Journal of Engineering Science and Technology Review 14 (6) (2021) 43 - 54

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

Research Article

Correlation Filter Tracking Algorithm Based on Object Saliency Guidance

Jinping Sun[1,*], Dan Li[1] and Ximin Wang[2]

[1]School of Information Engineering (School of Big Data), Xuzhou University of Technology, Xuzhou 221008, China
[2]School of Computer Science, University of Nottingham, Nottingham NG8 1AF, United Kingdom

Received 2 October 2021; Accepted 17 December 2021

Abstract

When dealing with object tracking in complex scenes, such as occlusion, illumination variation, deformation, and low resolution, the traditional correlation filter (CF) has some shortcomings in the object representation model and the relocation. To optimize the filter model and improve the success rate of the tracking algorithm, an appearance representation model and a saliency-guided relocation model were proposed in this study. An adaptive weighted multi-feature fusion model was designed based on the analysis of the internal correlation between different feature representation and CF responses. The weight coefficients were allocated adaptively according to the response value, and the final object position was obtained under the constraint of penalty term. With the first frame and the latest tracking result taken as the guidance, the saliency of the object was detected by using the updated multi-layer cellular automata method to achieve the purpose of object relocation after tracking drift or failure. Finally, the size of the object was predicted according to the strategy of size detection based on filter response. The accuracy of the proposed algorithm was verified by comparative experiments on benchmark data sets. Results show that the appearance representation model of the proposed algorithm is effective. The overlap success rate and distance precision rate of the proposed algorithm are 0.673 and 0.724, which are better than the results of other comparison algorithms. The proposed algorithm effectively solves the tracking drift or loss caused by complex scenes, such as occlusion, illumination variation, and low resolution. This study eliminates the problem of object recognition caused by environmental changes and provides references for the anomaly detection of real-time traffic.

Keywords: Object guidance, Significance detection, Relocation, Correlation filter, Object tracking

1. Introduction

Object tracking technology is an important research direction in the field of computer vision. It is a complex and challenging technology that integrates several fields, such as pattern recognition, image processing, and computer applications. In the past decades, classical object tracking algorithms mainly focused on the relevant theoretical knowledge of Kalman filter, mean shift, and particle filter. In recent years, the correlation filter (CF) [1-3] algorithm has gained popularity. CF algorithm uses the cyclic matrix to generate a large number of training samples and uses fast Fourier transform (FFT) in the frequency domain to speed up the processing, thereby providing a new research idea for object tracking. In particular, the combination of deep learning and correlation filtering greatly improves the robustness and accuracy of the tracking algorithm, which benefits from both the strong expression ability of depth features and the high-speed processing ability of the CF tracking framework.

The final task of the object tracking algorithm is to deal with the object tracking problem in actual application scenarios. However, unpredictable interference factors appear in actual tracking environments at any time, resulting in the complexity and variability of the tracking scene, which will in turn affect the tracking effect of the algorithm. At present, object tracking in complex scenes, such as occlusion, illumination variation, deformation, low resolution, low illumination, and rotation, may cause tracking drift or loss due to the interference of the object appearance representation model, a weak discrimination model, or an incorrect model update. Hence, tracking accuracy and robustness still need to be further improved to continuously meet the needs of practical applications.

On the basis of these aspects, researchers have conducted studies on the object appearance representation model and the relocation strategy [4-5], but many problems still need to be addressed, such as the unclear effect of multi-feature fusion and the object relocation offset. Therefore, determining how to preprocess the multiple features and allocate the contribution of different features as well as how to relocate after tracking loss are all urgent problems to be solved.

In this study, an appearance representation model integrating the histogram of oriented gradient (HOG) features, texture features, and color features is established by analyzing the relationship between filter response and feature contribution. With the first frame and the latest tracking result taken as the guidance, the proposed algorithm combines the saliency detection algorithm and the object relocation method to provide a reference for improving the effect of the tracking algorithm.

[*] E-mail address: sjp614@163.com

ISSN: 1791-2377 © 2021 School of Science, IHU. All rights reserved.
doi:10.25103/jestr.146.05

2. State of the art

CF has been applied to object tracking and has achieved excellent results in the latest public data set [6] and academic competition [7]. Henriques et al. [8] proposed the kernel correlation filter (KCF) algorithm, which uses HOG features to represent the object. The training samples were generated by cyclic shift for the basic samples, and FFT was used to accelerate the calculation of the algorithm, which improved tracking speed but led to the discontinuity of the sample boundary. Karunasekera et al. [9] and Meng et al. [10] described the research results and progress of the tracking algorithm in detail and compared the performance of tracker based on CF and other models. Their study provided an important reference for the research of object tracking algorithm.

Hare et al. [11] established a structured support vector machine (SVM) adaptive object tracking algorithm called Struck, which omitted the classification process and directly outputted the results. By limiting the growth of the support vector in the tracking process, the algorithm achieves high real-time performance, but its overlap rate is low. Bertinetto et al. [12] and Galoogahi et al. [13] both proposed a tracking algorithm that combined HOG features and color features; it had fast processing speed but poor robustness. Zhang et al. [14] established rotation and scale normalization descriptors and fused color and texture features to perform optimal similarity matching for candidate descriptors in adjacent frames, which improved the tracking accuracy but resulted in a contribution allocation that lacked flexibility. Zhao et al. [15] and Zhang et al. [16] used Kalman filter algorithm to predict the state of the object, judge whether the object was occluded, and predict that the object would still be occluded in the later stage. In the process of long-time object tracking, due to the changeable tracking environment, the target may have some problems, such as deformation and serious occlusion, which will result in tracking failure. How to recover tracking quickly is the key to realizing long-time object tracking. The mean shift algorithm [17] was widely used because of its simple calculation and high real-time performance. However, this algorithm lacks real-time update of the object model, a deficiency that easily leads to tracking failure due to object scale change in engineering application. Liu et al. [18] designed an algorithm for long-term object tracking which combined the attention mechanism et al. [19]. When tracking fails, the effectiveness of the reference frame is not considered when using the edge box to generate the recommendation area, which may cause tracking drift again. Xiong et al. [20] proposed a target scale and rotation parameter estimation method based on KCF to solve the problem of object scale and rotation changes caused by long-time object tracking. When the object tracking was lost, the object searching method combining color histogram and variance [21] was started to quickly determine the possible position of the object in the current frame and recover the later object tracking, but there was still the problem of positioning error. Yuan et al. [22] designed a focused object convolution regression model for the visual object tracking task. The object loss function can effectively balance the proportion of positive and negative samples and prevent the appearance model from overfitting the background samples. However, the high complexity of the algorithm results in poor real-time performance. Lukezic et al. discussed a tracking algorithm with channel and spatial reliability [23] which segmented the foreground and background of the candidate region and then processed it by spatial regularization. The tracked object is re-detected after long-time occlusion, but the execution speed of the algorithm is low.

The abovementioned methods based on re-detection are not universal. In complex scenes such as occlusion, scale transformation, deformation, illumination, and motion blur, tracking loss or tracking errors are prone to occur. Given the unsatisfactory or low efficiency of different tracking algorithms in solving different complex scenes, the object tracking technology requires further study to improve the tracking efficiency and effect.

This study mainly focuses on the tracking drift or loss caused by occlusion in complex scenes. It designs an adaptive weighted multi-feature fusion model based on the different advantages of HOG features, texture features, and color features. The weight coefficients are allocated adaptively according to the response values, and the final object position is obtained under the constraint of penalty term. When tracking drifts or fails, a significance re-detection model based on object guidance is designed. With the first frame and the latest tracking result taken as the guidance, the saliency of the object is detected by using the multi-layer cellular automata update method to achieve the purpose of object relocation.

The remainder of this study is organized as follows. Section 3 describes the overall framework of the proposed algorithm, and constructs the kernel correlation filter model of multi-feature fusion and the saliency re-detection model based on target guidance. Section 4 verifies the tracking effect of the algorithm in different video sequences, and compares different algorithms through experiments in two aspects, namely, quantitative and qualitative analyses. Section 5 summarizes the conclusions.

