IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 16, 2020, accepted April 20, 2020, date of publication May 11, 2020, date of current version May 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2993777

Correlation Filter With Motion Detection for Robust Tracking of Shape-Deformed Targets

CHENGYUAN LIU, JIANGLEI GONG, JIANG ZHU, JINXIN ZHANG, AND YUNYI YAN
School of Aerospace Science and Technology, Xidian University, Xi'an 710071 , China

Corresponding author: Yunyi Yan (yyyan@xidian.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61305041 and Grant 61704127.

ABSTRACT Target tracking is an important area of research in computer vision where stable target's tracking has been well solved. But in real world, it is difficult to ensure that the camera or lens could be fixed and the target could maintain its shape in whole video sequence. And as a result, in these unstable cases, robust tracking algorithms have to deal with the problem of target shape-deforming. Once the scenes video sequence contains shape-deformed target, tracking become a real challenging problem. Most previous tracking algorithms based on craft features only used HOG or/and CN features. This paper proposed an algorithm named as Correlation Filtering with Motion Detection (CFMD). This algorithm takes into account the camera shake and target motion information of the video sequence. After removing the effects of lens shake and camera movement, this algorithm can predict the motion information of the target, thereby effectively improving the tracking accuracy and robustness. In CFMD, the target position is determined by the weighted outputs of motion detection and correlation filter tracker. We evaluated our CMFD algorithm on the OTB-100 and VOT-2018 dataset compared with other target tracking algorithms, including Kernel Correlation Filter (KCF), Scale Adaptive with Multiple Features tracker (SAMF), Discriminative Scale Space Tracker (DSST), and Sum of Template and Pixel-wise LEarners (Staple), Learning Spatial-Temporal Regularized Correlation Filters for Visual Tracking(STRCF), Multi-Cue Correlation Filters for Robust Visual Tracking(MCCT). The experimental results showed that our algorithm owns the property of robust tracking of shape-deformed targets in video sequences containing lens shaking or camera moving and it achieves the state-of-the-art precision and tracking effects.

INDEX TERMS Robust target tracking, shape-deformed target, correlation filter, motion detection.

I. INTRODUCTION

Target tracking, which estimates the position of a target object in a video sequence, remains an important area of research in computer vision and is widely used in many fields, such as machine perception, video compression, human–computer interaction, etc. Existing tracking methods are mainly divided into two types. The first is training-based and the other is direct tracking. For training-based tracking, they gather lots of samples to training a model, e.g. Convolutional Neural Network (CNN) or other such things. This kind of solution needs high computing cost even that it often needs graphic process unit (GPU) to implement. But direct tracking is much lighter in view of computing complexity, and it is possible to be implemented in embedded system with relative low power consuming.

The direct tracking contains two strategies, one is the generative model algorithm such as particle filter [1], [2], Mean Shift [3] and Spatiogram method [4]. The algorithm framework is based on the idea of estimation of the target [5]. Under the condition of knowing the target information, the image of the current frame is evaluated to find the most likely target area. They models the target features and try to find the matched one in post-frame image(s) so as to track target in current frame. The other strategy is the discriminant model algorithm. Based on the idea of classification [6], the model framework uses the classifier learning method to distinguish background and target, such as TLD tracker [7], [8], L1APG algorithm [9] and Correlation Filter(CF) tracker [10].

Target tracking performance is often affected by several factors such as camera motion, lens shaking, scale change, illumination variations, partial occlusions, background clutter, and shape deformation. The CF (Correlation Filter) tracker solved these problems to some extent and showed

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

state-of-the-art performance. In the CF-based method, a correlation filter is generated in consecutive frames. Then, filtering is applied to obtain the response matrix in the next frame and the location of maximum value in the response matrix is the location of the target. The state-of-the-art correlation filtering based algorithms' features can be summarized as follows:

The KCF (Kernel Correlation Filter) [11] tracker combines the kernel method and HOG (Histogram of Oriented Gradient) [29] feature in the CF tracker to achieve high-precision tracking. On this basis, the CN (Color Name) [12] tracker improves the tracking effect when the target shape changes by multi-channel color features. The SAMF (Scale Adaptive with Multiple Features tracker) [13] algorithm fuses CN features with HOG features for tracking and calculates the target scale by matching the features of seven scales. Another scale adaptive method is proposed by the DSST (Discriminative Scale Space Tracker) [14] algorithm. In addition to establishing a filter to track the target position, it is found that separate filters for translation and scale estimation significantly improve the performance. Part-based tracker [15] enhances the ability of the algorithm to resist partial occlusion of the target by the Bayesian inference framework and a structural constraint mask. The RAJSSC tracker (Joint Scale-Spatial Correlation Tracking with Adaptive Rotation Estimation) [16] solved the problem of target scale and rotation change by combining scale-spatial correlation tracking with adaptive rotation. The Staple (Sum of Template and Pixel-wise Learners) [17] algorithm combines two image patch representations that are sensitive to complementary factors in order to learn a model that is inherently robust to both color changes and deformations. The response adaptation tracker [18] proposed a generic self-correction mechanism for correlation filter based trackers and solved the problem of large area occlusion. The context-aware method [19] enhances the adaptability of the CF tracker to complex environments by learning the background around the target. Long-term correlation tracker [20] address the problem of long-term visual tracking by using time temporal context information. By introducing time regularization, STRCF (Spatial-Temporal Regularized Correlation Filters) [21] can successfully track targets with small occlusions, and at the same time, it can tolerate large appearance changes. Because the performance of a single tracker is not stable enough, the fusion or combination of multiple trackers can effectively improve the robustness of tracking. MCCT (Multi-Cue Correlation Tracking) [22] proposes a multi-tracker fusion method where the optimal expert is chosen for each frame to determine the tracking result of current frame.

