Research Article

Scale-Adaptive Context-Aware Correlation Filter with Output Constraints for Visual Target Tracking

Jingxiang Xu, Xuedong Wu , Zhiyu Zhu, Kaiyun Yang, Yanchao Chang, Zhaoping Du, Zhengang Wan, and Lili Gu

School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu 212003, China

Correspondence should be addressed to Xuedong Wu; woolcn@163.com

Received 17 December 2019; Revised 7 May 2020; Accepted 7 May 2020; Published 9 June 2020

Academic Editor: Francesca Vipiana

Copyright © 2020 Jingxiang Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Context-aware correlation filter tracker is one of the most advanced target trackers, and it has significant improvement in tracking accuracy and success rate compared with traditional trackers. However, because the complexity of background in the process of tracking can lead to inaccurate output response of target tracking, an accurate tracking model is difficult to be established. Moreover, the drift problem is easy to occur during the tracking process due to the imprecise tracking model, especially when the target has large area occlusion, fast motion, and deformation. Aiming at the drift problem in the target tracking process, a novel algorithm is proposed in this paper. The developed method derives the specific representation of constraint output by assuming that the output response is Gaussian distribution, and a variable update parameter is obtained based on the output constraint relationship at first, then the tracking filter is selectively updated with changeable update parameters and fixed update parameters, and finally, the target scale is updated with maximizing posterior probability distribution. The effectiveness of developed algorithm is verified by comparing with other trackers on OTB-50 and OTB-100 evaluation benchmark datasets, and the experimental results have shown that the suggested tracker has higher overall object tracking performance than other trackers.

1. Introduction

With the rapid development of computer technology, computer vision plays an increasingly important role in our lives [1, 2]. The scope of computer vision research is quite extensive, including face recognition [3], vehicle or pedestrian detection [4, 5], target tracking [6, 7], and image generation [8–11]. Visual target tracking has become one of the most influential research fields in computer vision because it is widely used in video surveillance [12], intelligent transportation, and military guidance [13, 14]. With the successful application of visual tracking technology in human life, visual object tracking is used to more and more complex environment, such as illumination variation, occlusions, fast motion, and deformation and background clutter, and these complex factors bring great challenges to the stable tracking of targets [15–18].

The target tracking algorithm can be divided into the production tracking algorithm and discriminant tracking

algorithm according to whether the background information of the target is utilized or not [19–21]. The production tracking algorithm uses only target information and discards background information. Although the production algorithm can shorten the computation time, it gives up useful background information which can lead to lower accuracy of tracking the target [22]. The discriminating tracking algorithm takes the tracking problem as a binary classification problem, and the target area is marked with a positive sample and the background region is signed with a negative sample in the current frame [23]. The target area and the background area are distinguished by a classifier, which is obtained by training a large number of samples using a classification method of ridge regression [24].

Because the discriminant tracking algorithm has better tracking performance than the production tracking algorithm, it has been extensively used for target tracking in recent years. The context-aware correlation filtering (CACF) tracker is a discriminant tracking algorithm, and it has better

tracking performance than many other algorithms for dealing with some complex targets [25]. The algorithm uses target information and background information around target area as sample input and takes target position as sample output in the current frame. The target position is determined by using the correlation filter and the target position information of the previous frame. This algorithm can improve tracking accuracy and ensure real-time target tracking. However, the target is prone to drift and causes tracking to fail when the CACF tracker tracks a fast-moving target. Moreover, because the CACF tracker uses a fixed learning rate to update the model, it is easy to cause the model to be updated inaccurately when the tracked target has occlusion or deformation. This inaccurate model update method may cause the target to drift in the tracking process.

Although there are many trackers that have achieved excellent results in dealing with some of these problems, there are still some problems to be solved [26]. Aiming at the drift problem that is easy to occur in the tracking process, this paper proposes the scale adaptive context-aware filter with constraint for output response (OURS) to reduce target drift. The presented method in this paper mainly includes three innovations: (1) a variable parameter is found based on Gaussian distribution and correlation filtering at first, and then the filter is updated using the obtained variable parameter by assuming that the output response is Gaussian distribution. (2) The variable updating parameter with output response under Gaussian constraint and the fixed updating parameter are used to selectively update the filter. (3) The maximum posterior probability distribution is used to update the adaptive scale of the target.

2. Related Work

2.1. Correlation Filter Tracker. The correlation filtering algorithm can greatly reduce computational complexity by converting the time domain calculation into the frequency domain calculation, and it attracts attention of many researchers and applies it to visual target tracking. The principle of the correlation filtering algorithm is to produce a correlation response peak when an interesting object is encountered. The correlation filter has great application in the target tracking field because the tracking speed of traditional algorithms is slow. Bolme et al. [27] applied the correlation filter to target tracking for the first time and proposed the minimum output sum of the squared error tracker (MOSSE), and the tracker is obtained by calculating the least squared error between the actual output and the expected output. Then, many researchers were inspired by the MOSSE tracker and proposed a variety of improved target tracking algorithms. Henriques et al. [28] suggested the loop structure kernel correlation tracker (CSK) to improve the target tracking speed by using intensive sampling to get samples. Yuan et al. [29] proposed a metric learning model for visual tracking in the relevant filtering framework, which used the metric learning function to solve the target size problem. Ou et al. [30] developed a new method for selecting representative samples based on the coefficient-

constrained model, which regarded the template as a linear combination of representative samples.

In order to further improve the target tracking accuracy, Danelljan et al. [31] presented a real-time visual tracker with adaptive color attributes (CN) by embedding multichannel color attributes into kernel space to obtain an adaptive correlation filtering method. In order to deal with the impact of illumination variation on target tracking, Henriques et al. [32] developed the kernel correlation filter with high-speed tracking (KCF) by applying the histogram of oriented gradient (HOG) feature to the correlation filtering algorithm. The KCF tracker performs multichannel expansion of the linear correlation filter by introducing a linear kernel function, so the computational efficiency and the tracking performance were greatly improved.

The traditional correlation filtering algorithm uses a fixed-size window to construct tracker that can estimate the displacement change of tracking target, but it cannot effectively handle the scale change of the target. In response to changes of the target scale, Li et al. [33] proposed a scale-adaptive kernel correlation filtering tracker (SAMF) by fusing color features and HOG features as the new feature of the tracker and determined the optimal target position by comparing the maximum output amplitude with different scales. Taking the advantages of color features and HOG features, Bertinetto et al. [34] suggested a real-time tracking color supplement learning tracker that uses features consisted of color features and HOG features.

