sensors

MDPI

Article

Real-Time Object Tracking via Adaptive Correlation Filters

Chenjie Du [1,2], Mengyang Lan [1], Mingyu Gao [1,2], Zhekang Dong [1], Haibin Yu [1] and Zhiwei He [1,2,*]

1 School of Electronic Information, Hangzhou Dianzi University, Hangzhou 310018, China; ducj@hdu.edu.cn (C.D.); 182040114@hdu.edu.cn (M.L.); mackgao@hdu.edu.cn (M.G.); englishp@hdu.edu.cn (Z.D.); shoreyhb@hdu.edu.cn (H.Y.)
2 Zhejiang Provincial Key Lab of Equipment Electronics, Hangzhou 310018, China
* Correspondence: zwhe@hdu.edu.cn

Received: 24 May 2020; Accepted: 21 July 2020; Published: 24 July 2020

Abstract: Although correlation filter-based trackers (CFTs) have made great achievements on both robustness and accuracy, the performance of trackers can still be improved, because most of the existing trackers use either a sole filter template or fixed features fusion weight to represent a target. Herein, a real-time dual-template CFT for various challenge scenarios is proposed in this work. First, the color histograms, histogram of oriented gradient (HOG), and color naming (CN) features are extracted from the target image patch. Then, the dual-template is utilized based on the target response confidence. Meanwhile, in order to solve the various appearance variations in complicated challenge scenarios, the schemes of discriminative appearance model, multi-peaks target re-detection, and scale adaptive are integrated into the proposed tracker. Furthermore, the problem that the filter model may drift or even corrupt is solved by using high confidence template updating technique. In the experiment, 27 existing competitors, including 16 handcrafted features-based trackers (HFTs) and 11 deep features-based trackers (DFTs), are introduced for the comprehensive contrastive analysis on four benchmark databases. The experimental results demonstrate that the proposed tracker performs favorably against state-of-the-art HFTs and is comparable with the DFTs.

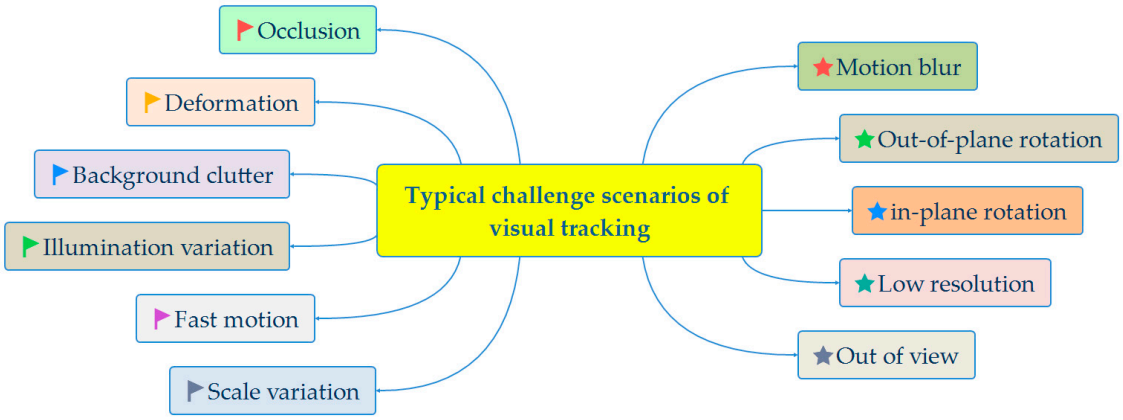
Keywords: correlation filter; histogram of oriented gradient; color naming; dual-template; target re-detection

1. Introduction

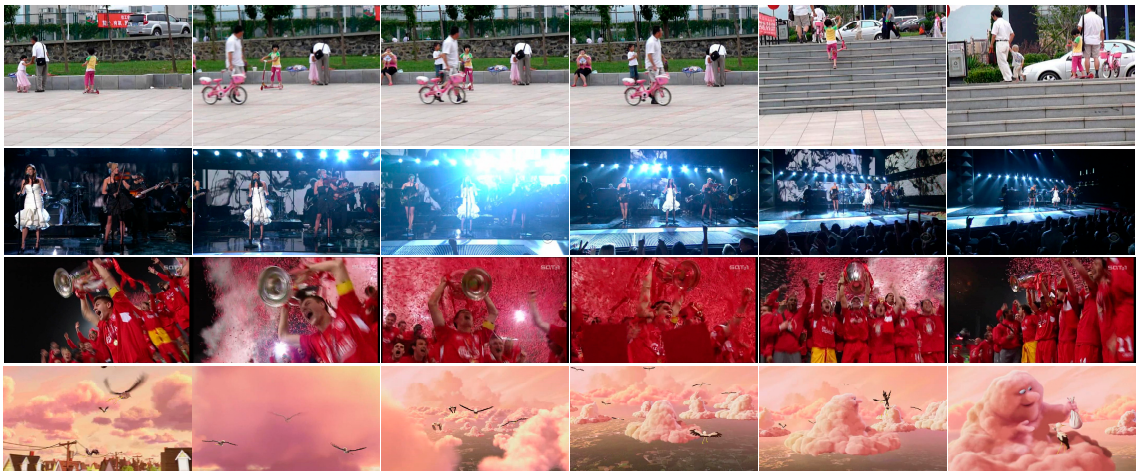
Visual object tracking is a challenging job and a hot research topic of the computer vision community. At present, with the technical improvement of hardware facilities and artificial intelligence, visual object tracking plays a critical role in innumerable applications (i.e., human-computer interaction, robot navigation, surveillance systems, handwritten recognition) [1]. Numerous object tracking works [2–7] have been widely researched to overcome the difficulty of tracking failures. Generally, object trackers can be divided into either generative trackers [8,9] or discriminant trackers [10,11], depending on whether a model utilizing background information is used to predict the target location. The results for the object tracking benchmark (OTB) [12,13] and visual object tracking (VOT) [14,15] show that discriminant tracking algorithms are superior to generative tracking algorithms in terms of tracking overall performance.

Recently, owing to the advantages of simplicity, robustness, and high accuracy, correlation filter-based trackers (CFTs) have received ample attention in visual object tracking [16,17]. Although the CFTs have made great achievements, tracking drifting or even loss may occur due to partial or complete occlusion, motion blur, deformation, illumination variation, and background clutter, etc. It is difficult

to improve the performance of the tracking algorithm in different challenge scenarios. A variety of complicated challenge scenarios are shown in Figure 1.



(**a**)



(**b**)

Figure 1. A variety of complicated challenge scenarios of visual tracking: (**a**) the 11 common video challenge scenarios of visual tracking; (**b**) the target objects are in different pose and appearance variations.

It is challenging to design a robust tracker. The CFT is a typical discriminant tracking algorithm. It evolved from the earliest minimum output sum of squared error (MOSSE) [18] to the circulant structure of tracking-by-detection with kernels (CSK) [19] and the kernelized correlation filter (KCF) [20]. KCF improves performance while ensuring a higher running speed. Learning spatially regularized correlation filters for visual tracking (SRDCF) [21] employ a large detection area and add weight constraints to filter coefficients to effectively alleviate the boundary effect. Generally, in complicated challenge scenarios, the above-mentioned CFTs still experience serious performance decline in the discrimination ability. In [22], multiple features were combined to improve the tracking performance and seek the optimal feature combination. In [23], Alexandros Makris et al. proposed a particle filter based tracking algorithm and fused the features of the salient points and color histogram. This tracker tackled the multimodal distributions emerging from cluttered scenes. Furthermore, most existing trackers suffer from many limitations: (1) the sole filter template is used; (2) the search range of the object is fixed; (3) the fusion weight coefficient is invariable during the updating process.

Additionally, owing to the strong representation ability of deep features, the correlation filter with depth features can achieve great success in visual object tracking, while the heavy computation

complexity inevitably results in low tracking speed and limited application in practical systems. Hence, a real-time dual-template CFT for various typical challenge scenarios is proposed in this work. In contrast to existing object tracking algorithms, the main contributions of this paper are as follows:

(1) To solve the limitation of the sole filter template, a dual-template method is proposed to improve the robustness of the tracker;
(2) In order to solve the various appearance variations in complicated challenge scenarios, the schemes of discriminative appearance model, multi-peaks target re-detection, and scale adaptive are integrated into the tracker;
(3) A high-confidence template updating technique is utilized to solve the problem that the filter model may be drift or even corruption.

The rest of this paper is organized as follows. Related work is discussed in Section 2. Then, the classical kernel correlation filter is presented in Section 3. The details of the proposed tracking method are illustrated in Section 4. Section 5 evaluates the performance of the proposed tracking method through experimental simulations. Finally, Section 6 concludes this work.

2. Related Work

In this section, several related methods are briefly described, including the early object tracking algorithms, the convolutional neural network (CNN)-based object tracking algorithms, and the correlation filter-based object tracking algorithms.

2.1. The Early Object Tracking Algorithms

In [8], texture features were added to the mean shift algorithm for tracking framework, because texture features can better describe the apparent features of the object and overcome the interference of background color similarity. In [24], a heuristic local anomaly factor was proposed to improve the tracking accuracy by using global matching and local tracking methods. Li et al. [25] proposed a patched based tracker by using reliable patches for object tracking. In [26], the scheme of mixed-integer programming was utilized to track a ball in team sports. The above object tracking algorithms are processed in the time domain. In the tracking process, it involves a complex matrix inversion calculation, which has a large amount of calculation and poor real-time performance.

2.2. The CNN-Based Object Tracking Algorithms

Hierarchical convolutional features (HCF) [27] reconciled the shallow and deep features of the visual geometry group (VGG) network and integrated them into a correlation filter, which obtained good tracking performance. However, HCF does not consider the scale variation and assumes that the target scale is constant in the whole tracking sequence. The multi-domain network (MDNET) [28] tracking algorithm exploits a small network to learn the convolution features and uses a softmax strategy to classify the samples. The performance of MDNET is excellent, yet the speed is only 1 frame per second (FPS). Adversarial deep tracking (ADT) [29] employs the deep convolutional generative adversarial networks, which are composed of the fully convolutional siamese network (SiamFC) [30] and the classification network. ADT can be trained and optimized end-to-end through adversarial learning. Wang et al. [31] integrated CNN features to the correlation filter framework. The peak-to-sidelobe ratio (PSR) is utilized to measure the differences between image patches. In [32], a sample-level generative adversarial network was used to expand the training samples, and a label smoothing loss regularization can obtain filter model regularization and reduce overfitting, the speed of the tracker is only 1.02 FPS on OTB-2013 dataset. Huang et al. proposed GlobalTrack, which is developed based on two-stage target detectors [33]. The GlobalTrack method can run without cumulative errors and obtain the best performance on four large-scale tracking datasets. In [34], a robust long-term tracking based on skimming and perusal modules (SPLT) was proposed. This robust algorithm ranked first on the VOT2018 long-term dataset. A novel long-term tracking framework based on deep regression and

verification networks was proposed by Zhang et al. [35]. The Distractor-aware Siamese Networks [36] exploited an effective sampling strategy to control the distribution of training samples and make the model focus on the semantic distractors. Li et al. proposed the SiamRPN++ approach [37] to take advantage of features from deep networks and improve the accuracy of the tracker. GlobalTrack, SiamRPN++, and DaSiam_LT rank first, third, and fourth in all the long-term tracking approaches at home and abroad, respectively. Generally, as the network layers number of DFTs increases, the computational complexity and the storage space of parameters are increased exponentially.

