



Overview of correlation filter based algorithms in object tracking

Shuai Liu¹ · Dongye Liu² · Gautam Srivastava^{3,5} · Dawid Połap⁴ · Marcin Woźniak⁴

Received: 10 March 2020 / Accepted: 27 May 2020

© The Author(s) 2020

Abstract

An important area of computer vision is real-time object tracking, which is now widely used in intelligent transportation and smart industry technologies. Although the correlation filter object tracking methods have a good real-time tracking effect, it still faces many challenges such as scale variation, occlusion, and boundary effects. Many scholars have continuously improved existing methods for better efficiency and tracking performance in some aspects. To provide a comprehensive understanding of the background, key technologies and algorithms of single object tracking, this article focuses on the correlation filter-based object tracking algorithms. Specifically, the background and current advancement of the object tracking methodologies, as well as the presentation of the main datasets are introduced. All kinds of methods are summarized to present tracking results in various vision problems, and a visual tracking method based on reliability is observed.

Keywords Artificial intelligence · Object tracking · Correlation filter algorithms · Dataset · Template update strategy

Introduction

Rapid developments of artificial intelligence and computer vision have been widely visible in various fields. Computer vision refers to the use of cameras and computers instead of

human eyes to visually recognize, track, and measure targets. First, image processing is performed so that the processed image is more suitable for human eye observation or instrument detection. Then, the process of visual object tracking is to track the target state in subsequent video sequence frames with the presented initial position and size of the target. Currently, object tracking is widely used in transportation hub monitoring, medical imaging, human-computer interaction, and other related fields. Scholars have done lots of work in these areas. Specific applications, among several other, are:

- Application of object tracking in unmanned aerial vehicles (UAVs). UAVs are commonly known as drones. Compared with the human eye, the drone has the advantages of stability and accurately capture. Therefore, drones using object tracking technology can achieve more robust tracking results [1]. Object tracking technology is also be used to identify and track specific target in a wide area to avoid accidents. For example, it is used for online grazing of grassland flocks, forest fire monitoring and early warning.
- Application of object tracking in industrial automation. All industries are developing intelligently, and object tracking is also widely used in industrial production [2]. Computer vision technology can be used to identify and track problematic products. However, in actual automated industrial production, the speed of product trans-

¹ College of Information Science and Engineering, & Hunan Provincial Key Laboratory of Intelligent Computing and Language Information Processing, Hunan Normal University, Changsha, China

² College of Computer Science, Inner Mongolia University, Hohhot, China

³ Department of Mathematics and Computer Science, Brandon University, Brandon, Canada

⁴ Institute of Mathematics, Silesian University of Technology, Gliwice, Poland

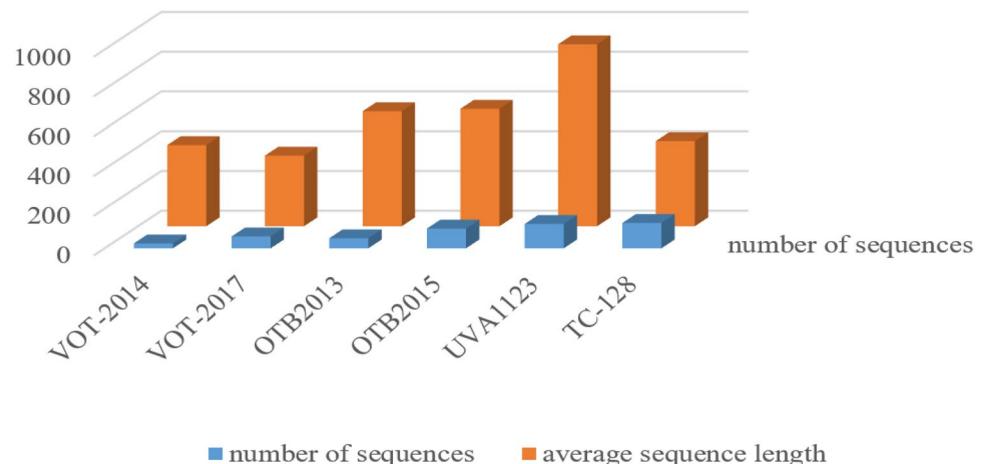
⁵ Research Centre for Interneual Computing, China Medical University, Taichung, Taiwan

mission is fast and camera model equipment is limited. Thence, it is easy to cause motion blurring and increase the tracking difficulty. Therefore, the realization of intelligent industrial production still needs to study the object tracking technology continuously.

- Application of object tracking in intelligent transportation. Today, an efficient transportation selection system is vital for the masses. In intelligent transportation, real-time monitoring and tracking of vehicles can be achieved using object detection and tracking technology [3]. The object detection and tracking technology can be used to obtain the current lane congestion and then develop or select an optimized travel plan.

In addition to drones, industrial and intelligent transportation, artificial intelligence technologies such as object tracking can also be used in mobile healthcare, military space and intelligent transportation. For example, Chen [4] proposed the FGM–ACO–FWA method to solve the problem of unsustainability using smart technologies in mobile medicine. According to the different applications of aircraft in various military and space industries, Pazooki [5] modeled and simulated this special type of UAV to meet the specific situation and location of arrival at an appropriate time. In addition, autonomous and intelligent systems have made progress in urban traffic management. Wuthishuwong et al. [6] modeled the transportation network using the concept of multi-agents to achieve a balanced and stable traffic volume at each intersection. Object tracking technology can also be used in the live broadcast of sports events [7]. Bai et al. [8] proposed a correlation filter with characteristic heterogeneous adaptation to improve the tracking ability of the system. Liu et al. [9] conducted extensive research on the developed template matching strategies, which improves the tracking performance. For most newly developed filters used for various inspection purposes, it is a useful strategy.

Fig. 1 Comparison of the overall situation of each data set



To easily and comprehensively understand the single target tracking technology and algorithm, this article focuses on the working principle and development of the correlation filter algorithm. In this paper, we provide a comprehensive introduction to existing data sets, and summarize the current correlation filter-based object tracking algorithms to present a comparison of models sourced in the domain of correlation filter tracking. In the following sections of this article, we first introduce four mainstream data sets for evaluation of tracking algorithms. Then, the key technologies of the correlation filter algorithms and the results are summarized. Moreover, we propose a template update strategy during object tracking. Finally, experiments show that this method improves the tracking effect without using complex mathematical operations to expand the model.

Data set introduction

In single object tracking, there are many types of datasets. Among them, the most authoritative and widely used are VOT and OTB datasets. Figure 1 shows the number of sequences and the average sequence length in the form of a histogram to allow a more intuitive comparison of the data sets.

In Fig. 1, the average sequence length of OTB2013, OTB2015 and TC128 are more than 500 frames. While the number of VOT2014 is the lowest. Among them, the OTB and VOT datasets are the most frequently used by people. The chapter is a detailed introduction to each dataset.

VOT dataset

The VOT dataset has been updated every year since 2013, which is composed of high-resolution color sequences. The latest VOT-2019 [10] introduces the challenges of VOT-RGBT and VOT-RGBD. The VOT-RGBT will evaluate the

use of a four-channel (RGB + IR) tracker in the tracking. The VOT-RGBD evaluates a tracker which uses four-channel (RGB + depth) in the tracking. The VOT uses the success rate and robustness evaluation as an evaluation index, where the success rate is the overlap rate between the bounding box which the tracker is tracking and the ground-truth on a single test sequence, and the robustness is the number of times that the tracker fails to track under a single test sequence during tracking.

To better evaluate the performance of the tracker, VOT is divided into five visual attributes: occlusion (OCC), illumination change (IC), motion change (MC), size change (SC), and camera motion (CM). When a frame does not belong to any of these five attributes, it is represented as a non-degraded (ND) attribute. These attributes allow the tracker to be compared on a subset of frames corresponding to the same attribute.

OTB dataset

The OTB2013 data set contains 51 videos. The data set contains a quarter of the grayscale image. 100 video sequences are contained in extended version OTB2015 [11] which is extended from the OTB2013 (50 videos). The ground-truth of the dataset is manually labeled. The OTB evaluation index uses both accuracy and success rate to evaluate tracker's performance. The accuracy refers to the average Euclidean distance between the center point of the bounding box in the ground-truth and the tracking result of the algorithm and the ratio between the number of frames in the threshold range and the total number of image frames in the entire sequence is then calculated. That is the average pixel error (APE). The threshold is typically set to 20 pixels. The success rate is the area overlap ratio of the bounded box of the ground-truth and the tracking result of the algorithm, and is the area under the image curve which overlap rate is greater than the threshold. That is the average overlap rate (AOR). The threshold is generally set to 0.5. The robustness evaluation of OTB indicators include one-pass evaluation (OPE), temporal robustness assessment (TRE), and spatial robustness assessment (SRE).

Given an initial bounding box of the initial frame, the tracker runs to the end of the sequence. TRE evaluates the

tracker on each segment and counts all the information. SRE is the sampling of the initial bounding box in the first frame by moving or scaling the ground-truth. SRE uses four center shifts and four angular shifts and four proportional changes. The amount of transfer is 10% of the target size, and the scale factor varies among 0.8, 0.9, 1.1, and 1.2. Therefore, the SRE evaluates each tracker 12 times. 11 sub-attribute are defined in the OTB, namely: illumination variation (IV), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-plane rotation (OPR), out-of-view (OV), background clutter (BC), and low resolution (LR). Figure 2 below shows some of the sub-attributes.

TColor-128 dataset

Most modern trackers rely purely on the grayscale version of the input image, ignoring the rich color information. TColor-128 [12] shows that color information is very helpful in improving visual tracking, and the improvements it brings are common to different algorithms. TColor-128 is systematically studied through algorithms and reference angles. In terms of algorithms, 16 state-of-the-art visual trackers are selected carefully with 10 color models fully coded. On the reference angles, 128 color sequences with the ground-truth and various challenge factor are annotated. This data set systematically combines various color models with state-of-the-art grayscale trackers and studies their performance. A color tracking benchmark is formed by creating a large color sequence reference using the annotations.

In addition, TColor-128 also performs color tracking evaluation through combinations of different color models and visual trackers. Finally, the success rate and accuracy are used as evaluation criteria. The success rate uses the area under the curve (AUC). The center location error (CLE) is used in the accuracy. The accuracy threshold is also set for 20 pixels. The results of the 160 color-coded tracker and the recently proposed color tracker on TColor-128 show that some color models (such as HSV and LAB) are generally more effective at improving tracking performance. When the target is in deformation or rotation, color information is most helpful.



Fig. 2 Some of the challenging sub-attributes in the OTB. In the left part shows illumination variation and in the right part shows scale variation

Like OTB, TColor-128 also contains 11 sub-attributes. In particular, fast motion in the TColor-128 refers to target motion greater than 20 pixels, and low resolution refers to less than 400 pixels in the ground-truth boundary box. Figure 3 shows a subsequence of the TColor-128 dataset.

UAV123 dataset

The UAV123 [13] dataset is a set of video sequences for drone tracking proposed in 2016. It contains 123 high-resolution aerial video sequences annotated, totaling more than 110K frames. The UAV123 data set contains three subsets. Using off-the-shelf professional-grade drones, 103 sequences on different objects at an altitude of 5–25 m are captured in Set1. The frame rate of video sequence is 30–96 FPS with resolution between 720p and 4K. All sequences are available in 720p and 30 FPS with an upright border of 30 FPS. The annotation is done manually at 10 FPS and then linearly interpolated at 30 FPS.

12 sequences are contained in set 2. Due to limited video transmission, these sequences have lower resolution and contain a reasonable amount of noise bandwidth. These subsets are annotated in the same way. Set 3 contains eight synthetic sequences captured by our proposed UAV simulator. From the perspective of a flying drone, the target moves along a predetermined trajectory in a different world rendered by the

Unreal4 game engine. Annotations are done automatically at 30 fps. The full object masks are also available. UAV123 can evaluate current advanced trackers using multiple metrics. The space tracking errors of tracker can also be evaluated for specific situations by annotating various attributes of the video sequence. The tracker is evaluated by a high-fidelity visual tracking simulator method. The combination of the simulator and extensive spatial benchmark provides a more comprehensive assessment toolbox for modern and advanced trackers and opens up new avenues for experimentation and analysis. Figure 4 below shows a partial subsequence of the UAV123 dataset.

These sequences contain common visual tracking challenges, including aspect ratio change, background clutter, camera motion, fast motion, full occlusion, illumination variation, low resolution, out-of-view, partial occlusion, similar object, scale variation and viewpoints change.