In recent years, tracking algorithms based on CNN features or deep frames have received increasing attention. As the best representative of tracking algorithms using CNN or deep structure, the Siamese trackers formulate the visual object tracking problem as to learn a general similarity map by cross-correlation between the feature representations from the target template and the search [23]. The CFNet tracker [24] and DSiam tracker [25] update the tracking model with the help of a running average template and a fast transformation module, respectively. The SiamRNN tracker [26] introduces the region proposal network(RPN) [26] after the Siamese network and performs joint classification and regression for tracking. The ATOM [27] algorithm uses a structure similar to siamNetwork as a discriminative network of pictures, and uses RPN network to regress the target position to improve tracking accuracy. The analysis of the results of VOT2019 shows that the top tracking algorithms use CNN features, most of which are based on ATOM or siam network structure [28]. However, because CNN feature extraction requires a large amount of computation and a large training data set, it is difficult to perform real-time inference and tracking on embedded devices. Especially in some sensitive or special field scenarios (such as desert, snowfield, grassland and other scenarios without much prior knowledge), it is difficult to obtain a large amount of training data, so it is difficult to deploy quickly. The algorithm discussed in this paper is mainly deployed in embedded terminals, so that it can be tracked in real time under the premise that it has a certain scene applicable ability. Due to the above reasons, our algorithm uses manual features instead of CNN features, so the following discussions and experiments only compare trackers based on manual features and DCF structure.

Existing algorithms can solve the problem of small range scale changes of the target during tracking. However, when the shape change is large due to the target's rapid movement, these algorithms often miss the target. And if the camera is moving fast or the lens are in large shaking, the existing methods' performance are reduced greatly. Figures 1(a–e) illustrate the problem by examples. In Figure 1, the above algorithms cannot accurately track the position of the target when the target moves quickly and its shape deforms largely, such as the flipping of a human body or the running of ants.

We combined moving target detection with correlation filtering to optimize the Staple tracker and proposed a Correlation Filter tracker with Motion Detection (CFMD). In our proposed tracking method, the Staple algorithm is first used to obtain a rough target position. Moreover, we detect the moving target near this location and obtain the position of the moving target. Finally, the coordinates of motion detection are used to average the results of correlation filtering to correct the output.

In our motion detection algorithm, frame differentiating [30], [31] is used to detect moving objects between two frames; i.e.,the greater the difference between the two frames, the greater the probability that the location of the moving object will be. However, lens shaking or camera moving will cause significant noise in frame differentiating, even the target does not move at all in the scene. To overcome this problem, we translate and zoom the current image several times, and match it with the previous frame image to effectively predict lens shaking. Then, the result of frame differentiating is weighted to the average to obtain the position of the

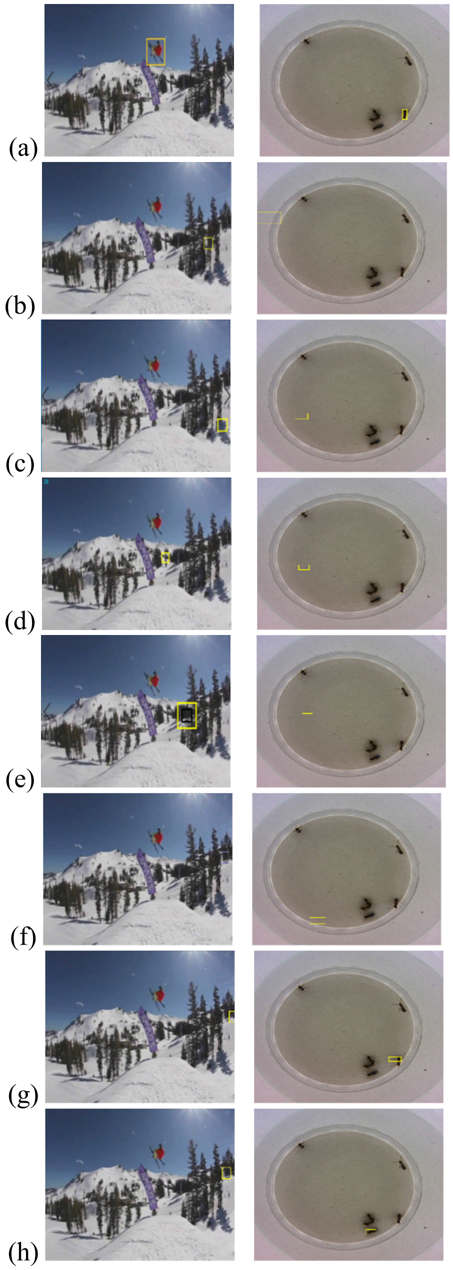


FIGURE 1. **Failures cases of classic algorithms: (a) ground truth; (b) KCF; (c) SAMF; (d) DSST; (e) Staple; (f) BACF; (g) ECO and (h) STRCF.**

moving target. The superiority of our algorithm is illustrated briefly in Figure 1, where all algorithms except for CFMD cannot track target robustly.

In fact, the major contributions of our CFMD algorithm can be listed as follows:

1. Predicting and eliminating lens shaking and camera motion by matching previous and current frames, together with target moving detection.
2. Combining a motion detection algorithm with correlation filter tracking. Motion detection is used to adjust the results of the tracking algorithm and obtain a robust tracking effect.
3. The proposed algorithm can deal with more difficult tracking tasks than existing ones, especially when the target deforms greatly with fast motion, the lens shakes severely or the camera moves rapidly.

II. THE STAPLE TRACKER

Staple [17] is a tracker that combines complementary cues in a ridge regression framework. An independent model is designed based on color statistics and it is combined with the traditional CF method using hog features. This algorithm is insensitive to illumination changes and adaptable to target deformation. Our proposed algorithm absorbs the merits of Staple.

A. CORRELATION FILTER RESPONSE

The CF tracker is designed to learn a discriminative filter that can transform the input feature map into a response matrix in order to infer the position of the target. The location of the highest point in the response matrix is the target location. The response matrix is generated as follows:

$$f_{tmpl}(x;h) = \sum\nolimits_{u} h[u]^{T} \phi_{x}[u], \tag{1}$$

where $f_{tmpl}(x;h)$ is the template correlation filter response, x is the patch in the input image, h is the parameter matrix of the filter, and u is a pixel location in x. After intercepting a patch x based on the target location of the previous frame, x first generates a multichannel feature map $\phi_x[u]$ through FHOG (Fast Histogram of Oriented Gradient) [32] feature extraction. After that the parameter matrix h is used to convolute with the feature map so as to obtain the response matrix f_{tmpl}. At this point, a value in f_{tmpl} is the probability of that point as the target center. In a traditional correlation filter tracker, the location of maximum filter response is the target position. In a Staple tracker, f_{tmpl} is used to combine color feature response to determine the target location.