2.3. The Correlation Filter-Based Object Tracking Algorithms

The earliest correlation filter tracker was the MOSSE filter, which was derived from signal correlation and simplified using Fourier correlation properties. Henriques proposed the CSK by introducing dense sampling and exploiting the Fourier diagonalization property of circulant matrices to accelerate the training and detection process. The CSK can run at an average speed of up to 362 FPS on the OTB-2015 dataset. The KCF was proposed based on the CSK by employing the kernel function and extending the multichannel histogram of oriented gradient (HOG). The surface texture feature and contour shape of the object can be well described by the HOG. Danelljan proposed a filter extended with color naming (CN) [38], which can effectively use the color information of the target. To improve the tracking speed, principal component analysis (PCA) is also used to reduce the 11-dimensional features to two-dimensional features. In [39], Zhu et al. presented a tracker that is not limited to a local search window and can efficiently probe the entire frame. In [40], fast discriminative scale space tracking (fDSST) employed a novel scale adaptive tracking method by learning separate discriminative correlation filters, which can improve the robustness and speed of tracker. Gao et al. proposed a maximum margin object tracking with weighted circulant feature maps (MMWCF) [41] to reduce the influence of inaccurate samples. An anti-occlusion correlation filter-based tracking method (AO-CF) [42] proposed an occlusion criterion and a new detection condition for detecting proposals. Long-term correlation tracking (LCT) [43] consists of translation and scale estimation and trains an online random fern classifier to re-detect objects in case of tracking failure. The CFTs possess the following superiorities: (1) CFTs skillfully use fast Fourier transform (FFT) to calculate the correlation characteristics between samples and filter template and find the target area with the greatest output response, making the trained filter model more reliable and accurate; (2) CFTs can transform the multiplication operation between matrices into element point multiplication, which can substantially improve the computation speed and achieve real-time tracking; (3) the tracking accuracy of correlation filter-based trackers can meet the needs of many applications and can be realized on embedded devices.

3. Kernel Correlation Filter Tracker

In this section, the principle of the kernel correlation filter tracker is explained in detail. It is noted that the introduction of the kernel correlation filter makes the tracker more robust and can deal with nonlinear classification problems.

3.1. Ridge Regression and Utilization of Circulant Matrix

In the kernel correlation filter tracker, Henriques et al. employed the form of ridge regression to solve filter coefficients. The ridge regression is a regularized least square method. Compared with traditional trackers [24–26], the ridge regression is more efficient and can obtain a simple closed-form optimal solution. The algorithm aims to learn the filter model from the training image patch x with the size of $M \times N$. At each location $(m, n)\{0, 1, \ldots, M-1\} \times \{0, 1, \ldots, N-1\}$, the training samples are generated by the circular shifts of image patch $x_{m,n}$. The Gaussian function label $y = \exp\{-((m - M/2)^2 + (n - N/2)^2/2\sigma^2)\}$. The objective function is defined as:

$$\min\left(\sum_i (f(x_i) - y_i)^2 + \lambda\|\omega\|^2\right), \quad (1)$$

where $f(x) = \omega^{\mathrm{T}} x$ and $\lambda > 0$ is a regularization parameter to control overfitting. Then, according to the computing method in [44], ω can be calculated directly as:

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

where each element x_i of the matrix X represents a sample, each element y_i of y represents the regression target value, and I is the unit matrix. In the frequency domain, Equation (2) is replaced by the complex form, which is now rewritten as:

$$\omega = \left(X^{\mathrm{H}} X + \lambda I\right)^{-1} X^{\mathrm{H}} y, \tag{3}$$

where X^{H} is the Hermitian transpose of X.

Through Equation (3), it can be found that the closed-form solution of the model parameter ω has the process of matrix inversion. The matrix inversion process will significantly reduce the tracker's speed, and become an obstacle factor in the realization of real-time target tracking. To solve this problem, the circulant matrix is introduced into the kernel correlation filter tracker. The circular shifts of image patches are used to generate the training samples. To simplify symbols, we assume that the positive sample data are $x = [x_1, x_2, x_3, \ldots, x_n]^{\mathrm{T}}$. Matrix P is called the cyclic translation operator, which can be expressed as:

$$P = \begin{bmatrix} 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}. \tag{4}$$

From Equation (4), a cyclic shift of the data can be represented as $Px = [x_{\mathrm{n}}, x_1, x_2, \ldots, x_{\mathrm{n-1}}]^{\mathrm{T}}$. All the cyclic sample data can be obtained as:

$$\{P^v x | v = 0, 1, \ldots, n-1\}, \tag{5}$$

where v is the number of cyclic shifts. All cyclic sample data can be concatenated as the circulant matrix $X = C(x)$, which is defined as:

$$X = C(x) = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_n & x_1 & x_2 & \cdots & x_{n-1} \\ x_{n-1} & x_n & x_1 & \cdots & x_{n-2} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ x_2 & x_3 & x_4 & \cdots & x_1 \end{bmatrix}. \tag{6}$$

The circulant matrix X has many special properties, which can be diagonalized by the discrete Fourier transform (DFT).

$$X = F\,diag(\hat{x}) F^{\mathrm{H}}, \tag{7}$$

where F means the DFT matrix. It is employed for transforming the sample data to the Fourier domain. F^{H} is the Hermitian transpose of F, and *diag* means the diagonal matrix. Equation (7) can be regarded as the characteristic decomposition of the cyclic matrix X. By employing the property of X in Equation (7), the covariance matrix $X^{\mathrm{H}}X$ can be computed by:

$$X^{\mathrm{H}} X = F\,diag(\hat{x}^* \odot \hat{x}) F^{\mathrm{H}}. \tag{8}$$

Then utilizing the properties of DFT and substituting Equation (8) into Equation (3), we can obtain:

$$\hat{\omega} = \frac{\hat{x}^* \odot \hat{y}}{\hat{x}^* \odot \hat{x} + \lambda}, \tag{9}$$

where $\hat{\omega}$, $\hat{x}$, and $\hat{y}$ are the DFT of ω, x, and y, respectively. $\hat{x}^*$ is the complex conjugation of $\hat{x}$, the operator $\odot$ denotes the Hadamard product, and λ is a regularization parameter for ridge regression.

3.2. Kernel Trick

The kernel trick is used to map the linear input to the nonlinear eigenspace and can improve the performance of the tracker. The classifier $f(x)$ is further transformed into:

$$f(x) = \omega^{\mathrm{T}}\varphi(x) = \sum_{i=1}^{n} \alpha_i \kappa(x_i, x_i'). \tag{10}$$

Then, for most of the kernel functions, the kernel form of the ridge regression is solved as:

$$\hat{\alpha} = \frac{\hat{y}}{\hat{k}^{xx'} + \lambda}, \tag{11}$$

where $k^{xx'}$ represents the first-row element of the kernel matrix $K = C(k^{xx'})$, and $k^{xx'}$ is computed by:

$$k^{xx'} = \exp\left\{-\frac{1}{\sigma^2}\left(\|x\|^2 + \|x'\|^2 - 2F^{-1}\left(\sum_d \hat{x}_d^* \odot \hat{x}_d'\right)\right)\right\}, \tag{12}$$

where d denotes the number of channels of the feature layer, σ is the parameter of the kernel function, and F^{-1} is inverse discrete Fourier transform (IDFT).

3.3. Fast Target Detection and Model Update

In the tracking process, the kernel correlation filter tracker extracts the features x from the image patch centered on the object location. When the next frame comes, the features z are extracted from the image patch in the same way as the previous frame. The tracker calculates the output response of the current frame in the Fourier domain as:

$$f(z) = F^{-1}(\hat{k}^{xz} \odot \hat{\alpha}), \tag{13}$$

where F^{-1} is the IDFT, $\odot$ is the dot product, and k^{xz} represents the kernel correlation of training sample x and testing sample z (i.e., features). The calculated maximum value of $f(z)$ demonstrates the object position of the current frame. To adapt to the changes of the object appearance model, the filter is updated by utilizing Equation (11) to train the α_{new}. Then, the updating process of the new filter can be linearly expressed as:

$$\hat{\alpha}^t = (1-\gamma)\hat{\alpha}^{t-1} + \gamma\hat{\alpha}_{new}, \tag{14}$$

where γ is the learning rate parameter.

4. The Proposed Tracker

In this section, based on the kernel correlation filter tracker, we integrate the dual-template, scale estimation, and adaptive template updating strategy components, and a detailed description of our proposed algorithm is provided.

4.1. The Framework of the Proposed Approach

In this paper, we propose a novel dual-template CFT. The main difference to the existing tracking algorithms is that our tracker utilizes the strategies of dual-template, discriminative appearance model, muti-peaks target re-detection, scale adaptive, and high-confidence adaptive template updating. The framework of the proposed algorithm is shown in Figure 2. First and foremost, the response fusion of color, HOG, and CN features effectively represent the target appearance. Secondly, a dual-template

strategy is presented to promote the discriminant ability of the algorithm. Next, a discriminative appearance model, a multi-peak target re-detection, and a scale adaptive scheme are integrated into the proposed tracker. Finally, the high-confidence adaptive template updating strategy is carried out to enormously decrease the risk of corruption of trackers.

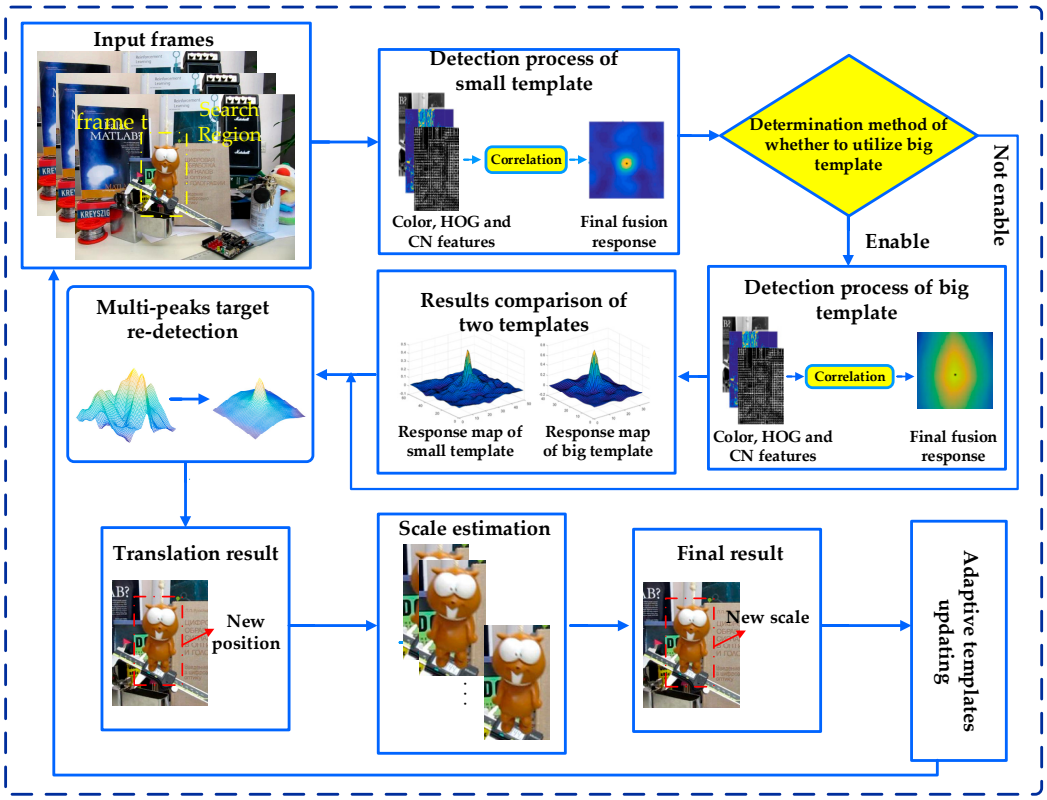


Figure 2. The framework of the proposed algorithm illustrated by the lemming sequence in frame t.