3. Methodology

The proposed algorithm mainly includes three parts, namely, CF model, multi-feature fusion model, and re-detection module. KCF (see Section 3.1 for details) uses the cyclic matrix to generate training negative samples to train the filter and designs the kernel function to predict the object position to obtain better tracking results. HOG features are robust to illumination variation. Texture features are generally not disturbed by illumination or background color in the complex environment where the background color is similar to the object color. Color features are not affected by image rotation, translation, and scale variation. A single feature can make it difficult to meet the needs of object tracking in complex scenes. Thus, an adaptive weighted appearance model integrating HOG features, texture features, and color features is proposed (see Section 3.2 for details). The center coordinates of the object obtained from different features are adaptively weighted average, and a penalty term is added to solve the problem of position weight redundancy caused by interference. In the process of long-time object tracking, the object's environment becomes complex and changeable, which will lead to tracking failure. After the object reappears, the re-detection module is started; it uses the multi-layer cellular automata method to detect the significance of the object according to the first frame and the latest tracking results in order to reposition the object (see Section 3.3 for details). The final size of the object is obtained according to the size detection mechanism based on the filter response (see Section 3.4 for details). The algorithm model is shown in Fig. 1. The blue box represents the search box, the red box represents the significance detection result, and the yellow box represents the final detection result.

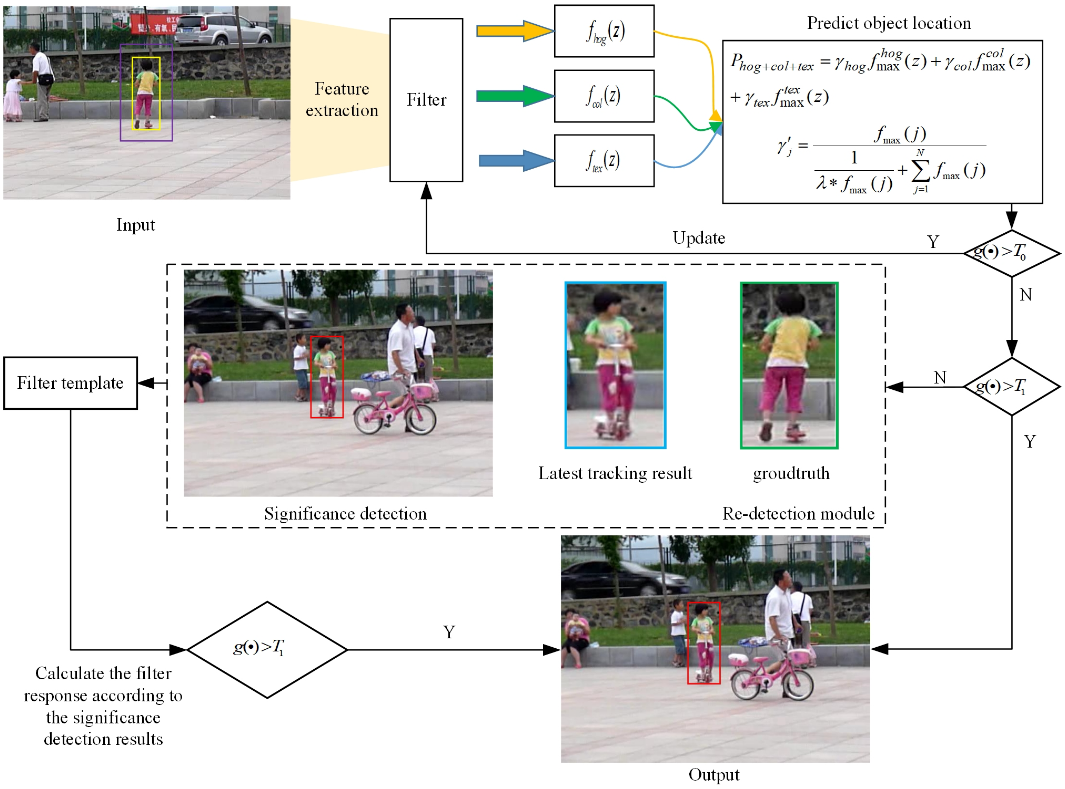

Input
Feature extraction
Filter
$f_{hog}(z)$
$f_{col}(z)$
$f_{tex}(z)$
Predict object location
$P_{hog+col+tex} = \gamma_{hog} f_{\max}^{hog}(z) + \gamma_{col} f_{\max}^{col}(z) + \gamma_{tex} f_{\max}^{tex}(z)$
$\gamma'_j = \dfrac{f_{\max}(j)}{\dfrac{1}{\lambda * f_{\max}(j)} + \sum_{j=1}^{N} f_{\max}(j)}$
Update
Y
$g(\bullet)>T_0$
N
Filter template
Latest tracking result
groudtruth
Significance detection
Re-detection module
N
$g(\bullet)>T_1$
Y
Calculate the filter response according to the significance detection results
$g(\bullet)>T_1$
Y
Output

Fig. 1. Algorithm model

3.1 Correlation filter model

The KCF uses the cyclic matrix to generate training negative samples and designs the kernel function to predict the object position, which solves the problem of insufficient training samples. Given the object position of the initial frame ($target$), a rectangular region is drawn around the $target$ to make it contain enough samples and background information and the trained filter template more robust. The rectangular region can be expressed as $(M,N) = sizeof(target) * (1 + padding)$, where M is the width of the region, N is the height of the region, and $(M/2, N/2)$ is the central position coordinate. The extracted feature map of the t-th frame is $x_t \in R^{M \times N}$, which is a tensor composed of the D-channel features extracted from the t-th frame, where D is the number of channels. The cyclic shift results of feature x_t along the M and N directions are used as training samples. Each shift sample $x_{ij}, (i,j) \in \{0,1,\cdots,M\text{-}1\} \times \{0,1,\cdots,N\text{-}1\}$ has a desired output. The desired output is generated by a Gaussian function, and its peak is the object center position. The desired output of the training image is

$$y_{ij} = e^{-\frac{(i-M/2)^2+(j-N/2)^2}{2\sigma^2}} \tag{1}$$

where σ is the core bandwidth. The center position has the highest score, represented as $Y_{(M/2,N/2)} = 1$. When the coordinate position (i,j) keeps away from the object center, the desired output y_{ij} decreases rapidly from 1 to 0.

A pair of training samples $\{x_t, \mathbf{Y}\}$ are used to learn the filter $\omega_t \in R^{W \times H \times D}$ of frame t, where $x_t \in R^{W \times H \times D}$ is the feature extracted from the image of frame t, including D channels, and the matrix $\mathbf{Y}$ represents the ideal response output. The filter ω_t is solved by finding a classifier x_t and a classifier w with the same size to minimize the error between the output of the filter and the expected output y_{ij}. The discriminative CFs describes it as a regularized least squares objective function.

$$\hat{\omega}_t = \arg\min_{\omega_t} \left\| \sum_{d=1}^{D} \omega_t^d * x_t^d \text{ - } \mathbf{Y} \right\|^2 + \lambda \sum_{d=1}^{D} \left\| \omega_t^d \right\|^2 \tag{2}$$

where x_t^d represents the channel characteristics of layer d of x_t, ω_t^d represents the filter corresponding to layer d, and $*$ represents the cyclic convolution operation. λ is the regularization parameter, and $\sum_{d=1}^{D} \left\| \omega_t^d \right\|^2$ is the regularization term. According to the properties of cyclic convolution structure, the optimized closed solution of Formula (2) can be directly obtained in frequency domain.

$$\hat{\omega}_{ijt} = (\mathbf{I} - \frac{\hat{x}_{ijt}\hat{x}_{ijt}^H}{\lambda n^2 + \hat{x}_{ijt}^H \hat{x}_{ijt}}) \frac{\hat{x}_{ijt}\hat{y}_{ij}}{\lambda n^2} \tag{3}$$

where ω_{ijt} represents the vector composed of the i-th row and j-th column position elements of filter ω_t in all channels. Similarly, x_{ijt} represents the vector composed of the i -th row and j -th column position elements of features x_t in all channels. y_{ij} represents the elements composed of the i -th row and j -th column of matrix $\mathbf{Y}$.

The ridge regression of linear space is mapped to nonlinear space through the kernel function. In nonlinear space, by solving a dual problem and some common constraints, the calculation can also be simplified through the diagonalization of the cyclic matrix Fourier space [24].