B. COLOR STATISTICS RESPONSE

The Staple tracker is proposed as a response model based on the color histogram response, which can be obtained through the following formula:

$$\begin{aligned} f_{hist}(x;\beta) &= g(\psi_x;\beta), \\ g(\psi;\beta) &= \beta^{T}\Big(\frac{1}{|H|}\sum\nolimits_{u\in \mathrm{H}} \psi[u]\Big), \end{aligned} \tag{2}$$

where $f_{hist}(x;\beta)$ is the color histogram response, β is the histogram weight vector, and ψ_x is the histogram feature pixels of x. For Formula (2), H represents the image, u is a pixel in the image, and $|H|$ is the vector value. We adopt a linear function of the (vector-valued) average feature pixel. The value in f_{hist} represents the probability of the point as a target location, which is predicted by the color statistical model [17].

C. OVERALL RESPONSE AND PARAMETER LEARNING

Two kinds of response matrix, f_{tmpl} and f_{hist}, are obtained through the above ways. The algorithm in staple integrates the two kinds of response matrices in a weighted average manner. The formula is as follows:

$$f(x) = \gamma_{tmpl} f_{tmpl}(x) + \gamma_{hist} f_{hist}(x), \tag{3}$$

where x is a patch-in input image at current frame, $f(x)$ is the overall response. f_{tmpl} and f_{hist}, respectively, represent the template correlation filter response matrix and color histogram response matrix. γ_{tmpl} is the weight of $f_{tmpl}(x)$ and γ_{hist} is the weight of f_{hist}. In the algorithm, γ_{tmpl} is set to 0.7 and γ_{hist} is set to 0.3. The position of the maximum value in $f(x)$ is the target position of the current frame.

III. MOTION DETECTION AND THE CFMD TRACKER

Staple tracker can work well for target tracking in smooth motion. But in cases that target deforms greatly with fast motion, the lens shakes severely or the camera moves rapidly, it can not achieve desired performance. We proposed an algorithm named Correlation Filter with Motion Detection algorithm (CFMD) to deals with these challenges. CFMD tracker combines one extension of correlation filter, i.e. the Staple tracker, and motion detection strategy together so that the motion detection precisely corrects the output of the Staple to resist large changes in shape. In this section, we first introduce our motion detection strategy and then describes the details of our entire CMFD tracking steps.

A. MOTION DETECTION FOR LENS SHAKING PREDICTION

Ideally, after subtracting the previous and current frames, the location where the pixels change is the location of the moving target. However, in real scenes, video may meet camera vibration and lens zoom, which alters the shooting background fast. Therefore, we have to detect motion or scene's change to predict the lens shake.

In order to detect motion, we take two images, i.e. the previous frame Ip and the current frame Ic, in our algorithm. For better implementation, we take lens shaking parameters $\theta = [\alpha, \beta, \varepsilon]$ as the result of motion detection, which can be determined by finding the best match between two frames:

$$\theta = \arg\min\{\frac{1}{(h-2\alpha)(w-2\beta)} \sum\nolimits_{i=\alpha}^{h-\alpha} \sum\nolimits_{j=\beta}^{w-\beta} |Z(Ic, \varepsilon)_{i,j} - Ip_{i+\alpha,j+\beta}|\}, \tag{4}$$

where h and w are the height and width of the scaled image at any frame, $\theta = \{\alpha, \beta, \varepsilon\}$ are the lens shaking parameters, $\alpha, \beta, \varepsilon$ represent the vertical translation, horizontal translation, and the scaling ration respectively. In fact θ contains displacements and scaling of the current frame image relative to the previous frame, which role as the most important three parameter to describe motions in video sequence. $Z(Ic, \varepsilon)$ means the scaling transformation and spatial translation function of the current image I_c with parameter setting ε. For better computing cost, we advise that the parameters should have $\varepsilon \in [0.8, 1.25]$, $\alpha \in [-20, 20]$, $\beta \in [-20, 20]$. Experiments on different videos show that the value of ε usually ranges from 0.84 to 1.23, and that of α from -13 to 13. β goes from minus 12 to 13. This shows that for most videos, the target scale between frames is small, and the target displacement between frames is generally within 15 units. And of course, these parameters' range can be set case by case. Its real value can be found using evolutionary computing methods by solving optimal problem with Equation (4) as the cost function.

The process of lens shaking prediction is to find the most suitable parameter $\theta = \{\alpha, \beta, \varepsilon\}$, which minimizes the distance between two adjacent frames after translating and scaling the images.

Based on the determined θ, we differentiate the images of two adjacent frames. We can use Equation (5) to calculate the difference map:

$$\mathrm{D} = \left|Z(Ic, \varepsilon)_{i,j} - Ip_{i+\alpha,j+\beta}\right| \tag{5}$$

where D is the difference map after applying the lens shaking parameters. Figure 2 shows the effect of the algorithm.

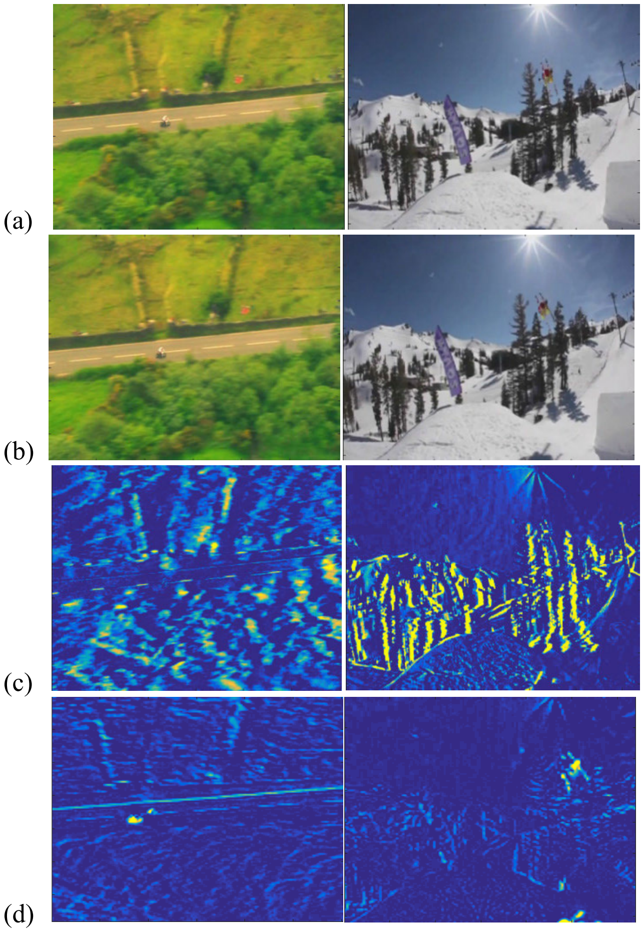


FIGURE 2. Frame differencing: (a) current frame; (b) previous frame; (c) direct difference map; and (d) difference map with lens shaking prediction.