4.2. Specific Solution

4.2.1. A Dual-Template Strategy

Due to the limitation of a sole filter template, a dual-template strategy is utilized. According to the kernel form of ridge regression in Equation (11) and the object information given in the first frame, the two filter templates with different sizes α_b and α_s are initialized as:

$$\hat{\alpha}_{type} = \frac{\hat{y}_{type}}{\hat{k}^{xx'}_{type} + \lambda} \quad type = s \; or \; b, \tag{15}$$

where α_b and α_s represent the big and small templates, respectively. Symbol $\hat{}$ represents the Fourier transform, y is the expected regression label, and λ is the regularization parameter. $k^{xx'}$ can be calculated as Equation (12).

In the subsequent frames, assuming that the detection center is the target position of the previous frame and that the size is the small template of the previous frame, the image patch z at the same location in the previous frame is intercepted according to the size of the small template α_s. The target region features are extracted and the output response α_s is calculated as Equation (16):

$$f_{type}(z) = F^{-1}\left(\hat{k}^{xz} \odot \hat{\alpha}_{type}\right) \quad type = s \; or \; b, \tag{16}$$

where the interpretation of F^{-1}, $\odot$ and k^{xz} is the same as that of Equation (13). r_s and r_b denote the maximum value of $f_s(z)$ and $f_b(z)$, respectively. The object position of the current frame can be obtained by using the r_s or r_b.

When r_s is bigger than a predefined threshold Th, the position corresponding to r_s is taken as the prediction position of the target in the current frame. The small template α_s can accurately predict the target position. When r_s is smaller than a predefined threshold Th, we use the big template α_b to calculate response $f_b(z)$ according to Equation (16) and can obtain the maximum response value r_b. The final prediction position of the object in the current frame is defined as:

$$p(x,y) = \begin{cases} L(f_s(z)) & r_b \le r_s \\ L(f_b(z)) & r_b > r_s \end{cases}, \tag{17}$$

where $p(x,y)$ is the prediction position of the object in the current frame. The prediction positions $L(f_s(z))$ and $L(f_b(z))$ are calculated by utilizing the small template and the big template, respectively.

4.2.2. A Discriminative Appearance Model

To effectively distinguish the foreground target from the surrounding background, the proposed tracker uses the Bayesian classifier based on a histogram to establish the color model of the target. For the given foreground target region O and its surrounding background region B, the probability that the pixel x belongs to the region O can be computed by:

$$P(x \in O|O,B,b_x) = \frac{P(b_x|x \in O)P(x \in O)}{\sum\limits_{\Omega \in \{O,B\}} P(b_x|x \in \Omega)P(x \in \Omega)}, \tag{18}$$

where b_x represents the bin index b assigned to the color components of the pixel point x. The probabilities $P(b_x|x \in O)$ and $P(b_x|x \in B)$ can be calculated as:

$$\begin{cases} P(b_x|x \in O) = \frac{H_O(b_x)}{|O|} \\ P(b_x|x \in B) = \frac{H_B(b_x)}{|B|} \end{cases}, \tag{19}$$

where $H_O(b_x)$ and $H_B(b_x)$ represent the histograms with the range b of the regions O and B, respectively. $|O|$ and $|B|$ represent the areas of foreground target and the background, respectively. Meanwhile, The probabilities $P(x \in O)$ and $P(x \in B)$ can be computed by:

$$\begin{cases} P(x \in O) = \frac{|O|}{|O|+|B|} \\ P(x \in B) = \frac{|B|}{|O|+|B|} \end{cases}. \tag{20}$$

Then, Equation (18) can be simplified as:

$$P(x \in O|O,B,b_x) = \frac{H_O(b_x)}{H_O(b_x) + H_B(b_x)}. \tag{21}$$

For instance, the color histograms of $H_O(b_x)$, $H_B(b_x)$, and $P(x \in O|O,B,b_x)$ of lemming sequence in frame 138 is shown in Figure 3. We can intuitively distinguish the foreground target region O and background region B by the color histogram of $P(x \in O|O,B,b_x)$.

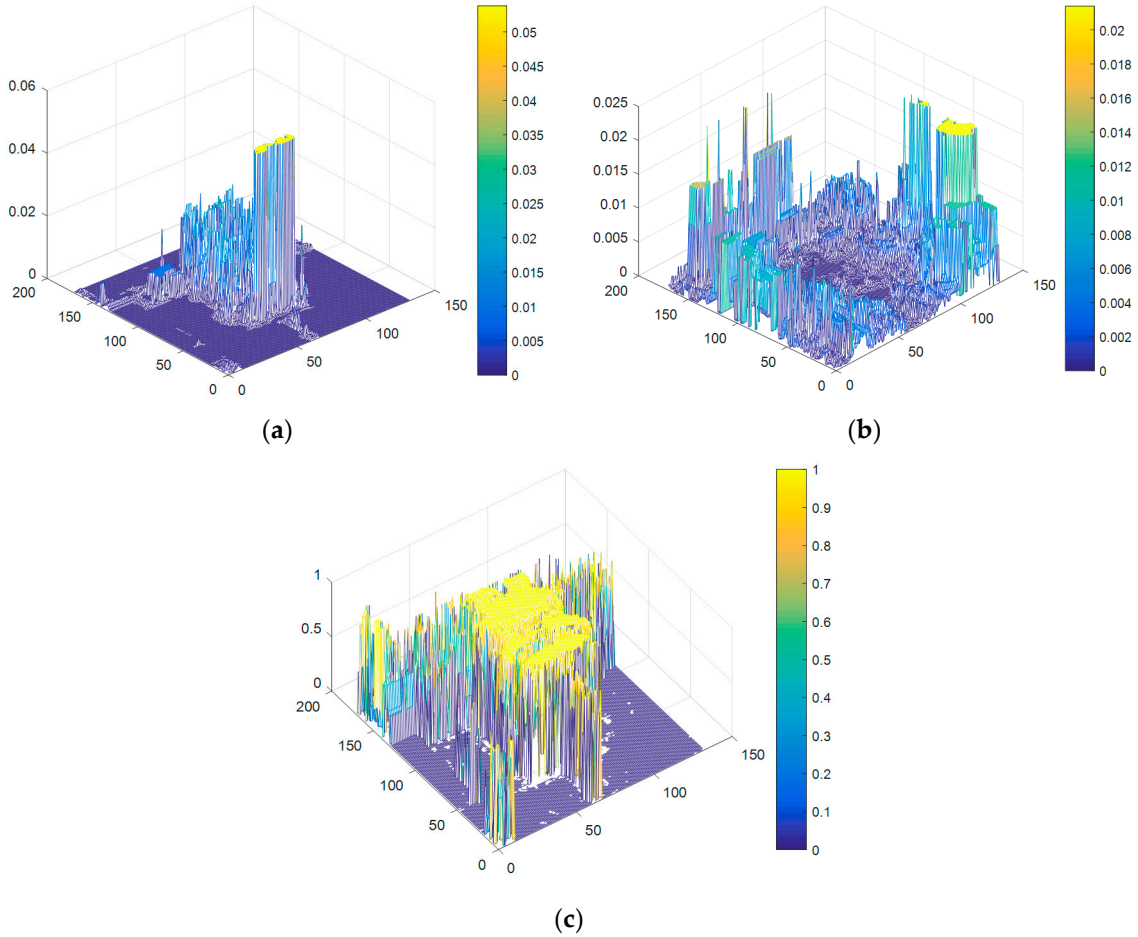


Figure 3. The color histograms of the lemming sequence in frame 138: (**a**) The color histogram of $H_O(b_x)$; (**b**) The color histogram of $H_B(b_x)$; (**c**) The color histogram of $P(x \in O|O, B, b_x)$

Subsequently, assuming that the detection center is the predicted target position of the previous frame. According to Equation (21), the probability $P_t[x(i,j)]$ of the pixel $x(i,j)$ belongs to the target region O in frame t can be computed. Then, the response of the fore-background color model f_{color} can be computed utilizing the average value of an integral image from the $P_t[x(i,j)]$. The response f_{color} can be denoted as:

$$f_{\text{color}} = \frac{1}{|M|} \sum_{X(i,j) \in M} P_t[x(i,j)], \tag{22}$$

where M represents the search region centered on the target center position of the previous frame, $|\bullet|$ defines the area of the region.

The tracking performance is sensitive to the changes of scenarios, consequently, we require that the parameters of the tracking model can change adaptively according to the change of the tracking scenarios. To improve the reliability of the tracking model, a feature fusion weight calculation method is utilized by combining peak-to-sidelobe ratio (PSR) and average correlation peak energy (APCE) [45] confidence index to adjust the changes of the challenge scenarios. The PSR is defined as the ratio of the peak intensity of the main lobe to that of the side lobe, which is:

$$PSR = \frac{\max(y_P) - u_\Phi(y_P)}{\sigma_\Phi(y_P)}, \tag{23}$$

where y_P is the confidence map, subscript $\Phi = 0.10$ is a constant, $u_\Phi(\cdot)$ and $\sigma_\Phi(\cdot)$ are the mean value and standard deviation of the confidence map, respectively.

Next, the APCE is defined as:

$$APCE = \frac{|F_{\max} - F_{\min}|^2}{mean\left(\sum_{w,h}(F_{w,h} - F_{\min})^2\right)}, \tag{24}$$

where $|\cdot|$ represents the Euclidean distance, $F_{\max}$, $F_{\min}$, and $F_{w,h}$ denote the maximum, minimum, and the w-th row h-th column elements of the filter response matrix with the size of $W \times H$, respectively. The numerator is the square of the difference between the $F_{\max}$ and the $F_{\min}$, and the denominator is the square mean of the difference between each element and the minimum value in the filter response matrix. The larger the PSR and APCE, the less occlusion and noise information in the filter template.

In various tracking challenge scenarios, the responses of the two features will change, and the confidence degree will fluctuate to different degrees. In this paper, by combining PSR and APCE, a feature fusion weight strategy is defined as:

$$\begin{cases} \psi_i = \mu_i \times PSR_i + \upsilon_i \times APCE_i \\ \zeta_i = \frac{\psi_i}{\sum_i \psi_i} \end{cases} \quad i = 1, 2, 3, \tag{25}$$

where ζ_1, ζ_2, and ζ_3 represent the adaptive fusion weight of the color, HOG, and CN features, ψ_1, ψ_2, and ψ_3 represent the confidence factor of the color, HOG and CN features, and μ_1, μ_2, and μ_3 are the PSR control factor of the color, HOG, and CN features, υ_1, υ_2, and υ_3 are the APCE control factor of the color, HOG, and CN features. After calculating the fusion weight of the three features, the final fusion response is obtained by combining the responses of the color, HOG, and CN features. The final fusion response is defined as:

$$f_{\text{fusion}} = \zeta_1 f_{\text{color}} + \zeta_2 f_{\text{HOG}} + \zeta_3 f_{\text{CN}}, \tag{26}$$

where f_{fusion} represents the final fusion response, ζ_1, ζ_2, and ζ_3 represent the adaptive fusion weight of color, HOG, and CN features. The responses f_{HOG} and f_{CN} can be calculated by Equation (16), and the response f_{color} can be calculated by Equation (22).

As an example, the response fusion of the lemming sequence is shown in Figure 4, the proposed tracker extracts the color, HOG, and CN features to obtain the response f_{color}, f_{HOG}, and f_{CN}, respectively. The maximum value position of the fusion response f_{fusion} is the predicted target center. Therefore, by fusing responses of color, HOG, and CN features, the proposed tracker owns the stronger target representation ability.

