Hindawi
Security and Communication Networks
Volume 2020, Article ID 8891035, 13 pages
https://doi.org/10.1155/2020/8891035

Research Article

Visual Object Multimodality Tracking Based on Correlation Filters for Edge Computing

Guosheng Yang and Qisheng Wei

School of Information Engineering, Minzu University of China, No. 27 Zhongguancun South Avenue, Beijing, China

Correspondence should be addressed to Qisheng Wei; 18301393@muc.edu.cn

Received 30 June 2020; Revised 23 September 2020; Accepted 6 November 2020; Published 15 December 2020

Academic Editor: Honghao Gao

In recent years, visual object tracking has become a very active research field which is mainly divided into the correlation filter-based tracking and deep learning (e.g., deep convolutional neural network and Siamese neural network) based tracking. For target tracking algorithms based on deep learning, a large amount of computation is required, usually deployed on expensive graphics cards. However, for the rich monitoring devices in the Internet of Things, it is difficult to capture all the moving targets in each device in real time, so it is necessary to perform hierarchical processing and use tracking based on correlation filtering in insensitive areas to alleviate the local computing pressure. In sensitive areas, upload the video stream to a cloud computing platform with a faster computing speed to perform an algorithm based on deep features. In this paper, we mainly focus on the correlation filter-based tracking. In the correlation filter-based tracking, the discriminative scale space tracker (DSST) is one of the most popular and typical ones which is successfully applied to many application fields. However, there are still some improvements that need to be further studied for DSST. One is that the algorithms do not consider the target rotation on purpose. The other is that it is a very heavy computational load to extract the histogram of oriented gradient (HOG) features from too many patches centered at the target position in order to ensure the scale estimation accuracy. To address these two problems, we introduce the alterable patch number for target scale tracking and the space searching for target rotation tracking into the standard DSST tracking method and propose a visual object multimodality tracker based on correlation filters (MTCF) to simultaneously cope with translation, scale, and rotation in plane for the tracked target and to obtain the target information of position, scale, and attitude angle at the same time. Finally, in Visual Tracker Benchmark data set, the experiments are performed on the proposed algorithms to show their effectiveness in multimodality tracking.

1. Introduction

Visual object tracking (VOT), the subfield of computer vision, is a process of continuously estimating the target state through video image sequence. In recent years, VOT has become a very active research domain due to its extensive applications in many sorts of fields such as intelligent surveillance [1], automatic driving [2], and traffic flow monitoring [3], to name a few.

In fields such as security monitoring and control, the traditional network architecture is difficult to deal with in terms of network delay and security reliability, and thus edge computing technology was born. Tasks with different attributes can be passed to different levels for processing. Zhan [4] shows that the first few feature extraction layers could run on edge device, and the others run on the cloud. And Gao [5, 6] divides tasks into different levels according to the business applications and using edge devices in one level.

As Figure 1 shows, for nonsensitive areas, video streams with lower resolutions can be processed on the local device; in the medium area, ordinary-resolution video streams can be used on the edge device; and in high-risk areas, high-resolution video streams can be used on the core cloud server, thereby reducing network bandwidth and improving the overall operating efficiency of the system. This article mainly explores the processing of video streams on edge clouds, and the tracking algorithm used is based on filtering.

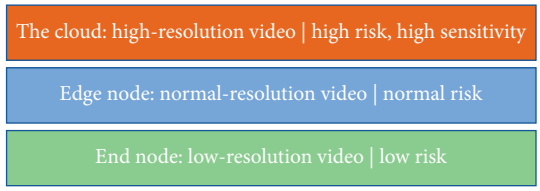


Figure 1: End-edge-cloud computing graphic.

A number of robust tracking algorithms have been proposed and developed to deal with the problems resulted from occlusion, illumination variation, background clutter, motion blur, and so on [7–14]. Such these algorithms are divided into deep learning-based category (DLC) and correlation filter- (CF-) based category [9].

For DLC, since the papers written by Geoff Hinton et al. [15, 16] were published, deep learning has become especially popular in the context of deep neural networks and has achieved impressive success on many applications, especially on feature extraction in computer vision. Inspired by such success, various deep-learning-based trackers [13, 14, 17–22] have been proposed and developed to cope with the problems encountered in tracking. Although most of trackers based on deep neural networks demonstrated the potential advantages for significantly improving the tracking performance which was testified by world VOT competitions [17], there are still some obvious limits. For example, there are fewer or even no training data available for the tracker because the prior information of the tracked object or the object bounding box is usually available only in the first frame. Even if the offline pretraining is employed to learn the target features for constructing a feature set of many targets, it is very possible for tracking a particular object whose features are not contained in the feature set. Nowadays, zero-shot and one-shot learning, as well as Siamese region proposal network, may be the most effective measures to cope with these problems [23–27].

And correlation filter-based tracking is also a solution. From its beginning of the minimum output sum of squared error (MOSSE) method [10] to the discriminative scale space tracker (DSST) method [12], a lot of improvements have continuously been made, which makes tracking based on CF achieve some highlighted tracking performances, such as lower computational load, being robust to the appearance variations of targets, and high tracking accuracy. However, there are still some improvements that need to be further studied for the tracking based on CF. One is that the algorithms do not consider the target rotation on purpose. The other is that the real-time property of DSST cannot always be ensured because it has a very heavy computational load to extract the histogram of oriented gradient (HOG) features from so many patches centered at the target position in order to ensure the scale estimation accuracy. This inspired us to think of an idea: on the premise of ensuring the tracking accuracy, appropriately decrease the number of patches in order to save the time for the introduction of the target rotation into DSST to form a multimodality tracking. It means that the tracker should simultaneously cope with translation, scale, and rotation in plane for the tracked target, which leads us to propose the visual object multi-modality tracking based on correlation filters (MTCF), to figure out these two problems, and at the same time to obtain the target information of position, scale, and attitude angle simultaneously.

In this paper, we design a correlation filter-based tracker aiming at tracking the target accurately and robustly with the tracking speed at 25 frames per second and tracking the rotation of target.

2. Related Materials

In this section, centering on tracking based on CF, we briefly list some relevant research works which have contributions to the tracking based on CF to highlight our motivations.

MOSSE method is taken as the earliest real-time CF-based tracker [28], which is an improved version of average synthetic exact filter (ASEF) [29] trained offline to detect objects. MOSSE tracker has strong robustness to target appearance and environment change, which can achieve very fast tracking speed. This is because that the correlation convolution of image in time domain is transformed into the multiplication of image in frequency domain, which greatly reduces the computation complexity and load. However, MOSSE method uses only grayscale samples to train CF and mainly focuses on translation without considering scale and rotation.

Based on the MOSSE, the circulant structure kernel CSK method [30] constructs a circulant matrix of training data by using cyclic shift of target window to maintain dense sampling around the target, rather than random sampling. On the other hand, CSK method maps ridge regression of linear space to nonlinear space through a kernel function and simplifies the calculation via solving a dual problem in nonlinear space to avoid inverse matrix operation, which leads to reducing the computation complexity and improving the tracking speed. The kernelized correlation filter (KCF) method [11] is an improved version of CSK. It introduces multichannel HOG features into CSK to enhance the feature representation ability and to improve the tracking performance significantly. Nevertheless, there exists a major imperfection for KCF method; i.e., it is not robust to the scale variation of the target. In addition, for the KCF-based tracking, the authenticity of negative samples will decrease along with the increase of cyclic displacement, which results in the tracker being trained on a portion of unreal samples. To address this issue, Danelljan et al. [18] introduce a spatial regularized term in the goal function of KCF-based tracker to penalize the filter coefficients near the margins of the bounding box. Based on [18], Dai et al. [28] propose a novel adaptive spatial regularized CF to make the tracker learn more reliable filter coefficients by fully exploiting the diversity information of different objects in different frames during the tracking process. However, just as the standard KCF-based trackers do, these two trackers are still not robust to the scale variation of the target.

