J. Meas. Sci. Instrum., 2024, 15(1): 9-22.
DOI: 10.62756/jmsi.1674-8042.2024002
http://jmsi.publish.founderss.cn/
jmsi@nuc.edu.cn

Spatial-temporal regularized correlation filtering algorithm with adaptive aspect ratio

XU Kai[1,2], LI Ting[1,2], GE Hongwei[1,2*]

1. *School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China;*
2. *Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence, Jiangnan University, Wuxi 214122, China*

***Corresponding author:** GE Hongwei (ghw@jiangnan.edu.cn)

Received: March 5, 2023 **Revised:** April 18, 2023 **Accepted:** April 20, 2023

Abstract: In object tracking, the traditional correlation filtering algorithm is unable to perceive the change of scale aspect ratio for moving targets, and it is easily affected by a complex environment, resulting in tracking failure. Therefore, a spatial-temporal regularized correlation filtering algorithm with adaptive aspect ratio (AAR-SRCF) was proposed. Firstly, the average peak-to-correlation energy (APCE) and peak score were used as references to weigh and fuse each feature response map to achieve accurate results. Additionally, a set of novel one-dimensional boundary filters were presented, integrating near-orthogonality and spatial regularization. These filters can adaptively detect changes in the target scale and aspect ratio by precisely locating the boundaries of the target's bounding box. Moreover, spatial regularization effectively mitigated the negative impact of the boundary effect for boundary filters. Finally, the learning rate of each boundary filter was adjusted separately according to the peak-to-sidelobe ratio (PSR) to prevent the model from degradation. Through extensive experiments on OTB datasets, the proposed algorithm shows excellent tracking performance, achieving better results than other excellent algorithms in each challenge attribute.

Key words: object tracking; correlation filter; spatial regularization; adaptive aspect ratio; template update; response fusion

0 Introduction

Object tracking is one of the important research topics in computer vision. It typically involves detecting changes in the position and scale of a target in video sequences in order to analyze and understand its behavior. This field has broad application prospects in various areas, such as military defense, autopilot, and human-computer interaction[1].

In 2010, Bolme et al.[2], proposed the minimum output sum of squared error tracker (MOSSE), which is the first to use correlation filtering in object tracking. Based on this, the kernelized correlation filter (KCF) [3] combines the histogram of oriented gradients (HOG) and kernel function to effectively address the problem of insufficient samples. Although the above algorithms have achieved good performance, they are limited in handling complex cases. For instance, for the scale variation problem, Danelljan et al.[4] made an accurate scale estimation for robust visual tracking (DSST) by using the scale filter that detects scalechanges by generating targets with different resolutions, but the tracking accuracy is poor. Li et al.[5] proposed the tracking model to integrate boundary and center correlation filters (IBCCF), which uses one-dimensional correlation filters to detect target boundary positions, but it is more computationally intensive. For the feature extraction problem, Danelljan et al.[6] proposed the continuous convolution operator-based tracking algorithm (C-COT) that uses a neural network to extract image features. Yan et al.[7] used the Siamese network for feature extraction and a region proposal subnetwork for target prediction. This is a visual tracking algorithm with the Siamese region proposal network (SiamRPN), which strikes a balance between tracking accuracy and tracking speed, and discards online fine-tuning of the model, making it difficult to perform long-time tracking tasks. To cope with the boundary effect problem, Danelljan et al.[8] proposed the spatial regularized correlation filter (SRDCF) by penalizing the learning at unreliable pixels, but the tracking speed is slow. Tian et al.[9] proposed the spatial-temporal regularized correlation filter (STRCF) by limiting the learning rate of the filter when the target state changes

abruptly, but the tracking performance is poor when occlusion occurs. Based on these works, Fu et al.[10] proposed the automatic spatial-temporal regularized tracking algorithm (AutoTrack), which adaptively adjusts model parameters based on the responses during tracking. This method can better cope with various unforeseen situations in object tracking that cannot be addressed by predefined parameter models.

However, the existing algorithms still have the following drawbacks: they fail to handle changes in target aspect ratio and may experience drift when tracking objects in complex environments, such as those with occlusions or deformation. To overcome these shortcomings, we proposed a spatial-temporal regularized correlation filtering algorithm with adaptive aspect ratio (AAR-SRCF). This algorithm presents three key improvements over AutoTrack. First, a weighted fusion mechanism was designed for the response maps of multiple image features extracted by AutoTrack to improve tracking accuracy. Furthermore, boundary filters were used to replace the scale detection method of AutoTrack, realizing aspect ratio variation detection. And then spatial weight constraints were imposed to mitigate the boundary effect. At last, the learning rate of each boundary filter was adjusted independently according to the peak-to-sidelobe ratio (PSR) score of the response map to improve the robustness.

1 Related work

AutoTrack defines local and global response variations that are calculated based on the response map of the current frame. These parameters are used to adjust the weight coefficients of spatial regularization and update coefficients of temporal regularization to achieve automatic optimization in the tracking process.