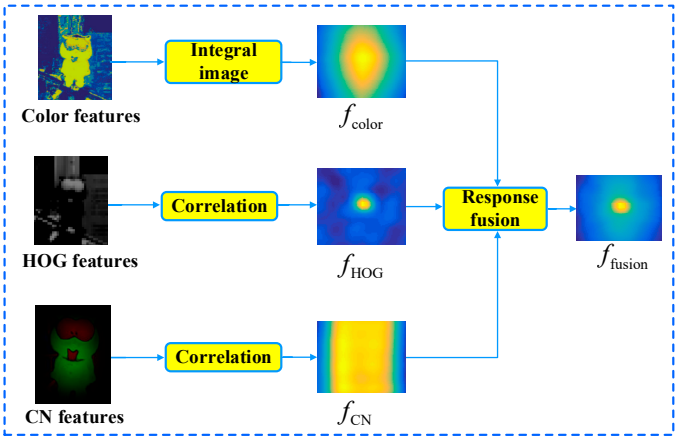


Figure 4. The response fusion of the lemming sequence.

4.2.3. A Multi-Peaks Target Re-Detection Technique

In the tracking process, the target may undergo interference of similar appearance, which will lead to inaccurate detection. Meanwhile, partial or full occlusion for a long time may further exacerbate the drift and the corruption of the target appearance model. The disturbance of similar objects and occlusion will fail to track the target successfully. Besides, the multiple peaks perhaps appear in the response map in this interference environment. Therefore, a multi-peaks target re-detection technique is proposed to redetect the tracked target and further improve tracking precision.

First, a spatial weight w is presented to computer drastic changes in consecutive frames during the tracking process, and the w can be computed by:

$$w^t = \exp\left(-\kappa^t \|C^t - C^{t-1}\|\right), \tag{27}$$

where C denotes the predicted target position in the frame t and $\|\bullet\|$ represents the Euclidean distance, t is the frame index. The spatial weight factor κ can be computed as:

$$\kappa = \frac{1}{2 * (L_{diag}\text{^}2)}, \tag{28}$$

where L_{diag} denotes the diagonal length of the target size. The spatial weight w is larger, the inter-frame movement is more drastic. In this circumstance, the target may be occluded or interfere with similar objects. When the w in frame t is less than predefined threshold $Thsw$, the multi-peaks target re-detection technique is activated.

Then, the local maxima locations with the multiple peaks in the fusion response map should be found out. A set of points M with these local maxima locations is defined as:

$$M = \{(i,j) \big| f_{\text{fusion}}(i,j) > \mu \max(f_{\text{fusion}})\}, \tag{29}$$

where μ represents the control parameter.

The responding image patches centered at those local maxima locations are redetected. Subsequently, the maxima response values of the local maxima locations are computed according to Equation (28) and are denoted as a vector score. Besides, the spatial weights of the local maxima locations can be computed by Equation (27) and are denoted as a vector w. The maximum confidence value $multi\_peak_{\max}$ of the local maxima locations can be formulated as:

$$multi\_peak_{\max} = \max(\mathbf{score} \odot \mathbf{w}), \tag{30}$$

where $\odot$ represents the Hadamard product operator. Finally, the final maximum response value $r_{\max}$ can be obtained by comparing the values of $\max(f_{\text{fusion}})$ and $multi\_peak_{\max}$, which can be written as:

$$r_{\max} = \begin{cases} muti\_peak_{\max}, & \max(f_{\text{fusion}}) < muti\_peak_{\max} \\ \max(f_{\text{fusion}}), & \max(f_{\text{fusion}}) \geq muti\_peak_{\max} \end{cases}. \tag{31}$$

The optimal target position can be estimated by the corresponding location of $r_{\max}$.

4.2.4. A Scale Adaptive Scheme

Owing to the drawbacks of the fixed search range of the object, we employ a scale adaptive scheme. To appraise the scale size of the object, the one-dimensional scale filter described in [46] is utilized to solve the scale problem. Firstly, the scale filter F_d is defined as:

$$\begin{aligned} F^d_{scale} &= \frac{Y_{scale} \odot (X^d_{scale})^*}{\sum_{d=1}^{D} X^d_{scale} \odot (X^d_{scale})^* + \lambda'} \\ &= \frac{A^d_{scale}}{B_{scale} + \lambda'}, \end{aligned} \tag{32}$$

where X_{scale} and Y_{scale} are the Fourier transform of the training sample x_{scale} and the expected regression label y_{scale}, respectively. D is the channel number of feature maps x_{scale}, and λ' is the regularization parameter. Then, after estimating the object position, we extract feature maps z_{scale} from the new predicted position of the object. The correlation response $f_{scale}(z)$ is computed by:

$$f_{scale}(z) = F^{-1}\left\{\frac{\sum_{d=1}^{D}\left(A_{scale}^{d}\right)^{*} \odot Z_{scale}^{d}}{B_{scale} + \lambda'}\right\}, \tag{33}$$

where Z_{scale} is the Fourier transform of the test sample z_{scale}. The maximum value of $f_{scale}(z)$ is the object scale. Then, to robustly learn the scale filter, the filter model is updated as:

$$\begin{cases} A_{scale,t}^{d} = (1-\eta_{scale})A_{scale,t-1}^{d} + \eta_{scale} Y_{scale} \odot \left(X_{scale,t}^{d}\right)^{*} \\ B_{scale,t} = (1-\eta_{scale})B_{scale,t-1} + \eta_{scale} \sum_{d=1}^{D} X_{scale,t}^{d} \odot \left(X_{scale,t}^{d}\right)^{*} \end{cases}, \tag{34}$$

where η_{scale} is a learning rate parameter, and the numerator $A_{scale,t}^{d}$ and $A_{scale,t-1}^{d}$ are the d-th channel filter template in the frames t and t-1, respectively. Besides, the denominators $B_{scale,t}$ and $B_{scale,t-1}$ are the templates in the frames t and t-1, respectively. The scale adaptive scheme further enhances the robustness of the tracker.

4.2.5. A High-Confidence Template Updating Technique

To further improve the reliability of the tracking model, APCE is taken as a judgment basis of whether the filter model needs to be updated. Because APCE measures the fluctuation degree of the response map, when the target is appearing in the search region, APCE will become larger, indicating that the tracking model almost has no disturbance. Otherwise, if the target is encountering challenges (e.g., occlusion or the interference of similar objects), APCE will become smaller and the fluctuation degree of the response map is fierce. In the situation of occlusion or the interference of similar objects, the filter model should not be updated.

An example of the APCE being sensitive to the visual tracking environment on the girl sequence is shown in Figure 5. For instance, we notice that APCE drops rapidly when occlusion occurs at frame 107. When the occlusion disappears at frame 120, APCE begins to increase. Subsequently, APCE drops rapidly when the target is disturbed by a similar object of the surrounding background in frame 878. Then, APCE begins to increase when the distractor disappears at frame 896 in the video sequence. According to the criteria in [41], the template will be not updated until the occlusion or similar object disappears. The evaluation criterion of the high-confidence template updating is given as:

$$\Omega = \begin{cases} 1, & if\ APCE_t > \frac{\sum_{i=t-m}^{t-1} APCE_i}{m} \times \xi \\ 0, & else \end{cases}, \tag{35}$$

where $APCE_t$ is the APCE of the frame t, proportion factor ξ is a constant, Ω means the control coefficient of model updating. Specifically, if $\Omega = 0$, it illustrates that the confidence degree of the model is low and the object is occluded or missing, the model will not be updated. If $\Omega = 1$, it illustrates that the confidence degree of the model is high, and the updating of small and big templates can be mathematically expressed as:

$$\hat{\alpha}_{type}^{t} = (1 - \Omega \cdot \tau\varepsilon_t)\hat{\alpha}_{type}^{t-1} + \Omega \cdot \tau\varepsilon_t \hat{\alpha}_{type}^{new} \quad type = s\ or\ b, \tag{36}$$

where $\varepsilon_t = APCE_t$ is the learning rate parameter in frame t, and τ is the control factor.

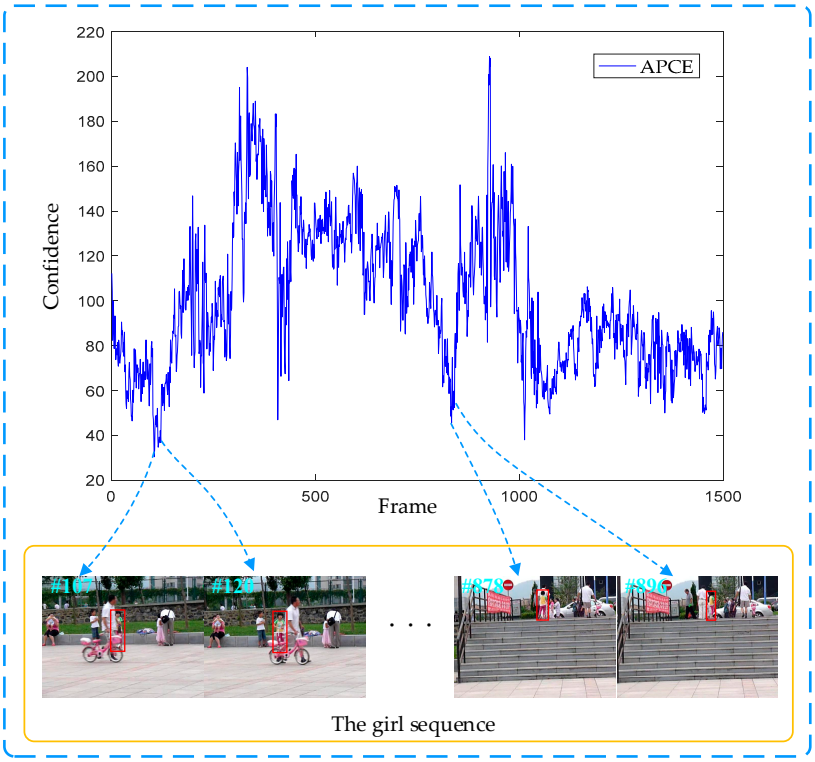


Figure 5. An example of the average correlation peak energy (APCE) being sensitive to the visual tracking environment on the girl sequence in the VOT2016 dataset.

5. Experimental Results and Analysis

In this section, a comprehensive experimental evaluation of various trackers is carried out. The specific process description is provided below.

5.1. Experiment Setup

The native MATLAB R2016b without optimization was utilized to implement the proposed tracking algorithm. The experiments were performed on an Intel (R) Core (TM) i7-9750H CPU (2.60 GHz) laptop with 16 GB main memory. The initial position and size of the target based on a bounding box centered on the target in the first frame were given in advance. The HOG and CN features were employed in the proposed tracking algorithm. In this paper, the tracking benchmark databases OTB2013, OTB2015, VOT2016, and LaSOT [47] were utilized for simulation experiments. In the proposed algorithm, in the fore-background color model, the color features were RGB and the number of histogram bins was 32. The cell size of the HOG features was 4×4 and the channel number of HOG features was 31. Then, the channel number of CN was 11. The spatial regularization parameter λ was set to 0.01. The small and large templates adopted 1.2- and 2.5-times the object size, respectively. The dual-template threshold was set to 0.65. In the scale filter, the scale level S was set to 33, the scale factor a was set to 1.02, and the feature of each scale level used the feature vector, which was obtained by 31-D HOG features, the learning rate of the scale filter was $\eta_{scale} = 0.025$. The proportion factor ξ was set to 0.707, and the control factor τ was set to 0.0003. Furthermore, the same parameter values and initialization were utilized for all the video sequences.

5.2. Compared Trackers

In order to comprehensively evaluate the proposed tracking algorithm, 27 state-of-the-art trackers were introduced for comparison purposes. These trackers can be divided into two categories: (a) 16 state-of-the-art handcrafted features-based trackers, including AO-CF, MMWCF, spatial-temporal

regularized correlation filters (STRCF) [48], parallel tracking and verifying (PTAV) [49], channel and spatial reliability correlation filter trackers (CSRDCF) [50], fDSST, spatially regularized correlation filters based on adaptive decontamination (SRDCFdecon) [51], sum of template and pixel-wise learners (Staple) [52], EdgeBox tracker (EBT), LCT, SRDCF, KCF, scale adaptive with multiple features tracker (SAMF) [53], DSST, CN, and CSK, (b) 11 deep features-based trackers including SPLT, siamese instance search tracker (SINT) [54], efficient convolution operators (ECO) [55], continuous convolution operator tracker (CCOT) [56], MDNET, SiamFC, structured siamese network (StructSiam) [57], dynamic siamese network (DSiam) [58], tracker based on context-aware deep feature compression with multiple auto-encoders (TRACA) [59], end-to-end representation learning for correlation filter (CFNet) [60], and HCF.