The successful application of correlation filter in target tracking can greatly improve the computational efficiency of the tracker, but there are still some unresolved problems, such as the drift problem is caused by fast motion, occlusion, and deformation.

2.2. Context-Aware Trackers. Context information can provide very significant ancillary information for tracking the target. The context-aware tracker not only selects the target area as sample but also selects background information around the target area as context area [35]. The whole sample space of the context-aware algorithm includes target area sample and context area sample. The specific situation is shown in Figure 1.

The context information may provide important supplementary information to target detection and tracking, and it can better identify the target's state. A context-aware visual tracking tracker (CAT) that found auxiliary target by online learning and provided context information for the target was proposed [36]. The CAT tracker can reduce uncertainty in the target tracking process and improve tracking performance. In order to effectively suppress the interference influence, a context-aware sparse tracker (CEST) that used context particle information to create dynamically updating dictionary template was suggested [37].

Although the CAT has achieved some successful results in target tracking, it is still unable to deal with the drift problem caused by fast motion, occlusion, and deformation. In order to solve the drift problem of tracking process, this

FIGURE 1: Illumination of context-aware area.

paper proposes a scale adaptive context-aware correlation filter with constraints for output response. The experimental results have shown that the proposed algorithm can reduce the drift of target tracking and improve the target tracking performance.

3. Scale-Adaptive CACF with Output Constraints

This section describes the principle of the context-aware correlation filtering (CACF) tracker, analyzes filter updating when the output response is Gaussian distribution, introduces the scale adaptive updating method, and provides the specific implementation steps of the suggested method.

3.1. Context-Aware Correlation Filter. The classifier of the traditional correlation filter tracking algorithm is trained by means of ridge regression. Its objective function is

$$\min_{\omega} \|\mathbf{A}_0 \omega - \mathbf{y}\|_2^2 + \lambda_1 \|\omega\|_2^2, \quad (1)$$

where $\mathbf{A}_0$ is a sample set which is obtained by cyclic sampling the target area, λ_1 is the regular term coefficient, $\mathbf{y}$ is the expected output, and ω is the correlation filter.

The symbol $\mathbf{A}_j (j \geq 0)$ is a cyclic matrix, and it is constructed with a special matrix $\mathbf{F}$. The $\mathbf{A}_j$ and its transposed matrix $\mathbf{A}_j^T$ can be obtained as follows

$$\begin{cases} \mathbf{A}_j = \mathbf{F} \text{diag}(\hat{a}_j) \mathbf{F}^H, \\ \mathbf{A}_j^T = \mathbf{F} \text{diag}(\hat{a}_j^*) \mathbf{F}^H, \end{cases} \quad (2)$$

where $a_j (j \geq 0)$ is the target sample, $\hat{a}_j$ is its Fourier transform, and $\hat{a}_j^*$ is its complex conjugate.

The correlation filter ω can be solved by equation (1) and equation (2) as

$$\omega = (\mathbf{A}_0^T \mathbf{A}_0 + \lambda_1 \mathbf{I})^{-1} \mathbf{A}_0^T \mathbf{y}. \quad (3)$$

According to equation (2) ($j = 0$) and equation (3), the Fourier form of correlation filter ω is

$$\hat{\omega} = \frac{\hat{\mathbf{a}}_0^* \odot \hat{\mathbf{y}}}{\hat{\mathbf{a}}_0^* \odot \hat{\mathbf{a}}_0 + \lambda_1}. \quad (4)$$

In order to estimate the target position, the response and the confidence map of sampling data need to be found. Let the input is $\mathbf{z}$, and the output is $f(\mathbf{z})$:

$$f(\mathbf{z}) = \mathbf{z} \cdot \omega. \quad (5)$$

Then, the Fourier transform of $f(\mathbf{z})$ is

$$\hat{\mathbf{y}} = \hat{\mathbf{f}}(\mathbf{z}) = \hat{\mathbf{z}} \odot \hat{\omega}, \quad (6)$$

where $\hat{\mathbf{y}}$ is the Fourier form of output response, $\hat{\mathbf{z}}$ is the Fourier form of input $\mathbf{z}$, and $\odot$ is the element-wise product. In order to further simplify the calculation, we set $\omega = \mathbf{A}_0^T \alpha$, and the simplified result of equation (3) is

$$\alpha = (\mathbf{A}_0^T \mathbf{A}_0 + \lambda_1 \mathbf{I})^{-1} \mathbf{y}. \quad (7)$$

According to equation (2) ($j = 0$) and equation (7), the Fourier form of correlation filter α is

$$\hat{\alpha} = \frac{\hat{\mathbf{y}}}{\hat{\mathbf{a}}_0^* \odot \hat{\mathbf{a}}_0 + \lambda_1}. \quad (8)$$

Because $\omega = \mathbf{A}_0^T \alpha$, equations (5) and (6) can be calculated as

$$\begin{aligned} f(\mathbf{z}) &= \mathbf{z} \cdot \mathbf{A}_0^T \cdot \alpha, \\ \hat{\mathbf{y}} &= \hat{\mathbf{z}} \odot \hat{\mathbf{a}}_0^* \odot \hat{\alpha}. \end{aligned} \quad (9)$$

The context-aware filter not only takes the target area as a sample, but also collects background information around the target area as an auxiliary sample. A new filter is obtained by ridge regression, and the objective function of context-aware filter is

$$\min_{\omega} \|\mathbf{A}_0 \omega - \mathbf{y}\|_2^2 + \lambda_1 \|\omega\|_2^2 + \lambda_2 \sum_{i=1}^k \|\mathbf{A}_i \omega\|_2^2, \quad (10)$$

where $\mathbf{A}_0$ is the sample set obtained by cyclic sampling of the target area, $\mathbf{A}_i$ is the set of samples gained by cyclic sampling of the context area, $\mathbf{y}$ is the expected output, ω is the new filter parameter, and λ_1 and λ_2 are the regular term coefficients of the tracking system.