1.1 Response variation

Local response variation is defined as $\Pi_1 = [\,|\Pi^1|,..,|\Pi^i|,..,|\Pi^T|\,]$, and the equation for its ith element is

$$\Pi^i = \frac{\boldsymbol{R}_t^i[\,\psi_\Delta\,] - \boldsymbol{R}_{t-1}^i}{\boldsymbol{R}_{t-1}^i}, \tag{1}$$

where T is the length of the response map; $[\,\psi_\Delta\,]$ is for overlapping the main peaks in two response maps; $\boldsymbol{R}_t^i$ represents the ith element in the response map at frame t; and Π_1 specifies the confidence of each pixel in the tracking area.

Global response variation is defined as $\Pi_2 = \left\| \boldsymbol{R}_t[\,\psi_\Delta\,] - \boldsymbol{R}_{t-1} \right\|^2$, which specifies the degree of difference in the overall response map between the current and the previous frame.

1.2 Automatic regularization

AutoTrack automatically optimizes the coefficient for spatial regularization based on Π_1, which is expressed as

$$\breve{\boldsymbol{u}} = \boldsymbol{P}^{\mathrm{T}} \delta \log(\Pi_1 + 1) + \boldsymbol{u}, \tag{2}$$

where $\boldsymbol{P}^{\mathrm{T}} \in \mathbb{R}^{T \times T}$ is used to crop the part of the target corresponding to the filter; δ is a constant to adjust the weight of Π_1; $\boldsymbol{u}$ is inherited from Ref. [9], and the filter penalizes pixels with drastic changes according to $\breve{\boldsymbol{u}}$ to avoid learning the wrong information.

AutoTrack automatically optimizes the update coefficient of temporal regularization by defining a reference $\breve{\theta}$ as

$$\breve{\theta} = \frac{\zeta}{1 + \log(\nu \Pi_2 + 1)}, \Pi_2 \leqslant \phi, \tag{3}$$

where ζ and ν are the hyper parameters. If Π_2 is greater than the threshold ϕ, indicating that there may be multiple peaks in the response map, the filter shall stop learning. Conversely, $\breve{\theta}$ decreases as Π_2 increases, thereby relaxing the filter and speeding up the learning of appearance changes.

Combining the above two points, the objective function of AutoTrack is got as

$$\varepsilon(\boldsymbol{W}_t, \theta_t) = \frac{1}{2}\left\| \boldsymbol{y} - \sum_{k=1}^{K} \boldsymbol{x}_t^k \otimes \boldsymbol{w}_t^k \right\|^2 + \frac{1}{2}\sum_{k=1}^{K}\left\| \breve{\boldsymbol{u}} \odot \boldsymbol{w}_t^k \right\|^2 + \frac{\theta_t}{2}\sum_{k=1}^{K}\left\| \boldsymbol{w}_t^k - \boldsymbol{w}_{t-1}^k \right\|^2 + \frac{1}{2}\left\| \theta_t - \breve{\theta} \right\|^2, \tag{4}$$

where $\boldsymbol{y}$ is the two-dimensional Gaussian label; $\boldsymbol{x}_t^k$ is the kth channel training sample at frame t; $\boldsymbol{W}_t = [\,\boldsymbol{w}_t^1,..,\boldsymbol{w}_t^k,..,\boldsymbol{w}_t^K\,]$ and θ_t are the filter and temporal regularization coefficient at frame t; $\otimes$ and $\odot$ denote the convolution operator and elemental dot product, respectively.

2 Proposed method

2.1 Response fusion

AutoTrack adds the response maps of HOG, color names (CN), and gray features directly and linearly, which cannot give full play to the advantages of each feature. Most of the fusion methods proposed by existing algorithms set the weight coefficients empirically[11] or consider only the response peaks[12], which are not

universal. We used a more expressive fused response map to improve the accuracy of position detection. In the localization process of each frame, the response maps corresponding to the three features were weighted and fused using the response confidence as a reference.

The two-dimensional response map contains not only the information of peak score, but also the information of oscillation degree and multi-peak situation. Therefore, we used average peak-to-correlation energy (APCE) and peak score as confidence indicators to calculate the weight of each feature response map. When the score of APCE is higher, it means that the degree of oscillation is smaller, i. e., the target is less likely affected by interference factors, and the detected target position is more accurate. APCE of HOG at frame t is calculated by

$$A_t^{\mathrm{HOG}}=\frac{\left|\max\left(\boldsymbol{R}_t^{\mathrm{HOG}}\right)-\min\left(\boldsymbol{R}_t^{\mathrm{HOG}}\right)\right|^2}{\frac{1}{T}\sum_{m,n}\left(\boldsymbol{R}_t^{\mathrm{HOG}}(m,n)-\min\left(\boldsymbol{R}_t^{\mathrm{HOG}}\right)\right)^2},\tag{5}$$