LASOT data set

With the rapid development of object tracking technology, the size and attribute of data sets have gradually increased. The general data set contains complete ground-truth and attribute comments as well as different evaluation criteria. Different fields have corresponding data sets. In addition to these basic data sets, the latest LASOT [14] data set consists



Michaeljackson_CE 0.329 IV, DEF, FM, IPR... Tiger 1 0.323 IV, OPR, OCC, FM...

Fig. 3 TColor-128 partial subsequences and the challenge factors involved. The red sequence is a new sequence, and the blue sequence is the original sequence



Fig. 4 The first frame of selected sequences from UAV123 dataset. The red bounding box indicates the ground-truth annotation

of 1400 sequences of more than 3.5 million frames. The average sequence length is more than 2500 frames. LASOT is the largest tracking benchmark with high-quality annotations by far. It is designed to train deep trackers and evaluate long-term tracking performance. The development of these data sets provides the possibility for object tracking to move forward.

Object tracking

Object tracking is an important component of computer vision. Object tracking can be divided into single object tracking and multi-object tracking. The process of single object tracking is to find a calibration target in the next sequence based on the given first frame information. In single object tracking, there are two non-depth methods: generative method and discriminative method. At present, the five steps of the discriminative method commonly used are motion model, feature extraction, observation model, model update and integration method. Figure 5 shows the general flow of object tracking.

At first, the image to be processed needs to be input and the motion model also needs to be selected. The motion model is used to generate candidate sample information which may contain the target. The speed and quality of sample generation affect the speed and effectiveness of the entire tracking. Motion models typically include particle filtering and sliding windows. Particle filtering uses particle sets to represent probabilities. It can be used for various state space models. This method approximates the probability density function by finding a set of random samples propagating in the state space, and replaces the integral operation with the sample mean. The sliding window generates a series of candidate samples on the calibration target frame through the cyclic matrix method.

Next, feature extraction is performed. Feature extraction is to find features in the candidate region which can uniquely identify the target. In object tracking, the quality

of the feature has the most direct impact on tracking results. Scholars have been working on the effects of different features on tracking and the effects of fusion of different features on tracking. Features generally include manual features and depth features. The manual features are obtained using information such as the shape of the image, geometric attributes and statistical histograms, Haar features, HOG features, LBP features and color features. The depth feature [15] is learned using a large number of training sets. In general, the depth feature has better robustness than manual feature.

The observation model is used to determine whether the current sample is matching tracked object. The generative model and the discriminative model are divided according to the observation model. The generated model is tracked by searching for the object most similar to the target in the current area. This is a template matching process, where sparse representation is the most commonly used. Finally, a discriminant model is obtained through learning as a discriminator, and the discriminator is used to discriminate whether the current sample is the target.

The template update is proposed to prevent the model from drifting due to changes in the appearance of the target. Template update can be considered primarily in terms of strategy and frequency. The update of the template frequency may be continuous for each frame or updated for multiple frames. The continuous update of each frame may introduce unwanted noise, thereby affecting the tracking effect. Interval multi-frame updates may speed up tracking. There are a variety of template strategies, and we can choose to adopt an update strategy which combines different methods.

The integration method is used to select the object. The final tracking result can directly select the highest confidence level, and can also refer to multiple prediction results to form the final result. Feature extraction is critical during tracking process. A sufficiently robust feature can handle most of the tracking challenges. In object tracking, most scholars are committed to the study of features. In addition, how to choose an effective tracking combination process is also a key issue in visual tracking.

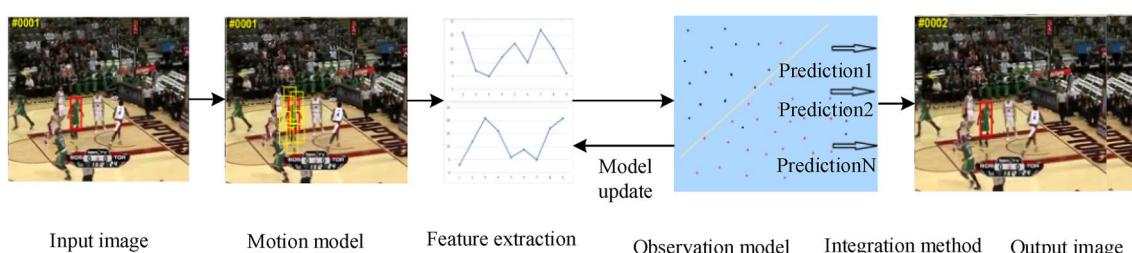


Fig. 5 The flowchart of object tracking. Firstly, the image to be processed is input, and then the motion model is established and the feature extraction operation is performed. Finally, the image is output according to the prediction result

The correlation filter tracking algorithm

In 2010, correlation filter method was used in object tracking for the first time. After nearly a decade of development, the correlation filter tracking algorithms now have matured. In this chapter, we will introduce the development of correlation filter algorithms. The specific development process is as follows.

By learning from gray images, the minimum output sum of squared error (MOSSE) [16] filter applies correlation filter to the tracking field for the first time. This filter is easy to calculate and can quickly track objects, but it does not guarantee to track accurately when the object's appearance changes. After that, Henriques et al. [17] proposed the circulant structure tracking with kernels (CSK) in 2012. Then, Danelljan et al. [18] proposed that the Kernels correlation filter (KCF) further adjusts the channel characteristics to multi-channel features and introduces CN features for tracking in 2014. The CN feature improves the filter's discriminative ability. However, the adaptability of the filter to rotation, out-of-view and fast motion still needs to be improved. Subsequently, Danelljan et al. [19] proposed a discriminative scale space tracker (DSST) using the feature pyramid to solve the multi-scale variation problem and also proposed the improved fDSST algorithm [20]. With the rapid rise of deep learning, the C-COT algorithm [21] effectively represents spatial position information with shallow CNN features, which is a combination of correlation filtering and CNN. The algorithm won the VOT2016 competition. Similar to C-COT, the CSR-DCF algorithm [22] also applies CNN features to the correlation filtering algorithm, which improved the robustness of the algorithms.

MOSSE algorithm

The MOSSE algorithm [16] introduced the correlation filter technology into the visual tracking field. This kind of algorithm can adapt to the problems of occlusion and rotation and achieve an amazing tracking speed of 669 fps. Running 26 times faster than the advanced MIL algorithm, the MOSSE filter is trained by the first frame and can have strong robust performance for illumination, scale and posture variation. When the target is occluded, the algorithm can determine the status of object tracking and update the filter parameters according to the PSR value. When the object reappears, it can be tracked again.

In the MOSSE algorithm, to create a fast tracker, the fast Fourier transform (FFT) is used to calculate the correlation in the Fourier domain. First, calculating the 2D Fourier transform of the input image ($F = F(f)$) and filter ($H = F(h)$). The convolution theorem states that correlation is the element multiplication in the Fourier domain. The symbol \odot

represents element-by-element multiplication, $*$ indicates complex conjugate and the representation of correlation is as follows:

$$g = f \otimes h, \quad (1)$$

where g , f and h represent response output, input image and filter template, respectively. It can be seen that we only need to determine the filter template h to get the response output. The fast Fourier transform (FFT) is used in Eq. (1). Therefore, the convolution operation becomes a point multiplication operation, which greatly reduces the amount of calculation. That is, the above formula becomes:

$$F(g) = F(f \otimes h) = F(f) \cdot F(h)^*. \quad (2)$$

Then, the above formula is abbreviated as follows: $G = F \cdot H^*$ and the next task to track is to find the filter template H^* : $H^* = \frac{G}{F}$.

In the process of actual tracking, we need to consider the influence of factors such as the appearance of the object. At the same time, considering the m images of the object as a reference can significantly improve the robustness of the filter template. The MOSSE model formula is as follows:

$$\min_{H^*} = \sum_{i=1}^m |H^* F_i - G_i|^2, \quad (3)$$

after a series of transformations, a closed solution is obtained:

$$H^* = \frac{\sum_i G_i \odot F_i^*}{\sum_i F_i \odot F_i^*}. \quad (4)$$

The algorithm tracks the object by correlating filters on the search window in the next frame. The new position of the object is represented by the maximum value of the associated output. Then performs an online update in the new location. The tracker update method uses the following formula:

$$H_i^* = \frac{A_i}{B_i}. \quad (5)$$

$$A_i = \eta G_i \odot F_i^* + (1 - \eta) A_{i-1}. \quad (6a)$$

$$B_i = \eta F_i \odot F_i^* + B_{i-1}. \quad (6b)$$

Finally, the PSR value is used for failure detection:

$$\text{PSR} = \frac{\text{peak} - \mu}{\sigma}. \quad (7)$$

In the experiment, the PSR value between 20 and 60 is considered to be a good tracking effect. When the PSR value

is lower than 7, it is judged as tracking failure and the template is not updated.

The MOSSE algorithm overall can adapt to small-scale variation, but it cannot adapt to large-scale variation. In addition, the MOSSE algorithm uses grayscale features that are not powerful enough and expressive in general. The sample sampling of the MOSSE algorithm is still a sparse sampling, and the training effect is general.

CSK algorithm

Unlike the traditional MOSSE algorithm using sparse sampling, the CSK algorithm [17] used a dense sampling method. The use of dense sampling leads to computational burden problems. Thus, the CSK algorithm uses the nature of the cyclic matrix to introduce a Fast Fourier Transform to speed up the algorithm. In addition, the CSK algorithm also introduces kernel techniques on MOSSE to improve the accuracy. A gaussian kernel is used in CSK to calculate the correlation between two adjacent frames. Specifically, the CSK linear classifier solves the correlation filter tracker expression as follows:

$$\min_{w,b} \sum_i^n L(y_i, f(x_i)) + \lambda \|w\|^2, \quad (8)$$

where i is the number of samples after dense sampling, w corresponds to the correlation filter H in MOSSE. The problem is solved by the ridge regression method, where L is the loss function of the least squares method. The calculation method of L is $L(y_i, f(x_i)) = (y_i - f(x_i))^2$, where $f(x_i) = \langle w, x_i \rangle + b$ is the ideal Gaussian response. $f(x_i)$ represents the dot product of the image x_i and the filter w in the frequency domain. $\langle \cdot, \cdot \rangle$ means dot product, the same as \odot . Therefore, $L(y_i, f(x_i))$ is $|H^* \odot F_i - G_i|^2$ in MOSSE. That is, the formula used by CSK is just to add a regular term $\lambda \|w\|^2$ behind MOSSE to prevent overfitting.

In addition, to improve the speed of classifying samples in the high-dimensional feature space, a kernel function is used in CSK. Let $\emptyset(x)$ denotes the feature space, $K(x, x') = \langle \emptyset(x), \emptyset(x') \rangle$ denotes its kernel function, according to the ridge regression $w = \sum_j^n \alpha_j \emptyset(x_j)$. Finally, after a series of solutions, we get α :

$$\alpha = (K - \lambda I)^{-1} y. \quad (9)$$

However, the target size of the algorithm is fixed and the robustness to scale variation is poor. Next, the nature of the circulant matrix is introduced.

