A target tracking method based on adaptive occlusion judgment and model updating strategy

Zhiming Cai [Corresp., Equal first author, 1, 2], **Zhuangzhuang Wang** [Equal first author, 1], **Jianchao Huang** [1], **Shujing Chen** [1], **Huabin He** [1]

[1] School of Electronic, Electrical Engineering and Physics, Fujian University of Technology, Fuzhou, Fujian, China

[2] National Demonstration Center for Experimental Electronic Information and Electrical Technology Education, Fujian University of Technology, Fuzhou, Fujian, China

Corresponding Author: Zhiming Cai
Email address: caizm@fjut.edu.cn

Target tracking is an important research in the field of computer vision. Despite the rapid development of technology, there still remains difficulties in balancing the overall performance for target occlusion, motion blur, etc. To address the above issue, we propose an improved kernel correlation filter tracking algorithm with adaptive occlusion judgement and model updating strategy (called Aojmus) to achieve robust target tracking. Firstly, the algorithm fuses color-naming (CN) and histogram of gradients (HOG) features as a feature extraction scheme and introduces a scale filter to estimate the target scale, which reduces tracking error caused by the variations of target features and scales. Secondly, the Aojmus introduces four evaluation indicators and a double thresholding mechanism to determine whether the target is occluded and the degree of occlusion respectively. The four evaluation results are weighted and fused to a final value. Finally, the updating strategy of the model is adaptively adjusted based on the weighted fusion value and the result of the scale estimation. Experimental evaluations on the OTB-2015 dataset are conducted to compare the performance of the Aojmus algorithm with four other comparable algorithms in terms of tracking precision, success rate, and speed. The experimental results show that the proposed Aojmus algorithm outperforms all the algorithms compared in terms of tracking precision. The Aojmus also exhibits excellent performance on attributes such as target occlusion and motion blur in terms of success rate. In addition, the processing speed reaches 74.85 fps, which also demonstrates good real-time performance.

A target tracking method based on adaptive occlusion judgment and model updating strategy

Zhiming Cai[1,2]✉*, Zhuangzhuang Wang[1*], Jianchao Huang[1], Shujing Chen[1], and Huabin He[1]

[1]**School of Electronic, Electrical Engineering and Physics, Fujian University of Technology, Fuzhou, China**
[2]**National Demonstration Center for Experimental Electronic Information and Electrical Technology Education, Fujian University of Technology, Fuzhou, China**

Corresponding author:
Zhiming Cai*

Email address: caizm@fjut.edu.cn

ABSTRACT

Target tracking is an important research in the field of computer vision. Despite the rapid development of technology, there still remains difficulties in balancing the overall performance for target occlusion, motion blur, etc. To address the above issue, we propose an improved kernel correlation filter tracking algorithm with adaptive occlusion judgement and model updating strategy (called Aojmus) to achieve robust target tracking. Firstly, the algorithm fuses color-naming (CN) and histogram of gradients (HOG) features as a feature extraction scheme and introduces a scale filter to estimate the target scale, which reduces tracking error caused by the variations of target features and scales. Secondly, the Aojmus introduces four evaluation indicators and a double thresholding mechanism to determine whether the target is occluded and the degree of occlusion respectively. The four evaluation results are weighted and fused to a final value. Finally, the updating strategy of the model is adaptively adjusted based on the weighted fusion value and the result of the scale estimation. Experimental evaluations on the OTB-2015 dataset are conducted to compare the performance of the Aojmus algorithm with four other comparable algorithms in terms of tracking precision, success rate, and speed. The experimental results show that the proposed Aojmus algorithm outperforms all the algorithms compared in terms of tracking precision. The Aojmus also exhibits excellent performance on attributes such as target occlusion and motion blur in terms of success rate. In addition, the processing speed reaches 74.85 fps, which also demonstrates good real-time performance.

INTRODUCTION

Motion target tracking (Lu and Xu, 2019; Wang et al., 2021) is one of the most active research areas in computer vision. With the continuous improvement of hardware facilities and the rapid development of artificial intelligence technology, motion target tracking technology is widely used in intelligent video surveillance (Zeng et al., 2020), human-computer interaction (Zhou and Liu, 2021), medical diagnosis (Al-Battal et al., 2021) and other fields. In the field of intelligent video surveillance, target tracking technology is commonly used in monitoring of vehicle violations and has been proved to be effective. In the field of medical diagnosis, tracking technology is frequently used in tracking microscopic items like cells. In terms of human-computer interaction, tracking technology is mostly utilized in robot vision and virtual environments, which primarily use visual technology to provide the tracking effect similar to human eyes. In the past two decades, target tracking technology has made tremendous developments. However, tracking targets are often limited by complex application environments, such as different illumination changes, interference from complex backgrounds, changes in their own scales, and occlusion by other objects. Therefore, improving the precision and robustness of tracking algorithms in complex environments and satisfying real-time applications become important research topics in visual target tracking.

Nowadays, the mainstream algorithms of target tracking can be classified into two categories. One is based on correlation filtering (Wei and Kang, 2017; Meng and Li, 2019), which determines the correlation region by establishing a correlation filter to find the maximum response value in the two adjacent frames, and then lock the target. Compared with the earlier tracking algorithms based on optical flow method (Xiao et al., 2016) and feature matching (Uzkent et al., 2015), the advantages for this category are fast speed and good robustness in the case of target occlusion, illumination change and motion blur (Liu et al., 2017). Another is based on deep learning (Li et al., 2016), which uses convolutional neural networks training to extract object features in the last frame and matches the object in the next frame. That is, the object is continually tracked during training. For one thing, the former is inferior to the latter when dealing with complex scenarios, such as target occlusion, out-of-view, scale variation etc. For another, the latter tracks object more slowly. Therefore, finding a solution not only meets the demands of accurate tracking in a variety of complex scenarios, but also achieves fast running are still an active research area. In this work, a target tracking algorithm based on adaptive occlusion judgment and model updating strategy, called Aojmus, is proposed to address the poor tracking performance in complex scenarios mentioned above. The Aojmus is designed on the basis of KCF algorithm and integrated with correlation filtering method which has the advantage in processing speed.

The contributions of this work can be summarized in three folds:

1. We propose to fuse CN and HOG features as the feature representation of the tracked target, which improves the discrimination and re-detection ability. Meanwhile, the scale filter is introduced to solve the defect of poor tracking precision of the target due to scale change.

2. We design four kinds of occlusion judgment indicators to solve tracking failure which caused by occlusion. These four indicators can adaptively judge the occlusion of the target during tracking. A double threshold mechanism is introduced to judge the degree of occlusion, which determines the update strategy of the tracker.

3. We use a weighted fusion strategy to fuse the results of occlusion judgments to ensure that the model update rate of the tracker changes dynamically with the judgment results of each frame, avoiding the tracking drift problem caused by fixed model update rate of most trackers.

The rest of this article is organized as follows. In “Literature Review”, related works about target tracking are surveyed. In “Preliminaries”, some prerequisites of the methodology are introduced. In ”Methodology”, we describe the architecture of the proposed algorithm, including statistical analysis, algorithm design and related parameter setting. In ”Experiments and analysis”, we compare and analyze the performance of the algorithm in quantitative and qualitative aspects. In “Conclusion”, we summarize this study and discuss possible future work.