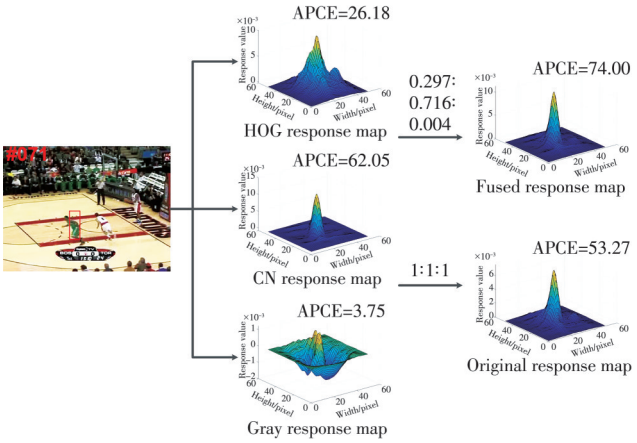
where $\boldsymbol{R}_t^{\mathrm{HOG}}$ is the HOG response map at frame t; and (m,n) is the coordinates of the element. The weight of each response map at frame t is calculated by

$$\begin{cases}A_t=A_t^{\mathrm{HOG}}\max\left(\boldsymbol{R}_t^{\mathrm{HOG}}\right)+A_t^{\mathrm{CN}}\max\left(\boldsymbol{R}_t^{\mathrm{CN}}\right)+\\ \qquad A_t^{\mathrm{Gray}}\max\left(\boldsymbol{R}_t^{\mathrm{Gray}}\right),\\ \omega_t^{\mathrm{HOG}}=A_t^{\mathrm{HOG}}\max\left(\boldsymbol{R}_t^{\mathrm{HOG}}\right)/A_t,\\ \omega_t^{\mathrm{CN}}=A_t^{\mathrm{CN}}\max\left(\boldsymbol{R}_t^{\mathrm{CN}}\right)/A_t,\\ \omega_t^{\mathrm{Gray}}=A_t^{\mathrm{Gray}}\max\left(\boldsymbol{R}_t^{\mathrm{Gray}}\right)/A_t.\end{cases}\tag{6}$$

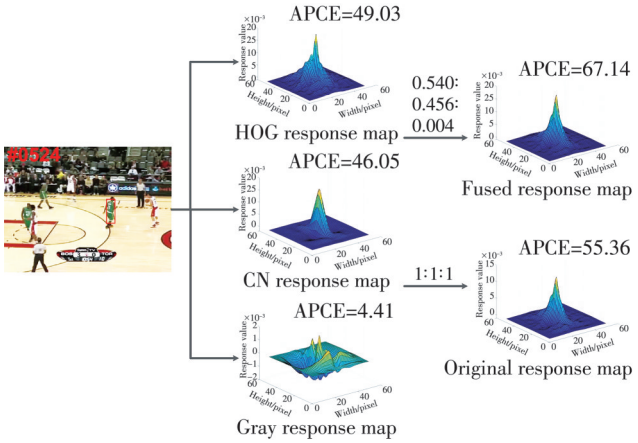
Finally, the fused response map for location detection is obtained by

$$\boldsymbol{R}_t^{\mathrm{Fused}}=\omega_t^{\mathrm{HOG}}\boldsymbol{R}_t^{\mathrm{HOG}}+\omega_t^{\mathrm{CN}}\boldsymbol{R}_t^{\mathrm{CN}}+\omega_t^{\mathrm{Gray}}\boldsymbol{R}_t^{\mathrm{Gray}}.\tag{7}$$

As shown in Fig. 1(a), when the video sequence proceeds to frame 71, the profile of the target changes dramatically and the HOG response map appears multi-peaked, which can be assumed that the tracking result of this feature is quite unreliable.



(a) Frame 71



(b) Frame 524

Fig. 1 Weighted fusion of feature response maps

The tracking result of the CN feature is the most reliable at this time, as shown by the APCE score, so the highest weight percentage is assigned. And in Fig. 1(b), when the video sequence proceeds to frame 524, the profile contour of the target is more consistent with the features learned previously, so the APCE score of the HOG response map is substantially increased. In addition, the fused response map obtained by the weighted fusion mechanism has a higher APCE score than the one obtained by direct fusion, reflecting the effectiveness of our method.

2.2 Adaptive aspect ratio

AutoTrack employs a scale filter to detect targets generated at multiple resolutions and selects the scale with the highest score. However, it is limited in its ability to detect only a fixed aspect ratio and to address object deformations or partial occlusions. This paper proposes a solution to enhance the detection capability of AutoTrack by incorporating boundary filters, which allow for greater flexibility in the size of the target bounding box, enabling both proportional enlargement or shrinkage as well as the independent change in height and width.

2.2.1 Boundary filters

In contrast to scale filters, boundary filters are four one-dimensional filters applied to the four sides of the bounding box. The response output is obtained by performing a correlation operation between the filter and the corresponding tracking region, and the boundary position is determined by the response peak score, thus obtaining the target scale. The tracking region is shown in Fig. 2. Given a bounding box with a center coordinate $(m_{\mathrm{c}},n_{\mathrm{c}})$ and height and width $(b_{\mathrm{height}},b_{\mathrm{width}})$, the center coordinate of the left tracking region is $(m_{\mathrm{left}},n_{\mathrm{left}})=(m_{\mathrm{c}}-b_{\mathrm{width}}/2,n_{\mathrm{c}})$, the height and width is $(l_{\mathrm{height}},l_{\mathrm{width}})=(\alpha_1 b_{\mathrm{height}},\alpha_2 b_{\mathrm{width}})$, where the magnification factor is

inherited from Ref. [5]. For the left and right filters, $(\alpha_1,\alpha_2)=(1.3,1.5)$, and for the up and down filters, $(\alpha_1,\alpha_2)=(1.5,1.3)$. According to the detection range, when the boundary positions change, the boundary filters can detect the changes in height and width from 0 to 2.5 times in each detection process, as well as 25×25 different aspect ratio combinations.