$\hat{x}=\sqrt{n}Fx$, F is a discrete Fourier constant matrix, which is expressed as follows:

$$F=\frac{1}{\sqrt{n}}\begin{bmatrix}1 & 1 & \cdots & 1\\ 1 & \omega & \cdots & \omega^{n-1}\\ \cdots & \cdots & \vdots & \cdots\\ 1 & \omega^{n-1} & \omega^{(n-1)(n-2)} & \omega^{(n-1)^2}\end{bmatrix} \quad (4)$$

To further improve the tracking accuracy, the nonlinear regression problem can be transformed into a linear solution by introducing a nonlinear mapping function $\varphi(x_i)$ to map the input data x_i to a high-dimensional space. ω is described as $\omega=\sum_i \alpha_i\varphi(x_i)$, and the kernel function $k(x_i,x_i')=<\varphi(x_i),\varphi(x_i')>$ is introduced. Formula (3) is modified to the following form:

$$\hat{\alpha}=\frac{\hat{y}}{\hat{k}^{xx}+\lambda} \quad (5)$$

where ^ represents the result of Fourier transform. k^{xx} represents the first row vector of the cyclic matrix and is described as $k^{xx}=[k(x,x),k(x,Px),\cdots k(x,P^{n-1}x)]$. A Gaussian kernel is described as $k(x,x')=\exp(-\|x-x'\|^2/\sigma^2)$. Therefore, $k^{xx'}$ is expressed as follows:

$$k^{xx'}=\exp\left(-\frac{1}{\sigma^2}(\|x\|^2+\|x'\|^2-2F^{-1}(\hat{x}^*\odot\hat{x}'))\right) \quad (6)$$

Given a frame of image, z is expressed as the feature vector, and the response will be calculated by Formula (7).

$$f(z)=\omega^T z=\sum_{i=0}^{n-1}\alpha_i k(z,x_i) \quad (7)$$

To speed up the calculation, Formula (7) is diagonalized to obtain the result of Formula (8).

$$\hat{f}(z)=\hat{k}^{xz}\odot\hat{\alpha} \quad (8)$$

$\hat{f}(z)$ is subjected to an inverse Fourier transform expressed as F^{-1}, and the final object region is found by finding the position corresponding to the maximum response.

$$f(z)=F^{-1}\left(\hat{f}(z)\right)=F^{-1}\left(\hat{k}^{xz}\odot\hat{\alpha}\right) \quad (9)$$

3.2 Adaptive weighted multi-feature fusion model

Section 3.1 reveals that the center coordinates $x(i,j)$ of the object will be calculated based on giving a frame x_t and filter ω_t. The n -th image feature is extracted, and the object center coordinate is expressed as $x_n(i,j)$. According to Bayesian principle, Formula (10) is as follows:

$$P(x(i,j)|x_t)=\int P(x(i,j)|B)P(B|x_t)dB\approx\sum_{n=1}^{N}\omega_n P(x(i,j)|B_n) \quad (10)$$

where N is the number of fused features, and ω_n is the confidence of likelihood distribution, which is expressed as $\omega_n=P(B_n|x_t)$, thus satisfying $\sum\omega_n=1$.

The adaptive weighted multi-feature fusion model is shown in Fig. 2. After the features in the search box are extracted, the filter responses of different features are calculated respectively, and the object position is finally determined according to the adaptive weighting coefficient.

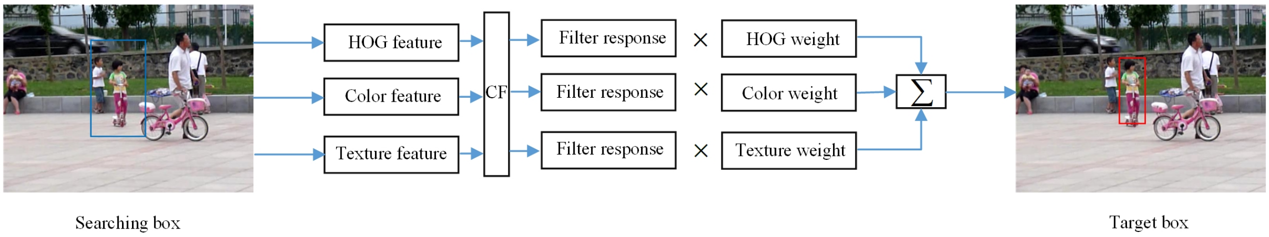


Fig. 2. Adaptive weighted multi-feature fused model

The probability density distribution of characteristic n ($n=1,2,\cdots,N$) is p_{ij}^n ($(i,j)\in\{1,2,\cdots,M\}\times\{1,2,\cdots,N\}$), satisfying $\sum p_{ij}^n=1$. The center coordinates of the object are obtained according to each feature extracted. Then, the final center position of the object is obtained by adaptive weighting according to the center coordinate, which is expressed as follows:

$$P_i=\gamma_1 P_{i1}+\gamma_2 P_{i2}+\cdots+\gamma_n P_{iN}$$
$$s.t.\ \sum_{j=1}^{N}\gamma_j=1 \quad (11)$$

P_i represents the center position of the object in frame i, P_{ij} represents the center position of the object obtained according to the j -th feature in frame i , and γ_j is the weight coefficient of the j -th feature.

The calculation of the weight coefficient in Formula (11) becomes the key to determine the final object position. Generally, a good feature representation will obtain higher response value in filter calculation. The predicted position is more accurate when the response value is large, and so the higher weight coefficient should be assigned. Therefore, the weight coefficient is automatically adjusted according to the maximum response value of each feature, which is calculated as follows:

$$\gamma_j = \frac{f_{\max}(j)}{\sum_{j=1}^{N} f_{\max}(j)} \tag{12}$$

$f_{\max}(j)$ is the maximum filter response of the j -th characteristic calculated using Formula (9). The way to obtain the weight coefficient by simple weighted average may lead to excessive position weight due to the influence of interference factors. Therefore, a simple penalty term is used to solve this problem.

$$\gamma'_j = \frac{f_{\max}(j)}{\frac{1}{\lambda * f_{\max}(j)} + \sum_{j=1}^{N} f_{\max}(j)} \tag{13}$$

$$\gamma''_j = \frac{\gamma'_j}{\sum \gamma'_j} \tag{14}$$

γ''_j is the weight coefficient of the j -th feature after adjustment, and λ represents a penalty term coefficient. $\frac{1}{\lambda * f_{\max}(j)}$ is a penalty term, which is used to assign high weight coefficients to features with high response values and low weight coefficients to features with low response values.

The positions of the maximum response P_{hog} , P_{col} , and P_{tex} of HOG features, color features, and texture features are calculated respectively according to Formula (9).

$$\begin{aligned} P_{hog} &\Leftarrow f_{\max}^{hog}(z) = \arg\max_{n}(f_{hog}(z^1), f_{hog}(z^2), \cdots f_{hog}(z^n)) \\ P_{col} &\Leftarrow f_{\max}^{col}(z) = \arg\max_{n}(f_{col}(z^1), f_{col}(z^2), \cdots f_{col}(z^n)) \\ P_{tex} &\Leftarrow f_{\max}^{tex}(z) = \arg\max_{n}(f_{tex}(z^1), f_{tex}(z^2), \cdots f_{tex}(z^n)) \end{aligned} \tag{15}$$

The final position of the object is obtained by fusing these three features, which is expressed as follows:

$$P_{hog+col+tex} = \gamma_{hog}P_{hog} + \gamma_{col}P_{col} + \gamma_{tex}P_{tex} \tag{16}$$

The current frame response peak $f_{\max}$ is calculated according to Formula (17), and when it is greater than the threshold T_0 , the object model is updated.

$$f_{\max} = \arg\max\left\{ f(z) = F^{-1}\left(\hat{f}(z)\right)\right\} \tag{17}$$

Assuming that the learning rate is γ , the update strategy is as follows:

$$x_t = (1-\gamma)x_{t-1} + \gamma x_t \tag{18}$$

$$\alpha_t = (1-\gamma)\alpha_{t-1} + \gamma\alpha_t \tag{19}$$

3.3 Re-detection module based on object guided saliency detection

$g(\bullet)$ is defined as the filter response, and the tracking result is detected by checking whether $g(\bullet)$ is lower than a given threshold T_1 . If $g(\bullet)$ is less than the threshold, it indicates that the tracking fails and so the re-detection program will be started. The model is shown in Fig. 3. First, the saliency map integrating texture features, color contrast, space features, and edge features is calculated according to the saliency calculation method and then the object is located by searching and calculating the maximum confidence value. The re-detection module is based on object guidance, and the input is the first frame image and the latest tracking result. These two images are selected as the input of re-detection for the following reasons. The object position of the first frame is manually marked, which may bring as little error as possible to the re-detection. However, the object tracking is a dynamic task. When tracking fails, the pose and background of the object may be quite different from those of the first frame. Therefore, the latest tracking result is also used as the input of re-detection.