KCF algorithm

The KCF [18] is a classic of traditional discriminant method. This series of algorithms learn filters from a series of training samples. Like CSK, the KCF sample generation method uses the cyclic shift method. Assuming one-dimensional data as $x = [x_1, x_2, \dots, x_n]$ the cyclic shift of x is denoted as $P_x = [x_n, x_1, \dots, x_{n-1}]$. All cyclic shift samples form a cyclic matrix are:

$$X = C(x) = \begin{bmatrix} x_1, x_2 & \cdots & x_n \\ \vdots & \ddots & \vdots \\ x_n, x_{n-1} & \cdots & x_1 \end{bmatrix}. \quad (10)$$

That is, it uses $(M \times N)$ image block x to train a filter $f(x) = \langle \omega, \phi_x \rangle$, which generates a training sample by performing a cyclic shift operation on x . The training samples include all cyclic shift forms P_i , where $i \in \{0, \dots, M-1\} \times \{0, \dots, N-1\}$. Each P_i generates a corresponding score y_i ($y_i \in [0, 1]$) which is generated by a Gaussian function based on the shift distance. Minimizing the regression error, the classifier is trained as:

$$w = \operatorname{argmin}_w \sum_i (\langle w, \phi(x) \rangle - y_i)^2 + \lambda \|w\|^2. \quad (11)$$

Among them, $\phi(x)$ is the mapping of Fourier space. $\lambda \geq 0$ is the regularization parameter, which shows the simplicity of the model. The periodic hypothesis achieves effective training and detection by using fast Fourier transform. If the translation invariance of the kernel function is used, α can be quickly obtained as $\hat{\alpha} = \frac{\hat{y}}{\hat{k}^{xx} + \lambda}$ for the special nature of the circulant matrix. In the filtering conversion process, a $m \times n$ candidate image block z for the search space is evaluated by the following formula:

$$f(z) = \mathcal{F}^{-1}(\hat{k}^{xz} \odot \hat{\alpha}), \quad (12)$$

where $f(z)$ is the filter response of all cyclic matrices z , and the highest response is the object of the current frame. The KCF algorithm generates a series of candidate samples by exploiting the properties of the cyclic matrix on the candidate window. It greatly improves the tracking speed compared to traditional window sampling. The problem is then converted to a fast operation in the frequency domain by Fourier transform. This turns the ridge regression problem in the time domain into a cross-correlation problem in the frequency domain. The KCF algorithm uses a multi-channel HOG feature instead of a single-channel grayscale feature. Due to the use of cyclic shift, the KCF algorithm has a boundary effect problem. In addition, the search area is fixed in KCF, so it is easy to exceed the search range in fast motion. Figure 6 is an effect diagram of the cyclic

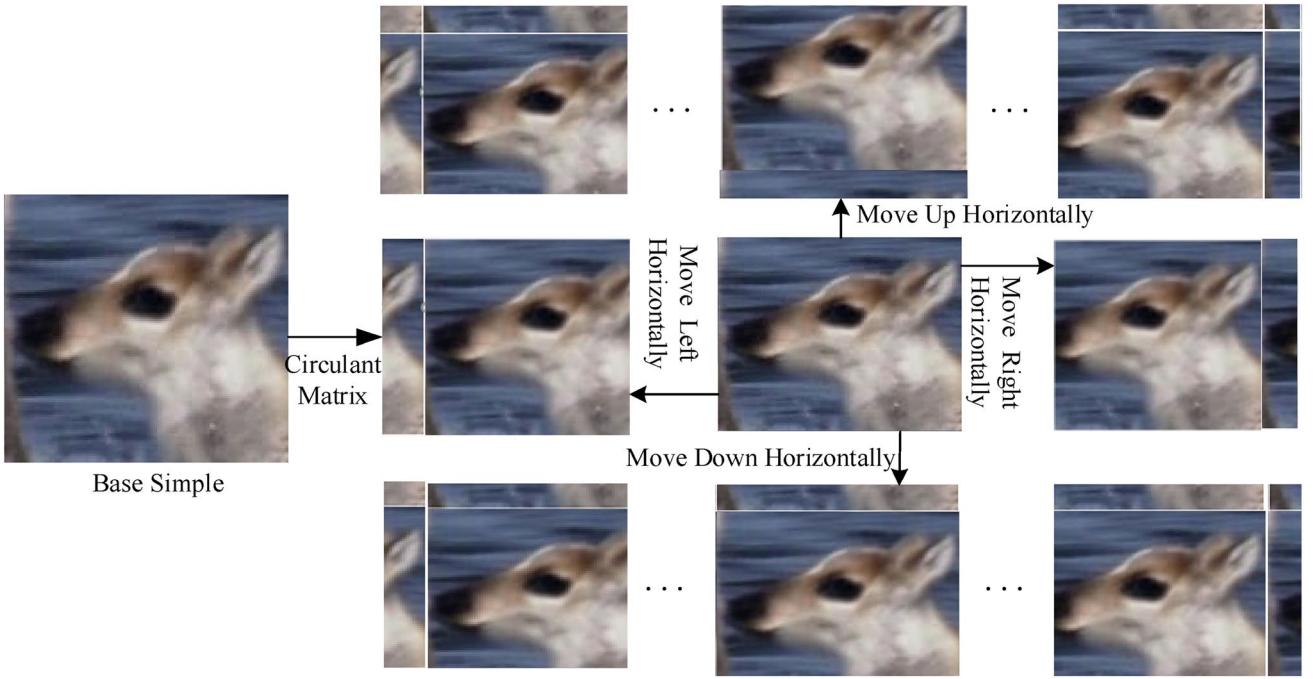


Fig. 6 The principle and effect diagram of cyclic shift. The upper left corner is the effect of the original sample moving to the left and up

matrix. After that, algorithms such as DSST improved the fixed scale problem of KCF.

DSST and fDSST algorithms

Robust scale estimation is a challenging issue in visual tracking. Most existing methods are unable to handle scale variation in complex image sequences. Therefore, the DSST algorithm [19] proposed a scale search and object estimation method based on one-dimensional independent correlation filter. Specifically, in a new frame, a two-dimensional position correlation filter is first used to determine a new candidate position of the target. A one-dimensional scale correlation filter is used to obtain candidate patches of different scales with the current center position as a center point, thereby finding the most matching scale. The scale filter of the DSST algorithm is learned by the scale pyramid representation. This scale estimation method is common to any tracking algorithm without scale variation. The loss function of DSST is as follows:

$$\varepsilon = \left\| \sum_{l=1}^d h^l * f_j - g \right\|^2 + \lambda \sum_{l=1}^d \|h^l\|^2. \quad (13)$$

Its solution in the frequency domain is:

$$H_t = \frac{\sum_{j=1}^t G_j F_j}{\sum_{j=1}^t F_j F_j}. \quad (14)$$

After dissolving, get the solution:

$$H^l = \frac{\bar{G} F^l}{\sum_{k=1}^d \bar{F^K} F^K + \lambda}, \quad l = 1, \dots, d. \quad (15)$$

Furthermore, the principle of scale selection is:

$$a^n p \times a^n R, n \in \left\{ \left[-\frac{S-1}{2} \right], \dots, \left[\frac{S-1}{2} \right] \right\}, \quad (16)$$

where P and R are the width and height of the object in the previous frame, a is the scale factor, and S is the number of scales. The DSST algorithm has a scale factor a of 1.02 and a scale number S of 33. Scale detection is gradually detected from fine to coarse.

Finally, the DSST algorithm uses the compressed training samples $\tilde{F}_t = \ell \{P_t, f_t\}$ and the compression object template $\tilde{U}_t = \ell \{P_t u_t\}$ to update the filter, resulting in:

$$A_t^l = (1 - \eta) A_{t-1}^l + \eta \bar{G}_t F_t^l \quad (17a)$$

$$B_t = (1 - \eta) B_{t-1} + \eta \sum_{k=1}^d \bar{F_t^k} F_t^k. \quad (17b)$$

Here, η is a learning rate parameter. The correlation score y at the rectangular area z of the feature map is calculated using the following formula. Then, it finds the new object state by maximizing the score y . Then, the new object state is found by maximizing the y score.

$$Y_t = \frac{\sum_{l=1}^d \overline{A_{t-1}^l} Z_t^l}{B_{t-1} + \lambda} \quad (18)$$

The correlation score at each location is then calculated by inverting DFT $y_t = \mathcal{F}^{-1}\{Y_t\}$. The estimation of the current target state is obtained by finding the maximum correlation score.

DSST uses 33 scale estimates increasing the computational burden. Therefore, the fDSST algorithm [20] accelerated the DSST. The fDSST algorithm used the techniques of feature dimension reduction and interpolation to greatly accelerate the algorithm. The search box of fDSST has become larger, thereby improving tracking accuracy. PCA features reduce dimensionality in positional filters. Based on computational considerations, the QR-decomposition reduction can reduce the loss of 1000×17 to 17×17 almost non-destructively in scaled filters. For insufficient samples, triangular interpolation is used to supplement to 33. In this way, the acceleration strategy greatly increases the speed of the algorithm, and extra time is spent to expand the search domain to improve the robustness. Unlike DSST, the response value of fDSST is calculated as

$$y = \mathcal{F}^{-1} \left\{ \frac{\sum_{l=1}^d \overline{A^l} Z^l}{B + \lambda} \right\}. \quad (19)$$

To train the filter, the feature mapping f of the patch is extracted. Then, the position of the new frame is estimated by extracting the feature map z at the predicted target position. Finally, the correlation score is calculated and updated. The proposed adaptive scale method also allows the algorithm to adaptively adapt to the scale variation. Figure 7 shows the

principle of the DSST algorithm with scale variation and the tracking effect after the scale variation.

SRDCF algorithm

To overcome the boundary effect appearing in the correlation filtering, the SRDCF algorithm [23] added a regular penalty term. The SRDCF algorithm divides the scale into several scales to overcome the scale variation. When solving the correlation filter, the SRDCF algorithm uses the iterative Gauss–Seidel method to learn online.

In the original DCF, the online training method is as follows:

$$\varepsilon_t(f) = \sum_{k=1}^t \alpha_k \|S_f(x_k) - y_k\|^2 + \lambda \sum_{l=1}^d \|f^l\|^2. \quad (20)$$

While SRDCF adds a regularization term, i.e. penalty term w :

$$\varepsilon(f) = \sum_{k=1}^t \alpha_k \|S_f(x_k) - y_k\|^2 + \sum_{l=1}^d \|w \cdot f^l\|^2, \quad (21)$$

where f is the filter template, l is the l th channel, and w is the regular coefficient matrix. Therefore, the background information is suppressed, and the filter can pay more attention to the object information. After normalization, we get:

$$\bar{\varepsilon}(\hat{f}) = \sum_{k=1}^t \alpha_k \left\| \sum_{l=1}^d \hat{x}_k^l \cdot \hat{f}^l - \hat{y}_k \right\|^2 + \left\| \sum_{l=1}^d \frac{\hat{w}}{MN} * \hat{f}^l \right\|^2. \quad (22)$$

The smoothed response graph is calculated by utilizing FFT and cyclic matrix properties for the above equation. After solving the function, the function is solved and Pascal's theorem is used to transform the objective function into the frequency domain. And the parameters are vectorized. To simplify the solution, we deal with as follows:

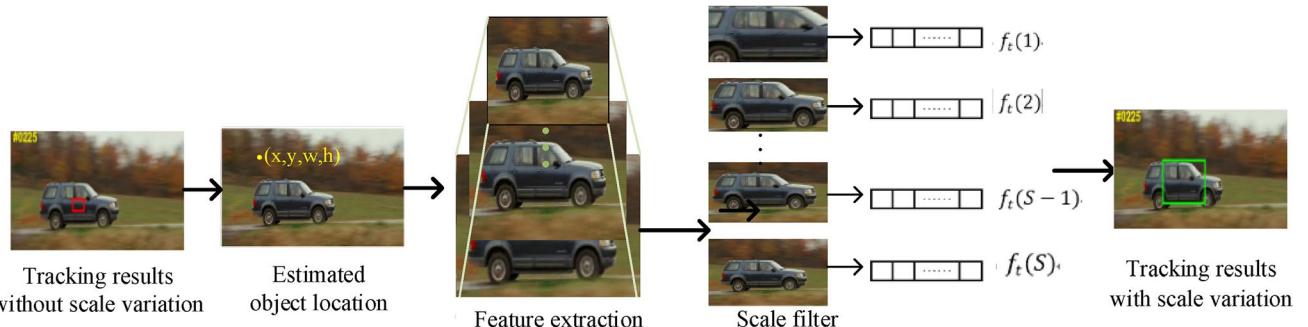


Fig. 7 DSST algorithm tracking schematic with scale variation

$$\bar{\epsilon}(\hat{f}) = \sum_{k=1}^t \alpha_k \left\| \sum_{l=1}^d D(\hat{x}_k^l) \hat{f}^l - \hat{y}_k \right\|^2 + \left\| \sum_{l=1}^d \frac{C(\hat{w})}{MN} \hat{f}^l \right\|^2, \quad (23)$$

where D is the diagonalization operation, C is the cycling operation, and k is the k -th sample. The convolutional symbol can be removed by a loop operation. Ultimately equivalent to solving linear equations:

$$A_t \tilde{f} = \tilde{b}_t. \quad (24)$$

Among them:

$$A_t = \sum_{k=1}^t \alpha_k D_k^T + W^T W. \quad (25a)$$

$$\tilde{b}_t = \sum_{k=1}^t \alpha_k D_k^T \tilde{y}_k. \quad (25b)$$

The Gaussian–Seidel method is used for simplified solving. In the tracking process, according to the ground-truth of the first frame, the training can be performed in an iterative manner:

$$A_t = (1 - \gamma) A_{t-1} + \gamma (D_t^T D_t + W^T W). \quad (26a)$$

$$\tilde{b}_t = (1 - \gamma) \tilde{b}_{t-1} + \gamma D_t^T \tilde{y}_k. \quad (26b)$$

The update method of A and b can reduce the amount of calculation. Scale detection uses the SAMF pyramid method. Down sample speeds up the calculation. Finally, the obtained response is interpolated to get the best scale, and then the Newton iteration method is used to find the maximum response point. After adding the regular coefficient matrix, the response value at the background is obviously suppressed. This makes it possible to expand the search domain for tracking. Figure 8 below shows the contrast effect of the SRDCF algorithm after adding the regular term constraint.