Figure 2(a) is the current frame image and Figure 2(b) is the previous frame image. Figure 2(c) is the difference map of direct difference between Figures 2(a) and 2(b). As we can see, it is difficult to find the target in Figure 2(c) due to the lens movement. However, after the correction of parameter, the difference map between the front and back frames is shown in Figure 2(d). Obviously, the moving object in the video sequence has a larger value in the difference map in Figure 2(d). Finally, we restore the size of the difference graph to the size of the input image.

B. TARGET CENTER'S LOCATING VIA WEIGHTED POSITION

Through the above methods, we obtain the difference map with lens shaking correction. Figure 3(a) shows the original image and location of the target. Figure 3(b) is the difference map corresponding to Figure 3(a) and Figure 3(c) is the display of Figure 3(b) in three-dimensional perspective. As can be seen from Figure 3(c), besides where the target is located there are also higher motion responses in other places. This is because there may be other moving objects in the field of vision that cause interference. To effectively distinguish the target from the jammer, the output of the Staple tracker is used to indicate the approximate location of the target. On the difference map, we intercept a search window twice the size of the target near the position indicated by the Staple tracker and find the moving target in it. As shown in Figure 3(d), the yellow window is the output of the Staple and the red window is the search window for motion detection.

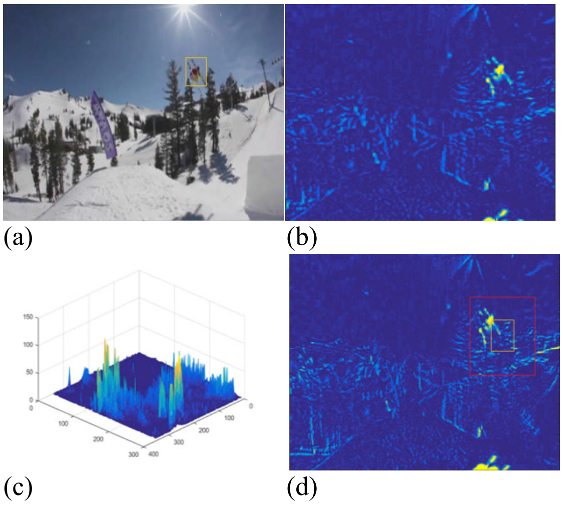

(a) (b) (c) (d)

FIGURE 3. Difference map with lens shaking prediction: (a) the target; (b) difference map; (c) differential map of three-dimensional view; and (d) search window.

In the search window of the difference map, the larger the value, the more likely it is to be the target center. Therefore, we obtain the coordinates of moving objects by statistics with weight. Assuming that the target coordinate of the Staple output is $[x_{cf}, y_{cf}]$ and the target size is $[size_x, size_y]$, the calculation is as follows:

$$\begin{cases} x_{md} = \displaystyle\sum_{i=x_{cf}-size_x}^{x_{cf}+size_x} \sum_{j=y_{cf}-size_y}^{y_{cf}+size_y} \frac{i \times D(i,j)}{D_{sum}} \\ y_{md} = \displaystyle\sum_{i=x_{cf}-size_x}^{x_{cf}+size_x} \sum_{j=y_{cf}-size_y}^{y_{cf}+size_y} \frac{j \times D(i,j)}{D_{sum}} \end{cases} \tag{6}$$

where x_{md} is the coordinate x of motion detection and y_{md} is the coordinate y. D_{ij} is the value at coordinate [i,j] in the differential map. D_{sum} is the sum of the value in the search window and is defined as follows:

$$D_{sum} = \sum_{i=x_{cf}-size_x}^{x_{cf}+size_x} \sum_{j=y_{cf}-size_y}^{y_{cf}+size_y} D(i,j), \tag{7}$$

Finally, we combine the output of Staple $[\boldsymbol{x_{cf}}, \boldsymbol{y_{cf}}]$ with the output of motion detection $[\boldsymbol{x_{md}}, \boldsymbol{y_{md}}]$:

$$\begin{cases} x_{final} = \rho \times x_{cf} + (1-\rho) \times x_{md} \\ y_{final} = \rho \times y_{cf} + (1-\rho) \times y_{md} \end{cases} \tag{8}$$

where ρ is the weight coefficient. Through experiments on multiple video sequences in different data sets, we found that the algorithm maintained good performance in various tracking scenarios when the parameter was about 0.5. $[x_{final}, y_{final}]$ are the final output of our algorithm and represent the coordinates of the target being tracked in the current frame?

C. CFMD ALGORITHM STEPS

In the proposed Correlation Filtering with Motion Detection (CFMD) algorithm, the Staple tracker is first used to indicate a rough target position. Then, we adapt motion detection to predict lens shaking and generate a difference map with lens shaking correction. Afterward, with the help of the Staple algorithm, a window is set up in D and the location of the moving target is obtained by statistics. Finally, the final position of the target is calculated by the weighted average of the Staple algorithm and our motion detection algorithm. Figure 4 shows the algorithm process.

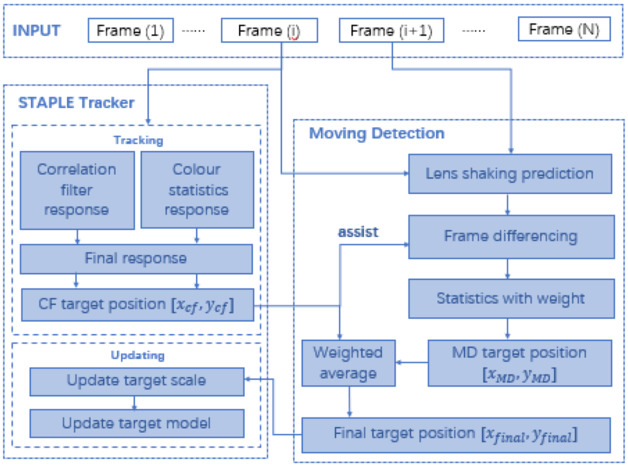


FIGURE 4. The algorithm process.