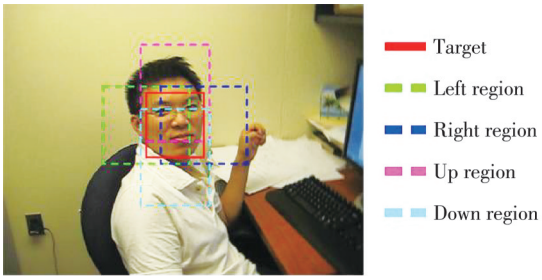


Fig. 2 Target area and boundary filter tracking areas

When training the left filter, the tracking region is treated as a multi-channel representation of one-dimensional $l_{\text{width}}\times1$ vector $\boldsymbol{x}_{\text{left}}=[\boldsymbol{x}_{\text{left}}^1,\boldsymbol{x}_{\text{left}}^2,\cdots,\boldsymbol{x}_{\text{left}}^{l_{\text{height}}}]$, corresponding to the filter $\boldsymbol{w}_{\text{left}}=[\boldsymbol{w}_{\text{left}}^1,\boldsymbol{w}_{\text{left}}^2,\cdots,\boldsymbol{w}_{\text{left}}^{l_{\text{height}}}]$. The objective function is expressed as

$$L_{\text{left}}(\boldsymbol{w}_{\text{left}})=\left\|\sum_{j=1}^{l_{\text{height}}}\boldsymbol{x}_{\text{left}}^j\otimes\boldsymbol{w}_{\text{left}}^j-\boldsymbol{y}_1\right\|^2+\lambda\|\boldsymbol{w}_{\text{left}}\|^2,\quad(8)$$

where $\boldsymbol{y}_1$ is the one-dimensional Gaussian label, λ is the regular term coefficient.

2.2.2 Near-orthogonality and spatial regularization

Ref.[5] shows that the boundary filters give the tracker the ability of aspect ratio adaption. But inaccurate tracking results lead to excessive distortion of the bounding box, which makes the tolerance accumulate during the tracking process and eventually leads to drift. Therefore, this study improves the boundary filters by using both near-orthogonality and spatial regularization.

From Fig. 2, we can see that there is an overlapping part between the boundary tracking region and the center region. $[\tilde{\boldsymbol{w}}_{\text{c}}]_{\text{vec}}$ and $[\tilde{\boldsymbol{w}}_{\text{left}}]_{\text{vec}}$ are defined as the vectorization of the center filter and the left filter in the overlapping part. The response output of the boundary filter should be highest at the boundary of the target and tends to zero at the rest. Moreover, the response output of the center filter should be highest at the center of the target. Therefore, $[\tilde{\boldsymbol{w}}_{\text{c}}]_{\text{vec}}$ and $[\tilde{\boldsymbol{w}}_{\text{left}}]_{\text{vec}}$ should be approximately orthogonal, i.e. $[\tilde{\boldsymbol{w}}_{\text{left}}]_{\text{vec}}^{\text{T}}[\tilde{\boldsymbol{w}}_{\text{c}}]_{\text{vec}}\approx0$.

The boundary effect is commonly found in correlation filtering algorithms, which is caused by the periodic expansion of training samples. When performing the convolution operation, boundary tracking region with half foreground and half background also generates false negative samples, which in turn reduces the discriminative ability of the filter. Therefore, in this study, the regular term in Eq. (8) is replaced by the Gikhonov regular term, implying the introduction of the spatial weight constraint. As shown in Fig. 3, image features close to the background are less reliable than those close to the target, so the high region indicates higher weight constraints assigned to the boundary filter, thereby increasing the penalty and reducing the emphasis on the background information, and the low region indicates low weight constraints assigned.

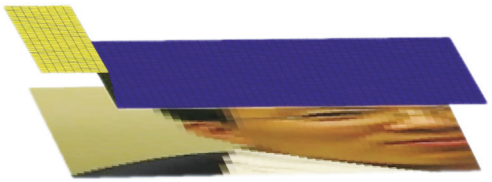


Fig. 3 Visualization of spatial weight constraints

Eq. (8) is rewritten as

$$L_{\text{left}}(\boldsymbol{w}_{\text{left}})=\left\|\sum_{j=1}^{l_{\text{height}}}\boldsymbol{x}_{\text{left}}^j\otimes\boldsymbol{w}_{\text{left}}^j-\boldsymbol{y}_1\right\|^2+\left\|\boldsymbol{\varphi}\odot\boldsymbol{w}_{\text{left}}\right\|^2,\quad(9)$$

where $\boldsymbol{\varphi}$ is the spatial weight constraints.