The cyclic matrix can be constructed with equation (2), and set $\mathbf{B}$ and $\bar{\mathbf{y}}$ as

$$\begin{aligned} \mathbf{B} &= \begin{bmatrix} \mathbf{A}_0 \\ \sqrt{\lambda_2} \mathbf{A}_1 \\ \vdots \\ \sqrt{\lambda_2} \mathbf{A}_k \end{bmatrix}, \\ \bar{\mathbf{y}} &= \begin{bmatrix} \mathbf{y} \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \end{aligned} \quad (11)$$

Then, the objective function $f(\omega, \mathbf{B})$ from equations (10) and (11) is

$$f(\omega, \mathbf{B}) = \mathbf{B}\omega - \bar{y}_2^2 + \lambda_1 \omega_2^2. \quad (12)$$

The solution ω of equation (12) can be computed as

$$\omega = (\mathbf{B}^T \mathbf{B} + \lambda_1 \mathbf{I})^{-1} \mathbf{B}^T \bar{y}. \quad (13)$$

According to equations (11) and (13), the Fourier form of correlation filter ω is

$$\hat{\omega} = \frac{\hat{\mathbf{a}}_0^* \odot \hat{\mathbf{y}}}{\hat{\mathbf{a}}_0^* \odot \hat{\mathbf{a}}_0 + \lambda_1 + \lambda_2 \sum_{i=1}^k \hat{\mathbf{a}}_i^* \odot \hat{\mathbf{a}}_i}. \quad (14)$$

Letting $\omega = \mathbf{B}^T \alpha$, we know from equations (13) and (14) that

$$\alpha = (\mathbf{B}^T \mathbf{B} + \lambda_1 \mathbf{I})^{-1} \bar{y}, \quad (15)$$

$$\hat{\alpha} = \frac{\hat{\mathbf{a}}_0^*}{\hat{\mathbf{a}}_0^* \odot \hat{\mathbf{a}}_0 + \lambda_1 + \lambda_2 \sum_{i=1}^k \hat{\mathbf{a}}_i^* \odot \hat{\mathbf{a}}_i}. \quad (16)$$

By taking equation (15) into equation (5), the following equation can be obtained:

$$f(\mathbf{z}) = \mathbf{z} \cdot \mathbf{B}^T \cdot \alpha. \quad (17)$$

Equation (18) can be computed according to equation (16) and equation (6):

$$\hat{\mathbf{y}} = \hat{\mathbf{z}} \odot \hat{\mathbf{a}}_0^* \odot \hat{\alpha}_0 + \sqrt{\lambda_2} \sum_{i=1}^k \hat{\mathbf{z}} \odot \hat{\mathbf{a}}_i^* \odot \hat{\alpha}_i. \quad (18)$$

3.2. Filter Updating with Gaussian Distribution. This section discusses the filter updating when the output is unconstrained and when the output response is constrained as Gaussian distribution.

3.2.1. Filter Updating with Unconstrained Output. Let y be the expected output and $f(\mathbf{z})$ be the actual output, and then the tracking problem can be expressed as

$$\begin{cases} \varepsilon_i = y_i - \omega^T \varphi(x_i), \\ \min_{\omega, \varepsilon} \varepsilon_i^2, \end{cases} \quad \|\omega\| \leq B, \quad (19)$$

where x_i is the $M \times N$ -sized image, $\varphi(x_i)$ is a nonlinear transformation and is recorded as the input, y_i is the output, and B is a small constant. Equation (20) is a Lagrangian equation which is transformed by equation (19):

$$\mathcal{L}_p = \sum_{i=1}^{M \times N} \varepsilon_i^2 + \sum_{i=1}^{M \times N} \beta_i (y_i - \omega^T \varphi(x) - \varepsilon_i) + \lambda (\|\omega\|^2 - B). \quad (20)$$

The expressions of variables α and ω can be obtained by solving the Lagrangian equation as follows:

$$\begin{cases} \alpha = (\mathbf{K} + \lambda \mathbf{I})^{-1} y, \\ \omega = \sum \alpha_i \varphi_i. \end{cases} \quad (21)$$

Both the model $\mathbf{x}^t$ (where t is the frame index) and the filter α^t are generally updated with a fixed coefficient η as

$$\hat{\mathbf{x}}^t = (1 - \eta) \hat{\mathbf{x}}^{t-1} + \eta \mathbf{x}^t, \quad (22)$$

$$\hat{\alpha}^t = (1 - \eta) \hat{\alpha}^{t-1} + \eta \alpha^t. \quad (23)$$

3.2.2. Filter Updating with Constrained Output. The focus of this section is to introduce a filter with constraints for output response. Assuming that the target output response is satisfied with Gaussian distribution, the objective function can be expressed as

$$\min_{\omega, \varepsilon} \sum_i \varepsilon_i^2 \quad \text{st:} \quad \begin{cases} \varepsilon_i = y_i - \omega^T \varphi_i, \\ \hat{\mathbf{y}}^t \sim N(\mu^t, \sigma^{2,t}), \\ \|\omega\| \leq B, \end{cases} \quad (24)$$

where y_i is the Gaussian representation of the i^{th} sample, ω^T is the correlation filter for the i^{th} frame, φ_i is a nonlinear transformation for the i^{th} frame, μ^t and $\sigma^{2,t}$ are the mean and the variance of the Gaussian model p , and $\hat{\mathbf{y}}^t$ is a variable defined by the Gaussian function.

Based on the maximum likelihood theory, the optimal solution of Gaussian distribution is

$$\frac{(\hat{\mathbf{y}}^t - \mu^t)^2}{2\sigma^t} + \frac{\ln(2\pi\sigma^t)}{2}. \quad (25)$$

It can be seen from equation (25) that only the change of $\hat{\mathbf{y}}^t$ value can affect the optimal solution of Gaussian distribution, and equation (25) can be simplified as

$$(\hat{\mathbf{y}}^t - \mu^t)^2. \quad (26)$$

Set $\hat{\mathbf{y}}^t = \omega^{T,t} \mathbf{x}^t$ (where $\mathbf{x}^t$ represents the t^{th} sample in this sample set); then,

$$(\omega^{T,t} \mathbf{x}^t - \mu^t)^2. \quad (27)$$

In order to simplify the calculation amount of optimization process, the following operations are performed on equation (27). According to the theory of Gaussian distribution, the mean μ^t and μ^{t+1} are updated as

$$\mu^t = (1 - \rho) \mu^{t-1} + \rho \omega^{T,t} \hat{\mathbf{x}}^t, \quad (28)$$

$$\mu^{t-1} = (1 - \rho) \mu^{t-2} + \rho \omega^{T,t-1} \hat{\mathbf{x}}^{t-1}, \quad (29)$$

where $\hat{\mathbf{x}}^t$ is the appearance of the learning target in the t^{th} frame, iteratively acquired by

$$\hat{\mathbf{x}}^t = (1 - \rho) \hat{\mathbf{x}}^{t-1} + \rho \mathbf{x}^t, \quad (30)$$

where $\mathbf{x}^t$ is the target in the t^{th} frame.