LITERATURE REVIEW

Bolme et al. (Bolme et al., 2010) were the first to apply the correlation filtering method to target tracking and proposed the MOSSE algorithm, which achieves tracking speed of 669 fps, but with a slightly poor precision of 43.1%. To solve the problem of insufficient samples of MOSSE algorithm, Henriques et al. (Henriques et al., 2012) proposed CSK algorithm, which acquired a large number of samples through the method of cyclic shift. Moreover, the computational complexity is reduced by frequency domain processing, and thereby a robust and accurate filter is obtained. Subsequently, Henriques et al. proposed the Kernelized Correlation Filter(KCF) (Henriques et al., 2015) tracking algorithm on the basis of CSK, which utilized HOG (Histogram of Oriented Gradients) feature instead of grayscale feature and introduced a circular matrix to reduce the computational effort. The algorithm also incorporates multi-channel data to improve the operation speed and meet the requirement of real-time in the process of tracking. Inspired by the scale pooling technique, Li et al. (Yang and Zhu, 2014) and Danelljan et al. (Danelljan et al., 2014a) proposed SAMF and DSST algorithms respectively, which solved the problem of scale adaptation of KCF algorithm. The SAMF algorithm fused HOG feature and CN (Danelljan et al., 2014b)(Color-Naming) feature for the first time on the basis of KCF, which improved the tracking precision, but the speed is significantly reduced. Similarly, the DSST algorithm also achieved scale adaption, but the overall performance is inferior.Danelljan et al. (Danelljan et al., 2017a) used convolutional neural network (CNN) to extract depth features on the feature model of the target while keeping the motion model (cyclic matrix)

and observation model (correlation filter) unchanged, achieving a significant increase in precision and success rate. The research interest in correlation filtering-based target tracking has declined because the precision is difficult to improve further when dealing with target occlusion, disappearance, or non-rigid object tracking. In recent years, some representative studies have still emerged, such as ASRCF (Dai et al., 2019), ARCF (Huang et al., 2019), and PRCF (Sun et al., 2019).

Another class of target tracking algorithms is based on deep learning (Li et al., 2016), which uses convolutional neural networks for feature extraction and classification of targets to achieve target tracking. Some of them incorporate correlation filtering and deep learning, such as the HCF (Ma et al., 2015). The MDNet algorithm proposed by Nam et al. (Nam and Han, 2016) was one of the early algorithms that used deep learning alone to implement target tracking. The algorithm trains each domain separately while updating the parameters of the shared layer during training so that these parameters can be adapted to all datasets. When tracking, MDNet uses a pre-trained CNN network to track the target and thereby locate the target. Here are some similar algorithms, such as SiamDW (Zhang and Peng, 2019), SiamCAR (Guo et al., 2020) and HiFT (Cao et al., 2021), etc. Despite the superior performance of deep learning-based tracking algorithms in achieving tracking precision, they still face the disadvantages of insufficient initial training samples and slow tracking speed.

PRELIMINARIES

In this section, to help establish an understanding of the essential elements involved in the proposed methodology, the Kernelized Correlation Filter (KCF), classifier training, fast detection and model updating are illustrated in advance.

Kernelized correlation filter

The core idea of the kernel correlation filtering (KCF) algorithm is to calculate the matching degree between the predicted region and the target by establishing a kernel function based on the ridge regression. By moving the complex calculation to the frequency domain with fast Fourier transform, the fast tracking for target is achieved. Similar to most discriminative tracking algorithms, KCF algorithm also performs target detection before filter model training. It firstly trains a model of the initial position of the target, then detects whether the target exists in the prediction region of the next frame, and finally uses Gaussian kernel to calculate the correlation between two adjacent frames and determines the position of the target according to its maximum response value in the target region. The basic principles of the kernel correlation filtering algorithm, including classifier training, fast detection and model update, are described below.

Classifier training

The classifier $f(x) = \langle w, \varphi(x) \rangle$ is obtained by training ridge regression. Let (x_i, y_i) be the training sample, where y_i is the regression expectation corresponding to sample x_i. The ridge regression on the training sample yields the linear regression function $f(x) = w^T x_i$. To prevent the overfitting phenomenon, the classifier needs to be regularized as follows:

$$\min_w \sum_{i=1}^{N} (f(x_i) - y_i)^2 + \lambda \|w\|^2 \tag{1}$$

where w is the classifier parameter and λ is the regularization parameter. The closed-form solution of the above equation is:

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

In the process of generating a large amount of information of target and background using the circular matrix, the feature space formed by the sample set appears nonlinear. Therefore, Gaussian kernel function $\varphi(x_i)$ is introduced for linear transformation, and get $f(x) = w^T x_i = w^T \varphi(x_i)$, where $w = \sum_{i=1}^{N} \alpha_i x_i$. So far, the solution of w is transformed to the solution of coefficient α, which eventually yields:

$$\alpha = (K + \lambda I)^{-1} y \tag{3}$$

where K is the kernel correlation matrix. To reduce the complexity of the calculation, Equation (3) is transformed into the frequency domain with the discrete Fourier transform(DFT). Then the solution

becomes:

$$\hat{\alpha} = \frac{\hat{y}}{k^{xx} + \lambda} \tag{4}$$

The purpose of classifier training is to solve the weight coefficient α, where k^{xx} is the first row element of the kernel cycle matrix K, and ˆ denotes the DFT of vector.

Fast detection and model updating

After classifier training, in order to locate the target position of the current frame, the KCF algorithm uses the target position of the previous frame as a template, then detects it in the candidate region z of the current frame and determines the target position by finding the maximum value of $f(z) = \alpha^T \varphi(X) \varphi(z)$. To increase the calculation speed, the KCF algorithm transfers the solution from the time domain to the frequency domain as follows:

$$\hat{f}(z) = \hat{k}^{xz} \odot \hat{\alpha} \tag{5}$$

where k^{xz} denotes the kernel correlation between the target sample x and the candidate detection region z, $\hat{f}(z)$ represents the response distribution in the candidate region and the position where its maximum value is located indicates the actual position of the target in the current frame.

To ensure each frame in the video sequence can be processed, the KCF algorithm uses linear interpolation to update the filter template $\hat{\alpha}_t$ and the target feature template $\hat{x}_t$ as follows:

$$\begin{cases} \hat{\alpha}_t = (1-\eta)\,\hat{\alpha}_{t-1} + \eta\,\hat{\alpha} \\ \hat{x}_t = (1-\eta)\,\hat{x}_{t-1} + \eta\,\hat{x} \end{cases} \tag{6}$$

where η denotes the model update rate and t is time stamp.

METHODOLOGY

As target occlusion, scale variation, illumination variation affect the performance of tracking, it is of great significance to conquer such problems. The KCF algorithm increases the training samples through the circular matrix, which in turn improves the tracking accuracy. Meanwhile, by transferring to the frequency domain to avoid matrix inversion operations, the computation is greatly reduced.

However, the KCF algorithm often fails to track in the case of target occlusion or target loss because the update model learns the features of the occluded object and causes the model to get the wrong target features in the accumulation of subsequent frames, which in turn leads to tracking failure. In addition, the tracking box of KCF algorithm cannot meet the scale variation of the target, which can also greatly reduce the precision of tracking. To address these problems, this paper proposes a target tracking algorithm, call Aojmus, based on an adaptive occlusion judgment and model update strategy. The flow chart of the algorithm is shown in Fig. 1.