(1) Texture feature extraction

To solve the problem of local binary pattern (LBP) mode being sensitive to noise, an improved LBP extraction mode is designed to add an offset to the central pixel g_c and improve the correlation between neighborhood pixels g_i and central pixels g_c . The coding method of the improved LBP feature extraction mode is as follows:

$$\begin{aligned} & S_{wen}(x_c, y_c) = \sum_{i=0}^{P-1} 2^p s(g_i - g_c) \\ & s(x) = \begin{cases} 1 & x \ge \kappa g_c \\ 0 & x < \kappa g_c \end{cases} \end{aligned} \tag{20}$$

where $\kappa \in [0,1]$ is an offset factor. When $\kappa = 0$, this mode is LBP. As κ increases, the influence of noise on the calculation result decreases.

(2) Salient edge extraction

The effective extraction of significant edges plays a key role in target detection. The significant edges of the image are obtained by combining the Kirsch detection operator and local contrast. Kirsch operator can preserve details and has good effect in minimizing noise. Local contrast highlights the image edge and makes up for the poor continuity of the image edge obtained by the Kirsch operator. Image edges are represented as follows:

$$S_{edge} = S_{\text{Kirsch}} \times (C^{loc})^2 \tag{21}$$

where S_{Kirsch} is the image edge extracted by the Kirsch operator, and C^{loc} represents the local contrast of the image.

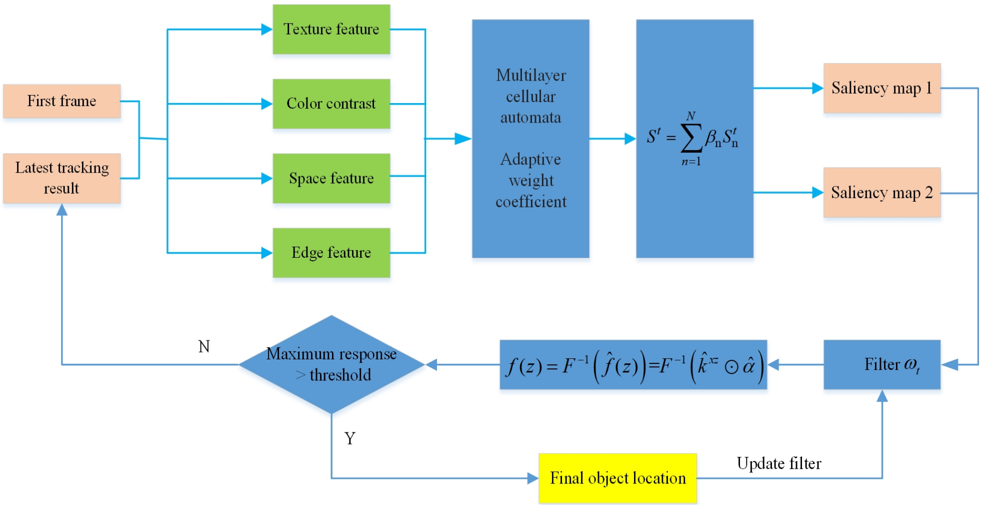

Texture feature
Color contrast
Space feature
Edge feature
First frame
Latest tracking result
Multilayer cellular automata
Adaptive weight coefficient
$S^t = \sum_{n=1}^{N} \beta_n S_n^t$
Saliency map 1
Saliency map 2
N
Maximum response > threshold
$f(z) = F^{-1}\left(\hat{f}(z)\right) = F^{-1}\left(\hat{k}^{xz} \odot \hat{\alpha}\right)$
Filter ω_t
Y
Final object location
Update filter

Fig. 3. Saliency detection model based on object guidanc

(3) Significance detection based on multi-layer cellular automata

Taking the object in the first frame and the latest tracking result as the input of significance detection, texture features, color contrast, spatial features and significant edge features are extracted, which are represented by S_{wen}, S_{cl}, S_{ct}, and S_{edge} respectively. The multi-layer cellular automata (MCA) synchronous update method is used to perform fusion optimization on different salient metrics for further improving the results of salience. For the input image, each pixel in the image represents a cell. The pixels at the same position in different scale saliency maps are neighbors to each other, and each neighbor has the same effect on the state of the cell at the next moment. The different saliency maps generated by various methods are spread to each layer of the cellular automata. The weights of different features are automatically generated according to the intersection of the histogram by calculating the histogram of the target, background, and entire areas of each image. The weights are used to optimize the texture features, color contrast, spatial features, and salient edge features. A final salience map S with better fusion effect is obtained.

$$S^t = \frac{1}{N}\sum_{n=1}^{N} S_n^t \tag{22}$$

where N is 4, which is the number of different scale salient maps. s_n^t iteratively updated by MCA mechanism represents the salient value of all pixels after time t, that is, $S_n^t \propto [S_{wen}^t, S_{cl}^t, S_{ct}^t, S_{edge}^t], n = 1 \to 4$.

The visual salience of an image is determined by the strongest response of multiple salient features, but unreasonable salience map fusion may further reduce the detection performance of the algorithm. In different scenes, the contribution of the salient feature map is different, so each feature needs to be weighted, and Formula (22) can be changed to

$$S^t = \sum_{n=1}^{N} \beta_n S_n^t = \beta_1 S_1^t + \ldots + \beta_n S_n^t \tag{23}$$

where $\sum \beta_n = 1, n = 1 \to N$, $\beta_1, \ldots, \beta_n$ is the weight coefficient of different salient feature maps. The proposed multi-feature template can reduce the intensity of background interference in different spaces. When the background changes, the weight coefficient $\beta_1, \ldots, \beta_n$ should also be different and can change adaptively. The value of each feature coefficient β_n is determined by comparing the histograms of different features, such as texture features, color contrast, and edge features of the salient area and the environmental background.

According to the above calculation method, two object saliency maps will be obtained, one is calculated based on the object position in the first frame, and the other is based on the object position provided by the latest tracking result. These two object saliency maps are not necessarily the final prediction results, and there may be some errors. Therefore, the saliency map also needs to be filtered to get the correct re-detection results. If the obtained response value meets the threshold, then the object position can be updated according to the saliency map position, and a new filter ω_{t+1} can be updated at the same time.

3.4 Size estimation mechanism

Object tracking task is realized by predicting the object size and position frame by frame given the initial frame. On the premise of eliminating the influence of interference, there is a certain relationship between the maximum filter response value of two adjacent frames and the size of the target. As the object size decreases, the response value increases. On the contrary, when the object size increases, the response value decreases. Hence, there is an opposite relationship between the object size and the maximum response value. Moreover, a mechanism is designed to determine the object size by using the response values of two adjacent frames. The object size corresponding to the three features is used as the weight to obtain the object size, and the change rate is expressed by C. Assuming that the given initial frame size is represented by Sz_1, the object size of frame t is represented by Sz_t, and the object size Sz_{t+1} of frame $t+1$ can be calculated by Formula (24).

$$\begin{aligned} Sz_{t+1} &= Sz_t \times C \\ C &= \frac{C_h + C_c + C_t}{2} \end{aligned} \tag{24}$$

C_h , C_c, and C_t represent the change rate of the HOG features, color features and texture features, respectively.

3.5 Algorithm flow

The object appearance representation model is constructed, and the discriminative filter is trained to predict the object location. When the detection result is unreliable, the saliency detection module is started for object re-detection to realize effective object tracking in complex scenes.

The algorithm is described as follows:

Step 1. Read the video frame I_{t+1} and initialize the initial object position (x_t, y_t, w_t, h_t) .

Step 2. Extract the searching candidate window centered on (x_t, y_t) in the I_{t+1} frame and calculate the HOG, color, and texture features of the search window.

Step 3. Use the optimized filter ω_t of the t -th frame in Section 3.1 to calculate the response of different features and predict the center position of the object respectively using Formula (15).

Step 4. Calculate the final location of fusion with adaptive weight coefficient using Formula (16).

Step 5. If the maximum response is greater than the threshold T_0 , go to Step 9; otherwise, proceed to the next steps.

Step 6. If the maximum response is less than the threshold T_1, then restart the detection module.

Step 7. Perform significance detection according to the first frame and the latest tracking result.

Step 8. Recalculate the filter response of the significance object. If the response is greater than the threshold T_0 , go to Step 9; otherwise, go to Step 10.

Step 9. Update the filter model ω_{t+1} with Formulas (18) and (19), and output the final location of the object.

Step 10. $t+1 \to t$. Turn to Step 3 to continue until the end of the video sequence.

4. Result Analysis and Discussion

To verify the effectiveness of the multi-feature fusion model proposed in Section 3.2 and the CF algorithm based on significance guided proposed in Section 3.3, the performance of the algorithm is analyzed and discussed through a large number of experiments. The proposed algorithm, KCF[8], Struck[11], TLD-CN[14], and MFFT[24] are compared on the OTB2015. These image sequences have various problems, such as deformation, occlusion, similar background color, scale variation, motion blur, and lighting effects. The experimental simulation environment is MATLAB R2018b. The computer configuration is as follows: Intel Core i7-8550U CPU, 2.0 GHZ frequency, 8GB memory, Windows 10 operating system.