By tacking equation (29) into equation (28), we can obtain

$$\mu^t = (1 - \rho) ((1 - \rho) \mu^{t-2} + \rho \omega^{T,t-1} \hat{\mathbf{x}}^{t-1}) + \rho \omega^{T,t} \hat{\mathbf{x}}^t. \quad (31)$$

Adding $(-\omega^{T,t} \mathbf{x}^t)$ to both sides of equation (31) and then

multiplying (-1) at both sides of the formula, we can obtain

$$\omega^{T,t}x^t - \mu^t = -\rho(1-\rho)\omega^{T,t-1}\hat{x}^{t-1} - (1-\rho)^2\mu^{t-2} - \rho\omega^{T,t}\hat{x}^t + \omega^{T,t}x^t. \quad (32)$$

Equation (33) can be acquired by replacing $\omega^{T,t}x^t - x^t$ with $\omega^{T,t}x^t - \hat{x}^t$:

$$\omega^{T,t}x^t - \mu^t = -\rho(1-\rho)\omega^{T,t-1}\hat{x}^{t-1} - (1-\rho)^2\mu^{t-2} - \rho\omega^{T,t}\hat{x}^t + \omega^{T,t}\hat{x}^t + \theta_1. \quad (33)$$

By taking equation (30) into equation (31), we can have

$$\begin{aligned} \omega^{T,t}x^t - \mu^t &= -\rho(1-\rho)\omega^{T,t-1}\hat{x}^{t-1} - (1-\rho)^2\mu^{t-2} - (1-\rho)^2\omega^{T,t}\hat{x}^{t-1} \\ &\quad + \rho(1-\rho)\omega^{T,t}x^t + \theta_1, \end{aligned} \quad (34)$$

where $\hat{x}^t$ can be approximately equal to $\hat{x}^{t-1}$ when there is no significant difference between two adjacent frames, and equation (34) can be approximated as

$$\omega^{T,t}x^t - \mu^t = \rho(1-\rho)(\omega^{T,t} - \omega^{T,t-1})\hat{x}^{t-1} - (1-\rho)^2(\omega^{T,t}\hat{x}^{t-1} - \mu^{t-2}) + \theta_2. \quad (35)$$

The simplified equation (35) is

$$\begin{aligned} \|\omega^{T,t}x^t - \mu^t\| &\leq \|\rho(1-\rho)(\omega^{T,t} - \omega^{T,t-1})\hat{x}^{t-1}\| + \|(1-\rho)^2(\omega^{T,t}\hat{x}^{t-1} - \mu^{t-2})\| + \|\theta_2\| \\ &\leq \rho(1-\rho)\|\omega^t - \omega^{t-1}\| \cdot \|\hat{x}^{t-1}\| + C. \end{aligned} \quad (36)$$

It can be seen from equation (36) that $\omega^{T,t}x^t - \mu^t$ also takes the minimum value when $\omega^t - \omega^{t-1}$ takes the minimum value. Therefore, as long as the optimal solution of $\omega^t - \omega^{t-1}$ is computed, the optimal solution of $\omega^{T,t}x^t - \mu^t$ is obtained as well.

Because the calculation of $\omega^t - \omega^{t-12}$ is still somewhat complicate, a new variable β_i^t is introduced for further simplifying the calculation, and according to equation (16), we can obtain by setting $\lambda = \lambda_1 + \lambda_2 \sum_{i=1}^k \hat{\mathbf{a}}_i^* \odot \hat{\mathbf{a}}_i$

$$\begin{cases} 2\varepsilon_i = \beta_i^t, \\ 2\lambda\omega = \sum_i \beta_i^t \varphi_i. \end{cases} \quad (37)$$

The solution of $\omega^t - \omega^{t-12}$ is transformed into the solution of $\beta^t - \beta^{t-12}$, and a new Lagrangian function can be inferred as

$$\mathcal{L}_p(\alpha | \mu^t, \sigma^{2,t}) = -\frac{1}{4} \sum_i \beta_i^{2,t} - \frac{1}{4\lambda} \sum_{i,j} \beta_i^t \beta_j^t \mathbf{K}_{ij} + \sum_i \beta_i^t y_i - s \sum_i (\beta_i^t - \beta_i^{t-1})^2 - \lambda B^2. \quad (38)$$

Letting $\omega = \sum_i \alpha_i^t \varphi_i$ and then $\beta_i^t = 2\lambda\alpha_i^t$, equation (38) can be computed as

$$\max_{\alpha, \lambda} -\lambda^2 \sum_i \alpha_i^{2,t} + 2\lambda \sum_i \alpha_i^t y_i - \lambda \sum_{i,j} \alpha_i^t \alpha_j^t \mathbf{K}_{ij} - 4\lambda^2 s \sum_i (\alpha_i^t - \alpha_i^{t-1})^2. \quad (39)$$

The partial derivative of λ in equation (39) is

$$(\lambda \mathbf{I} + 4\lambda s \mathbf{I} + \mathbf{K}) \boldsymbol{\alpha}^t = \mathbf{y} + 4\lambda s \boldsymbol{\alpha}^{t-1}. \quad (40)$$

Then, the Fourier transform of equation (40) is

$$F(\boldsymbol{\alpha} | \mu^t, \sigma^{2,t}) = \frac{F(\mathbf{y}) + 4\lambda s F(\boldsymbol{\alpha}^{t-1})}{F(\mathbf{k}) + \lambda + 4\lambda s}. \quad (41)$$

The simplification of equation (41) can be

$$F(\boldsymbol{\alpha} | \mu^t, \sigma^{2,t}) = \frac{F(\mathbf{k}) + \lambda}{F(\mathbf{k}) + \lambda + 4\lambda s} \odot \frac{F(\mathbf{y})}{F(\mathbf{k}) + \lambda} + \left(1 - \frac{F(\mathbf{k}) + \lambda}{F(\mathbf{k}) + \lambda + 4\lambda s}\right) \odot F(\boldsymbol{\alpha}^{t-1}). \quad (42)$$

Letting $\tau = (F(\mathbf{k}) + \lambda / F(\mathbf{k}) + \lambda + 4\lambda s)$ and taking τ into equation (42), equation (43) can be obtained as

$$F(\boldsymbol{\alpha} | \mu^t, \sigma^{2,t}) = \tau \odot F(\boldsymbol{\alpha}^t) + (1 - \tau) \odot F(\boldsymbol{\alpha}^{t-1}), \quad (43)$$

where μ^t and $\sigma^{2,t}$ are used to select the samples as shown in equation (44) and τ is a matrix with the same size as $F(\boldsymbol{\alpha})$.