In this section, we describe the specific implementation of Aojmus, including feature fusion, scale estimation, occlusion judgment and model updating.

Feature fusion design

The HOG features can effectively depict the local contour and shape information of the target and are very robust to illumination changes, but are poorly adapted to target deformation and fast motion.The CN features can well represent the global color information of the target and have excellent stability to target deformation and fast motion, but are sensitive to illumination and color changes. Therefore, we employ linear fusion of HOG feature and CN feature (Xie and Zhao, 2021) to achieve feature complementarity and improve tracking precision.

The process of linear weighted fusion of these two feature vectors is as follows:

$$v_{hc} = \delta v_{\text{hog}} + (1-\delta)\, v_{cn} \tag{7}$$

where v_{hog}, v_{cn}, v_{hc} represent HOG feature, CN feature and fused feature respectively, and δ is the weighted coefficient of feature fusion. In this paper, set $\delta = 0.5$ to ensure that the advantages of HOG and CN feature can be fully utilized.

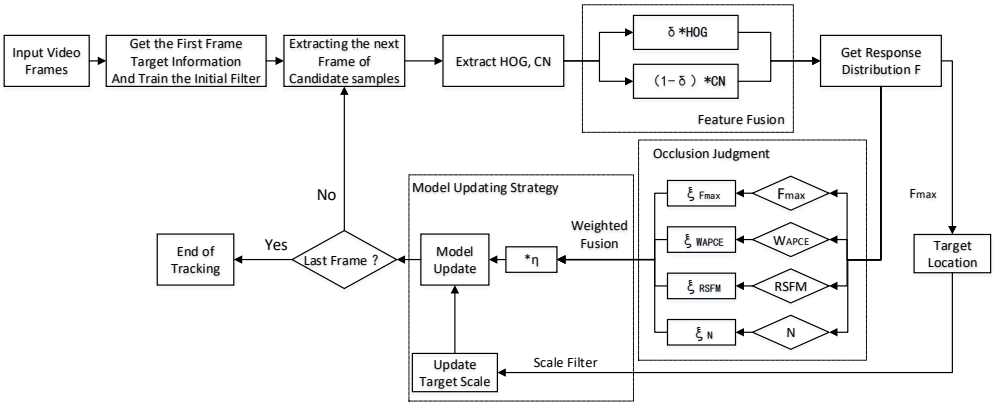


Figure 1. Flow chart of Aojmus algorithm.

Multi-scale estimation

The scale variation of target is one of the important factors affecting the tracking results. As the position change of two consecutive frames is often larger than the scale change, like DSST (Danelljan et al., 2014a), this paper first uses a two-dimensional position filter to determine the position information and then implements scale evaluation by training a one-dimensional scale filter.

Let f be the training sample and h be the optimal correlation filter. The minimum cost function is solved with ridge regression as follows:

$$\varepsilon = \left\| \sum_{l=1}^{d} h^l \odot f^l - g \right\|^2 + \lambda \sum_{l=1}^{d} \left\| h^l \right\|^2 \tag{8}$$

where $l \in \{1, \ldots, d\}$ is the feature dimension, g represents the regression expectation corresponding to the training sample f, and λ is the regularization factor. The scaling filter can be obtained by solving the above equation in Fourier domain:

$$H^l = \frac{\bar{G}F^l}{\sum_{k=1}^{d} \overline{F^k} F^k + \lambda} = \frac{A_t^l}{B_t} \tag{9}$$

where $\bar{G}$ represents complex conjugate of the DFTs of correlation outputs and λ is introduced to avoid zero denominator in case of the zero frequency component in f. By detecting the image block z in the new frame, we can obtain the response of scale filter as:

$$y = F^{-1} \left\{ \frac{\sum_{l=1}^{d} \overline{A^l} Z^l}{B + \lambda} \right\} \tag{10}$$

Up to this point, the response value of the scale filter can be calculated from Equation (10), and a new scale estimate can be determined based on the result of the maximum value. The selection principle of target sample size for scale evaluation is as follows:

$$a^n P \times a^n R, n \in \left\{ \left\lfloor \frac{-(s-1)}{2} \right\rfloor, \ldots, \left\lfloor \frac{(s-1)}{2} \right\rfloor \right\} \tag{11}$$

where P and R are respectively the width and height of the target in the previous frame, $a = 1.02$ is the scale factor, $s = 33$ is the length of the scale filter.

Adaptive occlusion judgment and model updating strategy

Target occlusion often occurs during tracking. In this section, we make a detailed analysis and propose an adaptive judgment method. Using the original model update rate in the KCF algorithm, the response of occlusion is analyzed with the FaceOcc1 image sequence in the OTB2015 (Wu et al., 2015) dataset as an example.

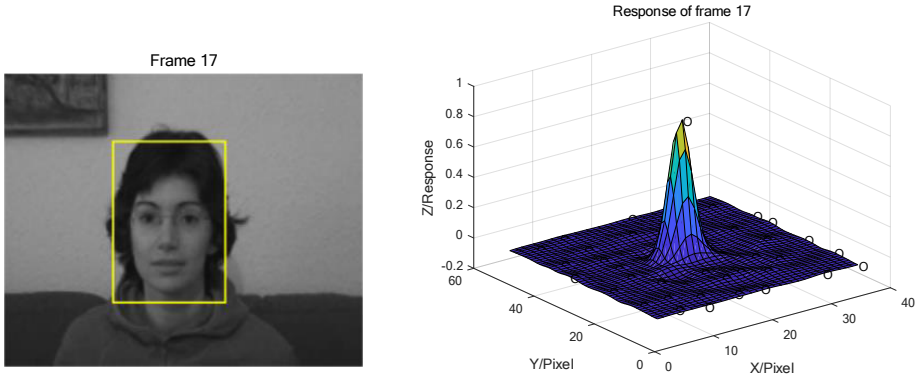


Figure 2. Tracking results and response distribution at frame 17 of the FaceOcc1 sequence without occlusion. Photo credit: Visual Tracker Benchmark.

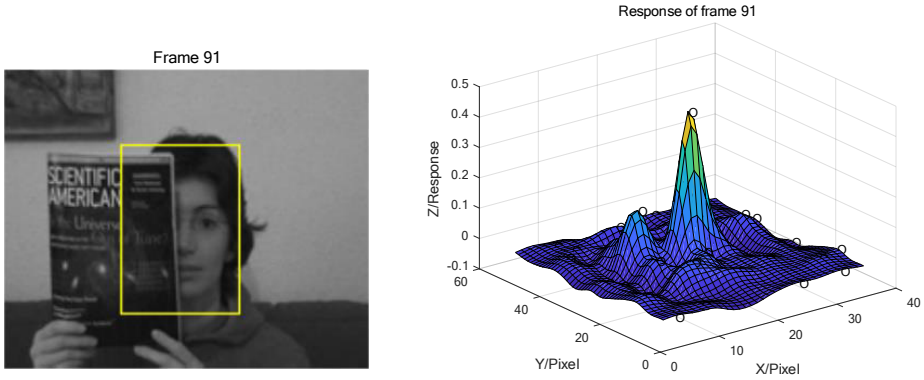


Figure 3. Tracking results and response distribution when occlusion appears at frame 91 of FaceOcc1 sequence. Photo credit: Visual Tracker Benchmark.