IV. EXPERIMENT

When the target shape change is large, existing algorithms cannot track the target precisely in whole process. In other words, these algorithms cannot accomplish robust track. In this section we evaluate the proposed CFMD algorithm's performance by real experiments. Video sequences containing large target shape changes or lens shaking were used to test the problem. We compared the proposed method (CFMD) with the state-of-the-art algorithms based on correlation filtering, including KCF [11], SAMF [13], DSST [14], Staple [17], STRCF [21], MCCT [22], BACF [34] and ECO [35]. All trackers are run on the same workstation (Intel Xeon CPU E5-2609 2.5GHz, 64GB RAM) using MATLAB.

A. EFFECT COMPARISON

We selected twelve video sequences (skiing, birds, ants, butterfly, traffic, road, car, BlurOwl, Board, Box, Dancer and Gym) from the OTB-100 [33] and VOT dataset to carry out our experiments. Among them, "skiing", "birds", "ants" and "butterfly" sequences contain target shape large change and/or lens shaking, camera motion. These four sequences are used to evaluate algorithms' performance in shape-deformed target tracking. And the other two, Other videos don't contain shape's changing. These two video sequences are used to test the performance in stable videos.

We propose an indicator named PSD (Probability of Shaking and Deformation) to represent the degree of shaking and deformation of the video sequence.

$$S=\sum \frac{\left|C_x^c-C_x^p\right|}{L_x^p}+\frac{\left|C_y^c-C_y^p\right|}{L_y^p}, \tag{9}$$

$$D=\sum \frac{\left|L_x^c-L_x^p\right|}{L_x^p}+\frac{\left|L_y^c-L_y^p\right|}{L_y^p}, \tag{10}$$

$$PS=\frac{S}{N_{frames}-1}, \tag{11}$$

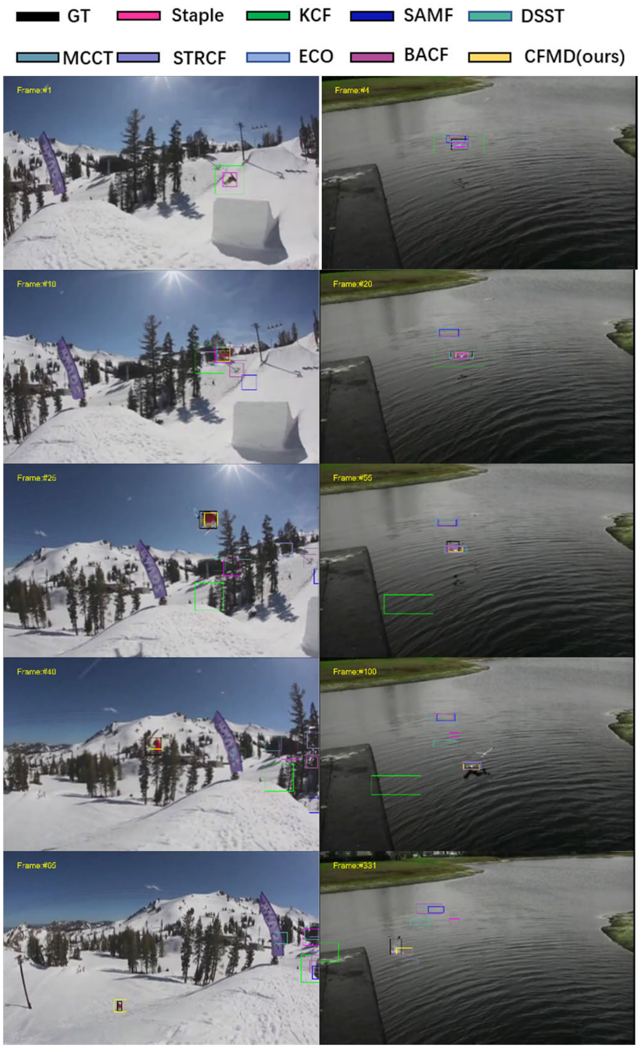
$$PD=\frac{D}{N_{frames}-1}, \tag{12}$$

$$PSD=PS\cdot PD \tag{13}$$

where C_x^c and C_y^c are the coordinates of ground truth (GT) target in the current frame, C_x^p and C_y^p are the coordinates of GT in the previous frame. L_x^c and L_y^c are the length and width of the GT in the current frame, L_x^p and L_y^p are the length and width of the GT in the previous frame. N_{frames} is the number of frames in the video sequence. S is the ratio of displacement of the target center position between adjacent frames to the target size. The displacement is the combined result of target motion and camera shake. The introduction of the target size normalizes the displacement. So S can reflect the shaking and motion of the sequence, PS is its average. D is the rate of change of the target size, it can be use to reflect the deformation of the target. PSD is a comprehensive indicator used to reflect target motion shaking and deformation, which defined as the product of PS and PD. When the shaking and deformation are relatively strong, value of PSD will be large.

