

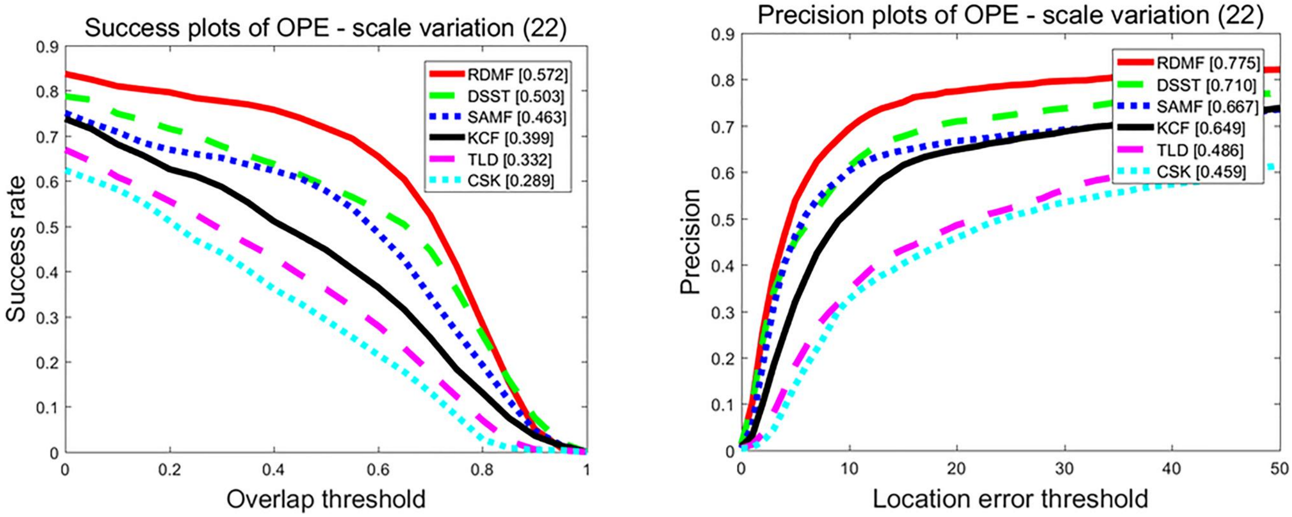

FIGURE 5 Comparision of the proposed re-detection multi-feature fusion (RDMF) algorithm with commonly used target tracking methods in terms of dealing with scale change

- Hardware conditions: the Central Processing Unit is Intel (R) core (TM) i5, the main frequency is 2.4 GHz, and the memory is 8 GB.
- Software conditions: Windows 7 operating system and the code environment is MATLAB 2016a.
- Test video: OTB-50 video test sequence in visual tracker benchmark library.

4.2 | Introduction to test dataset

In order to evaluate the efficacy of the proposed tracking algorithm in a comprehensive way, the data in the video library are marked with attributes. There are 11 kinds of video attributes, as shown in Table 1.

Each group of videos in the video set contains many of these 11 attributes, enabling us to evaluate the tracking algorithm objectively. The evaluation method includes both qualitative and quantitative evaluations. Qualitative evaluation refers to marking the target location of the tracking algorithm directly in the test video, and the effect of tracking algorithm can be observed directly by the human eyes. Because the qualitative evaluation is subjective, the quantitative evaluation method is also proposed. This method describes the effect of tracking algorithm's efficacy from two perspectives of the range accuracy and the overlap success rate.

The statistical method of distance accuracy is to calculate the average Euclidean distance between the actual target centre position and the target centre position determined by the algorithm in each frame. The distance accuracy value in the current video sequence is the ratio of the number of frames with Euclidean distance below the threshold to the total number of the frames.

The statistical method of overlapping success rate is to calculate the overlapping rate between the target position determined by the tracking algorithm and the actual target position, and the ratio of the number of frames, whose overlapping rate is greater than the specified threshold to the total number of frames is the overlapping success rate of the current video sequence. It can be calculated as

$$S = \frac{\gamma(t) \cap \gamma(a)}{\gamma(t) \cup \gamma(a)} \tag{15}$$

The numerator refers to the area intersection of the actual target and the target position determined by the tracking algorithm, and the denominator refers to the area union of the two positions; the ratio of the two determines the overlap rate S.