5.3. Experimental Results on the OTB2013 and OTB2015 Benchmark Databases

To gain more insights into the effectiveness of the proposed tracker, we evaluated and analyzed the overall performance, attribute-based evaluation, and qualitative comparison of ours and the other 12 trackers on the OTB2013 and OTB2015 benchmark databases. In the experiment, we compared our tracker against nine state-of-the-art HFTs: CSK, CN, DSST, SAMF, KCF, Staple, SRDCFdecon, fDSST, and AO-CF. Then, three DFTs, ECO, CCOT, and HCF, were also compared with our tracker.

5.3.1. The OTB2013 and OTB2015 Benchmark Databases

The OTB2013 database has 51 video sequences. Subsequently, the OTB2015 tracking benchmark database was expanded to 100 video sequences, even with some long sequences.

The OTB2015 dataset evaluates tracking algorithms in two aspects: precision plot and success plot. As shown in Figure 6, the precision plot represents different distance precisions (DP) under different thresholds for the target center location error (CLE), where CLE is the average Euclidean distance between the target prediction center location and the ground truth. DP is computed as the percentage of frames in the videos where the CLE is smaller than 20 pixels. The success plot represents different average overlap precisions (OP) under different thresholds for the overlap success score S, where $S = |R^{tr} \cap R^{gt}| / |R^{tr} \cup R^{gt}|$, R^{tr} denotes the prediction box, R^{gt} denotes the ground truth, $\cap$ and $\cup$ denote the intersection and union of two areas, and $|\cdot|$ denotes the number of pixels in the area. OP is the percentage of frames where the overlap between the prediction box and the ground truth exceeds 0.5. The area under the curve (AUC) was computed to evaluate the proposed tracker's performance.

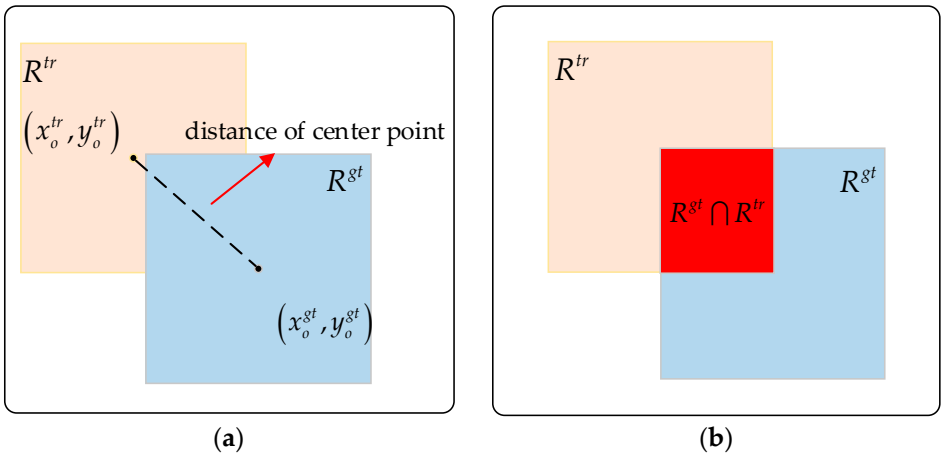
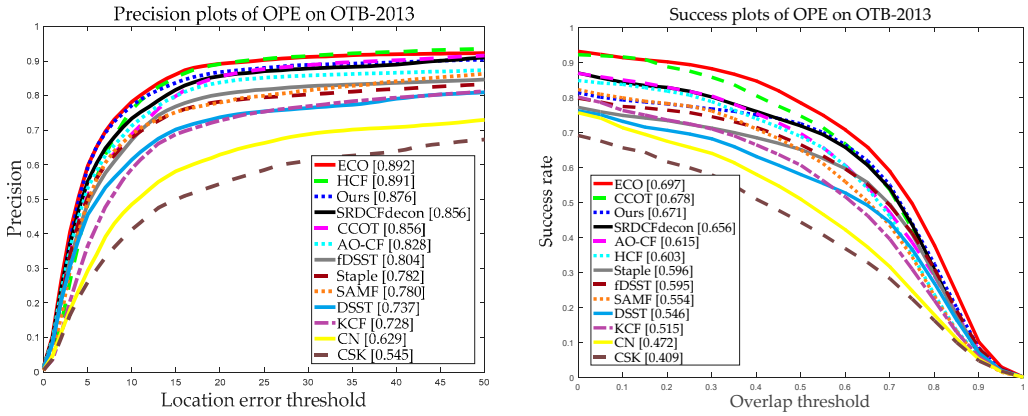


Figure 6. Evaluation criterion. (**a**) Distance of the center point. (**b**) Intersection over the union.

5.3.2. Overall Performance Evaluation

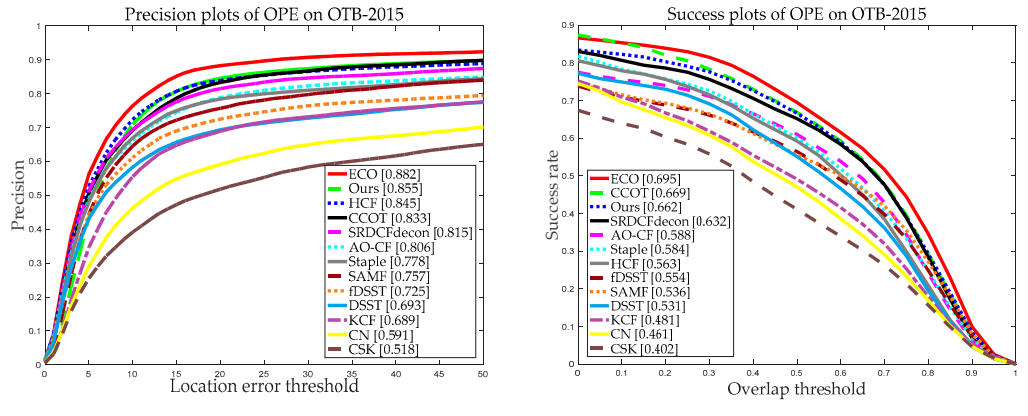
Figures 7 and 8 demonstrate the graphical performance results of our and the other 12 trackers using the two datasets. As we can see from Figures 7 and 8, our tracker achieved precision scores

of 0.876 and 0.855 and success scores of 0.671 and 0.662 for the OTB-2013 and OTB-2015 datasets, and the precision scores and success scores of our tracker ranked third of all 13 trackers. Compared with the classical KCF, our tracker made about 21.77% and 38.67% improvement in terms of precision scores and success scores on the OTB-2015 dataset. Besides, from Figures 7 and 8, we can see that our tracker performed favorably against nine state-of-the-art trackers with handcrafted features, compared with the ECO, CCOT, and HCF based on deep features, our tracker achieved comparable results. The running speed is also an important evaluation index for measuring the performance of the trackers. Tables 1 and 2 demonstrate the average tracking speed (FPS) comparison on OTB2013 and OTB2015. Particularly, due to the large computation overhead and complexity, the average FPS of ECO, CCOT, and HCF was approximately 1 FPS. Overall, the ECO, CCOT, and HCF performed well in terms of precision scores and success scores but at a lower tracking speed. Our tracker achieved an average tracking speed of 21.536 and 20.103 FPS on the OTB2013 and OTB2015, respectively, which is significantly faster than DFTs. Meanwhile, our tracker can meet real-time requirements.



(**a**) (**b**)

Figure 7. Precision and success plots on the dataset OTB-2013 using one-pass evaluation (OPE). (**a**) Precision plots of OPE on OTB-2013; (**b**) Success plots of OPE on OTB-2013. In the legends, the average distance precisions (DP) rates at a threshold of 20 pix and area under the curve (AUC) scores at a threshold of 0.5 are reported for the precision plots and success plots, respectively.



(**a**) (**b**)

Figure 8. Precision and success plots on the dataset OTB-2015 using OPE. (**a**) Precision plots of OPE on OTB-2015; (**b**) Success plots of OPE on OTB-2015. In the legends, the average DP rates at a threshold of 20 pix and AUC scores at a threshold of 0.5 are reported for the precision plots and success plots, respectively.

Table 1. The speeds in comparison of different trackers on the OTB-2013 dataset.

	Ours	ECO	CCOT	HCF	fDSST	SRDCFdecon	Staple	AO-CF	SAMF
Avg.FPS	21.536	1.271	0.524	0.740	71.079	2.203	83.484	52.107	18.469

Table 2. The speeds in comparison of different trackers on the OTB-2015 dataset.

	Ours	ECO	CCOT	HCF	fDSST	SRDCFdecon	Staple	AO-CF	SAMF
Avg.FPS	20.103	1.245	0.407	0.720	66.353	2.195	78.356	48.936	16.120

5.3.3. Attribute-Based Evaluation

In the two datasets, the 11 diverse challenging scenarios (as shown in Figure 1a) significantly affected the performance of the trackers. As these challenging scenarios come up from time to time, it may lead to the corruption of the filter model and eventually to track failure. Figure 9 demonstrates the attribute-based precision evaluation of our and the other 12 trackers using the OTB2015 dataset. As we can see from Figure 9, in terms of precision plots, our tracker ranked second for 4 out of the 11 challenging scenarios and ranked third for 5 out of the 11. Therefore, among all 11 challenging scenarios, our tracker achieved a comparative tracking performance. Compared with three DFTs, the performance of our tracker was comparable in most cases.