DSST [12, 31] trackers address the scale adaptation problem using multiscale searching strategies. It divides tracking into translation prediction and scale prediction.

Firstly, translation prediction is performed by applying a standard translation filter on the current frame to get the position of the target. Secondly, the target size is estimated by employing trained scale filter at the target location obtained from the translation filter. Translation filter and scale filter are two independent filters, and both are based on MOSSE. Although DSST tracker has improved the tracking performance and is robust to target scale variation, there exist some obvious limitations to be further perfected. One is that DSST does not consider the target rotation on purpose, which has strong negative impacts on the tracking performance. The other is that it is not necessary for guaranteeing the tracking speed to spend a lot of operation time on sampling too many patches centered on the target location.

Besides of the tracking method, features of the tracked target are also key components of a tracker, which has a very heavy influence on the tracking performance. Generally speaking, the richer the features are, the better the performance of the tracker is. The simplest feature is intensity matrix of the search image, which is used in MOSSE [10]. And SIFT features [32] and HOG features [33] are used in object tracking afterwards. In recent years, deep features [34] are widely used in object tracking. In this paper, HOG features combined with grayscale features rather than deep features are adopted because our focus is on the CF-based tracking. And we do not adopt SIFT because SIFT is scale-invariant and we need to explicitly capture the size change of the object.

Summarizing the analysis stated above, we propose the MTCF to alleviate the imperfections of the relevant CF-based trackers stated above. Aiming at tracking the target accurately and robustly with the tracking speed at 25 frames per second at least for practical visual object tracking, MTCF consists of 4 tasks. Firstly, based on the standard CF-based translation tracker, determine the target location in the current frame. Secondly, based on DSST, sample several patches (with alterable number of patches) with different resolutions, centered at the tracked target location determined by translation CF, figure out the feasible scale for patches, and seek out an optimal decision policy to find the final scale among feasible scales. Thirdly, based on the standard CF-based translation tracker, design a rotation tracker using space searching. Lastly, integrate the previous 3 tasks to form MTCF.

3. Methodology of the Tracking Design

3.1. Variable Symbols Used in This Paper. In this paper, f denotes the "feature" of one image patch cropped with specific bounding box, h denotes the correlation filter, and g denotes the response map of correlation. In this way, $f_{\text{trans},i}$ denotes the feature of i^{th} frame used to correlate with translation filter h_{trans} and we get the translation response map $g_{\text{trans},i}$.

And s_i denotes the scale of the target the tracker got after i^{th} frame, and r_i denotes the rotation angle of the target after tracking i^{th} frame.

In terms of the convolution theorem, the correlation in spatial domain can be transformed to element-wise manipulation, which will dramatically reduce the correlation computation load. Thus, for computation efficiency, correlation manipulation is proposed to use Fast Fourier Transform (FFT) method in frequency domain. So, let the uppercase variables be the Fourier transforms of their lowercase counterparts, i.e., $F_{\text{trans},i}$, $G_{\text{trans},i}$, and H_{trans} corresponding to $f_{\text{trans},i}$, $g_{\text{trans},i}$, and h_{trans}, respectively.

3.2. Standard Translation Tracker Based on Correlation Filter. As Figure 2 shows, given a video sequence, draw a rectangular bounding box (the very close same size as the target, the red one) around the target in first frame and extract a feature map $f_{\text{trans},i}$ from the chosen region (the green rectangle, two times the size of the red one). And then train a correlation filter h_{trans} to correlate with $f_{\text{trans},1}$ to get an ideal response $g_{\text{trans},1}$. In the next frames, use the correlation filter h_{trans} to correlate with extracted feature map from the chosen region and get a response map $g_{\text{trans},i}$ as follows:

$$g_{\text{trans},i} = f_{\text{trans},i} * h_{\text{trans}}, \tag{1}$$

where $*$ represents convolution operation.

In normal tracking process, there should be one peak in the response map. And the peak position is considered as the center of target (and in this sense tracking executes). The key of tracking is to find a robust feature extractor and maintain the correlation filter h_{trans} to counter a variety of adverse effects such as target appearance transformation, occlusion, and so on, using appropriate updating strategy.

3.3. Scale Tracker Based on Correlation Filter. Being different from that in the original DSST, the number of scales S (or the number of image patches) in this paper is an optional positive integer determined by the trade-off between tracking speed and tracking accuracy (i.e., smaller S is selected if tracking speed takes priority to tracking accuracy, vice versa). Let M, N be the shape of the target, and construct image patches centered on the target position p_i with different scales to form an image patch set

$$B_{\text{scale},i} = \left\{ \beta^n M \times \beta^n N \mid \ n = 0, \ \pm 1, \ \pm 2, \ldots, \ \pm \text{round}\left(\frac{S}{2}\right) \right\}, \tag{2}$$

where β is scale step. Resize each $\beta^n M \times \beta^n N$ from $B_{\text{scale},i}$ into the same shape to form a bounding box set $\text{patch}_{\text{scale},i}$. As Figure 3 shows, instead of extracting one feature map from a bounding box with fixed scale, the tracker extracts a feature map $f^n_{\text{scale},i}$ for each patch from the bounding box set $\text{patch}_{\text{scale},i}$ (the number of feature maps is 33 in Figure 3). Each feature map $f^n_{\text{scale},i}$ is concatenated into a vector, and all these vectors are combined into a feature map $f^n_{\text{scale},i}$. And we design a scale correlation filter to correlate the feature map $f^n_{\text{scale},i}$, and the scale where maximum response taking place is the predicting scale to match current scale of target.

3.4. Rotation Tracker Based on Correlation Filter. The target may rotate during tracking, so we use rotated bounding box

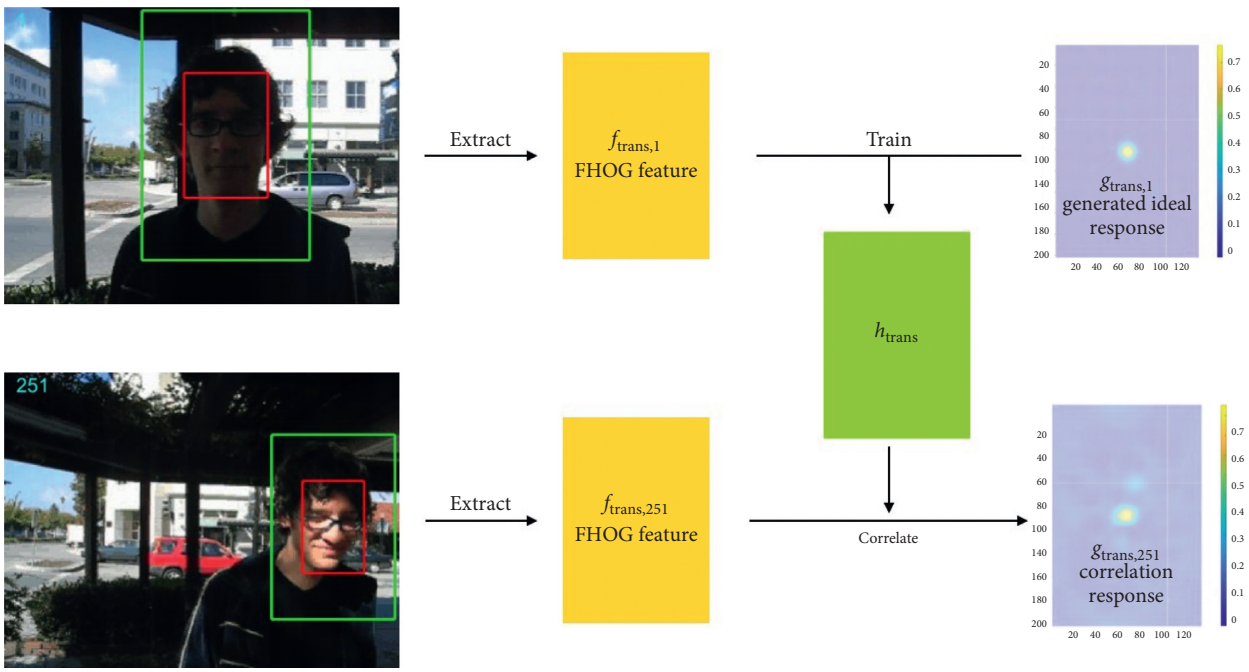


Figure 2: Correlation filter-based tracking diagram (the two frames are from sequence "Trellis" in [35]).