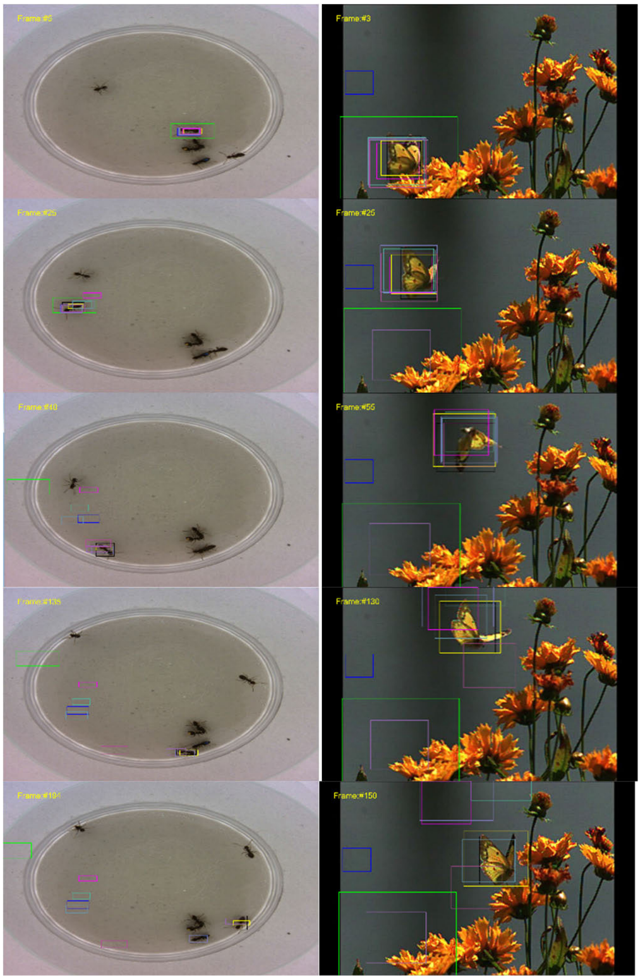
The higher the value of PS is, the greater the motion of the target in the video sequence has. Higher PD value indicates greater deformation of the target. PSD is the poduct of PS and PD, which means that only a video contains large motion and great deformation, its PSD can arrive relative large level. This suggests that PSD can be used to measure inter-frame movement and shaking, and it can work as an indicator to evaluate video sequences' motion and deform As Table 1 shows, the shake and motion of these three videos (Birds, BlurOwl, Ants3) are large. The video of birds has the most distortion. Comprehensive, in the bird video, the target moves and deforms the most. In fact, the bird flying speed is fast, and the incitement of the wings also causes a large deformation. The computing result in Table 1 shows the PSD is in line with human intuitive feelings in most cases. By calculating the PSD values, we choose the twelve videos from open dataset as our experimental videos to valuate our algorithm's performance dealing with fast motion and great deform.

The target tracking effects are shown in Figure 5, 6 and 7. In these figures, black rectangle is for the GT and the yellow one is our CFMD algorithm's results. The green, blue,



(a) Skiing (b) Birds

FIGURE 5. Tracking effect: (a) skiing and (b) birds.



(a) Ants (b) Butterfly

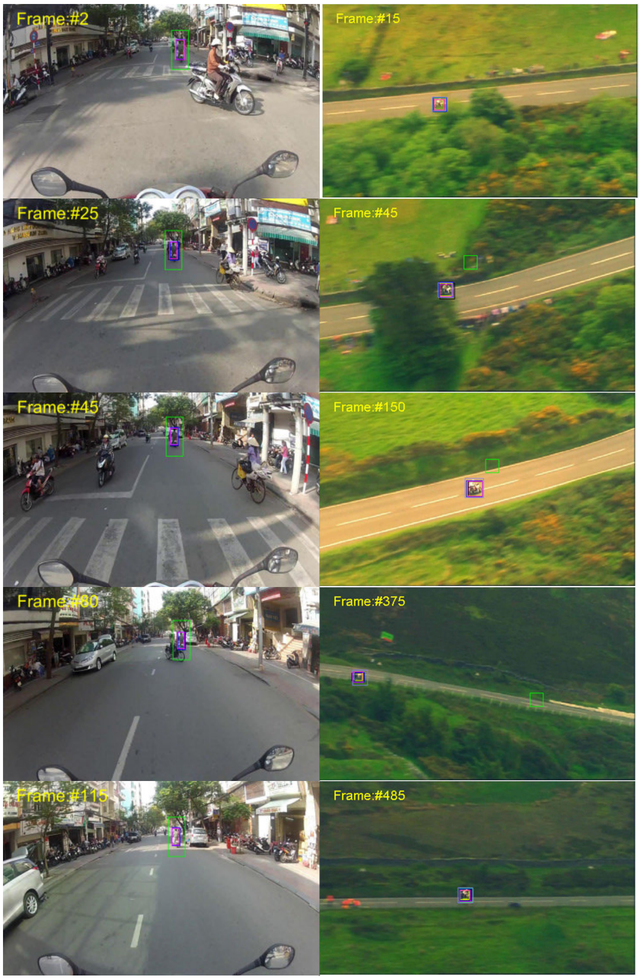
FIGURE 6. Tracking effect: (a) skiing and (b) butterfly.

TABLE 1. PSD for each sequence.

Video name	*PS*	*PD*	*PSD*
Ants3	0.29168	0.21668	0.0632
Birds	0.30209	0.59420	0.1795
Butterfly	0.11818	0.23971	0.0283
Skiing	0.28035	0.05669	0.0158
Road	0.12764	0.17999	0.0242
Traffic	0.08809	0.15470	0.0136
Car	0.23070	0.06481	0.0149
Blurowl	0.38987	0.10298	0.0401
Board	0.03607	0.03908	0.0014
Box	0.06006	0.04871	0.0029
Dancer	0.06531	0.08900	0.0058
Gym	0.06704	0.12558	0.0084

light-blue, and purple rectangles represent other state-of-the-arts correlation filtering algorithm's output.

Figure 5 and Figure 6 show the tracking effect results in shape-deformed tracking. In all of these four sequences, each algorithm except for CMFD lost their targets. As shown,