4.3 | Experimental results and analysis

First, the algorithm is evaluated quantitatively. Since the algorithm tracks the target based on the multi-feature fusion strategy, considering the robustness of the HOG, LBP, and CN feature maps, respectively, to illumination, MB, and target DEF, combining these three complementary features is beneficial to deal with the illumination change, target DEF and MB altogether. One pass evaluation represents the experimental results, which are drawn based on the tracking algorithm after one run on the dataset.

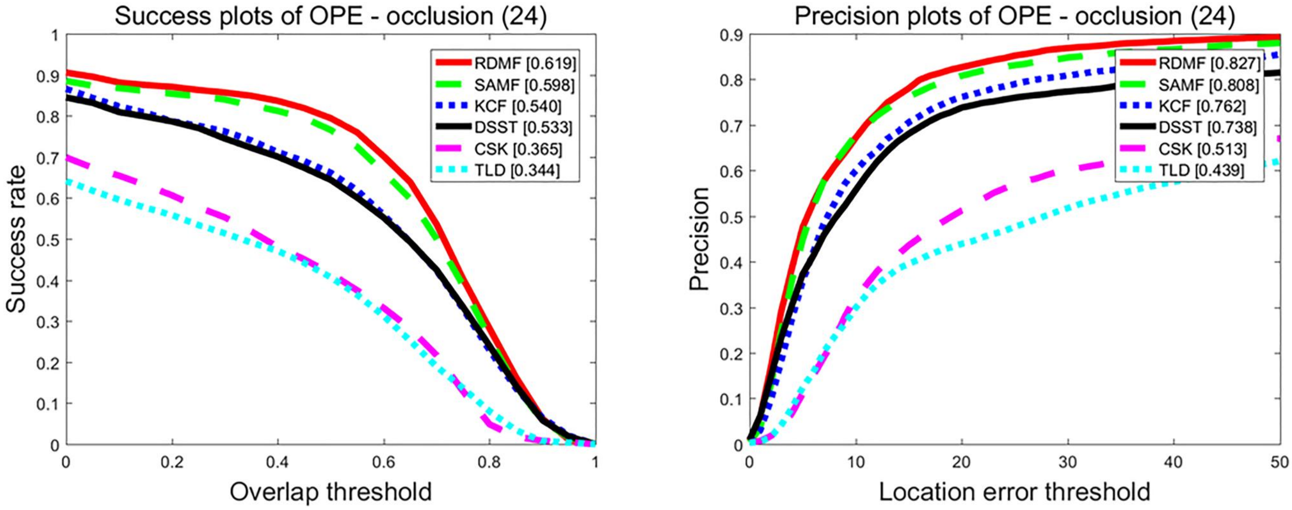

FIGURE 6 Comparision of the proposed re-detection multi-feature fusion (RDMF) algorithm with commonly used target tracking methods in terms of dealing with occlusion (OCC) effect