To solve this problem, SRDCF has a model on multiple training images, but this model limits efficiency. STRCF introduces time regularization into single-sample SRDCF, and uses the alternate direction method of the multiplier (ADMM) algorithm to make STRCF each sub-problem has closed solution. In addition, the use of manual features achieves a $5 \times$ acceleration, further solving the boundary effect. The SRDCFDecon algorithm [24] improves the sample and learning rate of the SRDCF algorithm. To overcome the problem of sample drift caused by the correlation filter samples being susceptible to contamination, the SRDCFDecon algorithm chooses to save historical samples. In the optimization objective function, the SRDCFDecon algorithm adds sample weight parameters and regular terms.

STAPLE algorithm

Luca Bertinetto et al. [25] found that the previous algorithm model learning relies on the spatial information of the tracking object, which is not robust to the deformation. However, the use of color features to learn the object can track well in the case of deformation and motion blur. When the light changes, the color features are not well expressed. The HOG features can track the object under the illumination variation. Therefore, the STAPLE algorithm achieves a relatively fast speed using the HOG and color features for fusion at a very good speed 80fps. The tracking effect is better than most existing tracking algorithms. The calculation of the STAPLE algorithm is obtained by linear combination of the template and the color histogram. The function is expressed as follows:

$$f(x) = \gamma_{\text{temp}} f_{\text{temp}}(x) + \gamma_{\text{hist}} f_{\text{hist}}(x). \quad (27)$$

The template score is a linear function of the K -channel feature image $\phi_x : \tau \rightarrow R^K$, obtained from x and defined on the finite grid $\tau \subset Z^2$:

$$f_{\text{temp}}(x; h) = \sum_{u \in \tau} h[u]^T \phi_x[u]. \quad (28)$$

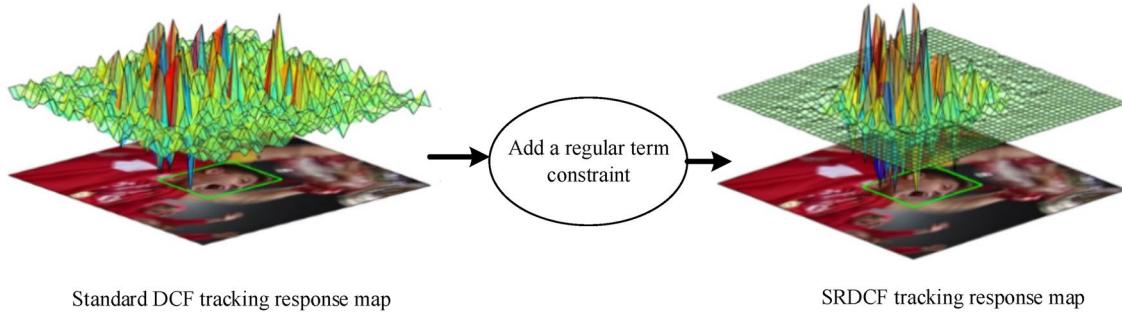


Fig. 8 The effect of the SRDCF algorithm

Among them, the template h is another K -channel image. The histogram score is calculated from the M channel feature image $\psi_x : \mathcal{H} \rightarrow R^M$, and x is obtained and defined in the finite mesh $\mathcal{H} \subset Z^2$:

$$f_{\text{hist}}(x; \beta) = g(\psi_x; \beta). \quad (29)$$

LMCF algorithm

The LMCF algorithm [26] was published on CVPR in 2017. Since the structured SVM has more discriminative power than the traditional SVM, the author combines the structured SVM with the correlation filter algorithm. In the tracking process, when there are similar interference objects around the target, the response graph usually has multiple peaks. The highest peak may be an interference object, which may cause misjudgment. Therefore, it uses multi-peak forward detection to overcome similar object interference.

In addition, LMCF improved the KCF algorithm from the perspective of model update for the first time. According to the current tracking situation, the model update is judged, thereby improving the accuracy of the tracking. In the traditional structured SVM struck algorithm, although the score is directly output through the online SVM, the algorithm is inefficient for using sparse mode. Therefore, LMCF uses a cyclic matrix instead of sparse sampling to increase the speed of structured SVM with CF. In addition, APCE is introduced for multi-peak judgment, the APCE formula is expressed as:

$$\text{APCE} = \frac{|F_{\max} - F_{\min}|^2}{\text{mean}\left(\sum_{w,h} (F_{w,h} - F_{\min})^2\right)}, \quad (30)$$

where F_{\max} , F_{\min} , $F_{w,h}$ represent the response at the highest, lowest and position, respectively. The APCE reflects the degree of oscillation of the response graph. According to the value of the APCE, the judgment of the target motion state can be judged, thereby determining the update of the template. If the APCE suddenly decreases, the target is likely to be occluded or lost. In this case, the model is not updated to avoid model drift. When the APCE and F_{\max} are greater than the historical mean by a certain ratio, the model is updated. This not only reduces the model drift and the number of model updates but also speeds up the operation of the algorithm.

$$h^* = \arg \min_h \left\{ \frac{\lambda}{2} \|h\|_2^2 + \hat{\zeta}^T (\hat{g} - \sqrt{T}(FP^T \otimes I_K)h) + \frac{\mu}{2} \|\hat{g} - \sqrt{T}(FP^T \otimes I_K)h\|_2^2 \right\} = \left(\mu + \frac{\lambda}{\sqrt{T}} \right)^{-1} (\mu g + \zeta), \quad (35)$$

BACF algorithm

For the traditional correlation filtering, the boundary problem caused by the training samples is generated by using cyclic matrix. The BACF algorithm [27] first enlarges the object search area, and then improves the quality of the generated samples. The traditional solution can be expressed as

$$E(h) = \frac{1}{2} \sum_{j=1}^D \left\| y(j) - \sum_{k=1}^K h_k^T X_k [\Delta T_j]_2^2 \right\|^2 + \sum_{l=1}^d \|\omega \cdot f^l\|^2 + \frac{\lambda}{2} \sum_{k=1}^K \|h_k\|_2^2. \quad (31)$$

The BACF algorithm adds the matrix P to the original algorithm instead:

$$E(h) = \frac{1}{2} \sum_{j=1}^D \left\| y(j) - \sum_{k=1}^K h_k^T P X_k [\Delta T_j]_2^2 \right\|^2 + \sum_{l=1}^d \|\omega \cdot f^l\|^2 + \frac{\lambda}{2} \sum_{k=1}^K \|h_k\|_2^2. \quad (32)$$

The addition of the matrix P is the process of secondary processing of the cyclic samples. The original and valid samples are preserved by P , which reduces the influence of the virtual samples on the tracking, then solves the formula:

$$E(h, \hat{g}) = \frac{1}{2} \|\hat{y} - \hat{X}\hat{g}\|_2^2 + \frac{\lambda}{2} \|h\|_2^2 \quad \text{s.t. } \hat{g} = \sqrt{T}(FP \otimes I_K)h \quad (33)$$

After applying fast Fourier transform to the frequency domain, the augmented Lagrange method (ALM) is used, the auxiliary variable g is constructed and g is subjected to the cropping operation. After using the augmented Lagrange method, we get:

$$\begin{aligned} \mathcal{L}(\hat{g}, h, \zeta) = & \frac{1}{2} \|\hat{y} - \hat{X}\hat{g}\|_2^2 + \frac{\lambda}{2} \|h\|_2^2 \\ & + \hat{\zeta}^T (\hat{g} - \sqrt{T}(FP^T \otimes I_K)h) \\ & + \frac{\mu}{2} \|\hat{g} - \sqrt{T}(FP^T \otimes I_K)h\|_2^2. \end{aligned} \quad (34)$$

Then, the ADMM optimization algorithm is used to transform the original problem into two sub-problems that solve the filter h :

$$\text{and the auxiliary variable } g: \quad (35)$$

$$\hat{g}^* = \arg \min_{\hat{g}} \left\{ \frac{1}{2} \|\hat{y} - \hat{X}\hat{g}\|_2^2 + \hat{\zeta}^T (\hat{g} - \sqrt{T}(FP^T \otimes I_K)h) + \frac{\mu}{2} \|\hat{g} - \sqrt{T}(FP^T \otimes I_K)h\|_2^2 \right\} = \left(\mu + \frac{\lambda}{\sqrt{T}} \right)^{-1} (\mu g + \zeta). \quad (36)$$

Then, when solving the sub-problem g , some simplification processing must be done to achieve the real-time performance of the tracking system since the calculation is too large. Finally, the solution problem of g is split into T independent objective functions:

$$\hat{g}(t)^* = \underset{\hat{g}(t)}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\hat{y}(t) - \hat{X}(t)\hat{g}(t)\|_2^2 + \hat{\zeta}(t)^T (\hat{g}(t) - h(t)) + \frac{\mu}{2} \|\hat{g}(t) - h(t)\|_2^2 \right\}. \quad (37)$$

$$\hat{g}(t)^* = \left(\hat{X}(t)\hat{X}(t)^T + T\mu I_K \right)^{-1} \left(\hat{y}(t)\hat{X}(t) - T\hat{\zeta}(t) + T\mu\hat{h}(t) \right). \quad (38)$$

This makes the complexity of \hat{g}^* from $O(K^3T^3)$ down to $O(K^3T)$. Finally, the Sherman–Morrison formula is used to simplify the inversion calculation:

$$\begin{aligned} \hat{g}(t)^* &= \frac{1}{\mu} \left(T\hat{y}(t)\hat{X}(t) - \hat{\zeta}(t) + \mu\hat{h}(t) \right) \\ &\quad - \frac{\hat{X}(t)}{\mu b} \left(T\hat{y}(t)\hat{s}_x(t) - \hat{s}_{\zeta}(t) + \mu\hat{s}_h(t) \right). \end{aligned} \quad (39)$$

Eventually, the complexity is reduced to $O(KT)$, where T is the dimension after the entire image is converted to a vector, and K is the number of layers of the feature. The model update strategy uses the traditional CF linear interpolation method:

$$\hat{X}_{\text{model}}^{(f)} = (1 - \eta)\hat{X}_{\text{model}}^{(f-1)} + \eta\hat{X}^f. \quad (40)$$

Figure 9 below shows the sample training of the DCF algorithm and the BACF algorithm processing operation and effects of the improved quality. Among them, the CS

operation refers to the cyclic shift operation, and the crop is the operation of cutting using the P matrix.

DRT algorithm

Existing CF methods usually focus on the discrimination of filters, while less attention is paid to reliability learning. This may cause the trained filter to be dominated by unexpectedly highlighted areas on the feature map, resulting in model degradation. To solve this problem, Sun et al. [28] proposed a new CF-based optimization problem to jointly simulate identification and reliability information. First, the filter is divided into elemental products of the underlying filter and reliability terminology. The base filter is used to learn the identification information between the target and the background, and the reliability term encourages the final filter to focus on a more reliable area. Second, general terminology for local response consistency is introduced to emphasize equal contributions from different regions and to prevent trackers from being controlled by unreliable regions. The proposed optimization problem can be solved using an alternating direction method and accelerated in the Fourier domain. In the model construction, the DRT algorithm mainly splits the original tracking template w into the point multiplication of the reliability weight map V_d and the original filter h_d , the formula is expressed as

$$W_d = h_d \odot V_d. \quad (41)$$

The reliability weight map is set to have a value in the area of the target frame, and the other area values are zero. The weight map is further divided into weighted sums of nine sub-areas, and each sub-weight graph only focuses on

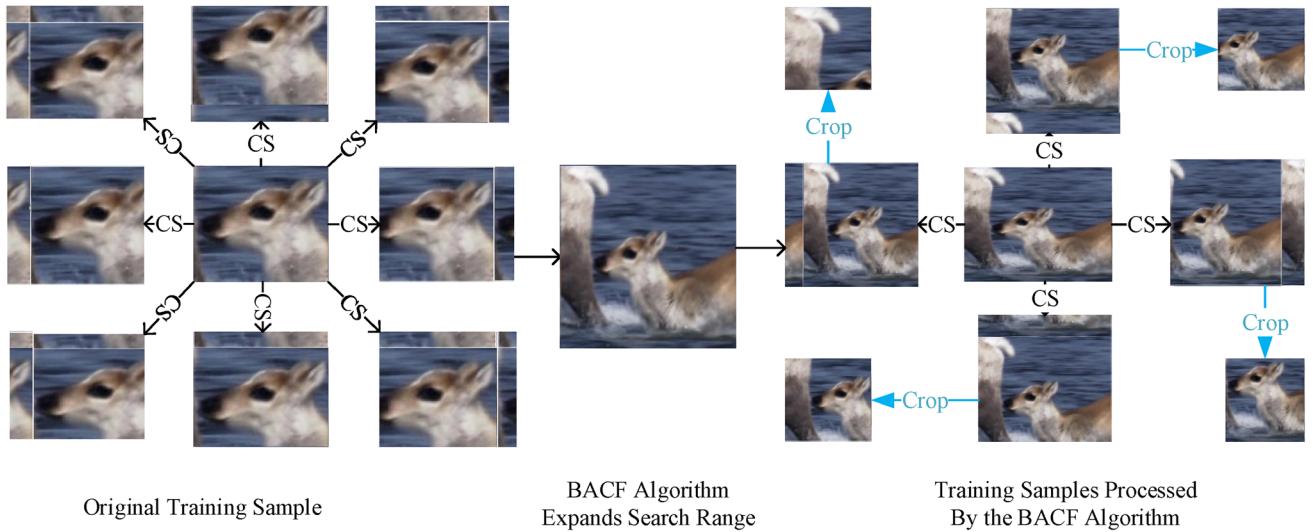


Fig. 9 Sample processing of the BACF algorithm

a part of the target area. The weight of this part of the area is 1, i.e.:

$$V_d = \sum_{m=1}^M \beta_m P_d^m, \quad (42)$$

where $P_d^m \in R^{K \times 1}$ is a binary mask, and the weights have upper and lower limits to ensure the stability of the tracker. Thus, a new tracking template is formed.