(a) Traffic (b) Road

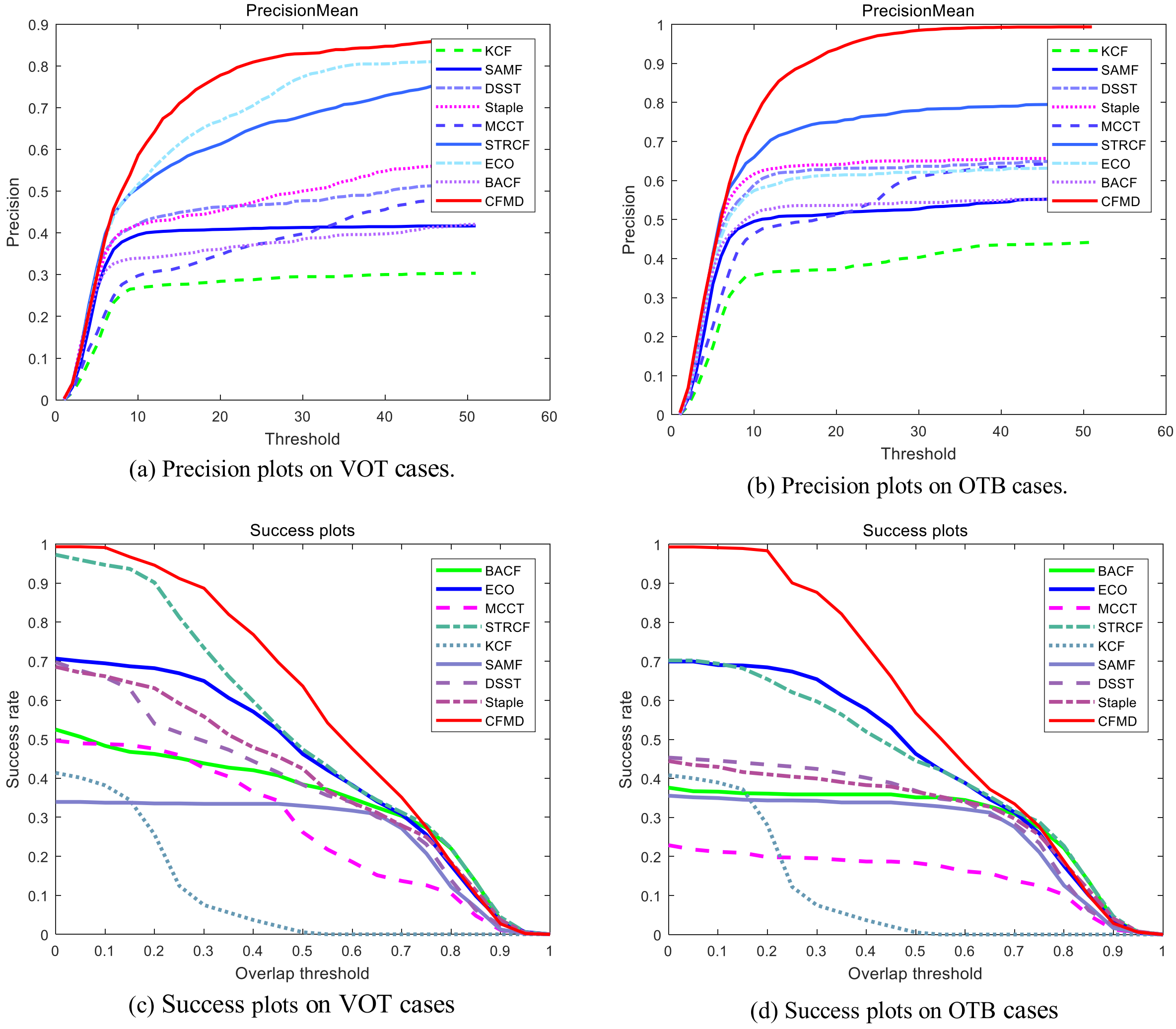
FIGURE 7. Tracking effect: (a) traffic and (b) road.

CFMD is robust and it can track the target without losing it in the entire video. In fact, when correlation filter misses the template or meets mismatch, the motion detection in CFMD will correct the target position so that the match can be well-done and the correlation filtering algorithm updates the right template in time. On the contrary, for the other algorithms Once shape deformed largely and fast, the template of the correlation filter tracking algorithm is hard to match properly and as a result, tracking is often to be seen as failure.

As shown in Figures 7, in the stable videos "traffic" and "road", each algorithm performs well in sequence in case of target shape maintaining well. The KCF meets target missing in some images. And we have to figure out that most videos in OTB-100 dataset are similar with these two ones. In this case, CMFD algorithm achieves comparable effect with other methods.

B. PERFORMANCE EVALUATION

We follow the evaluation protocol as in [13], [14], [17], where the CLE (center location error) is used to judge the accuracy



(a) Precision plots on VOT cases.

(b) Precision plots on OTB cases.

(c) Success plots on VOT cases

(d) Success plots on OTB cases

FIGURE 8. Precision curves and success curves.

of tracking. CLE is the distance between the output position and the ground truth:

$$CLE = \sqrt{(x_{final} - x_{gt})^2 + (y_{final} - y_{gt})^2}. \tag{14}$$

Then, we define a threshold *Th* with a range of 0 to 50. At each threshold, if CLE is less than *Th*, then it is determined to be successful tracking, otherwise, it is judged to be a tracking failure. In this way, all frames in a video are counted. The number of successful tracking is defined as *TP* (true positive), and the number of failures is set to *FP* (false positive). Precision can be defined as in Equation (15):

$$P = \frac{TP}{TP + FP}. \tag{15}$$

For these twelve sequences, we divide them into two groups, i.e. the VOT cases and OTB cases, according to their source dataset. We increase the threshold from 0 to 50 with step size 1. For every threshold, we calculate the tracking precision and Success rate. The Success rate can be defined as in Equation (16):

$$S = \frac{A_t \cap A_{gt}}{A_t \cup A_{gt}}. \tag{16}$$

where A_t is the area of the tracker prediction box, and A_{gt} is the area of GT.

And then, we list the precision curve and success curve for each experiment in Figure 8.

It can be seen from Figures 8(a–d) that CMFD's precision curve and success curve is higher than other algorithms. And, our CFMD algorithm can obtain much better tracking precision in the video sequences containing shape-deformed targets. In fact, these video sequences contain lens shaking, camera motion, or object shape changing, i.e. these four one have deformed shape and it is a real challenge for KCF, SAMF, DSST and Staple.

Table 2 records the results of different algorithms running in 12 data sets. F1 takes into account the accuracy and recall of the model. It can be seen as a harmonic average of model accuracy and recall and reflect the pros and cons of the

TABLE 2. F1-score, precision and success rate.

	F1-score	Precision	Success rate
KCF	0.15	0.23	0.11
SAMF	0.27	0.39	0.24
DSST	0.45	0.49	0.38
STAPLE	0.46	0.50	0.39
MCCT	0.50	0.50	0.42
STRCF	0.64	0.71	0.54
ECO	0.59	0.63	0.50
BACF	0.40	0.39	0.34
CFMD(ours)	**0.65**	**0.72**	**0.56**

tracking algorithm. Its maximum value is 1, and its minimum value is 0. Because most of these twelve video sequences have severe shake or fast moving or rapidly deforming targets. F1-scorce and accuracy of traditional correlation filtering algorithms such as KCF are not high. This does not mean these algorithms' performance in general stable sequences. The Precision reflects the fineness of the tracking algorithm. The higher the value, the better the tracking effect of the algorithm. The success rate reflects the ability of the tracking algorithm to complete the task, and is used to evaluate the performance of the algorithm when a certain error is allowed. The larger the value, the better.