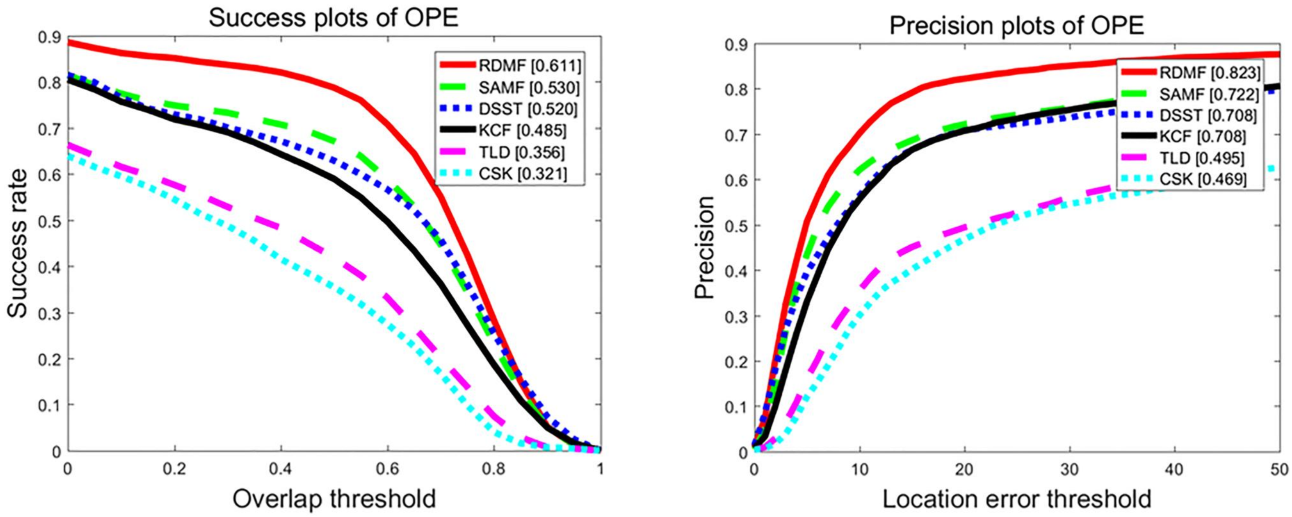

FIGURE 7 Overall effect comparison

In Figure 3, we compare the RDMF algorithm with classical target tracking algorithms such as KCF, DSST, SAMF, Track-Learning-Detection and CSK. It can be seen that due to the fusion of multiple features, the tracking accuracy of the proposed algorithm is among the best in the three challenging scenarios of illumination change, target DEF and MB. Due to using the multi-feature fusion, the proposed algorithm exhibits a good tracking accuracy