Different from the correlation filter using the threshold to detect the fault condition, this paper considers that the Gaussian prior property can prevent from the drift well. The proposed method uses Gaussian prior to select samples when their response output is Gaussian distribution, and it is satisfied with the following equation:

$$\left| \frac{\hat{y}^t - \mu^t}{\sigma^t} \right| < G, \quad (44)$$

where G is the empirical value, $\hat{y}^t$ is the maximum output response, μ^t is the mean, and σ^t is the standard deviation.

According to the basic principles of mean and variance, we know that μ^t and $\sigma^{2,t}$ are

$$\mu^t = \frac{\sum_{i=1}^t \hat{y}_i}{t}, \quad (45)$$

$$\sigma^{2,t} = \frac{\sum_{i=1}^t (\hat{y}_i - \mu^t)^2}{t}, \quad (46)$$

where $\hat{y}_i$ is the maximum output response of the i^{th} frame.

3.3. Scale Updating with Maximum Posterior Probability Distribution. At present, many scale updating methods are based on the likelihood principle. In the tracking detection stage, the maximum possibility of each scale is used to find the optimal scale in the scale pool. We can set the size of scale template as $S_T = (S_x, S_y)$ and its zoom pool as $S = \{t_1 \ t_2 \ \dots \ t_k\}$ at first, and then we can use k windows with the $\{t_i s_i | t_i \in S\}$ size to find the target and use bilinear interpolation to adjust the sample size to the fixed template size S_T by assuming that the target window is s_i in the original image. The response is calculated as follows:

$$f(z^{t_i}) = K^{xz} \odot \boldsymbol{\alpha}, \quad (47)$$

where z^{t_i} is the sample patch with the $t_i s_i$ size, K^{xz} is the correlation operation between the sample z and template x , and $\boldsymbol{\alpha}$ is the filter.

According to equation (47), the maximum response is

$$\arg \max_i F^{-1}(\hat{f}(z^{t_i})). \quad (48)$$

After a set of data is obtained using the response function $\hat{f}(z^{t_i})$, then the maximum function is used to find its maximum scalar. Because the target motion is hidden in the response graph, it is necessary to use t_i to adjust the final displacement to get the real motion deviation.

Although the maximum likelihood method has made some achievements in dealing with the problem of the target scale, it fails to solve the problem of the complex target scale. In order to solve the scale problem of the complex target, this paper adopts the maximum posterior probability distribution method instead of the maximum likelihood method, and the details are shown as

$$\max_i P(s_i | \mathbf{y}) = P(\mathbf{y} | s_i) P(s_i), \quad (49)$$

$$y_s = \arg \max_i F^{-1}(\hat{f}(z^{t_i})) P(s_i), \quad (50)$$

where s_i is the i^{th} scale, $\mathbf{y}$ is the output in the current frame, $P(\mathbf{y} | s_i)$ is the likelihood which can be calculated from the maximum response peak value at the i^{th} scale, y_s is the optimal scale response of the current frame, and $P(s_i)$ is a possibility to follow a Gaussian distribution, which can be obtained by centering around the previous scale and the standard deviation which is set in the experiment.

3.4. Steps of the Proposed Algorithm. The detailed steps of the proposed algorithm are summarized in Algorithm 1 based on the results of studies in Sections 3.1, 3.2, and 3.3.

4. Experiments

The proposed OURS tracker in this paper is compared with some excellent trackers in the standard test set OTB-50 and OTB-100. In the OTB dataset, each sequence has 11 attributes including illumination variation (IV), out-of-plane rotation (OPR), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-view (OV), background clutter (BC), and low resolution (LR) that represent challenging aspects of target tracking. The comparison algorithms include CSK [28], CACF [25], KCF [32], and Staple [34]. The precision curve is the percentage of estimated position which is within the given threshold of the ground truth, and the success curve is the coincidence rate between the predicted target position and the actual target position. The precision curve and the success curve are taken as evaluation criteria in our work. When the coincidence rate of

Inputs:

- $\mathbf{b}_0$: target location and size of the first frame
- $\hat{\mathbf{x}}_0$: the target model of the first frame
- $\hat{\alpha}_0$: the correlation filter of the first frame
- λ_1, λ_2 : the regular term coefficients of the tracking system

Outputs:

- $\mathbf{b}_{n-1}$: target location and size of the previous frame
- $\hat{\mathbf{x}}_n$: the target model of the current frame
- $\hat{\alpha}_n$: the correlation filter of the current frame
- P_{si} : maximum probability of the i^{th} scale
- y_{scale} : optimal scale response of the current frame
- $\mathbf{y}_n$: the maximum correlation filter response value of the current frame
- P_n : target location of the current frame

Preprocessing position and model

- (1) Initialize the bounding box of the target $\mathbf{b}_0 = [x_0, y_0, w, h]$
- (2) When the frame number n is less than 11, calculate the best mean and the variance of Gaussian constraint with equations (45) and (46)
- (3) Get a search window based on $\mathbf{b}_{n-1}$ ($n \geq 1$), calculate the maximum correlation response $\hat{\mathbf{y}}$ according to equation (16), and then mark the maximum correlation filter response value as $\mathbf{y}_n$, and the target position P_n is obtained from $\mathbf{y}_n$
- (4) Continue step (5) if $n < 11$, and go to step 8 if $n \geq 11$
- (5) Update the target model $\hat{\mathbf{x}}_n$ from equation (22)
- (6) Update the correlation filter $\hat{\alpha}_n$ with equation (23)
- (7) Return to step (3)

Model updating

- (8) Update the target model $\hat{\mathbf{x}}_n$ using equation (22)

Scale updating

- (9) Calculate the response P_{si} according to equations (47) and (48)
- (10) The maximum posterior probability is calculated by equation (47), and the optimal scale response y_{scale} is obtained from equation (50)

Filter updating

if $|\hat{\mathbf{y}} - \mu/\sigma| < G$

- (11) Update the correlation filter $\hat{\alpha}_n$ according to equation (43)
- else
- (12) Update the correlation filter $\hat{\alpha}_n$ using equation (23)
- end
- (13) Repeat step (3)

Until End of the video sequence

End

ALGORITHM 1: Scale adaptive context-aware correlation filter with output constraints for target tracking.

this frame is greater than the given threshold, it is recorded as successful tracking, otherwise it fails. In this study, the threshold of the precision curve is set to 20 px, and the threshold of the success curve is set to 0.5.