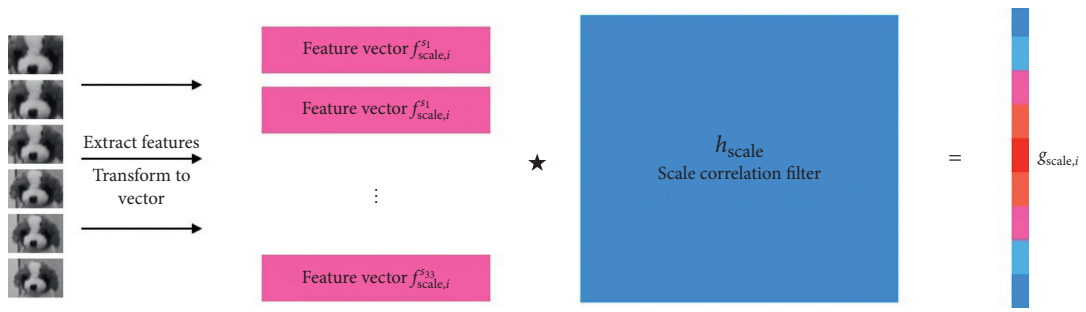


Figure 3: Scale tracking diagram.

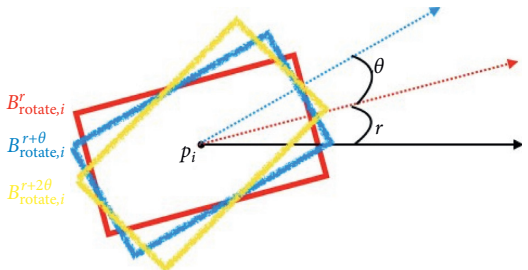


Figure 4: Bounding boxes rotate around the target center p_i.

centered on target to crop every frame. As shown in Figure 4, let p_i be the center of the target and r the current angle the target rotated, and $B^{r}_{\text{rotate},i}$ denotes the bounding box with the rotation in frame i. And using the rotation around target center, we construct a set of bounding boxes

$$B_{\text{rotate},i} = \left\{B^{r-\Theta}_{\text{rotate},i}, \ldots, B^{r-\theta}_{\text{rotate},i}, B^{r}_{\text{rotate},i}, B^{r+\theta}_{\text{rotate},i}, \ldots, B^{r+\Theta}_{\text{rotate},i}\right\}, \tag{3}$$

with the same size of $B^{r}_{\text{rotate},i}$; here Θ is the given maximum rotation angular displacement and θ is rotation step.

For each bounding box $B^{r}_{\text{rotate},i}$ from $B_{\text{rotate},i}$, extract feature map $f^{\text{rot}}_{\text{rotate},i}$, and correlate with the rotation correlation filter to get a maximum response value, where

$$\text{rot} \in \{r - \Theta, \ldots, r + \Theta\}. \tag{4}$$

Compare those values and find the largest one to get the predicting rotation angle r. Let r be the tracking result of frame i, as follows:

$$r_i = \max_{\text{rot}\in\{r-\Theta,\ldots,r+\Theta\}} \max g^{\text{rot}}_{\text{rotate},i}. \tag{5}$$

In addition to the methods we used here, we also envisioned the "1-dimensional correlation rotation tracking" in the Supplementary Materials. However, after testing, it shows that this method requires too much calculation and is not suitable for use at edge nodes.

3.5. Multimodality Tracking Based on Correlation Filter. Integrate translation, scale, and rotation stated in previous section to form MTCF whose iteration procedure at the i^{th}

frame is briefly outlined with the known parameters obtained in the $(i+1)^{\text{th}}$ frame, including target position p_{i-1}, translation filter h_{trans}, scale filter h_{scale}, scale s_{i-1}, rotation filter h_{rotate}, and rotation angle r_{i-1}.

3.5.1. Translation Estimation

(1) Construct bounding box $B_{\text{trans},i}$ with the scale s_{i-1}, centered at p_{i-1} in the i^{th} frame.

(2) Extract feature map $f_{\text{trans},i}$ from $B_{\text{trans},i}$.

(3) Calculate the correlation map $g_{\text{trans},i}$ using $f_{\text{trans},i}$ and h_{trans}.

(4) Obtain the target new position p_i corresponding to the position where the largest correlation value of $g_{\text{trans},i}$ taking place.

3.5.2. Scale Estimation

(1) Construct image patches $\text{patch}_{\text{scale},i}$ of different scales centered on the target position p_i in the i^{th}frame

(2) Extract feature map patches $\mathbf{f}_{\text{scale},i}$ from image patches $\text{patch}_{\text{scale},i}$, and concatenate each feature map $f_{\text{scale},i} \in \mathbf{f}_{\text{scale},i}$ to form an vector, and then combine those vectors to form a feature matrix $f_{\text{scale},i}$

(3) Calculate the correlation map $g_{\text{scale},i}$ using $f_{\text{scale},i}$ and h_{scale}

(4) Update the target scale with the optimal s corresponding to the position where the largest-scale correlation value is located

3.5.3. Rotation Estimation

(1) Construct image patches $\text{patch}_{\text{rotate},i}$ from the bounding box set $B_{\text{rotate},i}$ centered on target position p_i with rotation angle r_{i-1}

(2) Extract feature maps $f^{\text{rot}}_{\text{rotate},i}$ for every patch from $\text{patch}_{\text{rotate},i}$

(3) For every feature map $f^{\text{rot}}_{\text{rotate},i}$, make the correlation with the original rotation filter, and get a maximum response value $\text{score}_{\text{rotate},i}$

(4) Update the target rotation angle r_i with the optimal rot corresponding to the best $\text{score}^{\text{rot}}_{\text{rotate},i}$

3.5.4. Model Update

(1) Construct the bounding box $B_{\text{trans},i}$ centered on target position $p_i = (x_i, y_i)$ with scale s and rotation angle r

(2) Extract $f_{\text{trans},i}$, $f_{\text{scale},i}$, and $f_{\text{rotate},i}$

(3) Update translation model

(4) Update scale model

(5) Update rotation model

3.5.5. Keep Tracking. Output the tracking results of the i^{th} frame and return to the next frame tracking.

4. MTCF: The Entire Model

4.1. Translation Tracking Procedure. The simplest correlation-based tracking only focuses on translation of the target. In the first frame, we label a rectangular region $B_{\text{trans},1}$ centered on the target. So, the tracker can extract the feature map of target appearance. The feature map must maintain a spatial mapping because the tracker uses the position where the maximum response happens to predict new target position.

The simplest feature map is gray intensity matrix transformed from the specific region (for example, $B_{\text{trans},1}$) of original frame. Many researchers use 2-dimensional Hanning window (see Figure 5) to preprocess the primitive intensity matrix. After being processed by the Hanning window, the intensity matrix focuses on the central region of target and weakens the background information near the bounding box edge. Because in the first frame we draw the bounding box tightly around the target, the tracker may lose some features and behave unstable.