As table 2 shows, in these three indicators, our algorithm is ahead of other algorithms. Mainly because our algorithm performs well in fast-moving sequences, and for those general sequences we keep no lower than the level of other algorithms. The success key factor is the motion detection module.

If the video keeps stable and target shape maintains well in frames, e.g. the "traffic" and the "road" sequences, CFMD's precision performance is proved to be comparable with other state-of-the-art algorithms. An interesting fact is that the KCF's precision is much lower than other. This is due to that in the video sequence "road," the field-of-view is stable but KCF causes target lost in many frames where more details can be found in Figure 7(b). ECO and STRCF also perform well in experiments on other sequences, but they have lost targets in several videos with deformation and target movement. This leads to their result are not good enough.

Experiments show that our algorithm does not work worse than other classic algorithms on stable sequences. In sequences with lens shake and camera movement, our algorithm performs significantly better than other algorithms. Because our algorithm makes effective quantitative estimates of lens shake and camera movement, we can subtract errors from these external variables during the tracing process. Thereby improving the tracking accuracy.

V. CONCLUSION

Correlation filtering and its modifier can work well on stable video but cannot handle with the challenges from lens shaking, camera moving and deformed target shape. The proposed CFMD introduces motion detection to deal with these problems. The CFMD algorithm guides the moving target detection through the output of a correlation filtering algorithm (Staple tracker) and the outputs of the motion detection and the correlation filtering method are weighted by average to obtain reliable tracking results. The algorithm owns robust tracking performance and can locate target in video sequences that contain large changes in the target shape. We selected some targeted videos from the OTB-100 and VOT-2018 datasets. CFMD shows best performance in those videos containing lens shaking, camera moving and shape-deformed target. Even in stable videos, CFMD can also obtain comparable results with other popular algorithms. This means CFMD can suit more challenge in robust target tracking than other correlation filtering methods.

In fact, our algorithm has a motion detection module that solves the lens shake parameters from the video sequence and then combines the images to obtain the target's motion information. This is equivalent to adding a feature of the motion dimension. When the target is obviously moving, we can combine this feature with the traditional HOG and CN features for tracking. Experimental results show that this method has a great effect on lens shaking and fast moving situation. In particular, the motion detection module can be separated from our algorithms and combined with other excellent algorithms as an additional part of motion feature tracking. Adding this module to other algorithms can effectively improve the tracking effect and accuracy.

Because we use the motion detection module to analyze the motion of the target, the position information of the target can be obtained. Once the target is not moving, or the target is blocked by other objects, our algorithm may not be able to guarantee the tracking accuracy. During the experiment, we found that when the target was blocked, the accuracy and success rate of our algorithm decreased. You may need to add a bypass to the tracing framework to deal with occlusion. At present, there is no framework or algorithm with a good performance for tracking the blocked target, which is the direction we will study in the next step.

REFERENCES

[1] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," *Image Vis. Comput.*, vol. 21, no. 1, pp. 99–110, Jan. 2003.

[2] T. Li, S. Sun, T. P. Sattar, and J. M. Corchado, "Fight sample degeneracy and impoverishment in particle filters: A review of intelligent approaches," *Expert Syst. Appl.*, vol. 41, no. 8, pp. 3944–3954, Jun. 2014.

[3] A. Yilmaz, "Object tracking by asymmetric kernel mean shift with automatic scale and orientation selection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–6.

[4] C. O. Conaire, N. E. O'Connor, and A. F. Smeaton, "An improved spatiogram similarity measure for robust object localisation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2007, p. I-1069.

[5] Y. Wu, B. Jiang, and N. Lu, "A descriptor system approach for estimation of incipient faults with application to high-speed railway traction devices," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 10, pp. 2108–2118, Oct. 2019.

[6] F. Luo, L. Zhang, X. Zhou, T. Guo, Y. Cheng, and T. Yin, "Sparse-adaptive hypergraph discriminant analysis for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, pp. 1–5, 2019.

[7] Z. Kalal, K. Mikolajczyk, and J. Matas, "Face-TLD: Tracking-Learning-Detection applied to faces," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 3789–3792.

[8] Z. Kalal, K. Mikolajczyk, and J. Matas, "Forward-backward error: Automatic detection of tracking failures," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 2756–2759.

[9] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust l1 tracker using accelerated proximal gradient approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1830–1837.

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

[11] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[12] M. Danelljan, F. S. Khan, M. Felsberg, and J. V. D. Weijer, "Adaptive color attributes for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1090–1097.

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

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

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

[16] M. Zhang, J. Xing, J. Gao, X. Shi, Q. Wang, and W. Hu, "Joint scale-spatial correlation tracking with adaptive rotation estimation," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 32–40.

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

[18] A. Bibi, M. Mueller, and B. Ghanem, "Target response adaptation for correlation filter tracking," in *Proc. Eur. Conf. Comput. Vis.* Oct. 2016, pp. 419–433. Cham, Switzerland: Springer,

[19] M. Mueller, N. Smith, and B. Ghanem, "Context-aware correlation filter tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1396–1404.

[20] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5388–5396.

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

[22] N. Wang, W. Zhou, Q. Tian, R. Hong, M. Wang, and H. Li, "Multi-cue correlation filters for robust visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4844–4853.

[23] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of siamese visual tracking with very deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4282–4291.

[24] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-End representation learning for correlation filter based tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2805–2813.

[25] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang, "Learning dynamic siamese network for visual object tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1763–1771.

[26] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8971–8980.

[27] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ATOM: Accurate tracking by overlap maximization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4660–4669.

[28] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Oct. 2026, pp. 850–865.

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

[30] V. Markandey, A. Reid, and S. Wang, "Motion estimation for moving target detection," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 32, no. 3, pp. 866–874, Jul. 1996.

[31] S.-C. Huang, "An advanced motion detection algorithm with video quality analysis for video surveillance systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 1, pp. 1–14, Jan. 2011.

[32] D. Forsyth, "Object detection with discriminatively trained part-based models," *Computer*, vol. 47, no. 2, pp. 6–7, Feb. 2014.

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

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

[35] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6638–6646.