The regularization term is used to prevent model overfitting, and its value directly affects the tracking performance. If λ is too small, the regularization term is inactive. In contrast, if λ is too large, the regularization term dominates the overall error. The threshold T_0 and T_1 are used to determine the time to update the tracker model. If T_0 and T_1 are too small, the tracker will easily drift due to noisy updating. However, if T_0 and T_1 are too large, the tracker cannot adapt to the appearance changes of the tracking object. The parameter settings are as follows: the regularization parameter λ is 0.005, the update threshold T_0 of the object model is 0.35, the re-detection threshold T_1 is 0.2, and the learning rate γ is 0.80.

4.1 Evaluating indicator

Tracking performance, center position error, distance precision rate, overlap rate, and overlap success rate are selected as evaluation indexes to compare and analyze different algorithms.

(1) Center position error (CPE)

The center position error refers to the Euclidean distance between the estimated position (x', y') obtained by iteration and the true position (x, y) , which can be calculated as $D = \sqrt{(x-x')^2 + (y-y')^2}$. As D decreases, the accuracy and stability of the algorithm increases.

(2) Distance precision rate (DPR)

The tracking algorithm estimates the Euclidean distance between the center point of the target position and the center point of the manually marked target. Otherwise, the smaller the Euclidean distance, the higher the tracking accuracy. Range accuracy refers to the percentage of video frames whose Euclidean distance is less than a given threshold to the total number of frames. With different thresholds, ratios are different, a curve can be obtained, and the threshold is set to 20 pixels.

(3) Overlap rate (OR)

The overlap rate between the predicted bounding box S_P estimated by the tracking algorithm and the ground-truth bounding box S_G is calculated using Formula (25). As the overlap rate increases, the tracking success rate increases. The intersection and union of these two bounding boxes are represented by $\cap$ and $\cup$, and their area is represented by $Area(\bullet)$.

$$S = \frac{|Area(S_P \cap S_G)|}{|Area(S_P \cup S_G)|} \tag{25}$$

(4) Overlap success rate (OSR)

The overlap success rate represents the percentage of frames with overlap rate that is greater than a given threshold. With different thresholds, ratios differ, a curve can be obtained, and the threshold is set to 0.5.

(5) Tracking performance

Some tracking algorithms are designed very complex and maintain a high success rate, but the real-time performance can not meet the requirements. Some algorithms run faster, but they can not track accurately in various complex scenes. To evaluate the tracking performance of the algorithm, we need to consider both real-time performance and success rate.

4.2 Quantitative analysis

Images in the tracking data set are different in contrast, background interference, image noise, and so on. The adaptive weighted fusion algorithm can automatically adjust the weight according to the characteristics of each image to achieve better tracking effect.

(1) Center position error

To verify the relocation ability of the proposed algorithm, the video sequence Car4 with 659 images having obvious illumination variation is selected as the experimental data. The comparison results are shown in Fig. 4. CPE of the proposed algorithm is only 2.32 while CPE of the TLD-CN algorithm reaches 66.62, resulting in complete failure. When the object gradually moves away from the line of sight, it is affected by the interference. The center position offset predicted by other comparison algorithms is greater than 20 pixels, resulting in tracking drift and the inability to track the object correctly in subsequent frames.

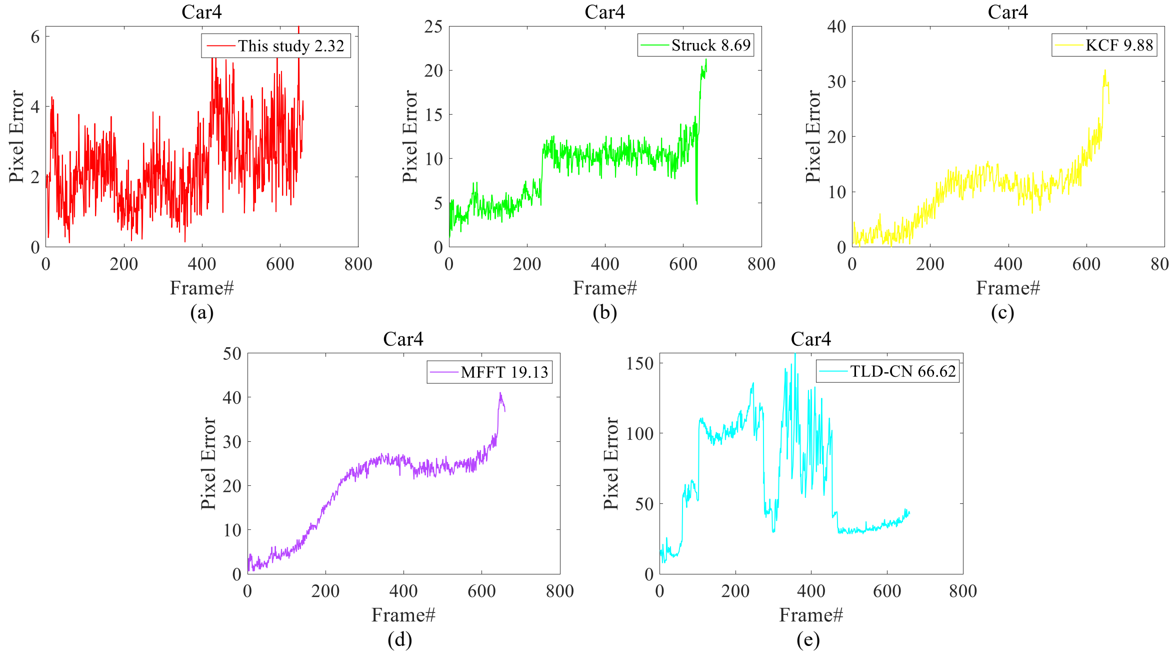


Fig. 4. Comparison of CPE of different algorithms in Car4. (a) CPE of this study. (b) CPE of Struck. (c) CPE of KCF. (d) CPE of MFFT. (e) CPE of TLD-CN

(2) Distance precision rate

Fig. 5 shows the DPR results of different algorithms in the video sequence Girls2, in which the object is occluded for a long time. The experimental results show that the DPR of the proposed algorithm and MFFT algorithm is high while the DPR of the other three comparison algorithms cannot meet the requirements. When the threshold is set to 20, the DPR of the proposed algorithm is close to 1. When the target is deformed, the background noise is large, illumination changes, and other complex effects occur, the proposed algorithm maintains good tracking ability and stability under complex environments with deformation, background clutters, and illumination variation.