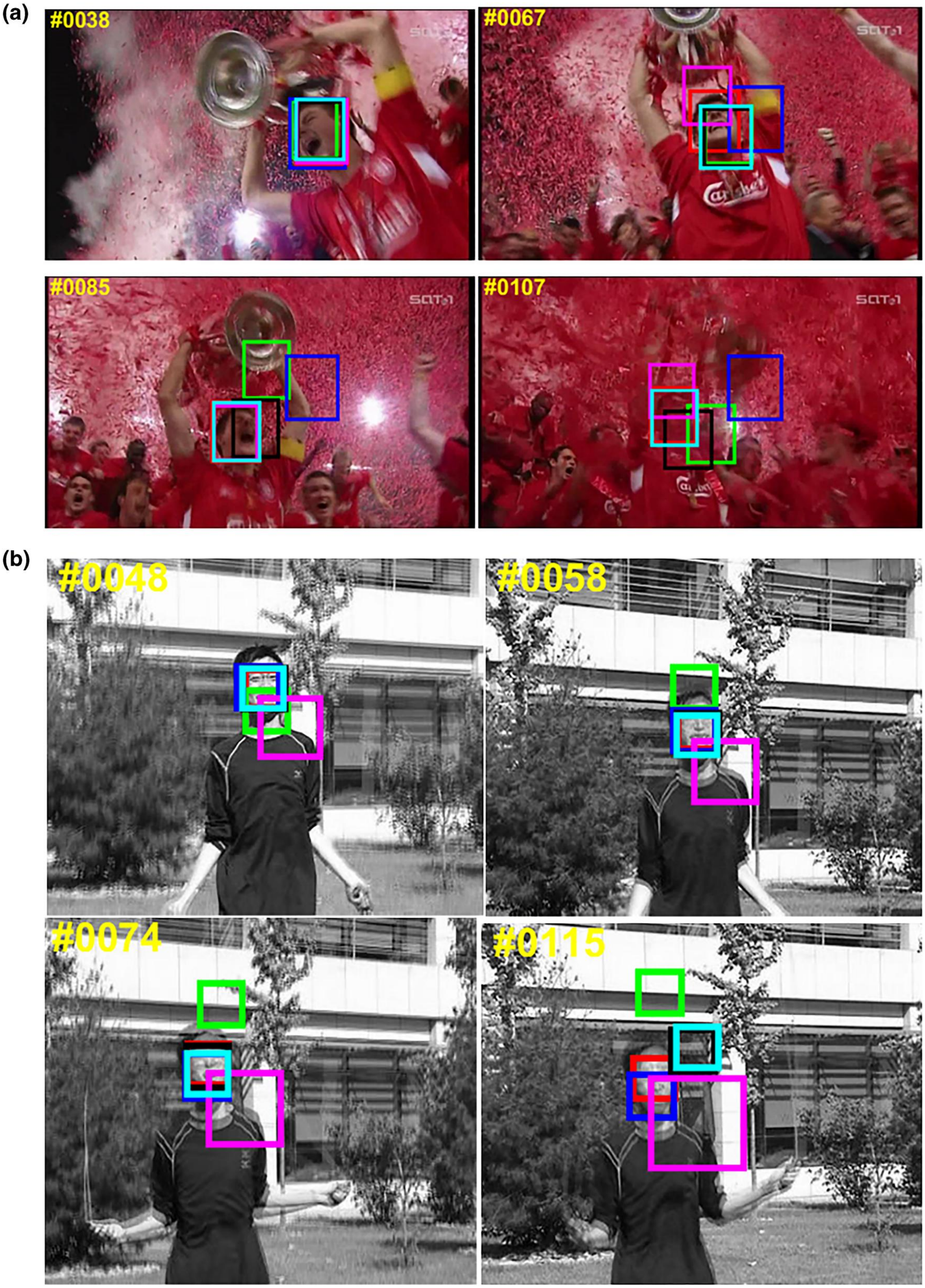

FIGURE 8 Comparision of effect of soccer sequence and jumping sequence: (a) soccer sequence and (b) jumping sequence

in the case of complex target background, as shown in Figure 4:

As can be seen in Figure 4, when dealing with a complex background, the overlapping success rate of the RDMF algorithm is 6.1% higher than that of KCF, and the range accuracy is 3.5% higher than that of KCF. The main reason is that the KCF only uses HOG features to describe the target, while our algorithm uses HOG, CN, and LBP features to describe the target. The advantage of this algorithm is not easy to introduce background noise when the object is similar.

Figure 5 presents the performance of the algorithms when subjected to the target scale change. It can be seen that, the overlapping success rate of the proposed RDMF algorithm is 6.9% higher than that of DSST algorithm equipped with scale estimation. Also, the range accuracy is 6.5% higher than that of the DSST algorithm. The justification is that the centre position filter of the RDMF algorithm is trained based on three features, while the DSST algorithm only uses HOG features, therefore, the accuracy of the RDMF algorithm is higher in determining the centre position of the target. The subsequent scale estimation is based on the centre position of the image multi-scale sampling, selecting the sample block with the best scale as the best scale of the target.

Figure 6 presents the operation of different algorithms subject to target OCC. It can be seen that the overlapping success rate of the RDMF algorithm is 2.1% higher than that of the SAMF algorithm, and the distance accuracy is 1.9% higher than that of the SAMF algorithm. This is due to using a re-detector in the RDMF algorithm. When the target is occluded and lost, the re-detector will start to search for the target, and when the target reappears, the re-detector will remove the target from the target, that is why the anti-occlusion ability of the algorithm is improved.

Figure 7 shows the overall quantitative evaluation effect comparison chart of each algorithm. We observe that the overlapping success rate of the RDMF algorithm is 8.1% higher than that of the SAMF algorithm, and the distance accuracy is 10.1% higher than that of the SAMF algorithm.