Subsequently, the basic filter $h = [h_1^T, \dots, h_D^T]^T$ and the reliability template are optimized:

$$[h, \beta] = \arg \min_{h, \beta} f(h, \beta; X)$$

$$\text{s.t. } \theta_{\min} \leq \beta_m \leq \theta_{\max}, \forall, \quad (43)$$

where the objective function is defined as

$$f(h, \beta; X) = f_1(h, \beta; X) + \eta f_2(h; X) + \gamma \|h\|_2^2. \quad (44)$$

Finally, the h and β are solved by the alternating direction method. The model update is using the conjugate gradient descent method to update h and updating β by solving the quadratic programming problem method. The SAMF method is used in scale detection, and features including a combination of traditional and deep features are also used. The ROI region at different scales centered on the estimated position of the last frame is extracted to obtain the multi-channel feature map X_d^s . Finally, the response of the object position of the scale s is calculated:

$$r_s = \sum_{d=1}^D \mathcal{F}^{-1} \left(\mathcal{F}(w_d) \odot (\mathcal{F}(x_d^s))^H \right). \quad (45)$$

Then, the target position and scale are jointly determined by finding the maximum value in the S response map. This joint estimation strategy shows a better performance. This

method first estimates the target location and then re-scales based on the estimated location. The process comparison between the DRT algorithm and the original algorithm is shown in Fig. 10.

ASRCF algorithm

The SRDCF and BACF algorithms have imposed additional spatial constraints on the filter coefficients, the boundary effects are mitigated to some extent. However, these constraints are usually fixed for different objects and cannot fully utilize the diversity information of the target. Moreover, object localization and scale estimation are usually performed on the same feature space, which requires extracting multi-scale feature maps during the tracking process. When the tracker takes advantage of some powerful and complex features, this strategy can significantly increase the computational load and slow down the tracking. Therefore, Dai et al. [29] proposed a new adaptive spatial regularization correlation filter (ASRCF) model, which can effectively estimate the object's perceived spatial regularization and obtain more reliable filter coefficients in the tracking process. ASRCF is a generic CF model. The ASRCF model is effectively optimized by the ADMM so that each sub-problem has an analytical solution. Finally, the method efficiently estimates the position and scale by two CF models: one uses shallow and deep features for precise position; the other uses shallow features for fast scale estimation. The target function is expressed as:

$$E(H, w) = \frac{1}{2} \left\| y - \sum_{k=1}^K x_k * (P^T h_k) \right\|_2^2 + \frac{\lambda_1}{2} \sum_{k=1}^K \|w \odot h_k\|_2^2 + \frac{\lambda_2}{2} \|w - w^r\|_2^2, \quad (46)$$

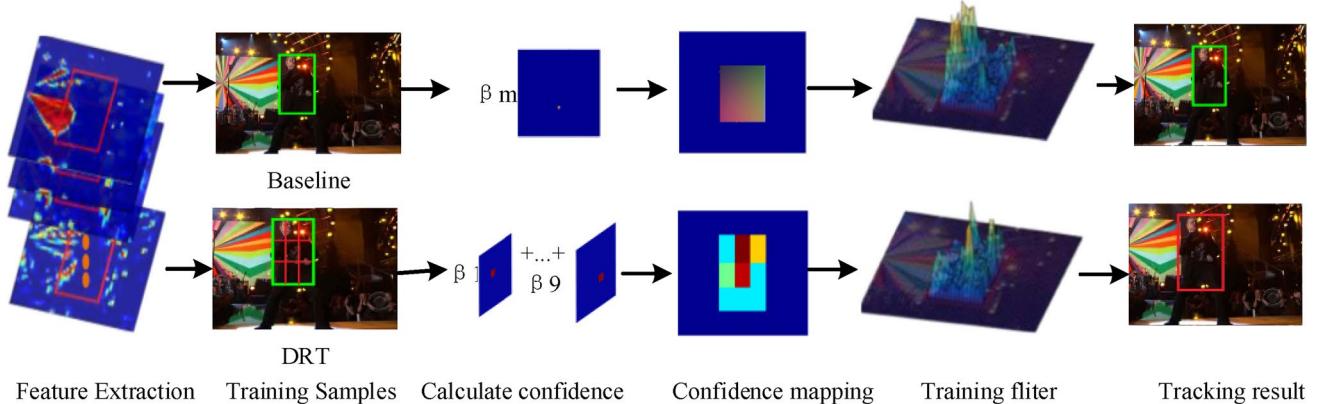


Fig. 10 Comparison of the tracking process and results of the DRT algorithm and the underlying algorithm. The confidence of blue in the confidence map is the lowest and the confidence of red is the highest

where the first term is a ridge regression term, which convolves the training data $X = [X_1, X_2, \dots, X_K]$ with the filter $H = [h_1, h_2, \dots, h_K]$ to the Gaussian distribution of ground-truth y . The second term is a regularization term that introduces adaptive spatial regularization on filter H , where the spatial weight w needs to be optimized. The third term attempts to make the adaptive spatial weight w similar to the reference weight w^r . This constraint introduces a priori information about w and avoids model degradation. λ_1 and λ_2 are the regularization parameters of the second and third terms, respectively. Inspired by SRDCF and BACF, the subsequent solution converts the objective function to the frequency domain and then uses the ADMM optimization algorithm to finally obtain the expression of the augmented Lagrange form:

$$\begin{aligned} L(H, \hat{G}, w, \hat{V}) = & E(H, \hat{G}, w) + \sum_{k=1}^K \hat{V}_k^T (\hat{g}_k - \sqrt{T} F P^T h_k) \\ & + \frac{\mu}{2} \sum_{k=1}^K \left\| \hat{g}_k - \sqrt{T} F P^T h_k \right\|_2^2. \end{aligned} \quad (47)$$

After the solution, the ADMM optimization algorithm is used to transform the original problem into two sub-problems for solving the filter h and the auxiliary variable g . The expression of sub-problems h is:

$$\begin{aligned} h_k^* = & \arg \min_{h_k} \left\{ \frac{\lambda_1}{2} w \odot h_k^2 + \frac{\mu}{2} \hat{g}_k - \sqrt{T} F P^T h_k + \hat{s}_k^2 \right\} \\ = & \frac{\mu T p \odot (s_k + g_k)}{\lambda_1 (w \odot w) + \mu T p}, \end{aligned} \quad (48)$$

and sub-problems g is:

$$\hat{G}^* = \arg \min_{\hat{G}} \left\{ \frac{1}{2} \left\| \hat{y} - \sum_{k=1}^K \hat{X}_k \odot \hat{g}_k \right\|_2^2 + \frac{\mu}{2} \sum_{k=1}^K \left\| \hat{g}_k - \sqrt{T} F P^T h_k + \hat{s}_k \right\|_2^2 \right\}. \quad (49)$$

When H, \hat{G} and \hat{S} are resolved, w can get a closed solution:

$$\begin{aligned} w^* = & \arg \min_w \left\{ \frac{\lambda_1}{2} \sum_{k=1}^K \|N_k w\|_2^2 + \frac{\lambda_2}{2} \|w - w^r\|_2^2 \right\} \\ = & \frac{\lambda_2 w^r}{\lambda_2 \sum_{k=1}^K h_k \odot h_k + \lambda_2 I}. \end{aligned} \quad (50)$$

Finally, the template update method is as follows:

$$\hat{X}_{\text{model}}^{\text{new}} = (1 - \eta) \hat{X}_{\text{model}}^{\text{old}} + \eta \hat{X}^*, \quad (51)$$

where $\hat{X}_{\text{model}}^{\text{old}}$ is the latest updated template, $\hat{X}_{\text{model}}^{\text{old}}$ is the old template, \hat{X}^* is the current observation, and η is the online learning rate.

In the tracking process, the BACF method applies this CF model to the search areas at five scales and obtains their relevant response graphs. Then, the optimal ratio is determined based on the ratio of the maximum score corresponding to the five response graphs. In each frame, the position is first estimated using the positional CF model with complex features, and then the scale is redefined to apply the scale CF model based on the five-dimensional HOG feature map. Figure 11 below shows the overall framework flow of the ASRCF algorithm.

Summary of correlation filtering algorithms

The correlation filter algorithm has the characteristics of high speed and precision, but it faces the challenges of boundary effect and scale effect. Therefore, each algorithm is improved by template updating strategy, feature improvement and area detection. For example, Li et al. [30] adopted an effective phase correlation scheme to simultaneously process scale and rotation changes in log polar coordinates, and achieved robust estimation of similar transformations of large displacements. With deep learning methods prevalent, correlation filtering algorithms still have a place in object tracking, which cannot be ignored. The overall performance of each algorithm is comprehensively analyzed and compared. Next, we show the accuracy and success rate of eight correlation filter algorithms, such as BACF and SRDCF, on OTB2013.

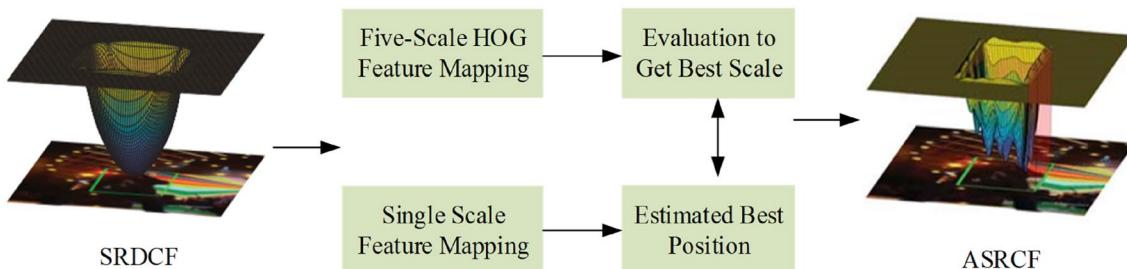


Fig. 11 Comparison of the overall process framework of the ASRCF algorithm

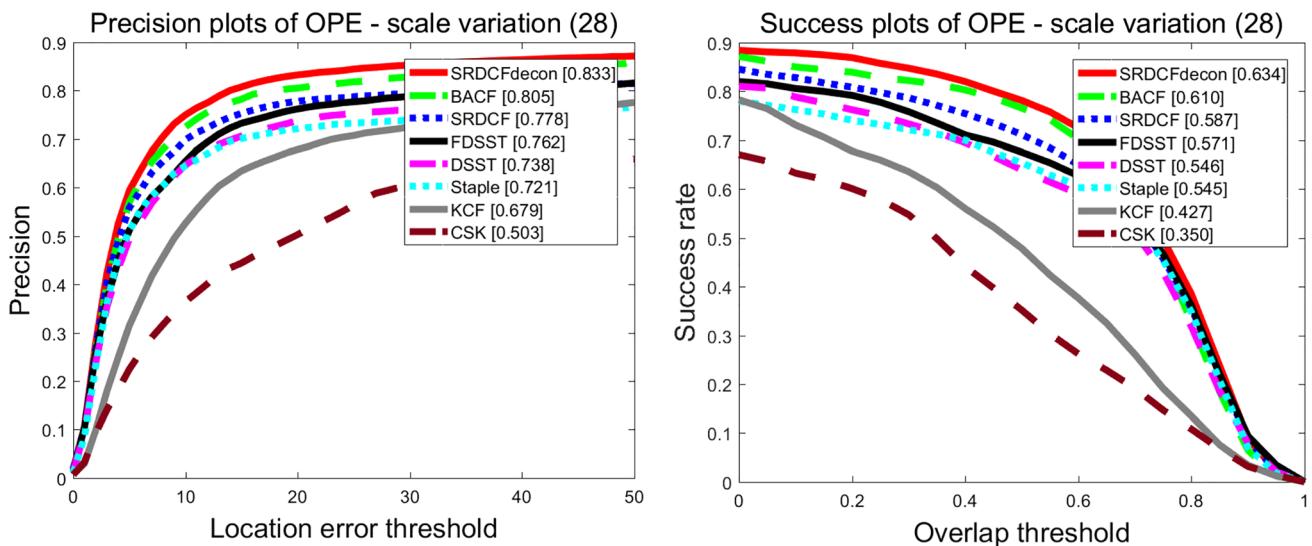


Fig. 12 Accuracy and success rate of the correlation filtering algorithm on OTB2013