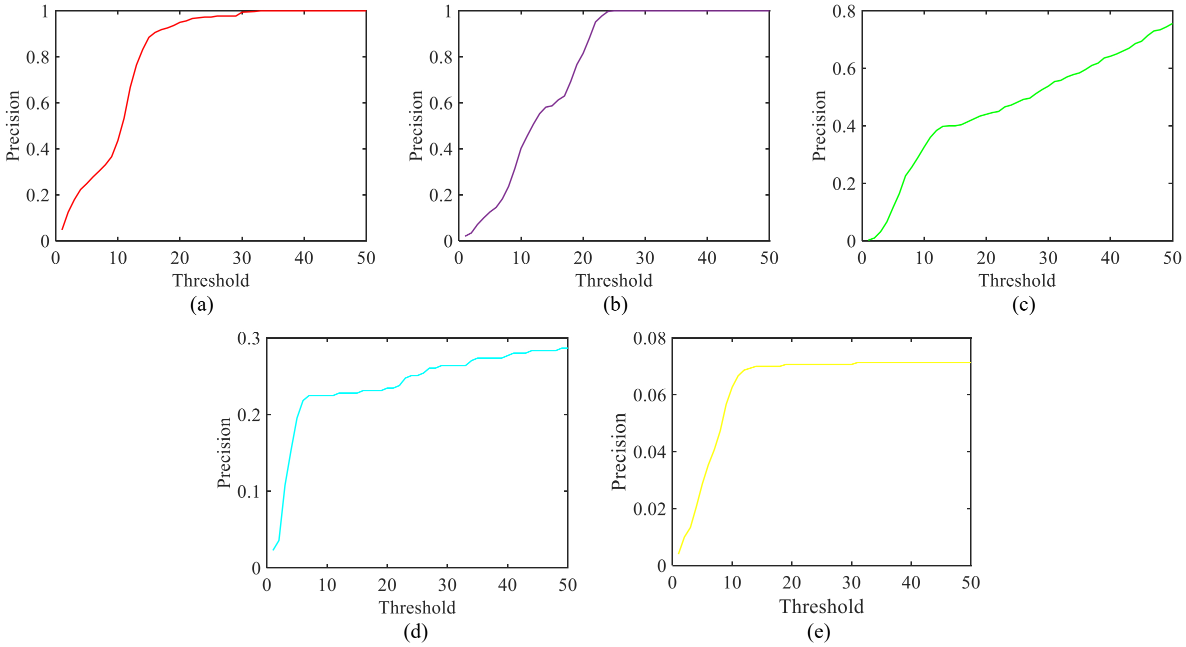
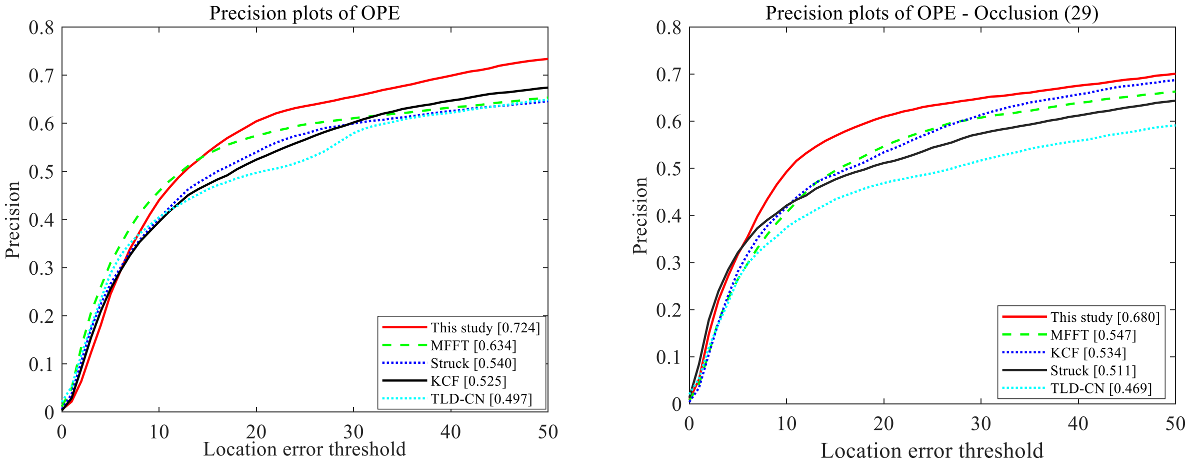


Fig. 5. Comparison of DPR of different algorithms in Girls2. (a) DPR of this study. (b) DPR of MFFT. (c) DPR of Struck. (d) DPR of TLD-CN. (e) DPR of KCF

Fig. 6 shows the comprehensive statistical results between the proposed algorithm and other comparison algorithms on OTB2015. The DPR of the proposed algorithm is shown to be the highest, reaching 0.724. The proposed algorithm with 0.680 scores achieves top rank in complex scenes with occlusion, which is 19.56% higher than that of the second-ranked MFFT (0.547), thus verifying the effectiveness of the algorithm for object relocation.

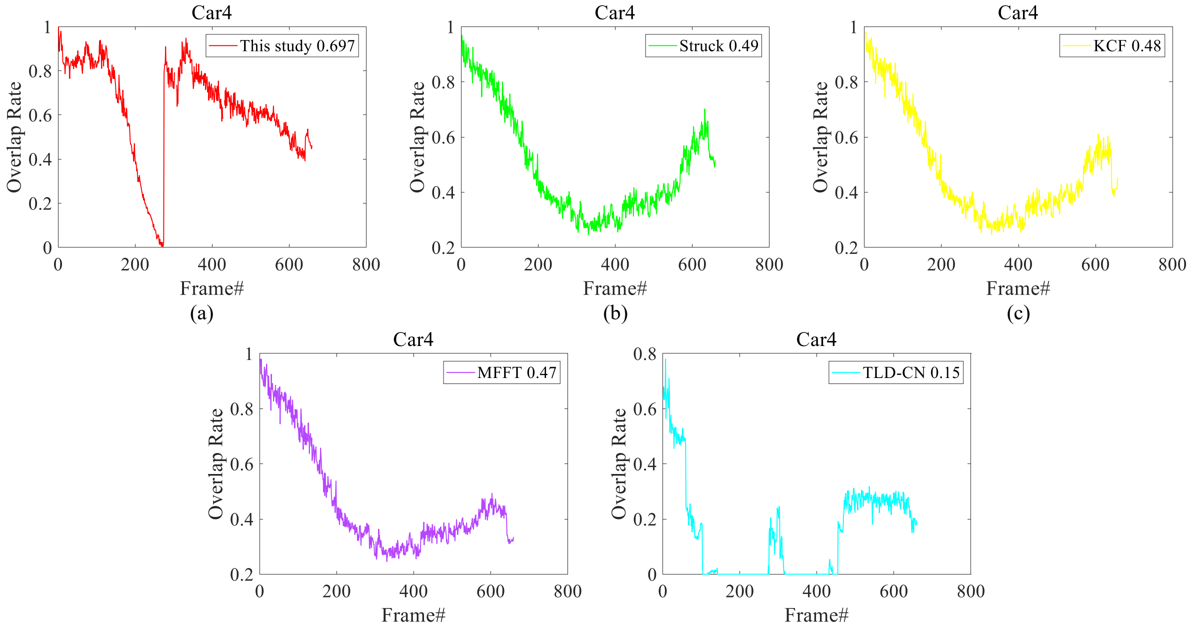


(a) (b)

Fig. 6. Comparison of DPR of different algorithms on OTB2015. (a) DPR of different algorithms. (b) Comparison of DPR of different algorithms under occlusion scenes

(3) Overlap rate

Fig. 7 shows the analysis result of the OR of the proposed algorithm and the comparison algorithms in the video sequence Car4. The average OR of the proposed algorithm is 0.697. When the fluctuation occurs near 270 frames, the target saliency guidance module is started, and the normal tracking mode is quickly restored. The OR of TLD-CN algorithm is only 0.15, which is less robust in the scene of background interference.



(a) (b) (c)

(d) (e)

Fig. 7. Comparison of OR of different algorithms in Car4. (a) OR of this study. (b) OR of Struck. (c) OR of KCF .(d) OR of MFFT. (e) OR of TLD-CN

(4) Overlap success rate

Fig. 8 shows the comparison curve of OSR of different algorithms on OTB2015. The proposed algorithm benefits from the adaptive appearance representation model, with the highest comprehensive score and an average OSR of 0.673. In the scene where the object is occluded, the overlap success rate is 0.624, which is 8.17% higher than that of the second-ranked MFFT (0.573) and 43.59% higher than that of the Struck algorithm (0.352).

(5) Tracking performance

Tracking speed is described by frames per second (FPS), as shown in Table 1. The tracking speed of the proposed algorithm is 57FPS, which is slightly worse than that of the KCF algorithm but can meet basic real-time tracking requirements. Although the designed tracker is not the fastest compared with the comparison algorithm, the overlap success rate is obviously better than the other comparison trackers.

Table. 1. Tracking performance of different algorithms

Performance index	This study	KCF	Struck	TLD-CN	MFFT
Tracking speed	57	89	6	20	50
OSR	0.673	0.474	0.477	0.417	0.588

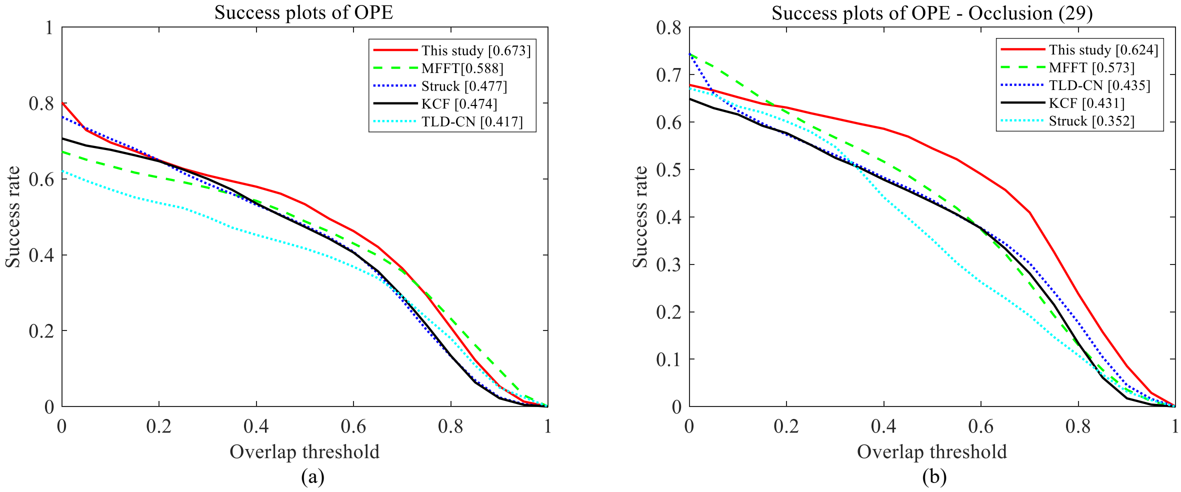


Fig. 8. Comparison of OSR of different algorithms on OTB2015. (a) OSR of different algorithms. (b) Comparison of OSR of different algorithms under occlusion scenes

4.3 Qualitative analysis

The comprehensive effect of the algorithm on OTB2015 is analyzed and discussed in Section 4.2. The visualization results of the algorithm will be analyzed in this section to more intuitively illustrate the accuracy of the proposed algorithm.