To address this issue, the simple way is to expand the search region. Define a parameter bb to determine how many times the size of $B_{\text{trans},1}$ the search window is. As a rule, greater parameter bb will contribute to extracting more features of the target and making the tracker stable. But much more time will be spent on the extracting features from the large search region.

This is clearly demonstrated by "S1" presented in the Supplementary Materials. In this paper, we adopt a trade-off policy and select the bb = 2.

From now on, we will use $B_{\text{trans},1}$ to represent the translation search windows.

From $B_{\text{trans},1}$ we get the $f_{\text{trans},1}$, and because we need to train the translation correlation filter h_{trans}, an initial $g_{\text{trans},1}$ is required. In prior papers, most of researchers take the Gauss-shaped response map as initialization, as follows:

$$\begin{aligned} g_{\text{expl}} &= e^{-(d/\sigma)}, \\ d &= \sqrt{(x - x_{\text{center}})^2 + (y - y_{\text{center}})^2}, \\ \sigma &= 1, \\ (x, y) &\in B_{\text{trans},1}, \end{aligned} \tag{6}$$

and Figure 6 shows an example of $g_{\text{trans},1}$.

Though the intensity feature is cheap in computation, it is unstable. Because it only takes advantage of little information in the frame. Recently, lots of deep features (for example, convolution neural network feature) are introduced to object tracking and behave well in accuracy and robustness. However, it is too computational expensive, and in this paper, we focus on FHOG [36] feature.

We use an FHOG feature extractor to get the feature map $f_{\text{trans},i}$. In translation step, 27 dimensions in FHOG and 1 dimension of intensity feature are taken into account. According to DSST [12], discriminative correlation filters for multidimensional features are applied as follows.

Minimize the cost function,

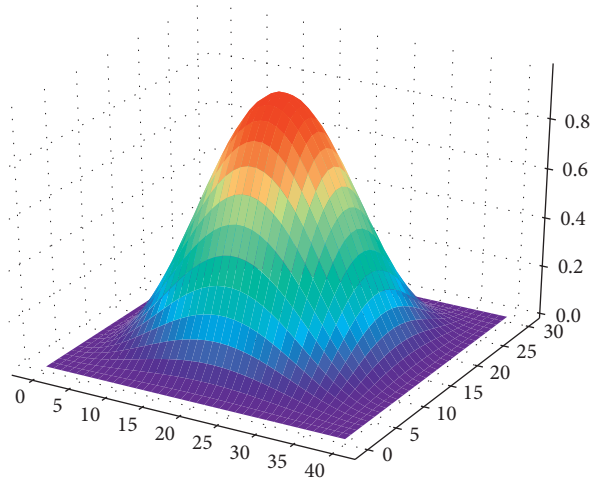


Figure 5: The 2-dimensional Hanning window (the shape is 40×30).

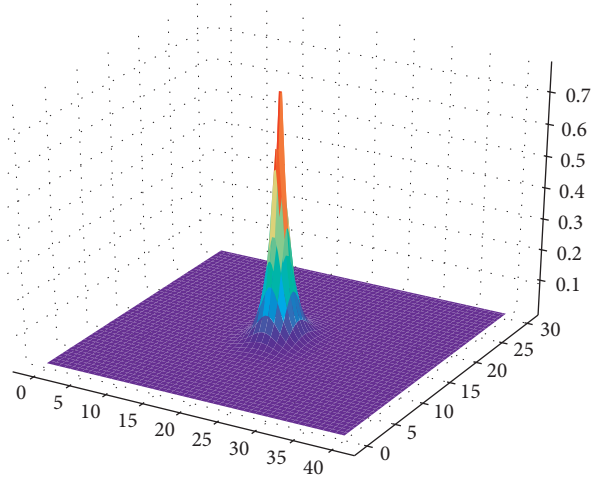


Figure 6: The Gauss-shaped response map ($\sigma = 1$, shape of B is 40×30).

$$\varepsilon = \left\| \sum_{l=1}^{d} f^l * h^l - g \right\| + \lambda \left\| \sum_{l=1}^{d} h^l \right\|. \tag{7}$$

Here, g is the ideal response about the correlation between feature map and filter, and the parameter $\lambda \geq 0$ is the regularization term. In FFT domain, the solution [12] can be written as

$$H^l = \frac{F^l \odot G^*}{\sum_{k=1}^{d} F^k \odot F^{k*} + \lambda}, \tag{8}$$

where $*$ indicates the complex conjugation, $\odot$ is for element-wise multiplication, and $l \in \{1, \ldots, d\}$ is the dimension number.

The translation filter H_{trans} can be solved as below:

$$H^*_{\text{trans}} = \frac{G_{\text{trans},i} \odot F^*_{\text{trans},i}}{\sum_l F^l_{\text{trans},i} \odot F^{l*}_{\text{trans},i} + \lambda}. \tag{9}$$

Equation (9) is employed in offline learning to obtain the correlation filter. In the practical tracking, the tracker (for example, MOSSE, KCF, and DSST) takes the target position in the $(i-1)^{\text{th}}$ frame as the center of bounding box B_i in the i^{th} frame, extracts feature map $f_{\text{trans},i}$ from B_i, and then calculates the correlation map

$$g_{\text{trans},i} = F^{-1}\left\{G_{\text{trans},i} = F_{\text{trans},i} H^*_{\text{trans},(i-1)}\right\}, \tag{10}$$

to determine the target new position p_i corresponding to the element with maximum value in $s_{\text{trans},i}$; here, F^{-1} indicates the inverse Fourier transform.

$$p_i = \max s_{\text{trans},i}(k, l). \tag{11}$$

Afterwards, reconstruct bounding box B_i centered on the target new position p_i from which feature map $f_{\text{trans},i}$ is extracted, and then update the translation CF to get $H^*_{\text{trans},i}$. Lastly, an iterative formula for equation (9) is presented as the following equations from equations (9)–(12) according to [10, 12]:

$$G_{\text{trans},i} = F_{\text{trans},i} \odot H^*_{\text{trans},(i-1)}, \tag{12}$$

$$A_{\text{trans},i} = \eta G_{\text{trans},i} \odot (1-\eta) A_{\text{trans},(i-1)}, \tag{13}$$

$$D_{\text{trans},i} = \eta F_{\text{trans},i} \odot F^*_{\text{trans},(i-1)} + (1-\eta) D_{\text{trans},(i-1)}, \tag{14}$$

$$H^*_{\text{trans},i} = \frac{A_{\text{trans},i}}{D_{\text{trans},i}}, \tag{15}$$

where η is the learning rate.

4.2. Scale Tracking Procedure. As for scale tracking procedure, two methods are commonly used. One is called "exhaustive scale tracking" and the other is "1-dimensional correlation filter scale tracking." In this paper, we use "1-dimensional correlation" method.

In the previous frame, we got the target position p_{i-1} and scale s_{i-1}.

Let M, N be the shape of the target, construct image patches centered on the target position p_i in terms of the method presented in Section 3, and resize to form a bounding box set $\text{patch}_{\text{scale},i}$. FHOG extractor is applied to extract a feature map $f^n_{\text{scale},i}$ for each patch from the bounding box set $\text{patch}_{\text{scale},i}$. Each feature map $f^n_{\text{scale},i}$ is concatenated into a vector, and all of these vectors are combined into an integrated vector $f_{\text{scale},i}$. Estimating the target scale can be solved by learning a separate 1-dimensional correlation filter. Design a 1-dimensional filter h_{scale} to correlate with $f_{\text{scale},i}$. The initial ideal response $g_{\text{scale},i}$ is a Gauss-shaped peak, as Figure 7 shows.