Fig.4 shows the tracking performance comparison of the algorithm before and after optimization in this subsection for two video sequences with aspect ratio variations (FleetFace, CarScale). Compared with AutoTrack, the algorithm in this paper is not only able to detect the scale of the target, but also to sense the aspect ratio variation. Moreover, the optimized algorithm encloses the range of the target more accurately and does not lose the target due to excessive deformation.

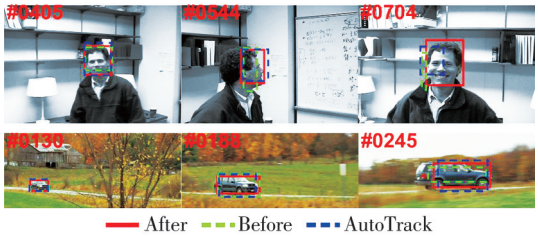


Fig. 4 Comparison of algorithms before and after optimization

2.2.3 ADMM optimization

The objective function of the proposed algorithm is as

$$\Phi(\boldsymbol{W})=\sum_{k\in\Psi}L_k(\boldsymbol{w}_k)+\varepsilon(\boldsymbol{W}_{\text{c}},\theta_{\text{c}})+\sum_{k\in\Psi}([\widetilde{\boldsymbol{W}}_{\text{c}}]_{\text{vec}}^{\text{T}}[\tilde{\boldsymbol{w}}_k]_{\text{vec}})^2,\quad(10)$$

where $\Psi=\{\text{left},\text{right},\text{up},\text{down}\}$; $\boldsymbol{W}=\{\boldsymbol{W}_{\text{c}},\boldsymbol{W}_{\text{left}},\boldsymbol{W}_{\text{up}},\boldsymbol{W}_{\text{down}}\}$; $L_k(\boldsymbol{w}_k)$ and $\varepsilon(\boldsymbol{W}_{\text{c}},\theta_{\text{c}})$ are as Eq. s(9) and (4), respectively; $\widetilde{\boldsymbol{W}}_{\text{c}}$ and $\tilde{\boldsymbol{w}}_k$ are center filter and boundary filters for the overlapping part, respectively. We use alternating direction method of multipliers (ADMM) to solve Eq. (10). The auxiliary variables $\boldsymbol{g}=\boldsymbol{W}_{\text{c}}$ and $\boldsymbol{v}_k=$

$\boldsymbol{w}_k$ are set to obtain the augmented Lagrangian function of Eq. (10), where $\boldsymbol{N}=\{\boldsymbol{W}, \boldsymbol{g}, \boldsymbol{p}, \boldsymbol{v}_k, \boldsymbol{q}_k\}$; $\boldsymbol{p}$ and $\boldsymbol{q}_k$ are Lagrange multipliers in scaled form; σ and τ_k are penalty coefficients.

The augmented Lagran gian function is expressed as

$$\Phi(\boldsymbol{N})=\sum_{k\in\Psi}L_k(\boldsymbol{w}_k)+\varepsilon(\boldsymbol{W}_c,\theta_c)+\sum_{k\in\Psi}([\tilde{\boldsymbol{g}}]_{\text{vec}}^{\mathrm{T}}[\tilde{\boldsymbol{v}}_k]_{\text{vec}})^2+\sigma\|\boldsymbol{W}_c-\boldsymbol{g}-\boldsymbol{p}\|^2+\sum_{k\in\Psi}\tau_k\|\boldsymbol{w}_k-\boldsymbol{v}_k-\boldsymbol{q}_k\|^2, \quad (11)$$

Then, the following sub-problems need to be solved in turn

$$\begin{cases}\boldsymbol{W}_c^{(I+1)}=\arg\min\limits_{\boldsymbol{W}_c}\left\{\varepsilon(\boldsymbol{W}_c,\theta_c)+\sigma^{(I)}\|\boldsymbol{W}_c-\boldsymbol{g}^{(I)}-\boldsymbol{p}^{(I)}\|^2\right\},\\ \boldsymbol{g}^{(I+1)}=\arg\min\limits_{\boldsymbol{g}}\left\{\|[\tilde{\boldsymbol{g}}]_{\text{vec}}{}^{\mathrm{T}}\boldsymbol{S}_k^{(I)}\|^2+\sigma^{(I)}\|\boldsymbol{W}_c^{(I)}-\boldsymbol{g}-\boldsymbol{p}^{(I)}\|^2\right\},\\ \boldsymbol{w}_k^{(I+1)}=\arg\min\limits_{\boldsymbol{w}_k}\left\{L_k(\boldsymbol{w}_k)+\tau_k^{(I)}\|\boldsymbol{w}_k-\boldsymbol{v}_k^{(I)}-\boldsymbol{q}_k^{(I)}\|^2\right\},\\ \boldsymbol{v}_k^{(I+1)}=\arg\min\limits_{\boldsymbol{v}_k}\left\{\|[\tilde{\boldsymbol{g}}^{(I)}]_{\text{vec}}{}^{\mathrm{T}}[\tilde{\boldsymbol{v}}_k]_{\text{vec}}\|^2+\tau_k^{(I)}\|\boldsymbol{w}_k^{(I)}-\boldsymbol{v}_k-\boldsymbol{q}_k^{(I)}\|^2\right\},\\ \boldsymbol{p}^{(I+1)}=\boldsymbol{p}+\sigma^{(I)}(\boldsymbol{g}^{(I+1)}-\boldsymbol{W}_c^{(I+1)}),\\ \boldsymbol{q}_k^{(I+1)}=\boldsymbol{q}_k+\tau_k^{(I)}(\boldsymbol{v}_k^{(I+1)}-\boldsymbol{w}_k^{(I+1)}),\end{cases} \quad (12)$$