Fig. 9 shows the comparison of other edge detection operators with the detection result of this study. Figs. 9(b–d) show that the edges obtained by the Sobel, Roberts, and Kirsch operators have poor continuity and many fractures. The edge map as shown in Figs. 9(e–f) obtained by log and canny operators contains the change of background texture, which cannot highlight the contour information of the foreground. Fig. 9(g) shows the edge detection results obtained by Harris operator, and the corner detection is not complete. Fig. 9(h) shows the edge extraction results proposed in this study. The algorithm obtains edge information with good continuity and eliminates too many background texture details.

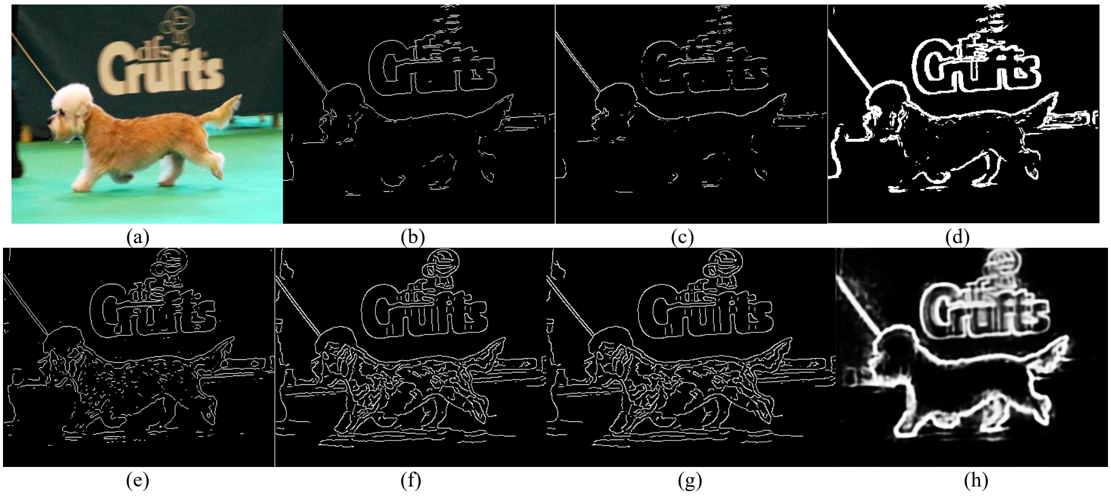
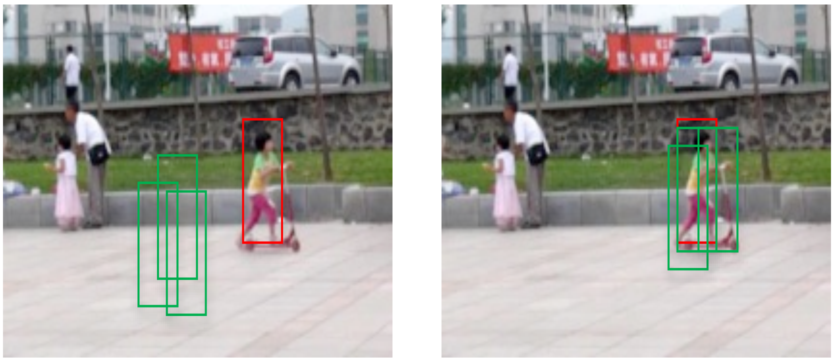


Fig. 9. Comparison of edge detection effect. (a) Original image. (b) Sobel operator. (c) Roberts operator. (d) Kirsch operator (e) log operator. (f) Canny operator. (g) zerocross operator. (h) This study

To re-detect the object from tracking failures, EdgeBox[18] method is used to generate region proposals of the whole image from tracking failures as well as for scale change estimation. The red bounding box represents the ground truth, and the green bounding box represents the proposals. The proposals generated by Edgebox are around the original location of the occlusion (Fig. 10(a)), but the proposals generated by the proposed method are around the tracked object (Fig. 10(b)). As can be seen from Fig. 10(a), when the tracking target reappears, the tracking still fails.

Fig. 11 shows the comparison results between the proposed algorithm and the other four algorithms (KCF, Struck, TLD-CN, and MFFT) in four typical video sequences, namely, Walking2, Girl2, Matrix, and Car4. The images in the Walking2 video sequence have problems such as low resolution, occlusion, and scale variation. The images in the Girl2 video sequence have problems such as scale variation, deformation, motion blur, and rotation. The images in the Matrix video sequence have problems such as illumination, occlusion, fast motion, scale variation, rotation, and background clutter. The images in the Car4 video sequence have problems such as uneven illumination, similar colors, scale changes, and fast motion.



(a) (b)

Fig. 10. Region proposals. (a) Proposals produced by EdgeBox. (b) Proposals produced by the proposed method

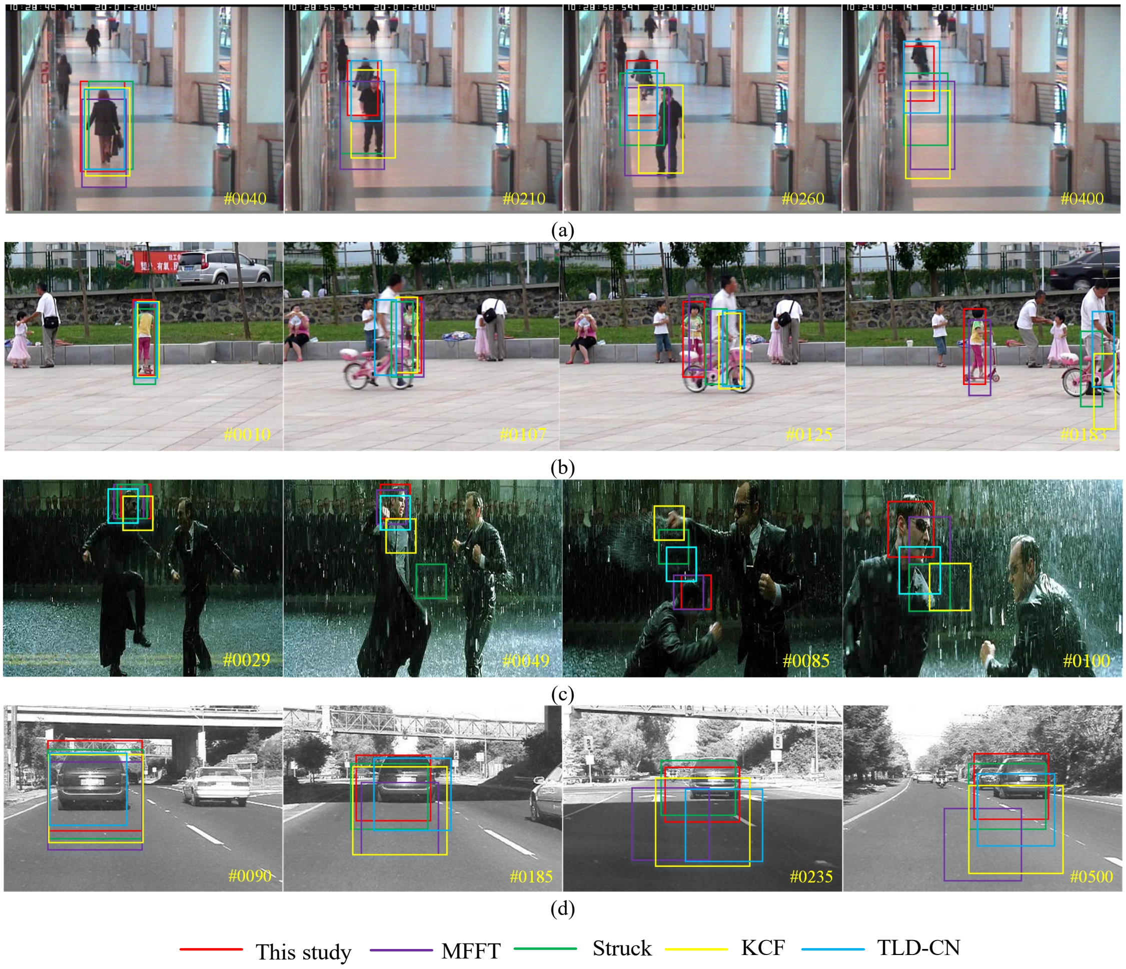


Fig. 11. Qualitative comparison of different algorithms in typical video sequences. (a) Video sequenceWalking2. (b) Video sequence Girl2. (c) Video sequence Matrix. (d) Video sequence Car4