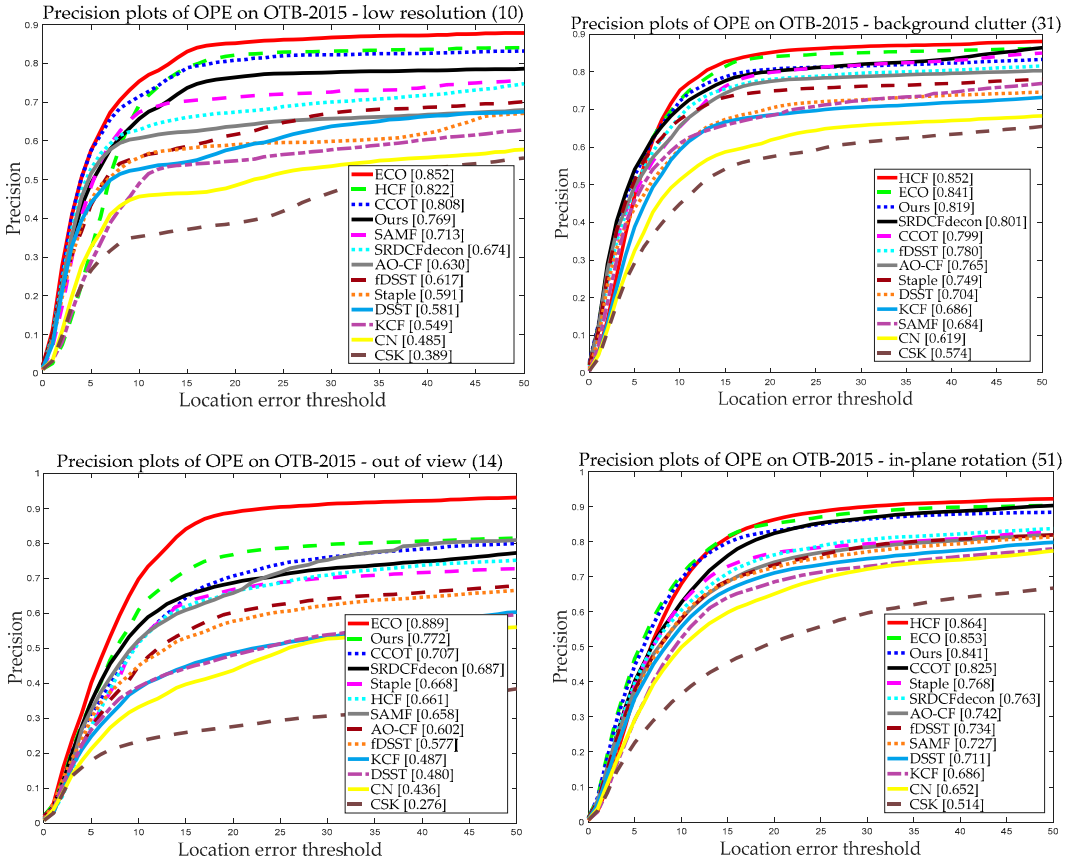


Figure 9. *Cont.*

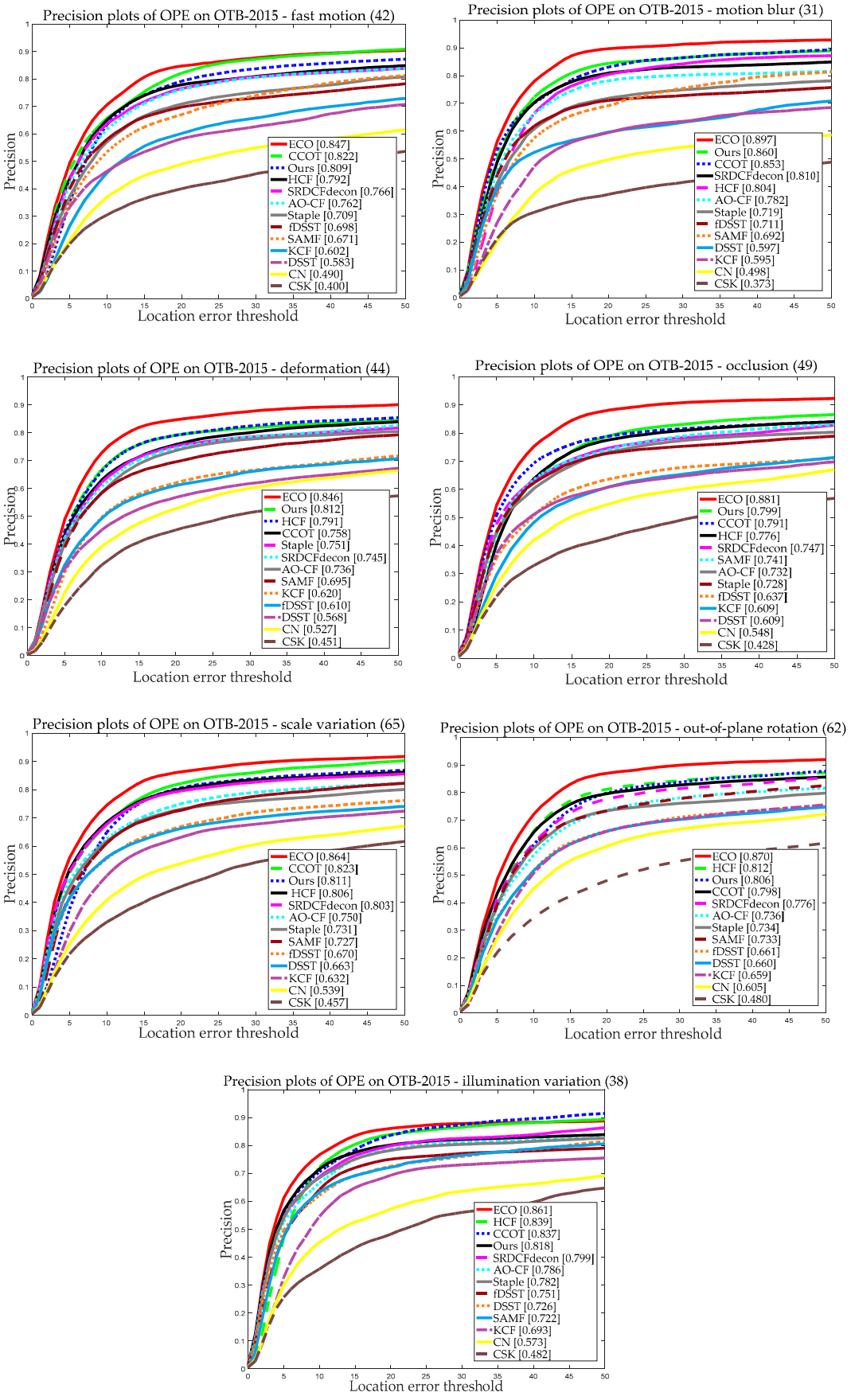


Figure 9. Precision plots for the compared trackers on OTB-2015 with 11 attributes using OPE.

5.3.4. Qualitative Comparison

In a comparative experiment, to better show the tracking performance, Figure 10 demonstrates the qualitative evaluation results of our, the other nine HFTs, and three DFTs. The frames from six representative video sequences with Bird1, Ironman, Diving, Jump, Human3, and Motorrolling were illustrated. As shown in Figure 10, at the beginning of video sequence frames, most tracking algorithms could keep up with the target. The targets of the Bird1 and Ironman video sequences mainly undergo the challenges of fast motion and motion blur, some trackers may lose target or pull-in unnecessary background information due to the fixed detection range, our approach can track the target successfully by utilizing the dual-template method. The main challenge of the Human3 and Diving video sequences is scale change; some trackers without scale estimation cannot deal with this scenario well, and the proposed method can overcome the limitation of large scale variations by using the scale adaptive scheme. Jump and Motorrolling video sequences are accompanied by occlusion and deformation, which may cause some trackers to drift; our algorithm can deal with occlusion better owing to the adaptive template updating. In general, the proposed approach can keep track of the target in the whole video sequence.

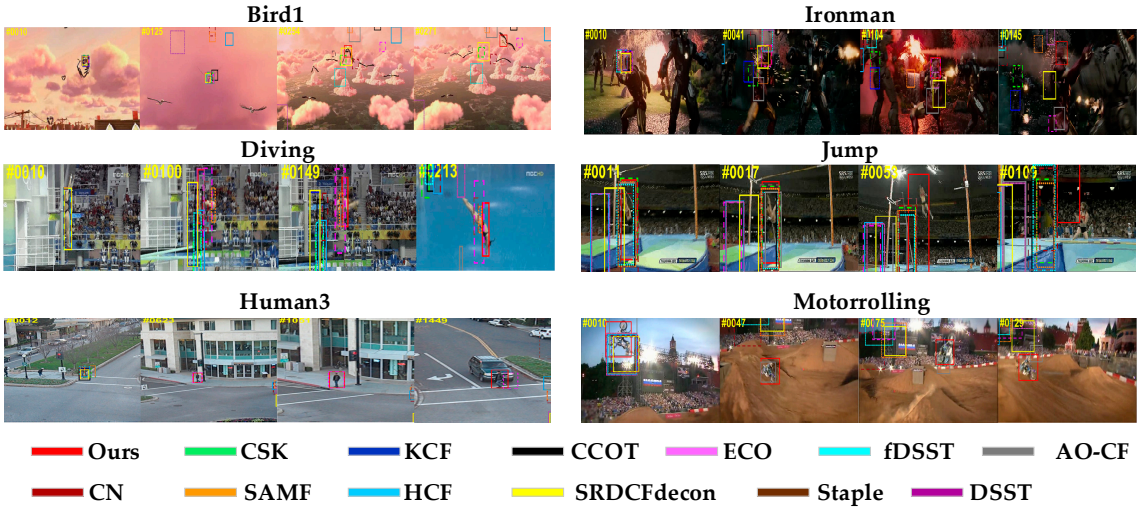


Figure 10. Qualitative evaluation of six video sequences (i.e., Bird1, Ironman, Diving, Jump, Human3, and Motorrolling). We show the results of ours and the other 12 trackers, including nine HFTs and three DFTs with different colors (our results are in red).

5.4. Experimental Results on the VOT2016 Benchmark Dataset

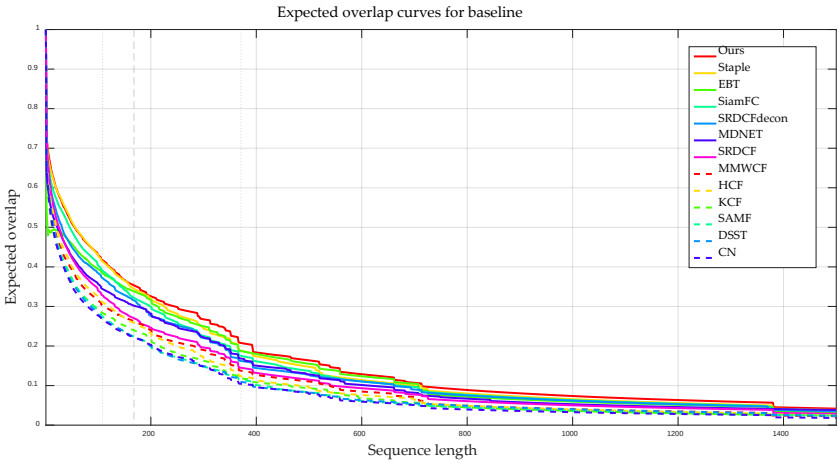
To gain more insights into the effectiveness of the proposed algorithm, we further exhibited the quantitative and qualitative comparison on the VOT2016 database. In the experiment, we compared our tracker against nine state-of-the-art HFTs (CN, DSST, SAMF, SRDCF, KCF, Staple, EBT, SRDCFdecon, and MMWCF) and three DFTs (SiamFC, MDNET, and HCF).

5.4.1. The VOT2016 Benchmark Database

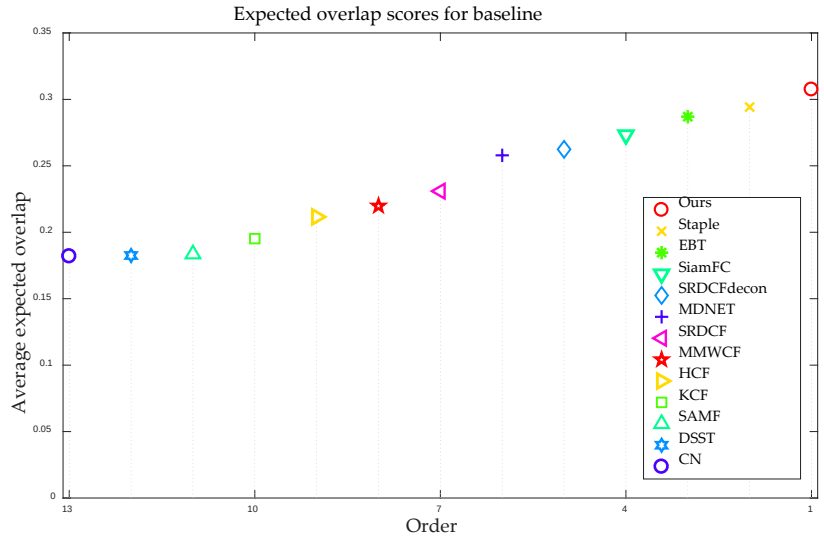
The VOT2016 database consists of 60 challenging sequences and is relatively a difficult sequence set in contrast to OTB2015. The VOT2016 dataset has five common video challenge scenarios: camera motion, illumination variation, motion change, occlusion, and size change. Two basic tracking evaluation indexes (i.e., accuracy and robustness) were utilized in this work. The accuracy denotes the average overlap rate of the tracking success state, while the robustness was measured by the total number of tracking failures. Additionally, the values of per-frame accuracy and robustness were combined as the expected average overlap (EAO) to measure the overall performance.