From the response distribution in Fig. 2, it can be seen that the main peak of the response in target box during tracking without occlusion is dominated and there is no other obvious peaks. The maximum peak response is close to 1. When the occlusion exists, as shown in Fig. 3, the maximum peak response decreases significantly, and the rest of the peak responses increase and become more prominent. It can be concluded that when the cclusion occurs, the fixed model update strategy learns the features of the occlusion and applies this feature to the search of the next frame which leading to the appearance of other peaks besides the main peak. Hence, the presence of occlusion can be determined based on the response distribution of the target.

Let F_{max} be the maximum peak response in each frame. The number of peak response points that exceed the maximum peak response by a certain proportion, denoted as N, can be expressed as:

$$N=\sum\left(\left(F_{\max}^{'}>\zeta F_{\max}\right)\in S\right) \tag{12}$$

where F'_{max} denotes the peak response other than the maximum peak, S represents the target area, and ζ is a proportionality coefficient which is set to 0.1 in this paper.

Besides the two judgment indicators of F_{max} and N, two more judgment indicators, *APCE* (Average Peak-to-Correlation Energy, (Wang et al., 2017)) and *RSFM* (Ratio between the Second and First Major mode, (Lukežic et al., 2017)) are introduced in this paper to ensure the robustness of the occlusion judgment.

The *APCE* can well reflect the variation of response and changes significantly when the target is obscured. It can be expressed as follows:

$$W_{APCE}=\frac{\left|F_{\max}-F_{\min}\right|^2}{mean\left(\sum_{w,h}\left(F_{w,h}-F_{\min}\right)^2\right)} \tag{13}$$

where F_{min} denotes the minimum value of the response, and $F_{w,h}$ denotes the response value of pixel in w-th row and h-th column. When occlusion appears, the value of W_{APCE} will decrease significantly.

The *RSFM* reflects the prominence of the main peak in the response map and is defined as follows:

$$RSFM=1-\min\left(\frac{F_{\sec ond}}{F_{\max}},\frac{1}{2}\right) \tag{14}$$

where F_{second} represents the response value of the second peak. The larger the value of *RSFM* is, the more prominent the main peak will be and the higher reliability the tracking will have, and vice versa.

In this paper, we use the four evaluation indicators mentioned above, F_{max}, N, W_{APCE} and *RSFM*, to determine whether the target is occluded or not. To verify the four indicators, we select the first 200 frames of the video sequence in FaceOcc1 for test. The relevant results of these four indicators for each frame are calculated and their cumulative averages are shown in Fig. 4.

As shown in Fig. 4, the evaluation indicators fluctuate significantly between frame 85 to 100. The F_{max}, W_{APCE}, *RSFM* are relatively low whereas N is relatively high, which indicates the existence of severe occlusion. By comparing the moments when occlusion appears in the original video sequence, it can be found that the above four indicators satisfy well for the judgment of occlusion as they are complementary to some extent.

In order to accurately determine whether occlusion exists and the degree of occlusion, this paper implements adaptive evaluation by setting dynamic double thresholds. The thresholds are selected based on the average value of each evaluation indicator for the previous $t-1$ frames, namely:

$$\theta(R)_n=\frac{\kappa_n}{t-1}\sum_{i=2}^{t}R^i,R\in\left(F_{\max},W_{APCE},RSFM,N\right),n=1,2 \tag{15}$$

where, κ denotes the weighted coefficient, and $\theta(R)$ denotes the threshold of the corresponding evaluation indicator. For every indicator, there are two thresholds. The weighted coefficient κ of each evaluation indicator is obtained by analyzing the Cumulative Average curve in Fig. 4, as shown in Table 1.

As can be seen from Equation (14), the value of F_{second} rises sharply when there is severe occlusion in current frame, resulting the output of *RSFM* to be 0.5. When it occurs, the lower limit of $\theta(RSFM)_2$

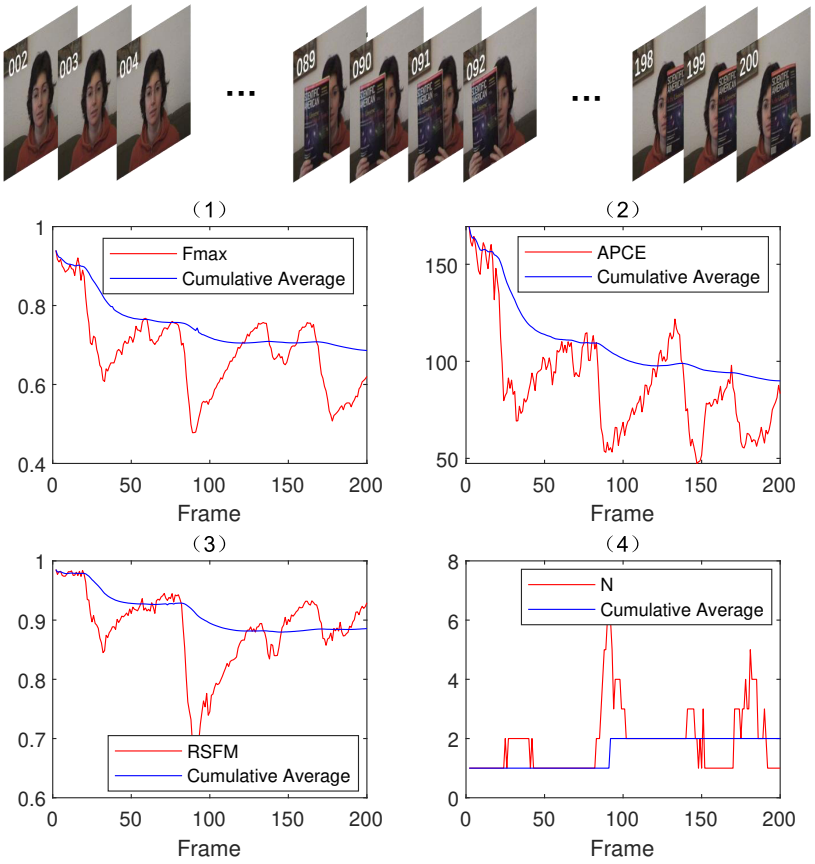


Figure 4. Line graphs of F_{max}, W_{APCE}, $RSFM$ and N for the front 200 frames of FaceOcc1 as an example. Photo credit: Visual Tracker Benchmark.

Table 1. Weighted coefficients of double thresholds for each evaluation indicator.

	κ_1	κ_2
$\theta\left(F_{max}\right)_n$	1	0.85
$\theta\left(W_{APCE}\right)_n$	1	0.7
$\theta\left(RSFM\right)_n$	1	
$\theta\left(N\right)_n$	1	

Table 2. Output values of the four judgment indicators μ_n.

	μ_1	μ_2	μ_3
$\xi_{F_{max}}$	1	1.2	1.5
$\xi_{W_{APCE}}$	1	0.8	0.5
ξ_{RSFM}	1	1.2	1.5
ξ_N	1.2		1

in Table 1 is set to 0.5 particularly. And as N is usually not sensitive to occlusion, it is set to a single threshold for simplicity.