4.1. Parameter Setup. This experiment tests the effect of parameters on the OURS tracker. For example, an experiment is completed based on a subset of OTB with different s values as shown in Figure 2, and the precision is the highest when s is 1050. In order to choose the best constraint G in equation (43), this study tested the different values in Figure 3, the precision is not changed much when G is varied from 1.0 to 1.5, and the precision is the highest when G is 1.7, so we take G is 1.7 in this work. In the relevant filtering part,

FIGURE 2: The evaluation of S based on precision.

FIGURE 3: The evaluation of G based on precision.

FIGURE 4: The average precision and success plot of 5 trackers (OTB-50). (a) Precision plots of OPE. (b) Success plots of OPE.

the feature is taken as a combination of the HOG feature and color feature, and the kernel function is Gaussian kernel function. Let $\lambda_1 = 10^{-4}$ and $\lambda_2 = 0.4$. Depending on experience, the standard deviation of Gaussian kernel function is $\sigma^2 = 0.5$. The selection of other parameters in the proposed OURS tracker is basically consistent with the CACF tracker [27].

4.2. Experiments on the Full Dataset OTB-50. This work is done by evaluating OPE and evaluates the tracker by running the tracker on 51 benchmark video sequences, and the actual ground position is initialized in the first frame and the average accuracy is derived. The precision and the success rate of various trackers are tested in the standard test set OTB-2013, and their test results are shown in Figure 4. The proposed OURS tracker achieves a score of 0.845 for the precision low threshold value of 20, and it

achieves a score of 0.774 for the success rate above the overlap threshold value of 0.5; the OURS tracker achieves the best performance in these trackers. The comparison of the OURS tracker and CACF tracker is summarized in Table 1. The OURS tracker gets higher precision than the CACF tracker when dealing with sequences containing illumination variation (IV), scale variation (SV), out-of-plane rotation (OPR), occlusion (OCC), deformation (DEF), in-plane rotation (IPR), motion blur (MB), and fast motion (FM) attributes. The running speed of the OURS tracker and other trackers is shown in Figure 5. It can be seen from Figure 5 that though the running speed of the CSK tracker is the highest and the running speed of the OURS tracker is the lowest, the running speed of the OURS tracker is only 13 frames slower than the CACF tracker. Comprehensive analysis from the high performance and running speed of the OURS tracker shows that the OURS tracker still has high application value.

TABLE 1: Performance (mean precision (20 px)) comparisons of the proposed method with the CACF tracker.

Method	Overall	IV	OPR	SV	OCC	DEF	MB	FM	IPR	OV	BC	LR
OURS (%)	84.5	81.3	84.8	79.6	82.9	82.4	74.6	73.9	79.7	75.7	78.6	51.5
CACF (%)	80.3	77.3	78.3	74.0	79.5	78.3	71.7	68.4	73.9	76.5	77.8	54.7

FIGURE 5: Speed comparisons of the proposed method with other trackers (OTB-50).

4.3. Experiments with Various Challenging Sequence Attributes on OTB-50. The attributes of standard test set annotations describe the challenges that the tracker faces within each video sequence. This test set allows the researcher to judge and characterize the behavior of the tracker without having to analyze each video sequence. In Figure 6, this experiment counts the precision of each attribute, and it is easy to see that the OURS tracker works the best properties. The success rate of each attribute is shown in Table 2, and the maximum value of success rates is represented by a bold font. It can be seen from Table 2 that the OURS tracker has the highest success rate when dealing with illumination variation (IV), out-of-plane rotation (OPR), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), and fast motion (FM). Although the success rate of Staple tracker is highest when dealing with in-plane rotation (IPR) and the success rate of CACF tracker has the highest success rate when dealing with out-of-view (OV), background clutter (BC), and low resolution (LR), the OURS tracker has also achieved the near-optimal performance in these four attributes. In a word, the OURS tracker achieves better performance in most cases.

4.4. Experiments on the Full Dataset OTB-100. To better verify the superiority of OURS tracker, the accuracy and success rate of the OURS tracker and other trackers are tested on the standard test set OTB-100, and the test results are shown in Figure 7. According to Figure 7, we know that the accuracy scores of OURS tracker and CACF tracker are 0.827 and 0.793 when the threshold value is lower than 20, so the tracking accuracy of OURS tracker is 0.034 higher than CACF tracker. The success rate scores of OURS tracker and CACF tracker are 0.718 and 0.699 when the overlap

threshold value is more than 0.5, so the tracking success rate of OURS tracker is 0.019 higher than that of CACF tracker. In order to better highlight the value of OURS tracker, the comprehensive performance of five trackers is shown in Table 3. It can be seen from Table 3 that the tracking accuracy and tracking success rate of OURS tracker is the highest among the 5 trackers. Although the running speed of the OURS tracker is lower, it is only 14.13 frames slower than the running speed of CACF tracker. In the case of low running speed requirement, the OURS tracker has better tracker performance than other trackers.

4.5. Experiments with Various Challenging Sequence Attributes on OTB-100. To verify the effectiveness of the OURS tracker, the OURS tracker is again compared with other trackers on the standard test set OTB-100. It can be seen from Figure 8 that the OURS tracker has achieved the best tracking performance when tracking objects with occlusion, geometric deformation, and fast movement. By solving the tracking performance of these main target attributes, the problem of tracking target drift is greatly alleviated. The OURS tracker has the best tracking performance on most attributes. In addition, we know from Figure 9 that the success rate of OURS tracker has obvious advantages compared with other trackers on most attributes. In summary, the OURS tracker achieves the best tracking performance compared with other trackers in almost all attributes.

4.6. Experiments under Different Video Sequences. In order to more accurately describe the performance of the OURS tracker, this paper selects nine video sequences and tests their precision. The precision of nine video sequences in this experiment is shown in Table 4, and the best results are reported in bold. According to the test results in Table 4, the OURS tracker has the highest precision when tracking Bolt, Coke, Couple, Jumping, Freeman1, Freeman4, Jogging, and Girl video sequences, and it also achieved near-highest precision when tracking Football video sequences. The OURS tracker performs better than other four trackers.