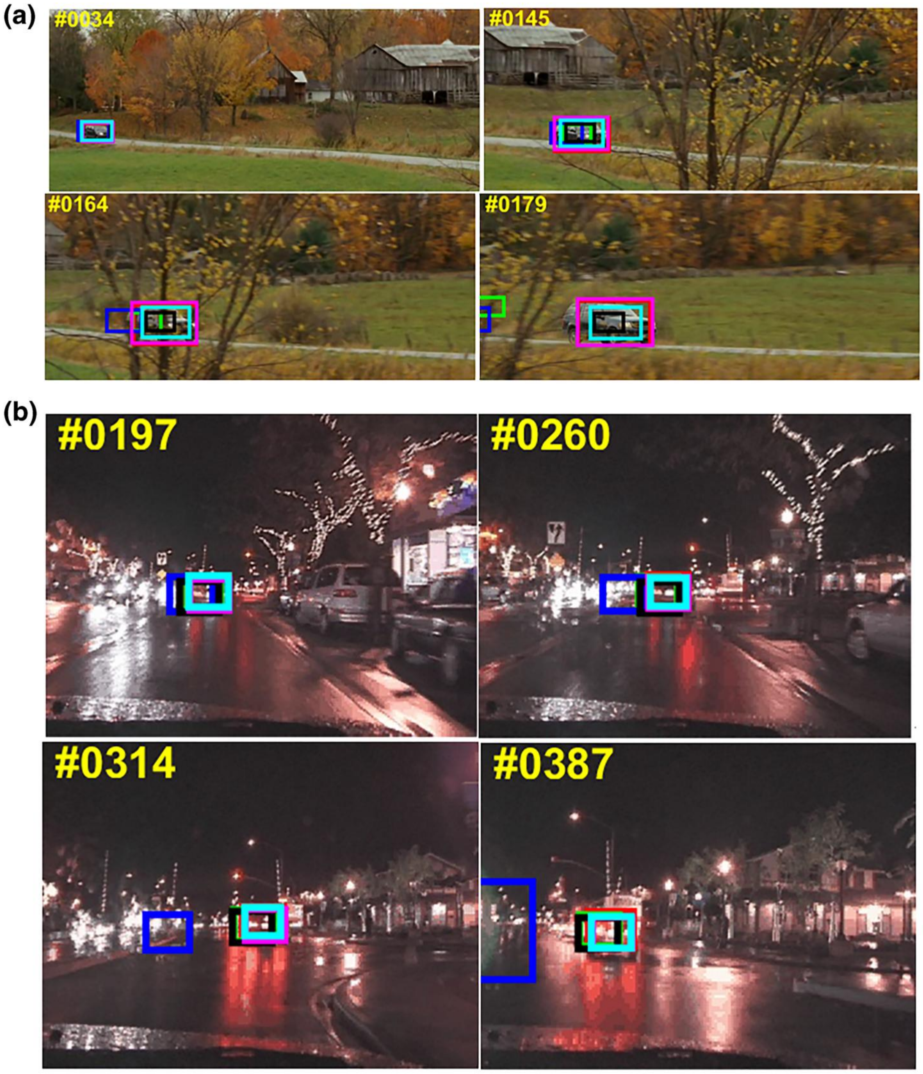

FIGURE 9 Comparision of car scale sequence and car dark sequence: (**a**) car scale sequence and (**b**) car dark sequence

In addition to the presented quantitative results, we evaluate each algorithm's tracking effect in a more intuitive qualitative manner.

The two sequences in Figure 8 present the soccer sequence and jumping sequence in OTB-50. In the soccer sequence, we see the results of each tracking algorithm when the target encounters the MB and complex background. In the 85th frame of the soccer sequence, the target becomes blurred due to the person's jump, and in the 107th frame, the target suffers from the interference of background noise. We can see the effect of this algorithm on these two sequences; all the challenges have good robustness. In the jumping sequence, the tracking target is always suffering from the MB due to the continuous jumping of characters. We observe that the RDMF algorithm can perfectly trace the target.

The two sequences in Figure 9 represent car scale and dark car sequences in OTB-50. In the car scale sequence, the main challenge we face is the scale change and OCC of the target. It can be seen that the RDMF algorithm can carry out scale adaptation in the process of tracking. In the 164th frame of the car scale sequence, the target is occluded, making some algorithms fail to track. The RDMF algorithm is implemented in sequential steps after OCC, and the target can still be tracked adaptively. In the car dark sequence, the main challenges we encounter are the illumination change and target scale change. We can see that the RDMF algorithm can still follow the target successfully when facing these two challenges.

The two sequences in Figure 10 include the jogging and tiger sequences in OTB-50. In the 72nd frame of the jogging sequence, the target disappears entirely due to OCC. At this time, the re-detector is invoked to re-locate the target. We can see that the target position is re-detected in the 79th frame. The challenge in the tiger sequence is the OCC and DEF of the target. In the 257th frame, the target is seriously occluded. We observe that in the 284th frame, the RDMF algorithm can resume the tracking of the target.

The sequence in Figure 11 is the singer2 sequence in OTB-50. It can be seen that the drastic change of illumination in the 41st frame leads to the failure of some trackers. In the 244th and 365th frames, it can be seen that the continuous movement of the target causes the constant DEF of the target. However, the RDMF algorithm can still follow the target robustly.