When occlusion appears, the fixed model update rate η will cause the tracker to learn the information of the occlusion object and lead to tracking drift. In this paper, the Aojmus algorithm adaptively generates a suitable model update rate according to the degree of occlusion in current frame. When occlusion occurs, the model update rate is appropriately increased to ensure that the target information of the part not occluded in the tracking frame is fully learned by the model so as to enhance the model's recognition capability, which can be used for accurate localization in next frame. The update strategy is defined as:

$$\xi_R = \begin{cases} \mu_1 & R \geq \theta(R)_1 \\ \mu_2 & \theta(R)_1 > R > \theta(R)_2 \\ \mu_3 & else \end{cases} \tag{16}$$

where, $R \in (F_{\max}, W_{APCE}, RSFM, N)$, $\mu_n (n = 1,2,3)$ represent the output values of occlusion judgment. They are concluded from experiments, as shown in Table 2.

There are four judgment results in the improved algorithm. In order to guarantee that ξ_R can accurately reflect the degree of occlusion, this paper uses a weighted fusion of the four output ξ_R to obtain the final model update rate as follows:

$$\xi = \frac{\xi_{F_{\max}}^2 + \xi_{W_{APCE}}^2 + \xi_{RSFM}^2 + \xi_N^2}{sum(\xi_R)} \tag{17}$$

Finally, we substitute the final model update rate into Equation (6) to obtain:

$$\begin{cases} \hat{\alpha}_t = (1 - \xi * \eta)\,\hat{\alpha}_{t-1} + \xi * \eta\,\hat{\alpha} \\ \hat{x}_t = (1 - \xi * \eta)\,\hat{x}_{t-1} + \xi * \eta\,\hat{x} \end{cases} \tag{18}$$

The proposed algorithm, Aojmus, is presented in Algorithm 1 .

EXPERIMENTS AND ANALYSIS

In this section the proposed Aojmus algorithm is compared with the other 4 relatively excellent tracking algorithms. We use three metrics to evaluate the performance of the algorithm, and select the representative video sequences to compare and analyze the tracking effect.

Experimental environment and parameters

The platform for the experiments in this paper is Matlab 2018a, and the hardware environment is a computer with Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz and 16GB RAM. The parameters of the algorithm are set as follows: the regularization parameter $\lambda = 10^{-4}$ and the initial model update rate $\eta = 0.02$.

In this paper, 66 video sequences provided by OTB-2015 are used for experimental verification. The dataset contains 11 attributes of common scenarios in target tracking, such as occlusion (OCC), deformation (DEF), illumination variation (IV), motion blur (MB), out-of-plane rotation (OPR), fast motion (FM), out-of-view (OV), in-plane rotation (IPR), low resolution (LR), scale variation (SV), and background clutter (BC). The performance evaluations are carried out with quantitative and qualitative analysis.

Algorithm 1 : Aojmus

Input:

Current frame I_t; The video for tracking S_{videos};
Target position p_{t-1} and scale s_{t-1} of previous frame;
The target feature template, $\hat{x}_{t-1}$ and the filter template, $\hat{\alpha}_{t-1}$;
The scale model A_{t-1}^{scale}, B_{t-1}^{scale}.

Output:

Target position p_t and scale s_t of current frame;
The updated target feature template, $\hat{x}_t$ and filter template, $\hat{\alpha}_t$;
The updated scale model A_t^{scale}, B_t^{scale}.

for each $I_t \in S_{videos}$ **do**
  Sample the new patch z^t from I_t at p_{t-1};
  Extract a scale sample z_{scale} from I_t at p_t and s_{t-1};
  Extract the HOG and CN features and fused with Equation (7);
  Calculate the response $\hat{f}(z^t)$ with Equation (5), and get F_{max};
  Calculate N , W_{APCE} and $RSFM$ with Equation (12), (13) and (14), and adaptively judge whether there is occlusion and the scope;
  Get the ξ_R and ξ by Equation (16) and (17);
  Compute the scale correlations y_{scale} using z_{scale}, A_{t-1}^{scale} and B_{t-1}^{scale} in Equation (10);
  Set s_t to the maximum of y_{scale};
  Use Equation (18) to update $\hat{x}_t$ and $\hat{\alpha}_t$ with $\hat{x}_{t-1}$ and $\hat{\alpha}_{t-1}$ adaptively;
  Use Equation (9) to update A_t^{scale} and B_t^{scale} with A_{t-1}^{scale} and B_{t-1}^{scale}.
  Return p_t and the updated $\hat{x}_t$, $\hat{\alpha}_t$, A_t^{scale}, B_t^{scale}.
end for

Experimental comparison and analysis

Quantitative analysis

In order to evaluate the performance of the algorithm in this paper, MSCF (Zheng et al., 2021), Staple (Bertinetto et al., 2016), fDSST (Danelljan et al., 2017b) and KCF algorithms with high performance were selected for comparison. Three statistical criteria of precision (Wu et al., 2013)(shorted as Pr) , success rate (Wu et al., 2013)(shorted as Sr) and tracking speed (shorted as Ts) were used for evaluation respectively. The Pr refers to the error of center position, namely the Euclidean distance in pixel unit, D_t , between the center of tracking box for each frame and the actual center in the benchmark. The final result is expressed with the average of errors.

$$Pr = \frac{D_t}{n} \tag{19}$$

The smaller the value of Pr is, the closer the tracked target center to the actual location is and the better the algorithm performs in terms of pricision.

Let O_t be the overlap between the tracking box in the current frame, B_t, and the actual box in benchmark, B_{bt}. It can be expressed as:

$$O_t = \frac{area(B_t \cap B_{bt})}{area(B_t \cup B_{bt})} \tag{20}$$

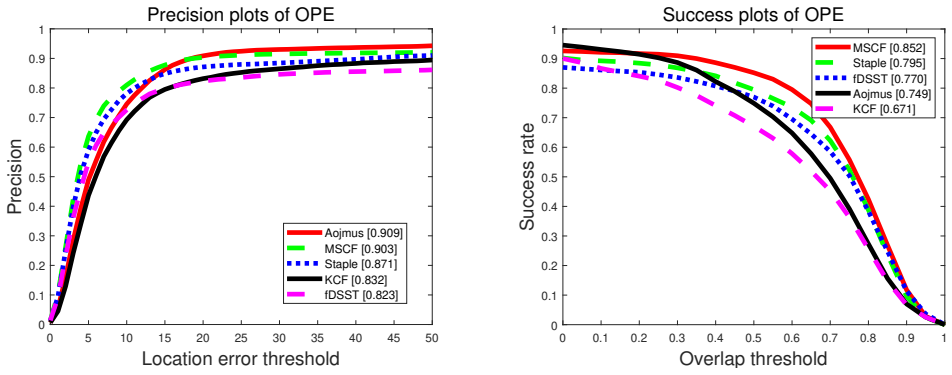
The success rate, Sr, is expressed as the average of O_t whose value is greater than the given threshold in the whole video sequence.

$$Sr = \frac{1}{n}\sum_{t=1}^{n} O_t \tag{21}$$

The value of Sr reflect the number of frames whose tracking box is closer to the real rectangle box. Obviously, the greater the Sr is, the better the performance of the algorithm will be.