This experiment provides a qualitative comparison of OURS tracker with the existed tracker in order to verify the stability of OURS tracker. These video sequences constitute different challenging situations including illumination variation (IV), out-of-plane rotation (OPR), scale variation (SV), in-plane rotation (IPR), background clutter (BC), occlusion (OCC), deformation (DEF), fast motion (FM), and motion blur (MB). The actual tracking results of OURS tracker, CACF tracker, and KCF tracker in four video sequences are shown in Figure 10. Figure 10(a) shows that the KCF tracker cannot stably track video sequences that contain scale variation attribute, but the OURS tracker and

FIGURE 6: Detailed results on each attribute (OTB-50). Precision plots of (a) OPE-fast motion (17). (b) OPE-occlusion (29). (c) OPE-deformation (19). (d) OPE-scale variation (28). (e) OPE-background clutter (21). (f) OPE-out-of-plane rotation (39). (g) OPE-in-plane rotation (31). (h) OPE-illumination variation (25). (i) OPE-motion blur (12). (j) OPE-out-of-view (6). (k) OPE-low resolution (4).

TABLE 2: Success rate comparisons of the tracker on different attributes.

Trackers	IV	OPR	SV	OCC	DEF	MB	FM	IPR	OV	BC	LR
OURS	73.7	75.6	68.9	76.4	80.4	71.4	69.3	70.9	76.1	72.1	51.2
Staple	69.2	70.6	65.4	73.2	77.4	62.8	60.5	71.4	58.6	69.8	49.9
KCF	58.1	60.8	47.9	61.8	67.1	59.5	55.7	61.5	65.0	67.2	35.7
CACF	71.9	72.0	65.9	74.3	78.5	69.0	65.5	67.8	76.8	75.4	54.2
CSK	38.8	43.9	35.2	40.0	37.0	33.6	38.0	45.7	41.0	49.1	39.7

FIGURE 7: The average precision and success plot of 5 trackers (OTB-100). (a) Precision plots of OPE. (b) Success plots of OPE.

TABLE 3: Comprehensive tracking performance of each tracker.

Method	OURS	CACF	Staple	KCF	CSK
Precision	0.827	0.793	0.784	0.696	0.518
Success	0.718	0.699	0.695	0.551	0.411
FPS	21.65	35.58	166.92	214.12	262.54

FIGURE 8: Detailed results on each attribute (OTB-100). Precision plots of (a) OPE-occlusion (48). (b) OPE-fast motion (39). (c) OPE-background clutter (31). (d) OPE-out-of-plane rotation (63). (e) OPE-in-plane rotation (51). (f) OPE-low resolution (9). (g) OPE-illumination variation (37). (h) OPE-occlusion (48). (i) OPE-motion blur (29). (j) OPE-out of view (14). (k) OPE-scale variation (63).

the CACF tracker can consistently track the Shaking video sequence. Figure 10(b) shows that the CACF tracker cannot successfully track the Bolt sequence when the video sequence contains in-plane rotation attributes, but the OURS tracker and the KCF tracker can handle these issues very well. Figure 10(c) shows that only the OURS tracker can stably

track Freeman video sequences and neither the KCF tracker nor CACF tracker can successfully track the sequence. Figure 10(d) shows that only the OURS tracker can successfully track the Jumping video sequence when the video contains fast motion and motion blur properties. In summary, the OURS tracker has excellent stability.

FIGURE 9: Comparison of tracker success rates (OTB-100).

TABLE 4: Performance (mean precision (20 px)) comparisons of different trackers in some video sequences.

Trackers	Bolt	Coke	Couple	Jumping	Freeman1	Freeman4	Jogging	Girl	Football
OURS	100	96.7	89.7	83.5	95.9	75.5	97.4	100	81.7
Staple	100	91.8	34.9	44.9	37.1	12.5	36.6	100	74.9
KCF	67.1	82.8	26.7	36.1	39.6	53.0	23.5	86.2	76.8
CACF	50.7	92.1	55.7	49.5	38.0	49.5	36.8	100	78.8
CSK	23.5	91.2	10.6	20.1	40.8	38.9	54.2	53.6	83.4

FIGURE 10: A visualization of the tracking results of OURS tracker and some visual trackers (CACF [25] and KCF [32]) on four benchmark sequences. (a) Shaking (IV, OPR, SV, and BC). (b) Bolt (OPR, OCC, DEF, and IPR). (c) Freeman1 (OPR, SV, and IPR). (d) Jumping (MB and FM).

5. Conclusion

In this work, a scale-adaptive context-aware correlation filtering algorithm with constrained output response for object tracking is proposed: (1) this study assumes that the output response is Gaussian distribution, and a variable updating parameter is found according to Gaussian output constraints. (2) the filter is updated with variable updating parameters when the output response is Gaussian constraint, and the filter is updated with fixed updating parameters when the output response is not any Gaussian constraint. (3) the optimal scale of target is obtained by using the maximum posterior probability distribution.

The proposed OURS tracker in this paper performs in the research cases because it has the following advantages. To begin with, the OURS tracker can provide a new filter that can get a more accurate model. In addition, the suggested tracker adopts a selectively updating strategy to effectively increase the tracking accuracy. Finally, the maximum posterior probability method is used to obtain a more accurate target scale. The experimental results have shown that the proposed tracker achieves better tracking performance than other trackers when dealing with drift problems caused by fast motion, deformation, and occlusion. Therefore, the developed tracker significantly improves the ability of the CACF tracker to handle drift problems and achieves better performance than many other trackers.

Further research will focus on two aspects: (1) we will focus on solving the problem for low resolution and out-of-view to achieve higher tracking performance. (2) We are only studying the single target tracking problem, so the next step is to study the tracking of multiple targets with output response constraint which is satisfied with Gaussian distribution.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 61671222) and the Postgraduate Research & Practice Innovation Program of Jiangsu Province (no. KYCX19_1693).