where $\boldsymbol{S}_k^{(I)}=[[\tilde{\boldsymbol{v}}_{\text{left}}^{(I)}]_{\text{vec}},[\tilde{\boldsymbol{v}}_{\text{right}}^{(I)}]_{\text{vec}},[\tilde{\boldsymbol{v}}_{\text{up}}^{(I)}]_{\text{vec}},[\tilde{\boldsymbol{v}}_{\text{down}}^{(I)}]_{\text{vec}}]$; (I) and $(I+1)$ are the iteration indexes.

1) Sub-problem $\boldsymbol{W}_c^{(I+1)}$

The first row of Eq. (12), which is the objective function of the center filter, satisfies the conditions for optimization using ADMM.

Let $\hat{\boldsymbol{h}}=\sqrt{T}\,\boldsymbol{F}\boldsymbol{w}_c$, ($\hat{\boldsymbol{H}}=[\hat{\boldsymbol{h}}^1,\hat{\boldsymbol{h}}^2,..,\hat{\boldsymbol{h}}^K]$) be the auxiliary variable, where $\boldsymbol{F}$ is the Fourier matrix, T is the length of the filter, and $\wedge$ denotes the Fourier form of the variable.

Then the augmented Lagrangian function in the frequency domain of the first row of Eq. (12) can be expressed as

$$E(\boldsymbol{W}_c,\theta_c,\hat{\boldsymbol{H}},\hat{\boldsymbol{M}})=\frac{1}{2}\left\|\hat{\boldsymbol{y}}-\sum_{k=1}^{K}\hat{\boldsymbol{x}}^k\odot\hat{\boldsymbol{h}}^k\right\|^2+\frac{1}{2}\sum_{k=1}^{K}\|\breve{\boldsymbol{u}}\odot\boldsymbol{w}_c^k\|^2+\frac{\theta_c}{2}\sum_{k=1}^{K}\|\hat{\boldsymbol{h}}^k-\hat{\boldsymbol{h}}_{\text{pre}}^k\|^2+\frac{1}{2}\|\theta_c-\breve{\theta}\|^2+\sigma^{(I)}\sum_{k=1}^{K}\|\sqrt{T}\,\boldsymbol{F}\boldsymbol{w}_c^k-\hat{\boldsymbol{g}}^{k,(I)}-\hat{\boldsymbol{p}}^{k,(I)}\|^2+\frac{\gamma}{2}\sum_{k=1}^{K}\|\sqrt{T}\,\boldsymbol{F}\boldsymbol{w}_c^k-\hat{\boldsymbol{h}}^k-\hat{\boldsymbol{m}}^k\|^2, \quad (13)$$

where $\boldsymbol{M}=[\boldsymbol{m}^1,\boldsymbol{m}^2,..,\boldsymbol{m}^K]$ is the Lagrange multiplier, $\hat{\boldsymbol{h}}_{\text{pre}}^k$ indicates the result calculated in the previous frame, and γ is the penalty coefficient. The sub-problems to be solved sequentially are

$$\begin{cases}\hat{\boldsymbol{H}}^{(i+1)}=\arg\min\limits_{\hat{\boldsymbol{H}}}\left\{\frac{1}{2}\left\|\hat{\boldsymbol{y}}-\sum_{k=1}^{K}\hat{\boldsymbol{x}}^k\odot\hat{\boldsymbol{h}}^k\right\|^2+\frac{\theta_c^{(i)}}{2}\sum_{k=1}^{K}\|\hat{\boldsymbol{h}}^k-\hat{\boldsymbol{h}}_{\text{pre}}^k\|^2+\right.\\ \left.\frac{\gamma^{(i)}}{2}\sum_{k=1}^{K}\|\sqrt{T}\,\boldsymbol{F}\boldsymbol{w}_c^{k,(i)}-\hat{\boldsymbol{h}}^k-\hat{\boldsymbol{m}}^{k,(i)}\|^2\right\},\\ \boldsymbol{w}_c^{k,(i+1)}=\arg\min\limits_{\boldsymbol{w}_c^k}\left\{\frac{1}{2}\|\breve{\boldsymbol{u}}\odot\boldsymbol{w}_c^k\|^2+\frac{\gamma^{(i)}}{2}\|\sqrt{T}\,\boldsymbol{F}\boldsymbol{w}_c^k-\hat{\boldsymbol{h}}^{k,(i)}-\hat{\boldsymbol{m}}^{k,(i)}\|^2+\right.\\ \left.\sigma^{(I)}\|\sqrt{T}\,\boldsymbol{F}\boldsymbol{w}_c^k-\hat{\boldsymbol{g}}^{k,(I)}-\hat{\boldsymbol{p}}^{k,(I)}\|^2\right\},\\ \theta_c^{(i+1)}=\arg\min\limits_{\theta_c}\left\{\frac{\theta_c}{2}\sum_{k=1}^{K}\|\hat{\boldsymbol{h}}^{k,(i)}-\hat{\boldsymbol{h}}_{\text{pre}}^{k,(i)}\|^2+\frac{1}{2}\|\theta_c-\breve{\theta}\|^2\right\},\\ \hat{\boldsymbol{M}}^{(i+1)}=\hat{\boldsymbol{M}}+\gamma^{(i)}(\hat{\boldsymbol{H}}^{(i+1)}-\hat{\boldsymbol{W}}_c^{(i+1)}),\end{cases} \quad (14)$$

where (i) and $(i+1)$ are the iteration indexes of the current sub-problem.