The scale with the largest correlation response value is taken as the optimal scale s_i.

Afterwards, extract feature map $f_{\text{scale},i}$ from the $\text{patch}^n_{\text{scale},i}$ centered on the target new position p_i with the

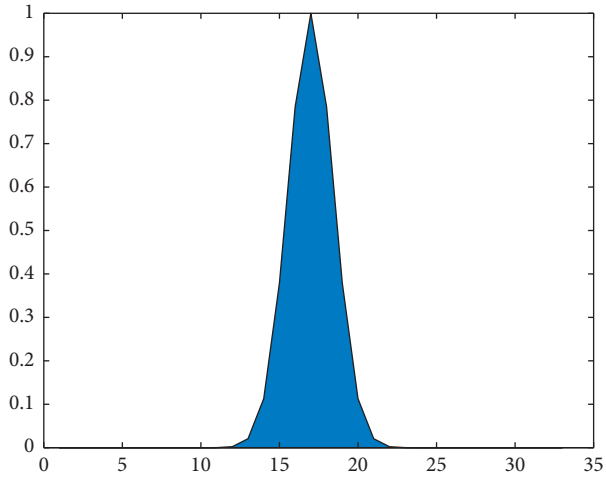


Figure 7: The 1-dimensional Gauss-shaped peak as the initial correlation response.

target final scale s_i, and then update the scale correlation filter to get $H^*_{\text{scale},i}$ using equations (13)–(17).

In this process, set the parameter "spatial bin size" to 4 to save time in the next process and use all FHOG dimensions. So, the length of S feature vector is $(M/4) \times (N/4) \times 31$.

Estimating the target scale can be solved by learning a separate 1-dimensional correlation filter. Treat the feature vector as multidimensional features and S vectors turn into 1-dimensional feature $f_{\text{scale},i} = \left\{f^{\text{sca}_1}_{\text{scale},i}, \ldots, f^{\text{sca}_S}_{\text{scale},i}\right\}$.

$$g_{\text{scale},i} = f_{\text{scale},i} * h_{\text{scale}}, \tag{16}$$

$$G_{\text{scale},i} = F_{\text{scale},i} \odot H^*_{\text{scale},(i-1)}, \tag{17}$$

$$A_{\text{scale},i} = \eta G_{\text{scale},i} \odot F^*_{\text{scale},(i-1)} + (1-\eta) A_{\text{scale},(i-1)}, \tag{18}$$

$$D_{\text{scale},i} = \eta F_{\text{scale},i} \odot F^*_{\text{scale},(i-1)} + (1-\eta) D_{\text{scale},(i-1)}, \tag{19}$$

$$H^*_{\text{scale},i} = \frac{A_{\text{scale},i}}{D_{\text{scale},i}}. \tag{20}$$

Construct different groups containing different number of patches. The number of patches varies from small to large (e.g., from 10 to 55), and all patches are centered at the tracked target location determined by translation CF in current frame. Let the basic DSST [12] perform on the visual track data set [35], and calculate the tracking speed and the tracking accuracy which is characterized by the Euclidean distance between tracking window center and ground truth center for each group. The experiment results are shown as in Figure 8.

From Figure 8, it can be seen that the tracking accuracy and speed are different with different numbers of patches. The larger number of the patches corresponds to a low tracking speed, and vice versa. Thus, on the premise of ensuring the tracking accuracy, appropriately select the number of patches in order to save the time for the introduction of the target rotation into DSST to form a multimodality tracking. After experiments, it is found that most of the time is spent in the feature extraction module.

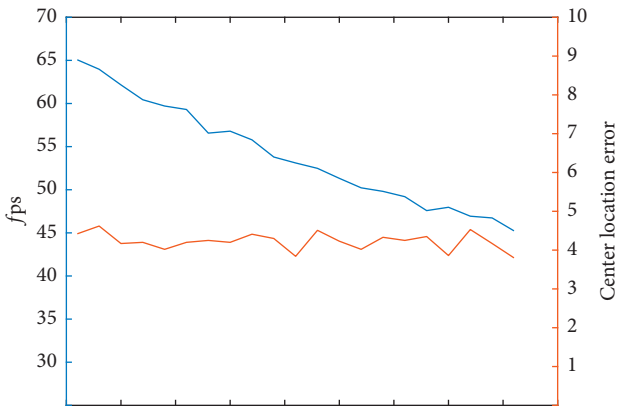


Figure 8: The tracking accuracy and speed results for groups with different scales.

4.3. Rotation Space Tracking Procedure. Set the target attitude with $r = 0$ in the first frame; construct a set of bounding boxes $B_{\text{rotate},i}$ as described in the previous section in successive frame. FHOG extractor is applied to extract a feature map $f^{\text{rot}}_{\text{rotate},i}$ for each patch $B^{\text{rot}}_{\text{rotate},i}$ from the bounding box set $B_{\text{rotate},i}$. Estimating the target rotation can be solved by learning a separate 1-dimensional correlation filter. Train a 1-dimensional a single rotation filter h_{rotate} as the similarity function to compute the maximum correlation response $\max g^{\text{rot}}_{\text{rotate},i}$ for each feature map $f^{\text{rot}}_{\text{rotate},i}$. Therefore, the best tracking angle r_i is calculated by using the following equation:

$$r_i = \max_{\text{rot}} \max g^{\text{rot}}_{\text{rotate},i}. \tag{21}$$

Afterwards, extract feature map $f_{\text{rotate},i}$ from the $B^{\text{rot}}_{\text{rotate},i}$ centered on the target new position p_i with the target final

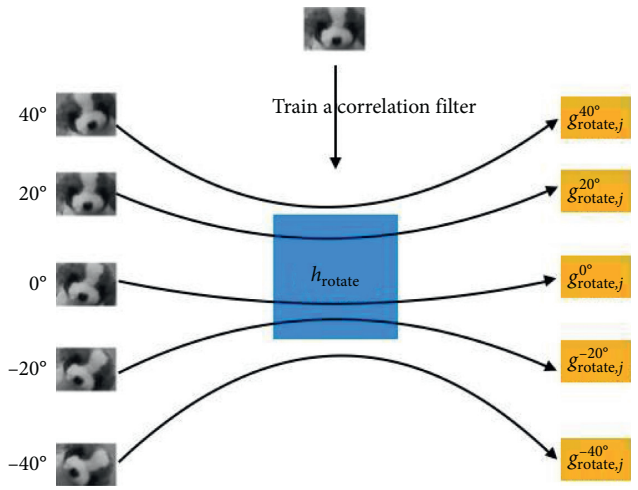


Figure 9: Exhausted method for tracking rotation.

rotation angle r_i, and then update the rotation correlation filter to get $H^*_{\text{rotate},i}$ using the following equations:

$$g_{\text{rotate},i} = f_{\text{rotate},i} * h_{\text{rotate}}, \tag{22}$$

$$G_{\text{rotate},i} = F_{\text{rotate},i} \odot H^*_{\text{rotate},(i-1)}, \tag{23}$$

$$A_{\text{rotate},i} = \eta G_{\text{rotate},i} \odot F^*_{\text{rotate},i} + (1-\eta) A_{\text{rotate},(i-1)}, \tag{24}$$

$$D_{\text{rotate},i} = \eta F_{\text{rotate},i} \odot F^*_{\text{rotate},(i-1)} + (1-\eta) D_{\text{rotate},(i-1)}, \tag{25}$$

$$H^*_{\text{rotate},i} = \frac{A_{\text{rotate},i}}{D_{\text{rotate},i}}. \tag{26}$$

We take Figure 9 as an example to demonstrate our search rotation angle. As Figure 9 shows, $r = 0°$, $\theta = 20°$, $\Theta = 40°$, androt $\in \{-40, -20, 0, 20, 40\}$. Construct a set of bounding boxes $B_{\text{rotate},i}$ with 5 patches $B^{\text{rot}}_{\text{rotate},i}$; train the rotation correlation filter h_{rotate} using the samples in the first frame. And Figure 10 shows the correlation response with each patch, where the $r + 20°$ corresponds to the highest response. Thus, we can make a conclusion that $r + 20°$ is the best predicting rotation angle in Figure 9, which demonstrate the effectiveness of our proposed search rotation method.