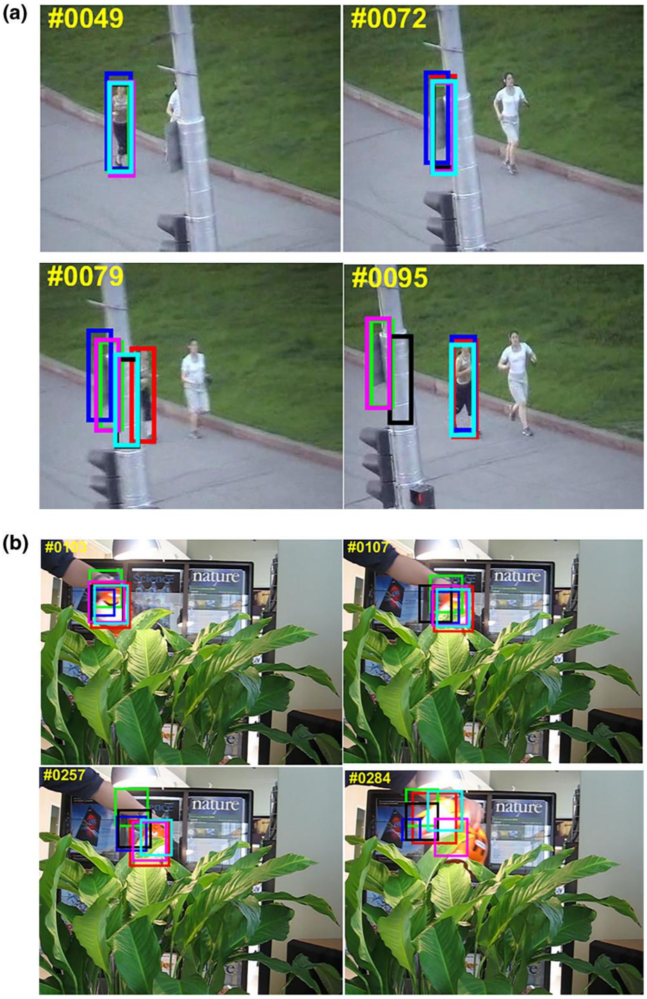

FIGURE 10 Comparision of effect of jogging sequence and tiger sequence: (**a**) jogging sequence and (**b**) tiger sequence

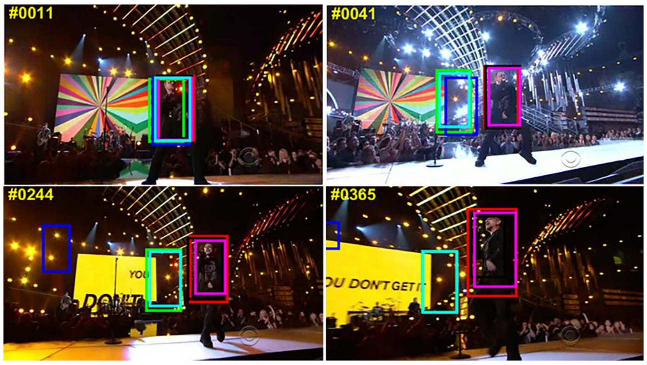

FIGURE 11 Comparision of singer2 sequence effect

5 | CONCLUSION

In this paper, we used the idea of fusing feature maps and invoking re-detection of targets to implement a robust long-term target tracking algorithm called RDMF. The results show that the proposed algorithm outperforms the most recently developed target trackers, especially when the target is subject to scale change, extreme illumination variation, and OCC. Compared with the SAMF algorithm, the overlap success rate and the range accuracy of the RDMF algorithm are improved by 8.1% and 10.1%, respectively. Scale adaptive kernel correlation filter is a scale adaptive target tracking algorithm, which combines HOG and CN features. Compared with the SAMF, we also integrate LBP features to improve the proposed algorithm's robustness to target MB. Besides, we also offered to train a re-detector online to enhance the algorithm's robustness against OCC. Compared with SAMF, the overlap success rate and range accuracy of RDMF algorithm are improved by 8.1% and 10.1%, respectively. Scale adaptive kernel correlation filter is a scale adaptive target tracking algorithm, which combines HOG and CN features. Compared with SAMF, we also integrate LBP features to improve the robustness of the proposed algorithm to target MB. In addition, we also proposed to train a re-detector online to improve the robustness of the algorithm against OCC. Compared with SAMF, the overlap success rate and range accuracy of RDMF algorithm are improved by 8.1% and 10.1%, respectively. Scale adaptive kernel correlation filter is a scale adaptive target tracking algorithm, which combines HOG and CN features. Compared with SAMF, we also integrate LBP features to improve the robustness of the proposed algorithm to target MB. In addition, we also proposed to train a re-detector online to improve the robustness of the algorithm against OCC.