1.1) Sub-problem $\boldsymbol{w}_c^{k,(i+1)}$

$$\boldsymbol{w}_c^{k,(i+1)}=\arg\min_{\boldsymbol{w}_c^k}\left\{\frac{1}{2}\|\breve{\boldsymbol{u}}\odot\boldsymbol{w}_c^k\|^2+\frac{\gamma^{(i)}}{2}\|\sqrt{T}\,\boldsymbol{F}\boldsymbol{w}_c^k-\hat{\boldsymbol{h}}^{k,(i)}-\hat{\boldsymbol{m}}^{k,(i)}\|^2+\sigma^{(I)}\|\sqrt{T}\,\boldsymbol{F}\boldsymbol{w}_c^k-\hat{\boldsymbol{g}}^{k,(I)}-\hat{\boldsymbol{p}}^{k,(I)}\|^2\right\}=\frac{2\sigma^{(I)}T(\boldsymbol{g}^{k,(I)}+\boldsymbol{p}^{k,(I)})+\gamma^{(i)}T(\boldsymbol{h}^{k,(i)}+\boldsymbol{m}^{k,(i)})}{\breve{\boldsymbol{u}}\odot\breve{\boldsymbol{u}}+2\sigma^{(I)}T+\gamma^{(i)}T}. \quad (15)$$

1.2) The solution of the sub-problems $\hat{\boldsymbol{H}}^{(i+1)}$ and $\theta_c{}^{(i+1)}$ and the update procedure of the Lagrange multiplier $\boldsymbol{M}^{(i+1)}$ are referred to Ref.[10].

2) Sub-problem $\boldsymbol{w}_k^{(I+1)}$

The third row of Eq. (12), which is the objective function of the boundary filter, also satisfies the conditions of the ADMM method. We set $\boldsymbol{v}_k'=\boldsymbol{w}_k$ as the auxiliary variable, and the augmented Lagrangian function is obtained as

$$\mathrm{L}(\boldsymbol{w}_k,\boldsymbol{v}_k',\boldsymbol{q}_k')=\left\|\boldsymbol{x}_k\otimes\boldsymbol{w}_k-\boldsymbol{y}_k\right\|^2+\left\|\boldsymbol{\varphi}\odot\boldsymbol{v}_k'\right\|^2+\tau_k^{(I)}\left\|\boldsymbol{w}_k-\boldsymbol{v}_k^{(I)}-\boldsymbol{q}_k^{(I)}\right\|^2+\tau_k'\left\|\boldsymbol{w}_k-\boldsymbol{v}_k'-\boldsymbol{q}_k'\right\|^2, \quad (16)$$

where $\boldsymbol{q}_k'$ denotes the Lagrange multiplier, τ_k' denotes the penalty coefficient of the current sub-problem. The following sub-problems need to be solved in turn

$$\begin{cases}\boldsymbol{w}_k^{(i+1)}=\arg\min\limits_{\boldsymbol{w}_k}\{\left\|\boldsymbol{x}_k\otimes\boldsymbol{w}_k-\boldsymbol{y}_k\right\|^2+\\ \tau_k^{(I)}\left\|\boldsymbol{w}_k-\boldsymbol{v}_k^{(I)}-\boldsymbol{q}_k^{(I)}\right\|^2+\tau_k'^{(i)}\left\|\boldsymbol{w}_k-\boldsymbol{v}_k'^{(i)}-\boldsymbol{q}_k'^{(i)}\right\|^2\},\\ \boldsymbol{v}_k'^{(i+1)}=\arg\min\limits_{\boldsymbol{v}_k'}\{\left\|\boldsymbol{\varphi}\odot\boldsymbol{v}_k'\right\|^2+\tau_k'^{(i)}\left\|\boldsymbol{w}_k^{(i)}-\boldsymbol{v}_k'-\boldsymbol{q}_k'^{(i)}\right\|^2\},\\ \boldsymbol{q}_k'^{(i+1)}=\boldsymbol{q}_k'+\tau_k'^{(i)}(\boldsymbol{v}_k'^{(i+1)}-\boldsymbol{w}_k^{(i+1)}).\end{cases} \quad (17)$$