In this process, how to set parameters of θ and Θ is very important to obtain the good tracking performance including tracking speed and tracking accuracy. Greater Θ and smaller θ will contribute to the good tracking performance, but much more time will be spent on the extracting features of the tracked target, which has a negative influence on the tracking speed. This is clearly demonstrated by "S2" presented in the Supplementary Materials. As a rule, parameters of θ and Θ are fixed by experiments according to the requirements of tracking tasks.

In this paper, we also adopt such a policy.

5. Experiment

5.1. Experiment Setup. In this paper, our method is implemented in MATLAB R2019a on Windows 10 system. The experiments are conducted on a PC with Intel Xeon® 2.4 GHz and 63.9 GB RAM. The data set is selected from the visual track data set [35]. Our experiment is divided into 3 groups with different parameters. All of them are used to testify our proposal method: on the premise of ensuring the tracking accuracy, appropriately decrease the number of patches in order to save the time for the introduction of the target rotation into DSST to form a multimodality tracking, to verify the effectiveness of our proposed rotation tracking algorithm, and to demonstrate the whole tracking performance of our proposed visual object multimodality tracking algorithm based on correlation filter.

In the experiment of each group, the visual track data set is selected to have target translation, scale, and rotation simultaneously. And the number of scales S, the scale factor β, and the learning rate η are kept unchanged in each group and are fixed as (33, 1.02, and 0.015) and (27, 1.0247, and 0.015), respectively, which means that maximum and minimum scale field of two groups are the same, as Figure 11 shows. And we test the influence of different size (bb) of searching window in the Supplementary Materials.

5.2. Experiment of the First Group. In this group experiment, Θ is selected to be 10°, and θ is selected to be 5°. As a result, the tracking speed is 31fps, and the experiment results are shown in Figures 12 and 13 consisting of some typical tracking frames.

From Figure 11, it can be seen that appropriately decreasing the number of patches completely can save the time for the introduction of the target rotation into DSST to form

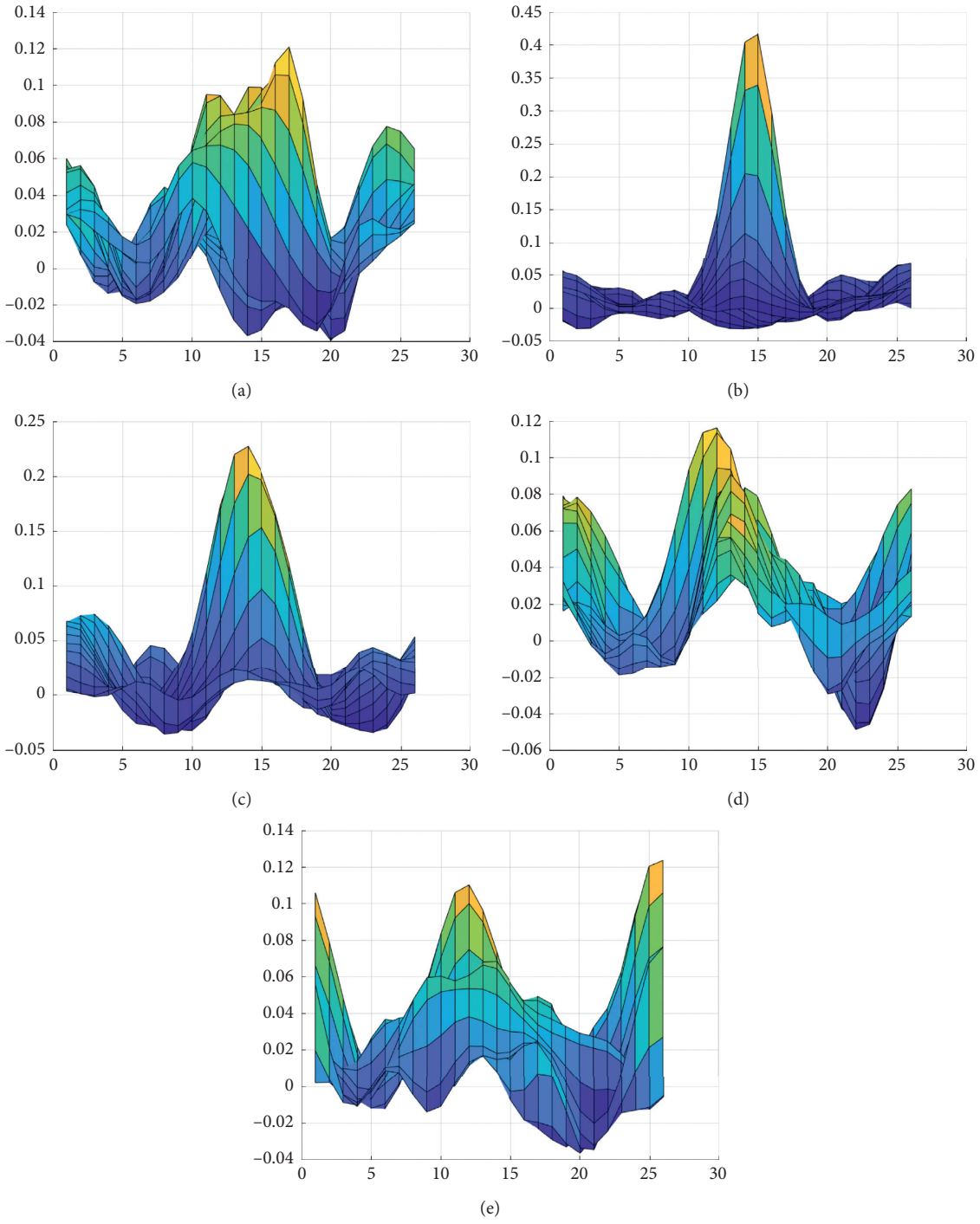


FIGURE 10: The correlation response with image patches cropped by different-rotation bounding box. (a) $r + 40^{\circ}$, (b) $r + 20^{\circ}$, (c) r, (d) $r - 20^{\circ}$, and (e) $r - 40^{\circ}$.

a multimodality tracking on the premise of ensuring the tracking accuracy and that our proposed rotation tracking algorithm can work well.