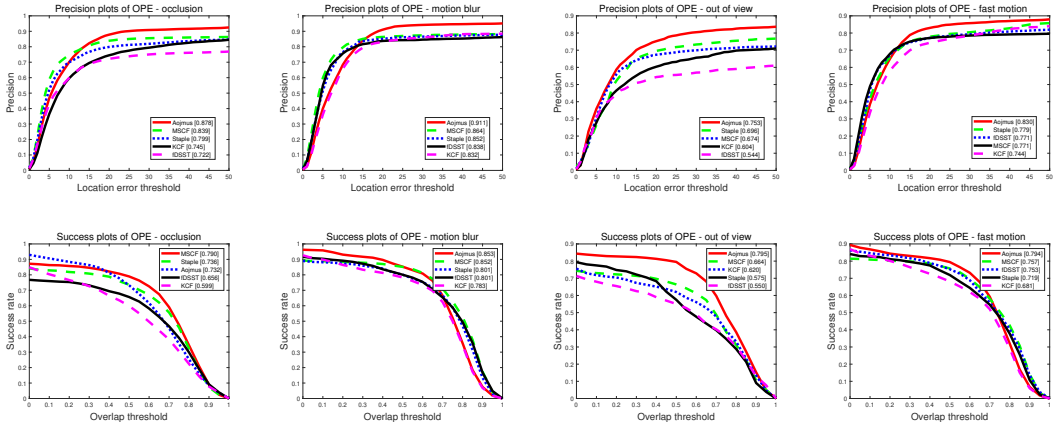
The tracking speed, Ts, refers to the number of video frames processed by the algorithm in each second, also known as frame rate with the unit of FPS (frames per second). It is defined as follows :

$$Ts = \frac{F_{total}}{t_{total}} \tag{22}$$



(a) (b)

Figure 5. The tracking precision plots (a) and success rate plots (b) of our algorithm and others on OTB-2015.



(a) OCC (b) MB (c) OV (d) FM

Figure 6. Precision plots (above) and success rate plots (below) under OCC (a), MB (b), OV (c), FM (d)

where, F_{total} indicates the total number of frames of the video sequence, and t_{total} is the time taken by the algorithm to run the whole video sequence.

To evaluate the performance of the proposed algorithm, Aojmus, we make a comparison with other four algorithms, MSCF, Staple, fDSST and KCF on OTB-2015 as shown in Fig. 5-Fig. 6 and Table 3 -Table 5. The evaluation method is OPE (One-Pass Evaluation), which means that after initializing the target, the whole video sequence is run at once. The location error threshold for precision plots and the overlap threshold for success rate plots are set as 20 pixels and 0.5, respectively.

Fig. 5 illustrates the precision and success rate of these five algorithms for running the whole sequences at once on the OTB-2015 dataset, from which the precision and success rate of Aojmus can be obtained are 0.909 and 0.749, respectively. Compared with others, the Aojmus performs the best in terms of precision and has improved 0.5% than the second ranked MSCF algorithm. Though the success rate is not outstanding, the Aojmus still shows high performance for OCC, MB, OV, and FM as shown in Fig. 6.

Table 3 and Table 4 respectively show the precision and success rate of the five algorithms under 11 attributes. As it can be seen that the Aojmus algorithm performs best on 10 of these attributes. In terms of success rate, the Aojmus appears more robust in dealing with fast motion, motion blur and out of view problems. Despite the disadvantages in other aspects, the Aojmus is not much inferior to other excellent algorithms. For the tracking speed, though the Aojmus is inferior to fDSST and KCF as shown in Table 5, it outperforms others in other aspects as shown in Fig. 5 and Fig. 6. From the above comparisons, the Aojmus exhibits good overall performance.

Table 3. Comparison of precision on different attributes.

Algorithms	OCC	DEF	FM	IPR	MB	OV	OPR	SV	IV	BC	LR
Aojmus	**0.878**	0.900	**0.830**	**0.916**	**0.911**	**0.753**	**0.895**	**0.992**	**0.906**	**0.892**	**0.996**
MSCF	0.839	**0.941**	0.771	0.878	0.864	0.674	0.881	0.868	0.876	0.877	0.988
Staple	0.799	0.878	0.779	0.860	0.852	0.696	0.816	0.851	0.863	0.860	0.797
fDSST	0.722	0.777	0.771	0.799	0.838	0.544	0.759	0.796	0.876	0.891	0.731
KCF	0.722	0.813	0.744	0.818	0.832	0.604	0.803	0.805	0.863	0.882	0.785

Table 4. Comparison of success rate on different attributes.

Algorithms	OCC	DEF	FM	IPR	MB	OV	OPR	SV	IV	BC	LR
Aojmus	0.732	0.727	**0.794**	0.762	**0.853**	**0.795**	0.718	0.651	0.684	0.744	0.662
MSCF	**0.790**	**0.891**	0.757	**0.805**	0.852	0.664	**0.799**	**0.794**	**0.851**	**0.863**	**0.931**
Staple	0.736	0.798	0.719	0.765	0.801	0.575	0.724	0.726	0.793	0.787	0.604
fDSST	0.656	0.710	0.753	0.733	0.801	0.550	0.678	0.715	0.815	0.827	0.705
KCF	0.599	0.661	0.681	0.662	0.783	0.620	0.628	0.520	0.644	0.750	0.285

Table 5. Comparison of tracking speed of our algorithm and others.

Algorithms	Frames Per Second (FPS)
Aojmus	74.85
MSCF	16.01
Staple	9.28
fDSST	83.29
KCF	219.03