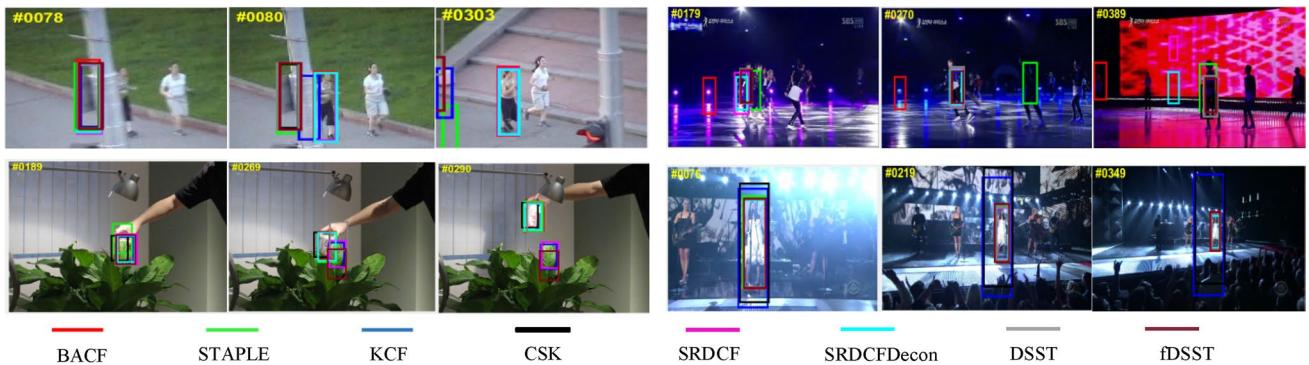


Fig. 13 The tracking effect of the correlation filtering algorithms on the video sequence. The video sets from left to right are Jogging 1, Shating 1, Coke and Singer 1

From Fig. 12 above, we can intuitively see that the overall accuracy and success rate of each algorithm after gradually improving the basic algorithm has gradually improved. In addition, Fig. 13 shows the tracking of these eight algorithms in the OTB2013 dataset.

From Fig. 13, we can see intuitively that the latest BACF and SRDCFDecon are robust in complex scenarios such as occlusion. Furthermore, most algorithms can better adapt to scale transformation 1 besides the original CSK and KCF algorithms. In addition, many scholars propose correlation

filter algorithms using deep features. For example, the C-COT algorithm [21] applies depth features to the correlation filtering, and achieves good tracking results. In the SRDCF algorithm, the STRCF algorithm further overcomes the boundary effect. It combines correlation filtering and deep convolutional neural networks. It is also the first work to unify flow extraction and tracking tasks in a network. This proves that the correlation filter algorithm still has strong vitality, and further research is needed.

Correlation filtering-based template update judgment

In object tracking, a number of samples are generated firstly based on given a priori information. These samples are subjected to feature extraction operations. In this way, the filter can be trained through online training. Finally, the trained filter is used to track the object. In the object tracking process, the template update generally uses a single template update strategy. To effectively deal with various problems and challenges in object tracking, our team proposes solution to improve the tracking effect using a template update strategy. In this chapter, we will detail the three tracking algorithms proposed by our team.

Object tracking strategy with visual attention features structure

Zheng [31] enhanced the predictive ability of the correlation filter algorithm for object position and scale algorithms by combining visual motion feature methods and spatiotemporal continuity. Merged KCF_VAF algorithm reduces the probability of tracking failures such as occlusion, scale variation, and motion blur. This method mainly updates the template according to the motion characteristics. The motion characteristics include the scale change rate, velocity and acceleration. The calculations of these three motion characteristics are introduced below.

(1) Scale change rate: It is used to measure the change of the moving target's own scale between successive frames. The scale change rate is calculated by using formulas (52):

$$S_w = \frac{W_o}{W_p}. \quad (52a)$$

$$S_h = \frac{H_o}{H_p}. \quad (52b)$$

The rate of scale change is measured by the coefficients of variation of the scale S_w and S_h . S_w represents the ratio of the target width W_o to the frame width W_p , and S_h represents the ratio of the target height H_o to the frame height H_p . When the scale of the target changes, its width and height will change, while the width and height of the frame will remain unchanged. Therefore, the ratio of the two can effectively measure the scale change of the target.

(2) Speed: The speed of the moving target can be expressed by the displacement between two adjacent frames. Considering the scale change of the target will also affect the speed, we can calculate the target speed using the formulas (53)–(55):

$$V_x = \frac{P_x^{(i)} - P_x^{(i-1)}}{S_w}, \quad (53a)$$

$$V_y = \frac{P_y^{(i)} - P_y^{(i-1)}}{S_h}, \quad (53b)$$

$$\|\vec{v}\| = \sqrt{V_x^2 + V_y^2}, \quad (54)$$

$$\tan\theta = \frac{V_y}{V_x}, \quad (55)$$

where $P^{(i)}$ and $P^{(i-1)}$ represent the central position of the target at the time i and $i-1$, S_w and S_h are the scale change coefficients, respectively. v_x and v_y are the object speed in the horizontal and vertical directions. $\|\vec{v}\|$ is the magnitude of the speed. The direction of velocity is expressed as $\tan\theta$. θ is the angle between the velocity direction and the horizontal direction. The units of velocity in Eqs. (53)–(55) are pixels per frame.

(3) Acceleration: The acceleration of the moving target is the first-order differential of the speed. The calculation method is similar to the speed. It can be expressed by the speed difference between two adjacent frames. The target acceleration is calculated by the formulas (56)–(58):

$$a_x = V_x^{(i)} - V_x^{(i-1)}, \quad (56a)$$

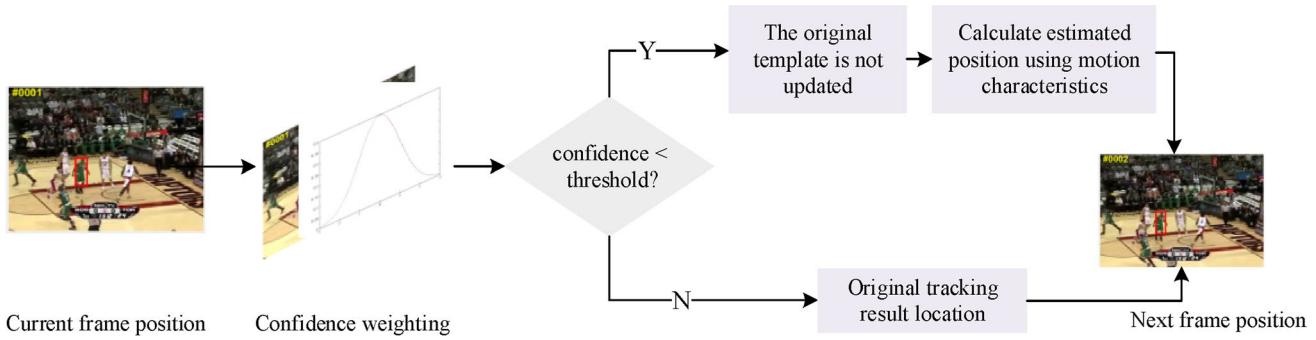
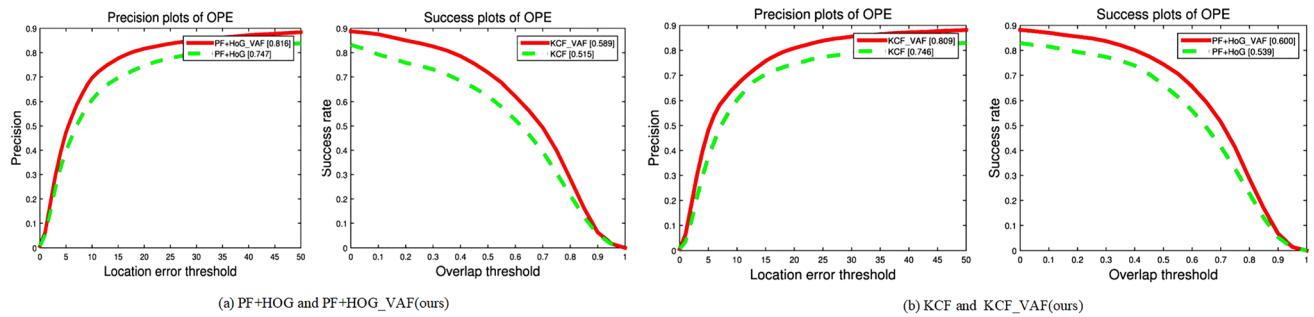
$$a_y = V_y^{(i)} - V_y^{(i-1)}, \quad (56b)$$

$$\|\vec{a}\| = \sqrt{a_x^2 + a_y^2}, \quad (57)$$

$$\tan\alpha = \frac{a_y}{a_x}, \quad (58)$$

where a_x and a_y are the acceleration in the horizontal and vertical directions, $\|\vec{a}\|$ is the magnitude of the acceleration, and the acceleration is represented by $\tan\alpha$, and α is the angle between the acceleration and the horizontal.

This paper first proposes to use the scale change rate to solve the scale problem of KCF. And velocity y and acceleration are used as motion features. In the K th frame, when the candidate region is generated, the scale change rate S_w and S_h , velocity v , and acceleration a are calculated according to the formulas (56)–(58). Based on this, the possible location of the object pos_k is calculated. Finally, the nearby pos_k candidate regions $\{C^1, C^2, \dots, C^N\}$ and the weights $\{w^1, w^2, \dots, w^n\}$ are assigned according to the weighted Gaussian distribution $N(\mu, \sigma^2)$ [32]. Then, the results are weighted according to the weights $\{w^1, w^2, \dots, w^n\}$ to get

**Fig. 14** Fusion visual motion feature tracking process diagram**Fig. 15** The accuracy and success rate comparison results on OTB2013 after the strategy is integrated into the tracking algorithm

the final predicted position \hat{p} . At this time, if the confidence of all candidate regions is lower than the threshold T_C , it is considered that the target may be lost due to occlusion, illumination changes and the like. Subsequently, the position of the target is predicted according to the motion characteristics. The target position \hat{p} is given with reference to the target category and the original template is not updated. The tracking process is shown in Fig. 14.

Figure 15 is the experimental results after integrating the proposed strategy into KCF and PF + HoG algorithms.