5.3. Rotation Tracking Performance Test. In this group experiment, θ is selected to be 12, and τ is selected to be 4. As a result, the tracking speed is 29 fps, and the experiment

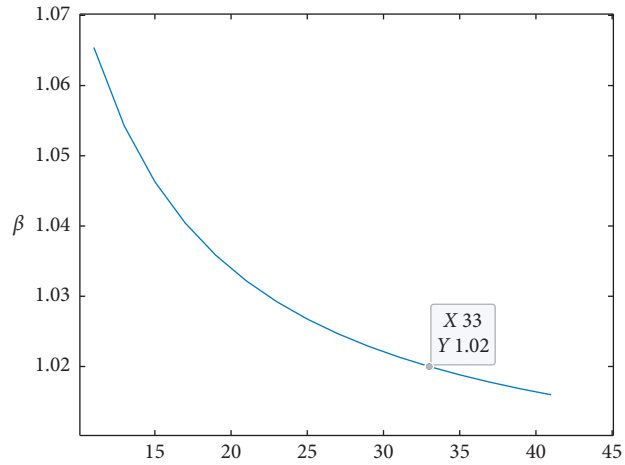
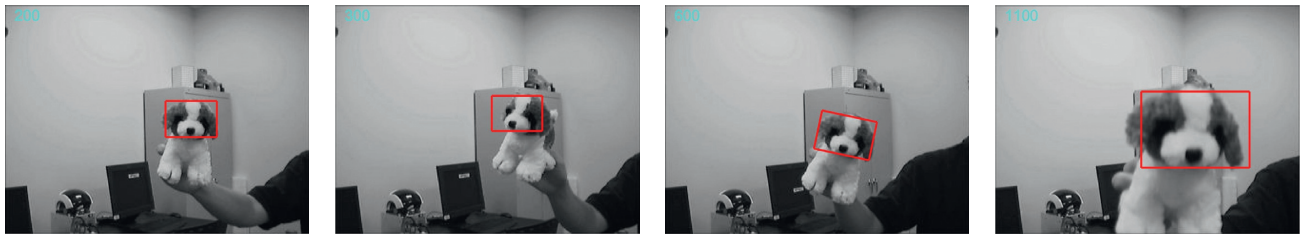
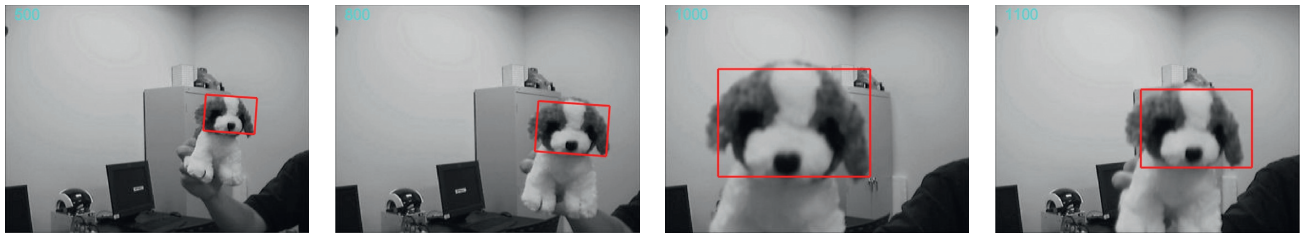


Figure 11: β is the function of S to guarantee fixed maximum and minimum scale field.



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

Figure 12: The tracking results with tracking speed 31fps. The red window represents that the target is tracked completely. (a) represents the original figure, (b) represents object translation, (c) represents object rotation, and (d) represents object scale.



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

Figure 13: The tracking results with tracking speed 29 fps. The red window represents that the target is tracked completely. (a) Rotation, (b) scale, (c) scale, and (d) translation.

results are shown in Figure 13 consisting of some typical tracking frames.

In this group experiment, the tracking speed is 29 fps which is lower than that in the first group experiment because τ is selected to be 4 which means the number of $B^{\rho}_{(\text{rotate},i)}$ is increased, resulting in much more time being spent on extracting feature map $f^{\rho}_{(\text{rotate},i)}$ from $B^{\rho}_{(\text{rotate},i)}$. But our proposal visual tracker still can work well in tracking the target with translation, scale, and rotation. This can be shown by Figure 13. From this perspective, we can say that the rotation step can be appropriately increased if tracking accuracy is preferred, and vice versa.

5.4. Multimodality Tracking Performance Test. In both of the two group experiments, our proposed MTCF algorithm is performed on the visual track data set [35] to demonstrate the multimodality tracking performance; the tracking results are shown in Figure 14.

From Figure 14, it can be seen that our proposed MTCF has good multimodality tracking performance, which can enable us to obtain the position, scale, and attitude angle of the tracked target simultaneously.

The generalization ability of this algorithm still maintains the same level as DSST and is very dependent on the HOG extraction algorithm.

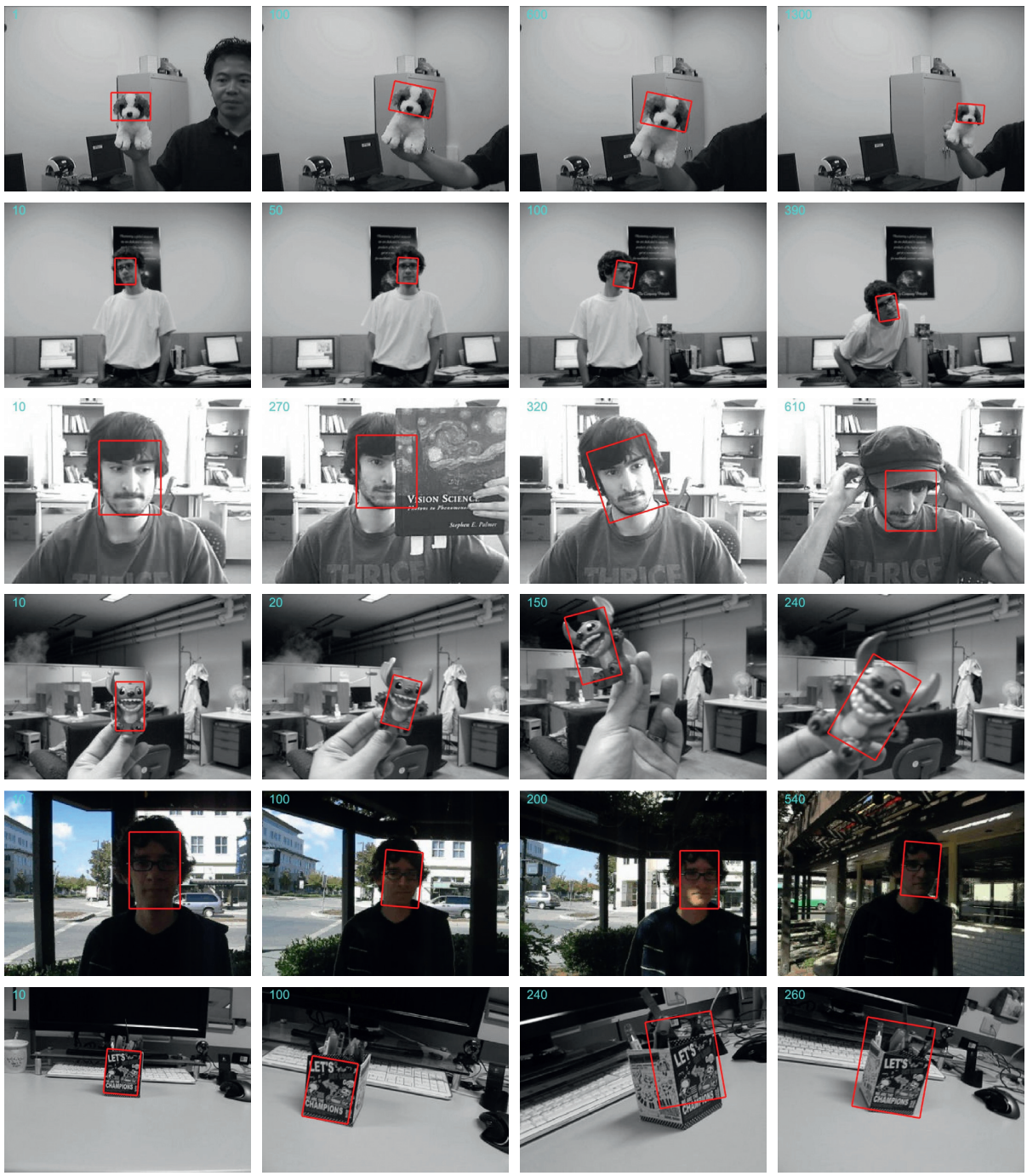


Figure 14: Some tracking results using MTCF, where the first five rows show that our tracker could track the target in scale variation, partially occlude occasion, and illumination variation, and the last row shows that the rotation tracking performance depends on scale tracking.