5.4.2. Quantitative and Qualitative Comparison

To further assess the effectiveness and accuracy of our tracker, Figure 11 demonstrates the comparison result of EAO under the different trackers. Figure 11a shows the ranking results of EAO and Figure 11b exhibits the EAO curves of 13 trackers over different sequences lengths. From Figure 11, we can see that the proposed tracker obtained an EAO score of 0.308 and ranked first among all the trackers. Table 3 presents the EAO score of each approach and the best three trackers are shown in red, blue, and green bold fonts. Particularly, compared with the Staple and EBT approach, our tracker made about 4.55% and 5.84% improvement, respectively. Table 4 demonstrates the accuracy evaluation of ours and other 12 state-of-the-art trackers on different challenging attributes. As shown in Table 4, our tracker obtained a significant gain in accuracy and achieved the top performer except for the camera motion and motion change. Overall, our approach performed better than the nine HFTs and three DFTs.



(**a**)



(**b**)

Figure 11. Comparison of the expected average overlap (EAO) for different trackers. (**a**) Ranking results of EAO, where the better trackers are located at the top and right. (**b**) EAO curves of trackers over different sequences lengths.

Table 3. The EAO performance comparison of ours and seven other state-of-the-art trackers on the VOT2016 dataset. The best two results are marked in red and blue bold fonts.

	Ours	SiamFC	EBT	MDNET	HCF	SRDCFdecon	Staple	SRDCF
EAO	**0.308**	0.277	0.290	0.258	0.220	0.262	**0.294**	0.231

Table 4. Accuracy evaluation of ours and seven other state-of-the-art trackers on different challenging attributes of the VOT 2016 benchmark database. The best two results are marked in red and blue bold fonts, respectively.

	Ours	SiamFC	EBT	MDNET	HCF	SRDCFdecon	Staple	SRDCF
Camera motion	**0.560**	**0.563**	0.491	0.547	0.438	0.530	0.551	0.551
Illumination change	**0.718**	0.672	0.407	0.639	0.462	**0.714**	0.709	0.680
Motion change	**0.528**	**0.530**	0.439	0.508	0.423	0.466	0.507	0.486
Occlusion	**0.499**	0.448	0.375	**0.491**	0.433	0.434	0.433	0.408
Size change	**0.528**	**0.514**	0.356	0.511	0.354	0.490	0.511	0.478
Empty	**0.597**	**0.586**	0.518	0.563	0.502	0.524	0.584	0.580
Mean accuracy	**0.572**	**0.552**	0.431	0.543	0.435	0.526	0.549	0.530
Weighted mean accuracy	**0.563**	**0.549**	0.453	0.537	0.437	0.509	0.540	0.523

Figure 12 shows qualitative evaluation results of our and the other 12 trackers on six video sequences, i.e., Bolt1, Soldier, Fish2, Fish4, Motocross2, and Gymnastics4. For instance, the Bolt1 and Soldier video sequences experience size change from large to small. When the object becomes smaller and the search range is the same, it will bring in considerably background noises and disturbance and make the drift; when the object becomes larger, the search range can only contain the local information of the object. Therefore, the fixed search range of object pollutes the object template and reduces the robustness of the tracker. The proposed approach can deal with size change well by the dual-template method and scale adaptive scheme. The Fish2 and Fish4 video sequences have the scenarios of moving swiftly or large movement range; due to the small detection range, most trackers will not be able to catch up with the target or even lose the target. Our tracker can track the object successfully due to the dual-template method. The Motocross2 and Gymnastics4 video sequences are accompanied by motion change and occlusion. The parameters of the tracking model are generally set as a fixed value in many trackers, which may cause the models to drift or even fail. Nevertheless, our algorithm can deal with the challenges due to the high-confidence adaptive template updating. As shown in Figure 12, the experimental results demonstrate that the proposed algorithm had the best tracking performance for the six video sequences.

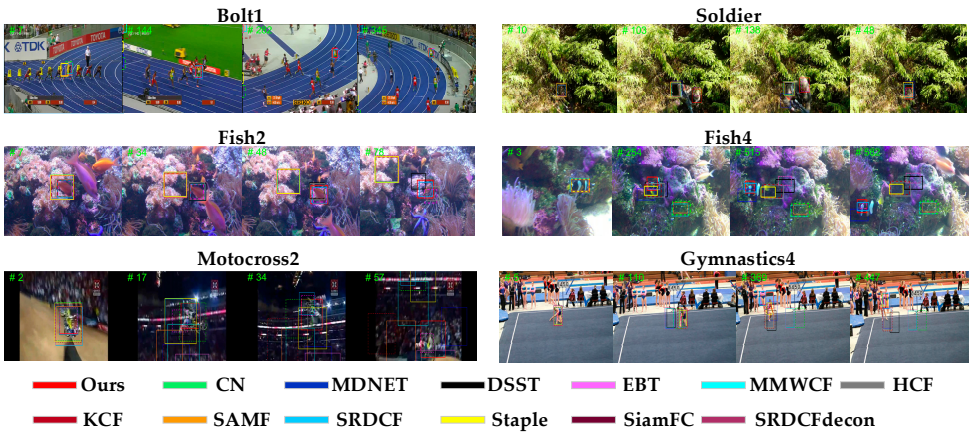
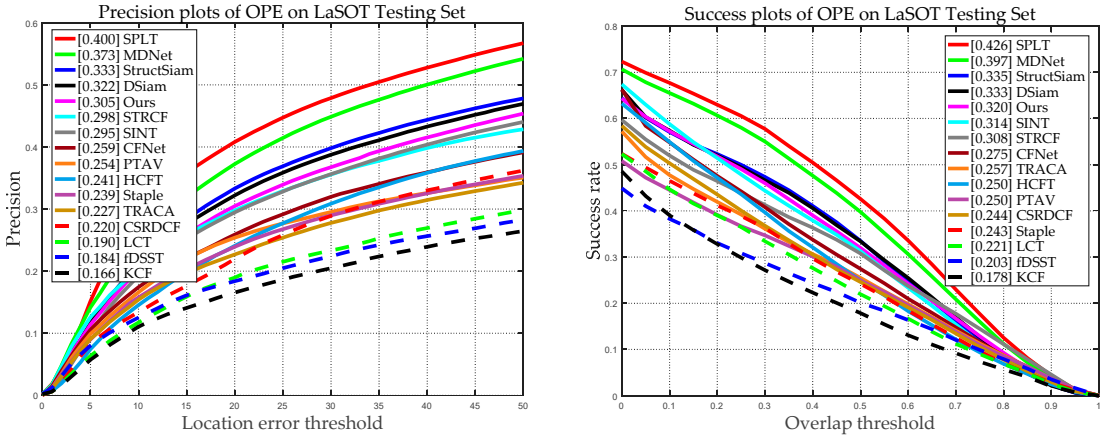


Figure 12. Qualitative evaluation on six video sequences (i.e., Bolt1, Soldier, Fish2, Fish4, Motocross2, and Gymnastics4) in the unsupervised experiment. We show the results of ours and the other 12 trackers, including 9 handcrafted features-based trackers and 3 deep features-based trackers with different colors (our results are in red).

5.5. Experimental Results on the LaSOT Benchmark Dataset

The LaSOT dataset is a high-quality long-term tracking dataset, which is much closer to realistic applications. The LaSOT dataset contains 1400 video sequences with more than 3.5M frames in all, and the average video length of the dataset is more than 2500 frames. Additionally, the number of object categories in LaSOT dataset is 70, which is two times more than the existing dense benchmark. The LaSOT dataset has 14 video challenges (i.e., aspect ratio change, deformation, fast motion, full occlusion, viewpoint change, and out-of-view). Besides, the most challenges of LaSOT dataset derive from the wild. During the tracking process, the trackers will encounter many complicated challenge scenarios. Similar to the OTB2013 and OTB2015 datasets, the LaSOT dataset evaluates tracking algorithms with precision and success plots.

In the experiment, we compared our tracker with the seven recent HFTs (LCT, KCF, STRCF, PTAV, Staple, fDSST, and CSRDCF) and eight DFTs (SPLT, MDNet, SINT, StructSiam, DSiam, TRACA, CFNet, and HCFT). Figure 13 shows the overall performance results of our and the other 15 state-of-the-arts tracking approaches using LaSOT dataset. As we can see from Figure 13, our approach obtained a precision score of 0.305 and a success score of 0.320, and the precision and success scores of the proposed tracker ranked fifth in total. Compared with the top four tracking algorithms (SPLT, MDNet, StructSiam, and DSiam), our tracking approach still has an accuracy gap. Compared with the top four deep learning-based tracking algorithms (SPLT, MDNet, StructSiam, and DSiam), our tracking approach still has an accuracy gap. However, with the increase in the network layers number of the deep learning-based trackers, the computational complexity and parameter storage space are increased exponentially. Usually, trackers are used in monitoring or removable embedded devices, which have limited computing power and storage and cannot meet the requirements of deep learning-based trackers. On the contrary, the tracking accuracy of our tracker can meet the needs of many applications and can be realized on embedded devices. Meanwhile, compared with the seven classical handcrafted features-based trackers, our tracker obtained a significant performance improvement. For instance, compared with LCT, which is a traditional long-term tracker, our approach made about 37.70% and 30.94% improvement in terms of precision and success scores.



(**a**) (**b**)

Figure 13. Precision and success plots on LaSOT using OPE. (**a**) Precision plots of OPE on LaSOT; (**b**) Success plots of OPE on LaSOT.

6. Conclusions and Discussion

Object tracking has been a hot research topic in various fields. However, in the face of complicated challenges, the performance of the trackers still needs to be improved. In this paper, a novel dual-template CFT for various typical challenge scenarios was investigated. Different from the traditional tracking approach, the proposed tracker employs the strategies of dual-template strategy,

discriminative appearance model, multi-peak target re-detection, scale adaptive scheme, and adaptive filter template updating, which can promote the effectiveness and robustness of the tracking approach. Meanwhile, 27 existing competitors, including 16 HFTs and 11 DFTs, were introduced for the comprehensive contrastive analysis on the OTB2013, OTB2015, VOT2016, and LaSOT benchmark databases. The experimental results show that the proposed tracker performs favorably against state-of-the-art HFTs and is comparable with the DFTs. In the future, we are considering the combination of a deep learning framework to further improve the accuracy and robustness of the proposed algorithm. To effectively track the target on a large-scale and long-term video dataset, we will further mine and study the re-detection mechanism and search region strategy.