Object tracking strategy based on visual memory mechanism

Liu et al. [33, 34] proposed the concept of visual memory mechanism and constructed a model of visual memory mechanism. The proposed concept is suitable for tracking targets in mobile scenes. The object will remain in mind as soon as it appears. When target reappears, the human can quickly and accurately match the target according to the memory [32, 35, 36]. The role of the alternate template is to make the tracking algorithm have the function of reappearing the target like in human vision. The environment where the target is located is determined by the proposed method firstly. When the target is in a complex environment, the exact template information in the $t - 1$ frame is stored

and saved as the standby template R_a . Then, using the optical flow method, a new target position P is calculated as formula (59):

$$P'_x = P_x^{t-1} + \mathcal{V}_y. \quad (59a)$$

$$P'_y = P_y^{t-1} + \mathcal{V}_y. \quad (59b)$$

Among them, v_x and v_y are the speeds of the target in the horizontal direction and the vertical direction respectively after using the optical flow method. Next, v_x and v_y are calculated as formulas (60):

$$\mathcal{V}_x = \mathcal{V}_y = \sum_i^{m+n} \frac{v_x^i}{m+n} \quad (60a)$$

$$\mathcal{V}_y = \sum_i^{m+n} \frac{v_y^i}{m+n} \quad (60b)$$

In current frame, when the confidence value of the selected target is lower than the threshold T_a , the template (context regression model) which has been trained from the previous frame is saved as an alternate template R_a . According to the speed of the object, multi-position detection is performed in the current frame to train another template

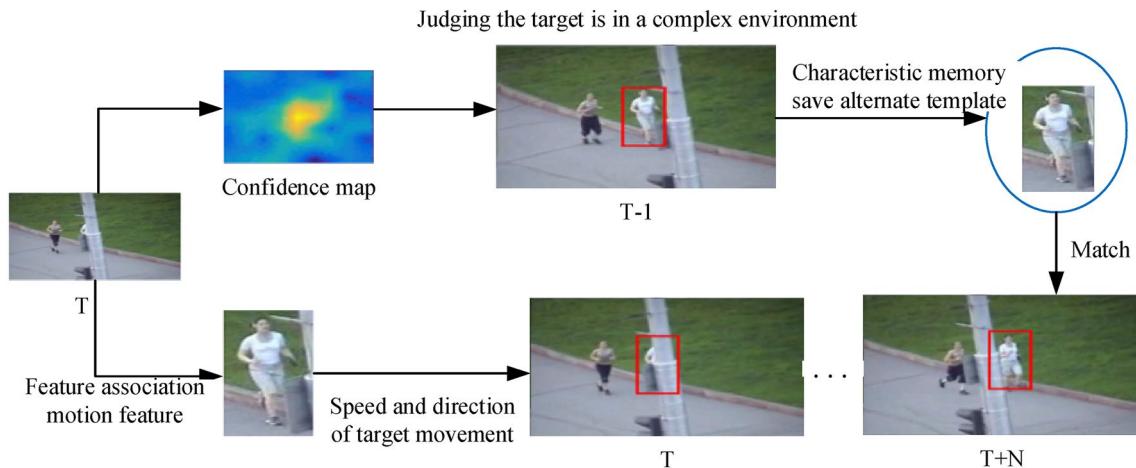


Fig. 16 Principle of the object tracking strategy of the fusion visual memory mechanism

(context regression model) R_c . In the next K frames, the existing trained filters in each frame are used to update the template. Meanwhile, the template R_a is reserved and not updated. After K frames, alternate template R_a is used to detect object and update the template.

$$\hat{x}^t = (1 - \theta)\hat{x}^{t-1} + \theta\hat{x}^t \quad (61)$$

$$\hat{A}^t = (1 - \theta)\hat{A}^{t-1} + \theta\hat{A}^t \quad (62)$$

Hence, the improvement process of the fusion template update method proposed in this paper is shown as follows:

- Initialize the first frame of the video to determine the tracking target. Then, the corresponding feature information is extracted and the classifier is trained.
- Using the classifier, the correlation of the candidate regions is calculated and the responsivity map is obtained.
- The maximum response value F_{\max} is calculated from the response graph.
- The value of F_{\max} is compared with the set threshold. If any values smaller than threshold, the motion features

are used to predict the position of the object. The target information of the previous frame is saved as a backup template through feature storage.

- After several frames, the target returns from the complex scene to the simple scene. When the value of F_{\max} is higher than set threshold, the alternate template R_a is used to match the target and the corresponding model is updated.

Figure 16 below shows the principle of the proposed improved object tracking strategy of fusing visual memory mechanism.

After incorporating this strategy into the FDSST and PF+HoG algorithms, the experimental results are shown in Fig. 17.

Proposed reliability-based visual tracking method

In the process of object tracking, the continuous changes of the object and its surrounding information result in the best matching position of the object (the position of the maximum response value) is not the actual object position.

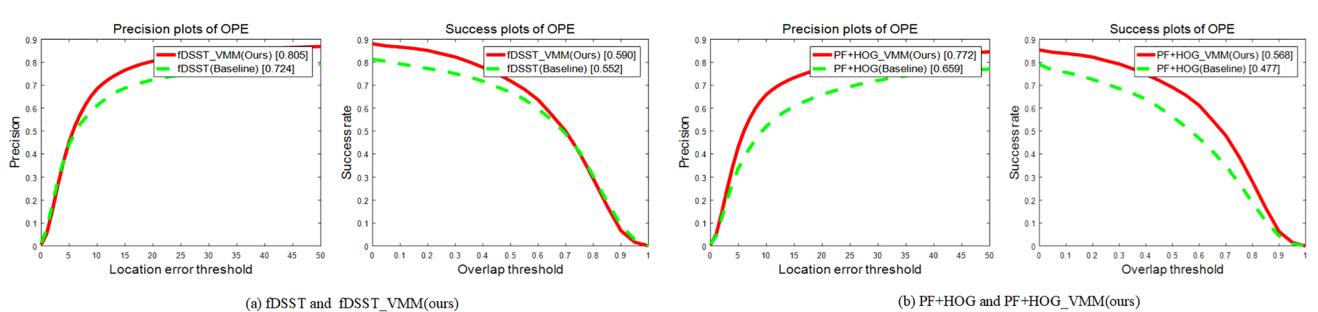


Fig. 17 Comparison of accuracy and success rate on OTB2015 after the strategy is integrated into the tracking algorithm

Therefore, the credibility of the response value is studied. First, the reliability of the information around the target location is calculated. The new strategy is re-selected, and the target location is located based on the predicted reliability. Then, nine features are extracted from the information around the target to obtain the reliability of the target location. These features can be used to digitize the differences and characteristics of information around the target. Because that selecting the best matching position as the target position leads to inaccurate tracking in the current frame, negative samples are chosen in first step. The use of negative samples in the maximum response value is a good indicator of positioning error. Therefore, negative samples are defined by formulas (63).

$$\sqrt{(x_n - gt(x_n))^2 + (y_n - gt(y_n))^2} > \sqrt{(ts(w)/\varepsilon)^2 + (ts(h)/\varepsilon)^2} \quad (63a)$$

$$\sqrt{(x_n - x_{n-1})^2 + (y_n - y_{n-1})^2} > \sqrt{(ts(w)/\gamma)^2 + (ts(h)/\gamma)^2} \quad (63b)$$

In formulas (63), x_n, y_n represent the position of the current frame of the tracking algorithm and $gt(x_n)$ represents the true location of the target. $ts(w), ts(h)$ represent the width and height of the target frame, x_{n-1}, y_{n-1} represents the target position of the previous frame of the tracking algorithm. The Euclidean distance is used here to measure the current position and the real position, as well as the difference in the position of the previous frame. When the formulas (63) are satisfied, such sample is used as a negative sample of this experiment. Then, nine tuple features are extracted from the surrounding information of multiple sequences to achieve reliability-assisted decision-making. Feature tuples are maximum values number, larger response value ratio, negative value ratio, difference, X and Y direction gradient (including two features), maximum response value and distance from the center of the corresponding position (including two functions). These nine features can describe and digitize the distribution of the response matrix from various aspects. The details are as follows:

(1) Peak number: When looking for the same target as the previous frame in a given area, each candidate frame in a given area is compared with the target template determined in the previous frame, thereby obtaining the response values of the respective candidate frames. Therefore, the response values are combined to form a response matrix R .

The centralized matrix rc is obtained by exchanging R matrix, as in formulas (64)–(66).

$$R = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix}, \quad (64)$$

$$rc = \begin{bmatrix} A_4 & A_3 \\ A_2 & A_1 \end{bmatrix}, \quad (65)$$

where the calculation methods of A_1, A_2, A_3 and A_4 are as follows:

$$A_1 = \begin{bmatrix} a_{1,1} & \cdots & a_{1,\lfloor m/2 \rfloor} \\ \vdots & \ddots & \vdots \\ a_{\lfloor n/2 \rfloor,1} & \cdots & a_{\lfloor n/2 \rfloor,\lfloor m/2 \rfloor} \end{bmatrix}. \quad (66a)$$

$$A_2 = \begin{bmatrix} a_{1,\lfloor \frac{m}{2}+1 \rfloor} & \cdots & a_{1,m} \\ \vdots & \ddots & \vdots \\ a_{\lfloor n/2 \rfloor,\lfloor \frac{m}{2}+1 \rfloor} & \cdots & a_{\lfloor n/2 \rfloor,m} \end{bmatrix}. \quad (66b)$$

$$A_3 = \begin{bmatrix} a_{\lfloor \frac{n}{2}+1 \rfloor,1} & \cdots & a_{\lfloor \frac{n}{2}+1 \rfloor,\lfloor m/2 \rfloor} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \cdots & a_{n,\lfloor m/2 \rfloor} \end{bmatrix}. \quad (66c)$$

$$A_4 = \begin{bmatrix} a_{\lfloor \frac{n}{2}+1 \rfloor,\lfloor \frac{m}{2}+1 \rfloor} & \cdots & a_{\lfloor \frac{n}{2}+1 \rfloor,m} \\ \vdots & \ddots & \vdots \\ a_{n,\lfloor \frac{m}{2}+1 \rfloor} & \cdots & a_{n,m} \end{bmatrix}. \quad (66d)$$

After the above centralization processing, the matrix rc has a plurality of extreme points. Since the target is likely to appear in one of many maxima, the maxima need to be processed. To obtain the maximum value matrix, the centered response value matrix rc is subjected to region maximization processing. The formula (67) defines a function to assign a value of 1 to the maximum value in the matrix rc , and the other is set to 0. That is, the 0–1 matrix B is obtained.

$$B = f(rc) = \begin{cases} 1 & rc(i,j) > \text{the adjacent eight response values} \\ 0 & \text{other} \end{cases} \quad (67)$$

The matrix B is the maximum matrix that we want to get. As shown in formula (68), the center matrix rc is multiplied by 0–1 matrix B to obtain the matrix $R1$.

$$R1 = rc \cdot * B \quad (68)$$

$R1$ is the maximum value response matrix in which the corresponding response value is stored at the maximum value and all other positions are zero. To avoid interference of the maximum value to the prediction, the threshold μ_1 is set to exclude these smaller maxima. That is, if the response value at the maximum value is too small, it is ignored. Only those maximum values greater than the threshold are saved. Therefore, it is only necessary to count the number nf of the $R1$ matrix that satisfies the threshold condition.

(2) Large response ratio: Here, a larger response value than the threshold μ_2 is defined. The information around many sequence targets is analyzed. The number of times greater than the threshold is counted and the ratio of the number of larger response values to the number of all samples is also obtained. During the tracking process, the response value of 1 is almost impossible to achieve. Therefore, the response value greater than the threshold μ_2 is set to 1, which is the ratio of the number of assignments to the total number of matrices. Equation (69) is its formulaic expression.

$$Pb = \frac{\text{number}_{a_{ij} > \mu_2}}{n*m} \quad (69)$$

(3) Negative value ratio: the negative ratio is similar to the ratio of larger response value of Eq. (70). The analysis of the response value of each frame leads to the conclusion that the response values in the response matrix are less than one and greater than zero and the distribution numbers of these response values are different. The negative number is more special. Therefore, the number of response values less than zero in each frame is divided by the number of all response values as a feature of the reliability degree judgment in the reliability network.

$$z = \frac{\text{number}_{a_{ij} < 0}}{n*m} \quad (70)$$

(4) Difference value: It is used to measure the difference between the maximum values under certain threshold conditions in formula (71). The difference between each maximum response value is measured using the square sum of the difference between each maximum response value greater than μ_3 and the maximum response value in the response matrix. In this way, the difference value can be used to describe the difference of the response values. The difference value is used as another characteristic information of the reliability network.

$$v = \sum_{i=1}^{nf} (R.\max - R(i))^2 \quad (71)$$

(5) Gradient values in the X and Y directions: As shown in formula (72), the position of the maximum response value is shifted by unit length in the X and Y directions to calculate the gradient. The steepness of the maximum response can be described by the gradient in both directions. The larger the gradient value, the steeper the position where the maximum response value is located. Since the boundary problem is considered, there are two cases in X direction.

$$\frac{\partial \text{response}}{\partial x} = \begin{cases} \frac{a_{xy} - a_{x-1,y}}{\Delta x} & x = n \\ \frac{a_{xy} - a_{x+1,y}}{\Delta x} & x \neq n \end{cases} \quad (72)$$

Here, x, y is used to indicate the position at the maximum response value and the response matrix is not centered. This gradient represents the difference between the position of the maximum response value and the surrounding information, so it is extracted as two of the reliability characteristic information of the reliability calculation. The same is true in the Y direction, for example, formula (73).

$$\frac{\partial \text{response}}{\partial y} = \begin{cases} \frac{a_{xy} - a_{x,y-1}}{\Delta y} & y = m \\ \frac{a_{xy} - a_{x,y+1}}{\Delta y} & y \neq m \end{cases} \quad (73)$$

(6) The maximum response value and the distance to center ratio of its position: The size of the maximum response is very important. The size of the maximum response reflects the similarity with the target of the previous frame. The maximum response value is used as a feature in the response matrix. Therefore, the center ratio of the target frame from the coordinates of x and y are used as the last two features of reliability-assisted decision.