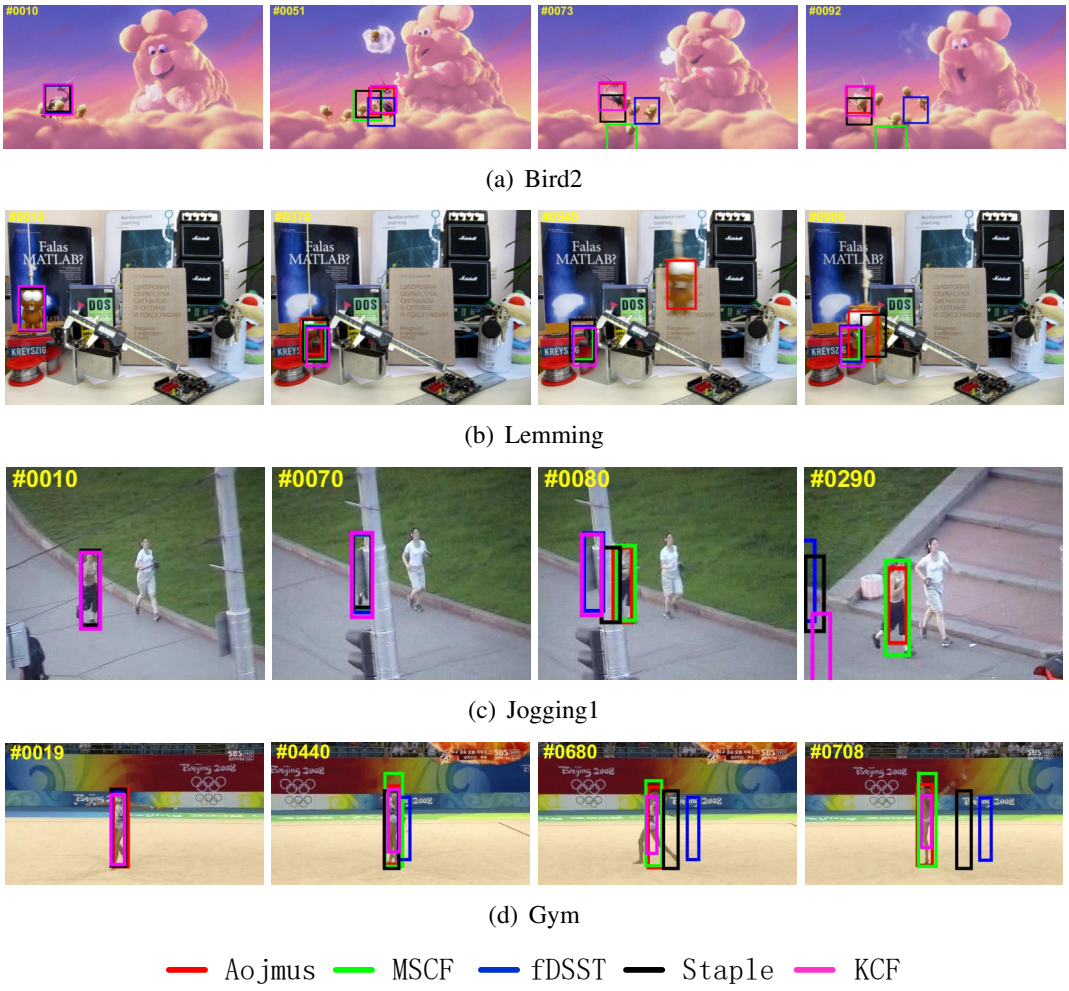


Figure 7. Tracking results of different algorithms on Bird2, Lemming, Jogging1 and Gym. Photo credit: Visual Tracker Benchmark.

Qualitative analysis

Like our previous work in (Wang et al., 2022), in order to better verify the advantages of the proposed algorithm, four groups of representative video sequences are selected for analysis, as shown in Fig. 7.

The video sequence of Bird2 contains OCC, DEF, FM, IPR, and OPR attributes. In frame 10, all the five tracking algorithms work properly. With the progress of tracking, the MSCF, fDSST, and KCF algorithms begin to show significant tracking drift at frame 51 affected by occlusion, deformation, and rotation problems. From frame 73, only the Aojmus and Staple can track accurately after the target(bird) flips.

The video sequence of Lemming contains IV, SV, OCC, FM, OPR, and OV attributes. From frame 10 to frame 370, as the target is not significantly affected by occlusion, fast motion and illumination changes, and the position of the target does not change after the occlusion, all the algorithms can track the target accurately. After the 370th frame, the occluded target reappears. As the tracking models of the other algorithms use fixed update rate and learn non-target information, they are unable to locate the target again in the subsequent frames, while the Aojmus can maintain accurate localization until the end of tracking. In addition, comparing the tracking boxes at 900th frame, it can be shown the Aojmus is also well adaptive to the scale change of the target.

The video sequence of Jogging1 contains OCC, DEF, and OPR attributes. The target is heavily occluded at frames 70 to 80. When the target reappears, the Aojmus is able to accurately locate the target using adaptive occlusion judgment and continue the model update to ensure that tracking is performed

reliably. The other algorithms except MSCF fail to cope with the occlusion problem.

The video sequence of Gym contains SV, DEF, IPR, and OPR attributes. At frame 19, the target begins to rotate and deform, and all algorithms can basically guarantee normal tracking. When it comes to 440th frame, the fDSST algorithm shows obvious drift, and fails to track the target. With the increase of target deformation and rotation, only the Aojmus, MSCF and KCF can keep tracking properly till frame 680. The Aojmus can adjust the tracking box with the scale of target dynamically whereas the MSCF and KCF fail to do so.

The above experimental results show that the Aojmus proposed in this paper can cope well with a variety of influence appearing in the tracking process, especially in the aspects of occlusion, scale change and deformation. On the whole, the Aojmus is robust for target tracking in complex scenes, and typically provides a new idea to deal with occlusion problems.

CONCLUSIONS

In target tracking, scholars have conducted in-depth research in many aspects to be able to predict the position of moving targets more accurately. However, due to the variability of the tracked target and scene, it is not easy to develop an algorithm that takes into account the above 11 influencing factors simultaneously, especially in solving the problems of target occlusion, deformation and scale variation. The previous researches, which typically uses one judgment indicator to address the occlusion problem, can't obtain outstanding overall performance. In this study, considering the complex scenarios and the requirement of mutual-complementarity of technologies, we propose four indicators, F_{max}, N, W_{APCE} and $RSFM$ as conditions to make the occlusion judgment more accurate. Moreover, we introduce an adaptive model updating strategy, fuse the results of the occlusion judgement and apply them into the model updating, which improves the precision in predicting the target position. As tracking is processed frame by frame where different influence factors may be encountered, this study presents a dynamic dual thresholds to compose the update strategy and achieves an accurate judgment of the existence and degree of occlusion, which solves the problem of tracking drift. In order to make full use of the feature information of target and reduce the influence of scale variation, we also incorporate a multi-feature fusion scheme and a scale estimation model in the backbone of the algorithm, which provides a good basis for later obscuration judgments and model updates.

The experimental results show that the Aojmus precedes the other typical tracking algorithms in terms of tracking precision, which has been increased by 0.6% and 3.8% respectively compared with the excellent algorithms, MSCF and Staple. Despite the Aojmus is not the best in terms of success rate, it surpasses the other four compared algorithms with respect to target occlusion, scale variation, fast motion, out-of-plane rotation and deformation. As the Aojmus is based on the kernel correlation filtering method, it runs well in real-time with high tracking speed of 74.85 frames per second, striking a good balance between tracking effectiveness and speed. It can be concluded that the kernel correlation filter-based multi-indicator occlusion judgement mechanism and adaptive model updating strategy can solve the common problems of target tracking while maintaining the overall performance. In future, we plan to investigate the feasibility of synthesizing our method with convolutional neural networks to improve the overall performance further and extend the application to indoor mobile robot and vehicle violation.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

This work is partially supported by the Natural Science Foundation of Fujian Province under the Grant[2018J01637], the Start-up Research Project under the Grant[GY-Z21064] of Fujian University of Technology, the New Engineering Research Project (Construction of Electronic and Electrical Practice Education System and Practice Platform Oriented to New Engineering) of Fujian University of Technology and the Research Project of Experimental Teaching Reform of Fujian University of Technology under the Grant [SJ2018002].

Competing Interests

The authors declare there are no competing interests.

Author Contributions

- Zhiming Cai designed the mathematical method of tracking algorithm, built the overall framework of the project, conceived and designed the experiments, authored and reviewed drafts of the paper, and approved the final draft.
- Zhuangzhuang Wang completed the whole part of the project, tested the feasibility of the algorithm, conceived and designed the experiments, prepared figures and tables, authored drafts of the paper, and approved the final draft.
- Jianchao Huang and Shujing Chen performed the experiments and processed the data, prepared the pictures and tables, and analyzed the results.
- Huabin He performed the experiments, analyzed the data, authored and reviewed drafts of the paper, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

The public test dataset, OTB-2015 (namely TB-100), used in this article are available on Visual Tracker Benchmark, at `http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html`.