ACKNOWLEDGEMENTS

This research study was supported in part by grants from the International Cooperation and Exchange Program of Shaanxi Province (2022KW-04), the Natural Science Foundation of Shaanxi Province (2018JM6120), the Xi'an Science and Technology Plan Project (21XJZZ0072), Major Science and Technology Projects of Xian Yang City (2017k01-25-12). This unnumbered section is used to identify people who have aided the authors in accomplishing the work presented and to acknowledge sources of funding.

CONFLICT OF INTEREST

The author declares no conflict of interest.

DATA AVAILABILITY STATEMENT

http://doi.org/10.1049/csy2.12042.

ORCID

Junsuo Qu https://orcid.org/0000-0002-4781-260X

REFERENCES

1. Artificial Intelligence: ICCV 2017 international conference on computer vision. NewsRx Health & Sci. (2017)
2. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. 37(9), 1834–1848 (2015)
3. Fu, L., et al.: Auto-exposure fusion for single-image shadow removal. https://arxiv.org/abs/2103.01255
4. Bolme, D.S., et al.: Visual object tracking using adaptive correlation filters. In: IEEE conference on computer vision and pattern recognition, pp. 2544–2550. IEEE, New York (2010)
5. Wang, Z., et al.: Scalable and adaptive reconstruction for video compressive sensing. https://arxiv.org/abs/2103.01786
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), vol. 1, pp. 886–893. IEEE (2005)
7. Danelljan, M., et al.: Adaptive color attributes for real-time visual tracking. In: Proceedings of IEEE conference on computer vision and pattern recognition, pp. 1090–1097 (2014)
8. Danelljan, M., et al.: Accurate scale estimation for robust visual tracking. In: British machine vision conference, nottingham, BMVA Press, Durham, 1–5 September 2014
9. Li, Y., Zhu, J.: A scale adaptive kernel correlation filter tracker with feature integration. In: European conference on computer vision, pp. 254–265. Springer, Cham (2014)
10. Ma, C., et al.: Adaptive correlation filters with long-term and short-term memory for object tracking. Int. J. Comput. Vis. 126(8), 771–796 (2018)
11. Wang, M., Liu, Y., Huang, Z.: Large margin object tracking with circulant feature maps. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4021–4029 (2017)
12. Danelljan, M., et al.: Learning spatially regularized correlation filters for visual tracking. In: Proceedings of the IEEE international conference on computer vision, pp. 4310–4318 (2015)
13. Li, F., et al.: Learning spatial-temporal regularized correlation filters for visual tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4904–4913 (2018)
14. Saffari, A., et al.: On-line random forests. In: 2009 IEEE 12th international conference on computer vision workshops, iccv workshops, pp. 1393–1400. IEEE (2009)
15. Henriques, J.F., et al.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. 37(3), 583–596 (2014)
16. Ma, C., et al.: Hierarchical convolutional features for visual tracking. In: Computer vision and pattern recognition (CVPR), 2015 IEEE conference on, pp. 3074–3082. New York, NY, USA: IEEE (2015)
17. Ozuysal, M., et al.: Fast keypoint recognition using random ferns. IEEE Trans. Pattern Anal. Mach. Intell. 32(3), 448–461 (2009)
18. Danelljan, M., et al.: Adaptive decontamination of the training set: a unified formulation for discriminative visual tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1430–1438 (2016)
19. Danelljan, M., et al.: Eco: efficient convolution operators for tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 6638–6646 (2017)
20. Van De Weijer, J., Schmid, C., Verbeek, J.: Learning color names from real-world images. In: 2007 IEEE conference on computer vision and pattern recognition, pp. 1–8. IEEE (2007)
21. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. 24(7), 971–987 (2002)

How to cite this article: Qu, J., et al.: Long-time target tracking algorithm based on re-detection multi-feature fusion. IET Cyber-Syst. Robot. 4(1), 38–50 (2022). https://doi.org/10.1049/csy2.12042