Then, the above nine features are put into the reliability-assisted decision-making model for training and testing. The reliability network model is a single-layer network structure consisting of nine feature inputs, two predictive outputs and three neurons. In the tracking process, the location information of the current frame target matching is extracted through nine features, and then the data is put into the input layer. The weight of the hidden layer saved in the previous training is used to calculate the reliability of the best matching position. According to the reliability of the output layer, it is determined whether the best matching position is reliable in the final. Therefore, the overall execution steps can be expressed as:

- A response matrix describing the target matching information of the kernel correlation filter tracker loop iteration is found in each frame.
- Perform reliability feature extraction on the response matrix to obtain feature data.
- Put the reliability feature data into the model of reliability-assisted decision-making to have a reasonable prediction.
- Give reasonable thresholds based on predictions and multiple experimental procedures.
- According to the difference in reliability obtained, the secondary positioning strategy should be adopted to achieve high-precision prediction.

Applying the reliability-assisted strategy in the tracking algorithm can judge the reliability of the current target. This can achieve more accurate target positioning. The experimental results in the case of fast motion after incorporating this strategy into the KCF algorithm are shown in Fig. 18 below.

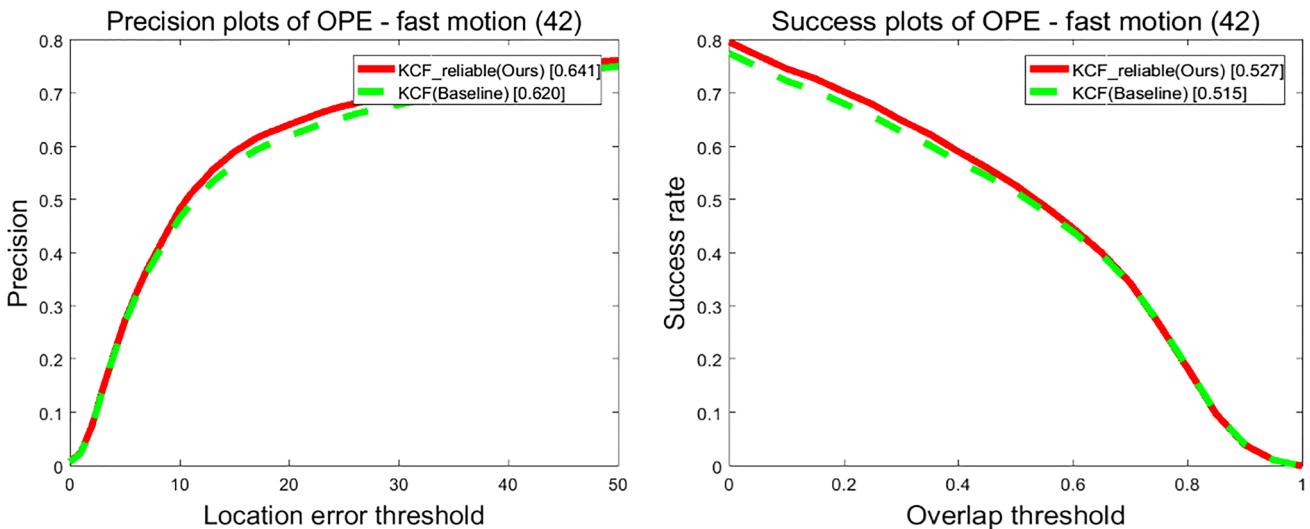


Fig. 18 Quantitative comparison of success rate and accuracy for fast motion in OTB2015

The results show that this method achieves better results than the basic algorithm. This improvement may be a very important advantage for system security in specific areas. Because of proposed gradient approach used in distribution of the response matrix only the maximum position is used in the response value to shift the window in directions of X and Y . Proposed tracking process is also boosted by matching model. We choose negative samples and measure distance between target, which helps to reduce similarity to other objects and improve matching to target object. Proposed method is efficient and results show positive effect in increasing precision and success rates.

Conclusions and future research

This article mainly introduces the correlation filter algorithms and the data sets used to evaluate the algorithms. The contributions of many research scholars and our team in object tracking are also introduced. Our team mainly uses the template update method based on correlation filter algorithm. The tracking accuracy and success rates have effectively improved after using this method, while the running speed decreases slightly. When encountering various problems, scholars' innovative solution can provide new ideas and inspiration for the development of object tracking algorithms.

The biggest advantage of the traditional correlation filter algorithm is that it runs fast and has good running effect, which can realize real-time tracking. Currently, deep

learning algorithms achieve the highest tracking accuracy. However, calculations of deep learning algorithms take much time and run slowly. Therefore, in the next stage, we can focus on three issues:

1. In traditional filter algorithms, we can continue to search for different template update strategies or feature fusion strategies or applying depth features to object tracking. For example, applying partial deep learning results in non-deep traditional algorithms can allow depth features to continue to take advantage of traditional algorithms.
2. In the deep learning algorithm, we can try to combine the new strategy with the classic feature processing method. In this way, future object tracking algorithms can be developed in the direction of real-time speed and high-precision tracking results.
3. We can summarize the object tracking algorithm of the deep learning or the tracking algorithm combined with deep learning and correlation filter. In the future, we can also consider trying to apply some techniques in object target tracking to multiple object tracking

In this way, the object tracking technology can achieve more efficient and faster real-world applications.

Acknowledgements This work was supported in part by the Key Scientific Research Projects of Department of Education of Hunan Province (19A312), Hunan Provincial Science and Technology Project Foundation (2018TP1018, 2018RS3065), National Natural Science Foundation of China under Grant 61502254, Open Project Program of the State Key Lab of CAD&CG under Grant A1926, Zhejiang University.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- KoubâaA QB (2018) DroneTrack: cloud-based real-time object tracking using unmanned aerial vehicles over the internet. *IEEE Access* 6:13810–13824
- Pérez L, Rodríguez Í, Rodríguez N, Usamentiaga R, García D (2016) Robot guidance using machine vision techniques in industrial environments: a comparative review. *Sensors* 16(3):335
- Murugan AS, Devi KS, Sivarajani A, Srinivasan P (2018) A study on various methods used for video summarization and moving object detection for video surveillance applications. *Multimed Tools Appl* 77(18):23273–23290
- Chen TCT (2019) Evaluating the sustainability of a smart technology application to mobile health care: the FGM–ACO–FWA approach. *Complex Intell Syst* 6:109–121
- Pazooki M, Mazinan AH (2018) Hybrid fuzzy-based sliding-mode control approach, optimized by genetic algorithm for quadrotor unmanned aerial vehicles. *Complex Intell Syst* 4(2):79–93
- Wuthishuwong C, Traechtler A (2019) Distributed control system architecture for balancing and stabilizing traffic in the network of multiple autonomous intersections using feedback consensus and route assignment method. *Complex Intell Syst* 6:165–187
- Kim W (2019) Multiple objects tracking in soccer videos using topographic surface analysis. *J Vis Commun Image Represent* 65:102683
- Bai B, ZhongB OG, Wang P, Liu X, Chen Z et al (2018) Kernel correlation filters for visual tracking with adaptive fusion of heterogeneous cues. *Neurocomputing* 286:109–120
- Liu F, Gong C, Huang X, Zhou T, Yang J, Tao D (2018) Robust visual tracking revisited: From correlation filter to template matching. *IEEE Trans Image Process* 27(6):2777–2790
- KristanM et al (2019) The seventh visual object tracking VOT2019 challenge results. In: International conference on computer vision workshops, pp 2206–2241
- Wu Y, Lim J, Yang MH (2015) Object tracking benchmark. *IEEE Trans Pattern Anal Mach Intell* 37(9):1834–1848
- Liang P, Blasch E, Ling H (2015) Encoding color information for visual tracking: algorithms and benchmark. *IEEE Trans Image Process* 24(12):5630–5644
- MuellerM, SmithN, Ghanem B (2016) A benchmark and simulator for UAV tracking. In: European conference on computer vision, pp 445–461
- Fan H, Lin L, Yang F, Chu P, Deng G, Yu S, Ling H (2019) LaSOT: a high-quality benchmark for large-scale single object tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5374–5383
- Kollias D, Tagaris A, Stafylopatis A, Kollias S, Tagaris G (2018) Deep neural architectures for prediction in healthcare. *Complex Intell Syst* 4(2):119–131
- Bolme DS, Beveridge JR, Draper BA, Lui YM (2010) Visual object tracking using adaptive correlation filters. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2544–2550
- Henriques JF, Caseiro R, Martins P, Batista J (2012) Exploiting the circulant structure of tracking-by-detection with kernels. In: European conference on computer vision, pp 702–715
- Henriques JF, Caseiro R, Martins P, Batista J (2015) High-speed tracking with kernelized correlation filters. *IEEE Trans Pattern Anal Mach Intell* 37(3):583–596
- DanelljanM HgerG, Khan FS, Felsberg M (2016) Discriminative scale space tracking. *IEEE Trans Pattern Anal Mach Intell* 39(8):1561–1575
- Danelljan M, Häger G, Khan F, Felsberg M (2014) Accurate scale estimation for robust visual tracking. In: British machine vision conference, pp 1–5
- Danelljan M, Robinson A, Khan F S, Felsberg M (2016) Beyond correlation filters: Learning continuous convolution operators for visual tracking. In: European conference on computer vision, pp 472–488
- Lukezic A, Vojir T, Čehovin Zajc L, Matas J, Kristan M (2017) Discriminative correlation filter with channel and spatial reliability. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6309–6318
- Danelljan M, Hager G, Shahbaz Khan F, Felsberg M (2015) Learning spatially regularized correlation filters for visual tracking. In: Proceedings of the IEEE international conference on computer vision, pp 4310–4318
- Danelljan M, Hager G, Shahbaz Khan F, Felsberg M (2016) 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
- Bertinetto L, Valmadre J, Golodetz S, Miksik O, Torr P H (2016) Staple: complementary learners for real-time tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1401–1409
- Wang M, Liu, Y, Huang Z (2017) Large margin object tracking with circulant feature maps. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4021–4029
- KianiGaloogahi H, Fagg A, Lucey S (2017) Learning background-aware correlation filters for visual tracking. In: Proceedings of the IEEE international conference on computer vision, pp 1135–1143
- Sun C, Wang D, Lu H, Yang MH (2018) Correlation tracking via joint discrimination and reliability learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 489–497
- Dai K, Wang D, Lu H, Sun C, Li J (2019) Visual tracking via adaptive spatially-regularized correlation filters. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4670–4679
- Li Y, Zhu J, Hoi S C, Song W, Wang Z, Liu H (2019) Robust estimation of similarity transformation for visual object tracking. In: Proceedings of the AAAI conference on artificial intelligence, pp 8666–8673
- Pan Z, Liu S, Sangaiah AK, Muhammad K (2018) Visual attention feature (VAF): A novel strategy for visual tracking based on cloud platform in intelligent surveillance systems. *J Parallel Distrib Comput* 120:182–194
- Bryce R, Shaw T, Srivastava G (2018) Mqtt-g: a publish/subscribe protocol with geolocation. In: 2018 41st international conference on telecommunications and signal processing (TSP). IEEE, pp 1–4
- Liu S, Liu G, Zhou H (2019) A robust parallel object tracking method for illumination variations. *Mobile Netw Appl* 24(1):5–17
- Shuai L, Chunli G, Fadi A et al (2020) Reliability of response region: a novel mechanism in visual tracking by edge computing for IIoT environments. *Mech Syst Signal Process* 138:106537

35. Bryce R, Srivastava G (2018) The addition of geolocation to sensor networks. In: ICSOFT 2018, pp 796–802
36. Praznik L, Srivastava G, Mendhe C, Mago V (2019) Vertex-weighted measures for link prediction in hashtag graphs. In: Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining, pp 1034–1041

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.