Author Contributions: C.D. constructed the framework of our approach, performed the experiment, and wrote the original manuscript. M.L. and Z.D. analyzed the experiment results. M.G., Z.H. and H.Y. provided suggestions about the revision of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (No. 61671194), Fundamental Research Funds for the Provincial Universities (No. GK199900299012-010) and Key R&D Program of Zhejiang Province (No. 2020C03098).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Han, Y.Q.; Deng, C.W.; Zhao, B.Y.; Zhao, B.J. Spatial-temporal context-aware tracking. *IEEE Signal Process. Lett.* **2019**, *26*, 500–504. [CrossRef]
2. Dong, X.P.; Shen, J.B.; Wang, W.G.; Liu, Y.; Shao, L.; Porikli, F. Hyperparameter optimization for tracking with continuous deep Q-learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 518–527. [CrossRef]
3. Dong, X.P.; Shen, J.B. Triplet loss in siamese network for object tracking. In Proceedings of the 14th European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 472–488. [CrossRef]
4. Li, C.P.; Xing, Q.J.; Ma, Z.G. HKSiamFC: Visual-tracking framework using prior information provided by staple and kalman filter. *Sensors* **2020**, *20*, 2137. [CrossRef] [PubMed]
5. Mercorelli, P. Denoising and harmonic detection using nonorthogonal wavelet packets in industrial applications. *J. Syst. Sci. Complex.* **2007**, *20*, 325–343. [CrossRef]
6. Mercorelli, P. Biorthogonal wavelet trees in the classification of embedded signal classes for intelligent sensors using machine learning applications. *J. Frankl. Inst.* **2007**, *344*, 813–829. [CrossRef]
7. Kim, B.H.; Lukezic, A.; Lee, J.H.; Jung, H.M.; Kim, M.Y. Global motion-aware robust visual object tracking for electro optical targeting systems. *Sensors* **2020**, *20*, 566. [CrossRef]
8. Du, K.; Ju, Y.F.; Jin, Y.L.; Li, G.; Li, Y.Y.; Qian, S.L. Object tracking based on improved mean shift and SIFT. In Proceedings of the 2nd International Conference on Consumer Electronics, Communications and Networks, Yichang, China, 21–23 April 2012; pp. 2716–2719. [CrossRef]
9. Zhang, T.; Liu, S.; Xu, C.; Yan, S.C.; Ghanem, B.; Ahuja, N.; Yang, M.H. Structural sparse tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 150–158. [CrossRef]
10. Babenko, B.; Yang, M.H.; Belongie, S. Visual tracking with online multiple instance learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami Beach, FL, USA, 20–25 June 2009; pp. 983–990. [CrossRef]
11. Hare, S.; Saffari, A.; Torr, P. Struck: Structured output tracking with kernels. In Proceedings of the International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 263–270. [CrossRef]
12. Wu, Y.; Lim, J.; Yang, M. Online object tracking: A benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 2411–2418. [CrossRef]
13. Wu, Y.; Lim, J.; Yang, M. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef]

14. Kristan, K.; Jiri, M.; Leonardis, A. The visual object tracking VOT2015 challenge results. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 11–18 December 2015; pp. 564–586. [CrossRef]
15. Kristan, M.; Matas, J.; Leonardis, A.; Felsberg, M.; Pflugfelder, R.; Čehovin, L.; Vojír, T.; Häger, G.; Lukežič, A.; Fernández, G.; et al. The visual object tracking VOT2016 challenge results. In Proceedings of the 14th European Conference on Computer Vision (ECCV), Amsterdam, Holland, 8–16 October 2016; pp. 777–823. [CrossRef]
16. Liu, F.; Gong, C.; Huang, X.; Zhou, T.; Yang, J.; Tao, D. Robust visual tracking revisited: From correlation filter to template matching. *IEEE Trans. Image Process.* **2018**, *27*, 2777–2790. [CrossRef]
17. Han, Z.J.; Wang, P.; Ye, Q.X. Adaptive discriminative deep correlation filter for visual object tracking. *IEEE Trans. Circuits Sys. Video Technol.* **2020**, *30*, 155–166. [CrossRef]
18. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550. [CrossRef]
19. Henriques, J.F.; Rui, C.; Martins, P.; Batista, J. Exploiting the circulant structure of tracking-by-detection with kernels. In Proceedings of the 12th European Conference on Computer Vision (ECCV), Florence, Italy, 7–13 October 2012; pp. 702–715. [CrossRef]
20. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596. [CrossRef]
21. Danelljan, M.; Hager, G.; Khan, F.S.; Felsberg, M. Learning spatially regularized correlation filters for visual tracking. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 11–18 December 2015; pp. 4310–4318. [CrossRef]
22. Ma, L.; Lu, J.; Feng, J.J.; Zhou, J. Multiple feature fusion via weighted entropy for visual tracking. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 11–18 December 2015; pp. 3128–3136. [CrossRef]
23. Makris, A.; Kosmopoulos, D.; Perantonis, S.; Theodoridis, S. Hierarchical feature fusion for visual tracking. In Proceedings of the 2007 IEEE International Conference on Image Processing (ICIP), San Antonio, TX, USA, 16–19 September 2007. [CrossRef]
24. Fu, C.H.; Duan, R.; Kircali, D.; Kayacan, E. Onboard robust visual tracking for UAVs using a reliable global-local object model. *Sensors* **2016**, *16*, 1406. [CrossRef]
25. Wang, W.; Wang, C.P.; Liu, S.; Zhang, T.Z.; Cao, X.C. Robust target tracking by online random forests and superpixels. *IEEE Trans. Circuits Sys. Video Technol.* **2018**, *28*, 1609–1622. [CrossRef]
26. Jiang, N.; Liu, W.Y.; Wu, Y. Learning adaptive metric for robust visual tracking. *IEEE Trans. Image Process* **2011**, *20*, 2288–2300. [CrossRef] [PubMed]
27. Ma, C.; Huang, J.B.; Yang, X.K.; Yang, M.H. Hierarchical convolutional features for visual tracking. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 11–18 December 2015; pp. 3074–3082. [CrossRef]
28. Nam, H.; Han, B. Learning multi-domain convolutional neural networks for visual tracking. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4293–4302. [CrossRef]
29. Zhao, F.; Wang, J.Q.; Wu, Y.; Tang, M. Adversarial deep tracking. *IEEE Trans. Circuits Sys. Video Technol.* **2019**, *29*, 1998–2011. [CrossRef]
30. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H.S. Fully-convolutional siamese networks for object tracking. In Proceedings of the 14th European Conference on Computer Vision (ECCV), Amsterdam, Holland, 8–16 October 2016; pp. 850–865. [CrossRef]
31. Wang, Y.; Wei, X.; Shen, H.; Tang, X.; Yu, H. Adaptive model updating for robust object tracking. *Signal Process. Image Commun.* **2020**, *80*, 115656. [CrossRef]
32. Han, Y.M.; Zhang, P.; Huang, W.; Zha, Y.F.; Cooper, G.D.; Zhang, Y.N. Robust visual tracking based on adversarial unlabeled instance generation with label smoothing loss regularization. *Pattern Recognit.* **2020**, *97*, 107027. [CrossRef]
33. Huang, L.H.; Zhao, X.; Huang, K.Q. GlobalTrack: A simple and strong baseline for long-term tracking. *arXiv* **2019**, arXiv:1912.08531. Available online: https://arxiv.org/abs/1912.08531 (accessed on 24 July 2020).

34. Yan, B.; Zhao, H.J.; Wang, D.; Lu, H.C.; Yang, X.Y. 'Skimming-perusal' tracking: A framework for real-time and robust long-term tracking. *arXiv* **2019**, arXiv:1909.01840. Available online: https://arxiv.org/abs/1909.01840 (accessed on 24 July 2020).
35. Zhang, Y.H.; Wang, D.; Wang, L.J.; Qi, J.Q.; Lu, H.C. Learning regression and verification networks for long-term visual tracking. *arXiv* **2018**, arXiv:1809.04320. Available online: https://arxiv.org/abs/1809.04320 (accessed on 24 July 2020).
36. Zhu, Z.; Wang, Q.; Li, B.; Wu, W.; Yan, J.; Hu, W. Distractor-aware siamese networks for visual object tracking. *arXiv* **2018**, arXiv:1808.06048. Available online: https://arxiv.org/abs/1808.06048 (accessed on 24 July 2020).
37. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; Yan, J. SiamRPN++: Evolution of siamese visual tracking with very deep networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 4282–4291. [CrossRef]
38. Danelljan, M.; Khan, F.S.; Felsberg, M.; Weijer, J. Adaptive color attributes for real-time visual tracking. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 1090–1097. [CrossRef]
39. Zhu, G.; Porikli, F.; Li, H. Beyond local search: Tracking objects everywhere with instance-specific proposals. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 943–951. [CrossRef]
40. Danelljan, M.; Hager, G.; Khan, F.S.; Felsberg, M. Discriminative scale space tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1561–1575. [CrossRef]
41. Gao, L.; Li, Y.S.; Ning, J.F. Maximum margin object tracking with weighted circulant feature maps. *IET Comput. Vis.* **2019**, *13*, 71–78. [CrossRef]
42. Liu, J.; Xiao, G.; Zhang, X.C.; Ye, P.; Xiong, X.Z.; Peng, S.Y. Anti-occlusion object tracking based on correlation filter. *Signal, Image Video Process.* **2020**, *14*, 753–761. [CrossRef]
43. Ma, C.; Yang, X.K.; Zhang, C.Y.; Yang, M.H. Long-term correlation tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA USA, 7–12 June 2015; pp. 5388–5396. [CrossRef]
44. Rifkin, R.; Yeo, G.; Poggio, T. Regularized least-squares classification. *Nato Sci. Ser. Sub Ser. III Comput. Syst. Sci.* **2003**, *190*, 131–154. [CrossRef]
45. Wang, M.M.; Liu, Y.; Huang, Z.Y. Large margin object tracking with circulant feature maps. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4800–4808. [CrossRef]
46. Danelljan, M.; Häger, G.; Khan, F.S.; Felsberg, M. Accurate scale estimation for robust visual tracking. In Proceedings of the British Machine Vision Conference (BMVC), Nottingham, UK, 1–5 September 2014. [CrossRef]
47. Fan, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Bai, H.; Xu, Y.; Liao, C.; Ling, H. Lasot: A high-quality benchmark for large-scale single object tracking. *arXiv* **2018**, arXiv:1809.07845. Available online: https://arxiv.org/abs/1809.07845 (accessed on 24 July 2020).
48. Li, F.; Tian, C.; Zuo, W.M.; Zhang, L.; Yang, M.H. Learning spatial-temporal regularized correlation filters for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 4904–4913. [CrossRef]
49. Fan, H.; Ling, H. Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking. *arXiv* **2017**, arXiv:1708.00153. Available online: https://arxiv.org/abs/1708.00153 (accessed on 24 July 2020).
50. Lukezic, A.; Vojir, T.; Zajc, L.C.; Matas, J.; Kristan, M. Discriminative correlation filter with channel and spatial reliability. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4847–4856. [CrossRef]
51. Danelljan, M.; Hager, G.; Khan, F.S.; Felsberg, M. Adaptive decontamination of the training set: A unified equation for discriminative visual tracking. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1430–1438. [CrossRef]
52. Bertinetto, L.; Valmadre, J.; Golodetz, S.; Miksik, O.; Torr, P.H.S. Staple: Complementary learners for real-time tracking. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 27–30 June 2016; pp. 1401–1409. [CrossRef]

53. Li, Y.; Zhu, J.K. A scale adaptive kernel correlation filter tracker with feature integration. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 254–265. [CrossRef]
54. Tao, R.; Gavves, E.; Smeulders, A.W.M. Siamese Instance Search for Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1420–1429. [CrossRef]
55. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ECO: Efficient convolution operators for tracking. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6931–6939. [CrossRef]
56. Danelljan, M.; Robinson, A.; Khan, F.S.; Felsberg, M. Beyond correlation filters learning continuous convolution operators for visual tracking. In Proceedings of the 14th European Conference on Computer Vision (ECCV), Amsterdam, Holland, 8–16 October 2016; pp. 472–488. [CrossRef]
57. Zhang, Y.H.; Wang, L.J.; Qi, J.Q.; Wang, D.; Feng, M.Y.; Lu, H.C. Structured siamese network for real-time visual tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 355–370.
58. Guo, Q.; Feng, W.; Zhou, C.; Huang, R.; Wan, L.; Wang, S. Learning dynamic siamese network for visual object tracking. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1781–1789. [CrossRef]
59. Choi, J.; Chang, H.; Fischer, T.; Yun, S.; Lee, K.; Jeong, J.; Demiris, Y.; Choi, J.Y. Context-aware deep feature compression for high-speed visual tracking. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 479–488. [CrossRef]
60. Valmadre, J.; Bertinetto, L.; Henriques, J.; Vedaldi, A.; Torr, P.H. End-to-end representation learning for correlation filter based tracking. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5000–5008. [CrossRef]