The experiment results underlying this article and the source codes are available at GitHub `https://github.com/fzxincai/Aojmus` Zenodo. `https://doi.org/10.5281/zenodo.7563961`

REFERENCES

Al-Battal, A. F., Gong, Y., Xu, L., Morton, T., Du, C., Bu, Y., Lerman, I. R., Madhavan, R., and Nguyen, T. Q. (2021). A cnn segmentation-based approach to object detection and tracking in ultrasound scans with application to the vagus nerve detection. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, pages 3322–3327, Mexico. IEEE.

Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., and Torr, P. (2016). Staple: Complementary learners for real-time tracking. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1404–1409, Las Vegas, NV, USA. IEEE.

Bolme, D. S., Beveridge, J. R., Draper, B. A., and Lui, Y. M. (2010). Visual object tracking using adaptive correlation filters. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010*, pages 2544–2550, San Francisco, CA, USA. IEEE.

Cao, Z., Fu, C., Ye, J., Li, B., and Li, Y. (2021). Hift: Hierarchical feature transformer for aerial tracking. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15437–15446, Montreal, QC, Canada. IEEE.

Dai, K., Wang, D., Lu, H., Sun, C., and Li, J. (2019). Visual tracking via adaptive spatially-regularized correlation filters. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4665–4674, Long Beach, CA, USA. IEEE.

Danelljan, M., Bhat, G., Khan, F. S., and Felsberg, M. (2017a). Eco: Efficient convolution operators for tracking. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6931–6939, Honolulu, HI, USA. IEEE.

Danelljan, M., Häger, G., Khan, F. S., and Felsberg, M. (2014a). Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference*, Nottingham, United kingdom. British Machine Vision Association, BMVA.

Danelljan, M., Häger, G., Khan, F. S., and Felsberg, M. (2017b). Discriminative scale space tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(8):1561–1575.

Danelljan, M., Khan, F. S., Felsberg, M., and Van De Weijer, J. (2014b). Adaptive color attributes for real-time visual tracking. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1090–1097, Columbus, OH, USA. IEEE.

Guo, D., Wang, J., Cui, Y., Wang, Z., and Chen, S. (2020). Siamcar: Siamese fully convolutional classification and regression for visual tracking. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6268–6276, Seattle, WA, USA. IEEE.

Henriques, J. F., Caseiro, R., Martins, P., and Batista, J. (2015). High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596.

Henriques, J. F., Rui, C., Martins, P., and Batista, J. (2012). Exploiting the circulant structure of tracking-by-detection with kernels. In *Proceedings of the 12th European conference on Computer Vision - Volume Part IV*, pages 702–715, Florence, Italy. Springer Verlag.

Huang, Z., Fu, C., Li, Y., Lin, F., and Lu, P. (2019). Learning aberrance repressed correlation filters for real-time uav tracking. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2891–2900, Seoul, Korea (South). IEEE.

Li, H., Li, Y., and Porikli, F. (2016). Deeptrack: Learning discriminative feature representations online for robust visual tracking. *IEEE Transactions on Image Processing*, 25(4):1834–1848.

Liu, C., Yao, X., Zhu, Z., Peng, S., and Zheng, W. (2017). A robust tracking method based on the correlation filter and correcting strategy. In *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, pages 698–702, Chengdu. IEEE.

Lu, M. and Xu, Y. (2019). A survey of object tracking algorithms. *Acta Automatica Sinica*, 45(7):1244–1260.

Lukežic, A., Vojír, T., Zajc, L. C., Matas, J., and Kristan, M. (2017). Discriminative correlation filter with channel and spatial reliability. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4847–4856, Honolulu, HI, USA. IEEE.

Ma, C., Huang, J.-B., Yang, X., and Yang, M.-H. (2015). Hierarchical convolutional features for visual tracking. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3074–3082, Santiago, Chile. IEEE.

Meng, L. and Li, C.-X. (2019). Brief review of object tracking algorithms in recent years: correlated filtering and deep learning. *Journal Of Image And Graphics*, 24(7):1011–1016.

Nam, H. and Han, B. (2016). Learning multi-domain convolutional neural networks for visual tracking. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4293–4302, Las Vegas, NV, USA. IEEE.

Sun, Y., Sun, C., Wang, D., He, Y., and Lu, H. (2019). Roi pooled correlation filters for visual tracking. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5776–5784, Long Beach, CA, USA. IEEE.

Uzkent, B., Hoffman, M. J., Vodacek, A., and Chen, B. (2015). Feature matching with an adaptive optical sensor in a ground target tracking system. *IEEE Sensors Journal*, 15(1):510–519.

Wang, M., Liu, Y., and Huang, Z. (2017). Large margin object tracking with circulant feature maps. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4800–4808, Honolulu, HI, USA. IEEE.

Wang, N., Xi, M., Zhou, W.-G., Li, L., and Li, H.-Q. (2021). Recent advance in deep visual object tracking. *Journal of University of Science and Technology of China*, 51(4):335–344.

Wang, Z., Huang, J., Chen, S., Zhang, Z., and Cai, Z. (2022). Target tracking algorithm based on multi-feature adaptive fusion in complex scenes. In *Proceedings of the 2022 2nd International Conference on Control and Intelligent Robotics*, page 310–315, Nanjing, China. Association for Computing Machinery.

Wei, Z. and Kang, B. (2017). Recent advances in correlation filter-based object tracking: a review. *Journal of Image and Graphics*, 22(8):1017–1033.

Wu, Y., Lim, J., and Yang, M.-H. (2013). Online object tracking: A benchmark. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2411–2418, Portland, OR, USA. IEEE.

Wu, Y., Lim, J., and Yang, M. H. (2015). Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848.

Xiao, J., Zhu, S. P., Huang, H., and Xie, Y. N. (2016). Object detecting and tracking algorithm based on optical flow. *Journal of Northeastern University(Natural Science)*, 37(6):770–774.

Xie, W.-X. and Zhao, T. (2021). Multi-feature adaptive fusion based target tracking algorithm. *Journal of Signal Processing*, 37(4):603–615.

Yang, L. and Zhu, J. (2014). A scale adaptive kernel correlation filter tracker with feature integration. In *13th European Conference on Computer Vision, ECCV 2014*, pages 254–265, Zurich, Switzerland. Springer Verlag.

Zeng, Y., Fu, X., Gao, L., Zhu, J., Li, H., and Li, Y. (2020). Robust multivehicle tracking with wasserstein association metric in surveillance videos. *IEEE Access*, 8:47863–47876.

Zhang, Z. and Peng, H. (2019). Deeper and wider siamese networks for real-time visual tracking. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4586–4595, Long Beach, CA, USA. IEEE.

Zheng, G., Fu, C., Ye, J., Lin, F., and Ding, F. (2021). Mutation sensitive correlation filter for real-time uav tracking with adaptive hybrid label. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 503–509, Xi'an, China. IEEE.

Zhou, T. and Liu, Y. (2021). Long-term person tracking for unmanned aerial vehicle based on human-machine collaboration. *IEEE Access*, 9:161181–161193.