6. Conclusion and Future Work

In this paper, on the premise of ensuring the tracking accuracy, we introduce the alterable patch number for target scale tracking and the space searching for target rotation tracking into the standard DSST tracking method and propose a multimodality tracking MTCF to simultaneously cope with translation, scale, and rotation in plane for the tracked target and to obtain the target information of position, scale, and attitude angle at the same time. Experimental results demonstrate that the proposed multimodal target tracking algorithm MTCF (1) can reach the approving tracking speed which is largely exceeded 25 fps at least for practical visual object tracking by appropriately decreasing the number of patches for target scale tracking and (2) can obtain good tracking performance for translation, scale, and rotation simultaneously. In the future, our work will focus on the distributed hardware and software

implementation of the proposed multimodal comprehensive tracking algorithm.

For terminal devices not equipped with GPU units, low-resolution video is used to reduce the computational pressure on target features. For edge devices with certain computing capabilities, they are responsible for the main target tracking tasks. Finally, for a few critical and high-risk areas, the network bandwidth saved by the above two is used to upload to the central cloud processor for calculation to achieve hierarchical governance coordination.

Data Availability

All the source codes and related pictures will be uploaded to GitHub and will be available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61772575, National Key R&D Program of China under Grant 2017YFB1402101, and Independent Research Projects of Minzu University of China.

Supplementary Materials

S1: influence of different size of searching window and analysis of the results. S2: 1-dimensional correlation rotation tracking, $\theta = 30^{\circ}$. (*Supplementary Materials*)

References

[1] G. Liu, S. Liu, K. Muhammad, A. K. Sangaiah, and F. Doctor, "Object tracking in vary lighting conditions for fog based intelligent surveillance of public spaces," *IEEE Access*, vol. 6, 2018.

[2] Y. Nishida, T. Sonoda, S. Yasukawa et al., "Underwater platform for intelligent robotics and its application in two visual tracking systems," *Journal of Robotics and Mechatronics*, vol. 30, no. 2, pp. 238–247, 2018.

[3] A. De Bruin and M. J. Booysen, "Drone-based traffic flow estimation and tracking using computer vision," 2015.

[4] Z. Zhang, X. Q. Zhang, D. C. Zuo, and G. D. Fu, "Research on target tracking application deployment strategy for edge computing," *Ruan Jian Xue Bao/Journal of Software*, vol. 31, no. 9, 2020.

[5] H. Gao, L. Kuang, Y. Yin, B. Guo, and K. Dou, "Mining consuming behaviors with temporal evolution for personalized recommendation in mobile marketing apps," *Mobile Networks and Applications*, vol. 25, no. 4, pp. 1233–1248, 2020.

[6] X. Ma, H. Gao, H. Xu, and M. Bian, "An IoT-based task scheduling optimization scheme considering the deadline and cost-aware scientific workflow for cloud computing," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, 2019.

[7] X. Sheng, Y. Liu, H. Liang, F. Li, and Y. Man, "Robust visual tracking via an improved background aware correlation filter," *IEEE Access*, vol. 7, 2019.

[8] X. Dong, J. Shen, D. Yu, W. Wang, J. Liu, and H. Huang, "Occlusion-aware real-time object tracking," *IEEE Transactions on Multimedia*, vol. 19, no. 4, pp. 763–771, 2016.

[9] M. Fiaz, A. Mahmood, S. Javed, and S. K. Jung, "Handcrafted and deep trackers," *ACM Computing Surveys*, vol. 52, no. 2, pp. 1–44, 2019.

[10] D. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2544–2550, IEEE, New York, NY, USA, 2010, http://ieeexplore.ieee.org/document/5539960/.

[11] 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, 2014.

[12] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1561–1575, 2016.

[13] R. Tao, E. Gavves, and A. W. M. Smeulders, "Siamese instance search for tracking," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1420–1429, IEEE, New York, NY, USA, 2016, http://ieeexplore.ieee.org/document/7780527/%20.

[14] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++-evolution of siamese visual tracking with very deep networks," *CoRR*, vol. 1812, 2018.

[15] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 502–504, 2006.

[16] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[17] P. Li, D. Wang, L. Wang, and H. Lu, "Deep visual tracking: review and experimental comparison," *Pattern Recognition*, vol. 13, no. 2, pp. 117–126, 2017.

[18] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Learning Spatially Regularized Correlation Filters for Visual Tracking," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4310–4318, IEEE, New York, NY, USA, 2015, http://ieeexplore.ieee.org/document/7410847/.

[19] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, "Action-decision networks for visual tracking with deep reinforcement learning," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1349–1358, IEEE, New York, NY, USA, 2017, http://ieeexplore.ieee.org/document/8099631/.

[20] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *Proceedings of the Computer Vision-ECCV 2016 Workshops*, pp. 850–865, Springer, Berlin, Germany, 2016.

[21] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4293–4302, IEEE, Berlin, Germany, 2016, http://ieeexplore.ieee.org/document/7780834/.

[22] H. Fan and H. Ling, "SANet: structure-aware network for visual tracking," in *Proceedings of the 2017 IEEE Conference on*

Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 2217–2224, IEEE, Berlin, Germany, May 2017, http://ieeexplore.ieee.org/document/8015009/.

[23] Z. Al-Halah and R. Stiefelhagen, "How to transfer? Zero-shot object recognition via hierarchical transfer of semantic attributes," 2015.

[24] O. Vinyals, C. Blundell, T. Lillicrap, and D. Wierstra, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems*, pp. 3630–3638, IEEE, Berlin, Germany, 2016.

[25] X. Dong, J. Shen, D. Wu, K. Guo, X. Jin, and F. Porikli, "Quadruplet network with one-shot learning for fast visual object tracking," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3516–3527, 2019.

[26] H. Zhang, W. Ni, W. Yan, J. Wu, H. Bian, and D. Xiang, "Visual tracking using siamese convolutional neural network with region proposal and domain specific updating," *Neurocomputing*, vol. 275, pp. 2645–2655, 2018.

[27] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8971–8980, New York, NY, USA, 2018.

[28] K. Dai, D. Wang, H. Lu, C. Sun, and J. Li, "Visual tracking via adaptive spatially-regularized correlation filters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4670–4679, London, UK, 2019.

[29] D. S. Bolme, B. A. Draper, and J. R. Beveridge, "Average of synthetic exact filters," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2105–2112, IEEE, Berlin, Germany, 2009.

[30] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proceedings of the European Conference on Computer Vision*, pp. 702–715, Springer, Berlin, Germany, 2012.

[31] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proceedings of the British Machine Vision Conference*, BMVA Press, Nottingham, UK, 2014.

[32] H. Zhou, Y. Yuan, and C. Shi, "Object tracking using sift features and mean shift," *Computer Vision and Image Understanding*, vol. 113, no. 3, pp. 345–352, 2009.

[33] C. Gárate, P. Bilinsky, and F. Bremond, "Crowd event recognition using hog tracker," in *Proceedings of the 2009 Twelfth IEEE International Workshop on PerforMance Evaluation of Tracking and Surveillance*, pp. 1–6, Berlin, Germany, 2009.

[34] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proceedings of the the IEEE International Conference on Computer Vision (ICCV) Workshops*, Berlin, Germany, December 2015.

[35] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: a benchmark," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, London, UK, 2013.

[36] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester, "Discriminatively trained deformable part models," 2012.