The object in video sequence Walking2 is occluded from 197 frames, resulting in the tracking drift in the comparison algorithm as shown in Fig. 11(a). The traditional KCF algorithm fails completely, and the tracking box still stays in the position before occlusion. The proposed algorithm can locate the object when the object is occluded. The TLD-CN algorithm benefits from adaptive size adjustment, and the tracking effect is also ideal. The target has video sequence Girl2 with severe occlusion and long-time occlusion. The proposed algorithm and MFFT algorithm achieve good tracking results in video sequence Girl2 with severe occlusion and long-time occlusion. Other comparison algorithms have tracking drift and do not have the ability to re-track. The proposed algorithm effectively removes the background information to establish an effective model to smooth the filter as well as achieves better tracking results in the video sequence Matrix, including background interference and rotation as shown in Fig. 11(c), whereas other comparison algorithms have tracking drift. Fig. 11(d) shows the tracking comparison results of different algorithms in video sequence Car4. The object in video sequence Car4 has fast speed and large illumination changes, which lead the proposed algorithm to some errors in tracking when the target changes lanes; however, no failure occurs. The Struck and KCF algorithms show varying degrees of drift and even lead to tracking failures. The qualitative analysis results show the obvious advantages of the proposed algorithm, which further verify the effectiveness of the re-detection module in the tracking process.

5. Conclusions

Aiming at the shortcomings of traditional CF in the object appearance model and the relocation method, this study constructs a discriminative feature model and explores an adaptive multi-feature fusion algorithm by combining saliency detection technology and object tracking method. The following conclusions can be drawn:

(1) There is a correlation between the different feature representation and the CF responses. The contribution of features is adaptively adjusted according to the actual tracking environment so that different features will benefit from one another in different scenes. The learned filter model is used to calculate the fused position of the object.

(2) The different significant prior features are integrated into the tracking framework. The first frame and the latest tracking result are used as the object guidance to obtain the significant position of the object and achieve the relocation.

(3) On the premise of excluding the influence of interference, there is an opposite relationship between the maximum filter response of the two adjacent frames. The change of the object size is determined according to the response values of the two adjacent frames, and the object size is predicted.

The proposed algorithm combines image saliency detection and object detection. The established object appearance model and the relocation model make the proposed algorithm more robust in various complex scenes, which is of great benefit to the subsequent development of traffic event detection algorithms. Owing to the lack of testing and verification on the video sequences in the actual application scenes, the proposed algorithm will be verified in a real-time video for optimization in future studies.

Acknowledgments

This work was supported by Xuzhou Science and Technology Plan Project (No. KC21303).

References

1. Majd, M., Safabakhsh, R., "Correlational convolutional LSTM for human action recognition". *Neurocomputing*, 396, 2020, pp.224-229.
2. Howard, W., Nguang, S. K., Wen, J. W., "Robust video tracking algorithm: a multifeature fusion approach". *IET Computer Vision*, 12(5), 2018, pp.640-650.
3. Razzaq, M. A., Quero, J. M., Cleland, I., Nugent, C., Lee, S., "uMoDT: An unobtrusive multi-occupant detection and tracking using robust kalman filter for real-time activity recognition". *Multimedia Systems*, 26(5), 2020, pp.553-569.
4. Li, S. N., Qin, Z., Song, H. B., "A temporal-spatial method for group detection, locating and tracking". *IEEE Access*, 4, 2016, pp.4484-4494.
5. Xu, T. Y., Feng, Z. H., Wu, X. J., Kittler, J., "Joint group feature selection and discriminative filter learning for robust visual object tracking". In: *IEEE International Conference on Computer Vision (ICCV)*, Seoul, Korea: IEEE, 2019, pp.7949-7959.
6. Muller, M., Bibi, A., Giancola S., Alsubaihi， S., Ghanem, B., "TrackingNet: A large-scale dataset and benchmark for object tracking in the wild". In: *European Conference on Computer Vision(ECCV)*, Germany, Munich: Springer, 2018, pp.300-317.
7. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., He, Z., "The sixth visual object tracking VOT2018 challenge results". In: *European Conference on Computer Vision(ECCV)*, Germany, Munich: Springer, 2018, pp. 3-53.
8. Henriques, J. F., Caseiro, R., Martins, P., Batista, J., "High-speed tracking with kernelized correlation filters". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3), 2015, pp.583-596.
9. Karunasekera, H., Wang, H., Zhang, H., "Multiple object tracking with attention to appearance, structure, motion and size". *IEEE Access*, 7, 2019, pp.104423-104434.
10. Meng, L., Yang, X., "A Survey of Object Tracking Algorithms". *Acta Automatica Sinica*, 45(7), 2019, pp.1244-1260.
11. Hare, S., Saffari, A., Torr, P. H. S., "Struck: Structured output tracking with kernels". *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 38(10), 2015, pp.2096-2109.
12. Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., Torr, P. H. S., "Staple: complementary learners for real-time tracking". In: *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, Las Vegas, NV, USA: IEEE, 2016, pp.1401-1409.
13. Galoogahi, H. K., Fagg, A., Lucey, S., "Learning Background-Aware Correlation Filters for Visual Tracking". In: *IEEE International Conference on Computer Vision(ICCV)*, Venice, Italy: IEEE, 2017, pp.1369-1378.
14. Zhang, J., Huang, H. M.,Wang, J. M., Bao, J. R., "An improved tld real-time target tracking algorithm based on cn algorithm". *Computer Engineering & Science*, 42(7), 2020, pp.1215-1225.
15. Zhao, G. H., Zhuo, S., Xu, X. L., "Multi-object tracking algorithm based on kalman filter". *Computer Science*, 45(8), 2018, pp.253-257.
16. Zhang, Z. L.,Wang, Y. X., "SiamRPN target tracking method based on kalman filter". *Intelligent Computer and Applications*, 10(3), 2020, pp.44-50.
17. Akhtar, J., Bulent, B., "The delineation of tea gardens from high resolution digital orthoimages using mean-shift and supervised machine learning methods". *Geocarto International*, 36(7), 2021, pp.758-772.
18. Liu, H., Hu, Q. Y., Li, B., Guo, Y., "Robust long-term tracking via instance-specific proposals". *IEEE Transactions on Instrumentation and Measurement*, 69(4), 2020, pp.950-962.
19. Tünnermann, J., Born, C., Mertsching, B., "Saliency From Growing Neural Gas: Learning Pre-Attentional Structures for a Flexible Attention System". *IEEE Transactions on Image Processing*, 28(11), 2019, pp.5296-5307.
20. Xiong, D., Lu, H. M., Xiao, J. H., Zheng, Z. Q., "Robust long-term object tracking with adaptive scale and rotation estimation". *Acta Automatica Sinica*, 45(2), 2019, pp.289-304.
21. Di, N., Zhu, M., Han, G. L., "Research on the Robust Illumination of the Likelihood Similarity Function in Tracking Target". *Journal of Software*, 26(1), 2015, pp.52-61.
22. Yuan, D., Fan, N. N., He, Z. Y., "Learning target-focusing convolutional regression model for visual object tracking". *Knowledge-Based Systems*, 194, 2020, pp.105526.
23. Lukezic, A., Vojir, T., Zajc, L. C., Matas, J., Kristan, M., "Discriminative correlation filter with channel and spatial reliability". In: *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, Honolulu, HI, USA: IEEE, 2017, pp.6309-6318.
24. Yuan, D., Zhang, X. M., Liu, J. Q., Li, D. H., "A multiple feature fused model for visual object tracking via correlation filters". *Multimedia Tools and Applications*, 78, 2019, pp.27271-27290.