References

- [1] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust Tracking via Multiple Experts using Entropy Minimization," in *Proceedings of the European Conference on Computer Vision*, pp. 188–203, Zurich, Switzerland, September 2014.
- [2] D. Benarab, T. Napoléon, A. Alfalou, A. Verney, and P. Hellard, "Optimized swimmer tracking system by a dynamic fusion of correlation and color histogram techniques," *Optics Communications*, vol. 356, pp. 256–268, 2015.
- [3] C. Ding, J. Choi, D. Tao, and L. S. Davis, "Multi-directional multi-level dual-cross patterns for robust face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 3, pp. 518–531, 2014.
- [4] H. Zhou, L. Wei, and C. P. Lim, "Robust vehicle detection in aerial images using bag-of-words and orientation aware scanning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 99, pp. 1–12, 2018.
- [5] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Towards reaching human performance in pedestrian detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 973–986, 2018.
- [6] D. Yuan, N. Fan, and Z. He, "Learning target-focusing convolutional regression model for visual object tracking," *Knowledge Based Systems*, vol. 194, Article ID 105526, 2020.
- [7] D. Yuan, X. Zhang, J. Liu, and D. Li, "A multiple feature fused model for visual object tracking via correlation filters," *Multimedia Tools and Applications*, vol. 78, no. 19, pp. 27271–27290, 2019.
- [8] O. Z. Kraus and B. J. Frey, "Computer vision for high content screening," *Critical Reviews in Biochemistry and Molecular Biology*, vol. 51, no. 2, pp. 102–109, 2016.
- [9] L. N. Gaxiola, V. H. Diaz-Ramirez, J. J. Tapia, and P. García-Martínez, "Target tracking with dynamically adaptive correlation," *Optics Communications*, vol. 365, pp. 140–149, 2016.
- [10] C. Tian, Y. Xu, and W. Zuo, "Image denoising using deep CNN with batch renormalization," *Neural Networks*, vol. 121, pp. 461–473, 2020.
- [11] C. Tian, Y. Xu, Z. Li, W. Zuo, L. Fei, and H. Liu, "Attention-guided CNN for image denoising," *Neural Networks*, vol. 124, pp. 117–129, 2020.
- [12] K. Dimitropoulos, P. Barmoutis, and N. Grammalidis, "Higher order linear dynamical systems for smoke detection in video surveillance applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 5, pp. 1143–1154, 2017.
- [13] D. Yuan, L. Xin, Z. He, L. Qiao, and S. Wu, "Visual object tracking with adaptive structural convolutional network," *Knowledge-Based Systems*, vol. 194, Article ID 105554, 2020.
- [14] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. van den Hengel, "Part-based robust tracking using online latent structured learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 6, pp. 1235–1248, 2017.
- [15] O. Akin, E. Erdem, A. Erdem, and K. Mikolajczyk, "Deformable part-based tracking by coupled global and local correlation filters," *Journal of Visual Communication and Image Representation*, vol. 38, pp. 763–774, 2016.
- [16] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015.
- [17] Y. Huang, T. L. Song, and W. J. Lee, "Multiple detection joint integrated track splitting for multiple extended target tracking," *Signal Processing*, vol. 163, pp. 126–140, 2016.
- [18] H. K. Zhang, L. Zhang, and M. H. Yang, "Fast compressive tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 10, pp. 125–141, 2014.
- [19] P.-Y. Lv, S.-L. Sun, C.-Q. Lin, and G.-R. Liu, "Space moving target detection and tracking method in complex background," *Infrared Physics & Technology*, vol. 91, pp. 107–118, 2018.
- [20] D. Yuan, X. Lu, D. Li, Y. Liang, and X. Zhang, "Particle filter re-detection for visual tracking via correlation filters," *Multimedia Tools and Applications*, vol. 78, no. 11, pp. 14277–14301, 2019.

- [21] W. J. Liu, D. Q. Liu, and B. W. Fei, "Optimal matching tracking algorithm based on discriminant appearance model," *Pattern Recognition and Artificial Intelligence*, vol. 30, no. 9, pp. 791–802, 2017.
- [22] K. Nithin and F. Bremond, "Globality-locality-based consistent discriminant feature ensemble for multicamera tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 3, pp. 431–440, 2017.
- [23] A. El-Fergany, "Multi-objective allocation of multi-type distributed generators along distribution networks using backtracking search algorithm and fuzzy expert rules," *Electric Power Components and Systems*, vol. 44, no. 3, pp. 252–267, 2016.
- [24] Y. Cong, B. Fang, and J. Liu, "Speeded up low-rank online metric learning for object tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 6, pp. 922–934, 2015.
- [25] M. Mueller, N. Smith, and B. Ghanem, "Context-aware correlation filter tracking," in *Proceedings of the IEEE Conference on Computer Vision*, pp. 1387–1395, Venice, Italy, October 2017.
- [26] S. Hare, S. Golodetz, A. Saffari et al., "Struck: structured output tracking with kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2096–2109, 2016.
- [27] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proceedings of the IEEE Conference Computer Vision*, pp. 2544–2550, San Francisco, CA, USA, June 2010.
- [28] J. F. Henriques, C. Rui, and P. Martins, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proceedings of the European Conference on Computer Vision*, pp. 702–715, Florence, Italy, October 2012.
- [29] D. Yuan, W. Kang, and Z. He, "Robust visual tracking with correlation filters and metric learning," *Knowledge Based Systems*, vol. 195, Article ID 105697, 2020.
- [30] W. Ou, D. Yuan, Q. Liu, and Y. Cao, "Object tracking based on online representative sample selection via non-negative least square," *Multimedia Tools and Applications*, vol. 77, no. 9, pp. 10569–10587, 2018.
- [31] M. Danelljan, K. F. Shahbaz, and M. Felsberg, "Adaptive color attributes for real-time visual tracking," in *Proceedings of the IEEE Conference on Computer Vision*, pp. 1090–1097, Columbus, OH, USA, June 2014.
- [32] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.
- [33] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proceedings of the European Conference on Computer Vision*, pp. 254–265, Zurich, Switzerland, September 2014.
- [34] L. Bertinetto, J. Valmadre, and S. Golodetz, "Staple: complementary learners for real-time tracking," in *Proceedings of the IEEE Conference on Computer Vision*, pp. 1401–1409, Las Vegas, NV, USA, June 2016.
- [35] H. Li, H. Wu, H. Zhang, S. Lin, X. Luo, and R. Wang, "Distortion-aware correlation tracking," *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5421–5434, 2017.
- [36] M. Yang, Y. Wu, and G. Hua, "Context-aware visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 7, pp. 1195–1209, 2009.
- [37] T. Zhang, B. Ghanem, S. Liu, C. Xu, and N. Ahuja, "Robust visual tracking via exclusive context modeling," *IEEE Transactions on Cybernetics*, vol. 46, no. 1, pp. 51–63, 2016.