2.1) Sub-problem $\boldsymbol{w}_k^{(i+1)}$

$$\boldsymbol{w}_k^{(i+1)}=\arg\min_{\boldsymbol{w}_k}\{\left\|\boldsymbol{x}_k\otimes\boldsymbol{w}_k-\boldsymbol{y}_k\right\|^2+\tau_k^{(I)}\left\|\boldsymbol{w}_k-\boldsymbol{v}_k^{(I)}-\boldsymbol{q}_k^{(I)}\right\|^2+\tau_k'^{(i)}\left\|\boldsymbol{w}_k-\boldsymbol{v}_k'^{(i)}-\boldsymbol{q}_k'^{(i)}\right\|^2\}. \quad (18)$$

Since the calculation of a response element depends only on the corresponding elements of the filter and sample in all channels, Eq. (18) can be vectorized and then solved as

$$\Gamma_j^{(i+1)}(\hat{\boldsymbol{w}}_k)=\frac{1}{T(\tau_k^{(I)}+\tau_k'^{(i)})}\left(\mathrm{I}-\frac{\Gamma_j(\hat{\boldsymbol{x}}_k)\Gamma_j(\hat{\boldsymbol{x}}_k)^{\mathrm{T}}}{T(\tau_k^{(I)}+\tau_k'^{(i)})+\Gamma_j(\hat{\boldsymbol{x}}_k)^{\mathrm{T}}\Gamma_j(\hat{\boldsymbol{x}}_k)}\right)\left(\Gamma_j(\hat{\boldsymbol{x}}_k)\hat{\boldsymbol{y}}_j+T\tau_k^{(I)}(\Gamma_j(\boldsymbol{v}_k^{(I)})+\Gamma_j(\boldsymbol{q}_k^{(I)}))+T\tau_k'^{(i)}(\Gamma_j(\boldsymbol{v}_k'^{(i)})+\Gamma_j(\boldsymbol{q}_k'^{(i)}))\right), \quad (19)$$

where $\Gamma_j(\,)$ denotes the vectorization of the j th pixel over all channels.

2.2) Sub-problem $\boldsymbol{v}_k'^{(i+1)}$

$$\boldsymbol{v}_k'^{(i+1)}=\arg\min_{\boldsymbol{v}_k'}\{\left\|\boldsymbol{\varphi}\odot\boldsymbol{v}_k'\right\|^2+\tau_k'^{(i)}\left\|\boldsymbol{w}_k^{(i)}-\boldsymbol{v}_k'-\boldsymbol{q}_k'^{(i)}\right\|^2\}=\frac{\tau_k'^{(i)}(\boldsymbol{w}_k^{(i)}-\boldsymbol{q}_k'^{(i)})}{\boldsymbol{\varphi}\odot\boldsymbol{\varphi}+\tau_k'^{(i)}}. \quad (20)$$

2.3) Lagrange multiplier update

$$\boldsymbol{q}_k'^{(i+1)}=\boldsymbol{q}_k'+\tau_k'^{(i)}(\boldsymbol{v}_k'^{(i+1)}-\boldsymbol{w}_k^{(i+1)}), \quad (21)$$

where the penalty coefficient τ_k' starting at 1 has the following iterative formula

$$\tau_k'^{(i+1)}=\min(\tau_{\max},\beta_k\tau_k'^{(i)}),\quad \tau_{\max}=10\,000,\ \beta_k=10. \quad (22)$$

3) The solution of the sub-problems $\boldsymbol{g}^{(I+1)}$ and $\boldsymbol{v}_k^{(I+1)}$, the update procedure of the Lagrange multipliers $\boldsymbol{p}^{(I+1)}$ and $\boldsymbol{q}_k^{(I+1)}$ are referred to Ref.[5].

After the above iterative solution, we can obtain the center filter and the boundary filters, and use them for the detection in the next frame, the center filter response is calculated by

$$\boldsymbol{R}_t=F^{-1}[\hat{\boldsymbol{z}}_t\odot\hat{\boldsymbol{W}}_{t-1,\mathrm{c}}], \quad (23)$$

where F^{-1} denotes the inverse Fourier transform, $\hat{\boldsymbol{z}}_t$ denotes the Fourier form of the center tracking region detection sample at frame t. Each boundary filter response is calculated by

$$\boldsymbol{R}_{t,k}=F^{-1}[\hat{\boldsymbol{z}}_{t,k}\odot\hat{\boldsymbol{w}}_{t-1,k}], \quad (24)$$

where $\hat{\boldsymbol{z}}_{t,k}$ denotes the Fourier form of the detection sample for each boundary tracking region at frame t.

2.3 Template update strategy

There are two issues with updating the templates of each boundary filter using the same and fixed learning rate: Firstly, in the case of complex motion scenes, the learning rate should be influenced by the reliability of the tracking results and should be variable. Secondly, in the case of target occlusion, the learning rate of each boundary filter should be adjusted according to its own condition, so that the occluded boundary can avoid making more tolerance while ensuring that the correct boundary can quickly learn.

Since the one-dimensional response maps of the boundary filters are typically single-peak, we use PSR, which focuses on the peak score, as a confidence indicator to design a flexible boundary filter template update strategy. If the ratio of the at the current frame to the historical average is greater PSR score than the threshold, it is considered that the tracking result is reliable, and the template is updated with a larger learning rate; otherwise, the template is updated with a smaller learning rate. The PSR calculation equation of the left filter is expressed as

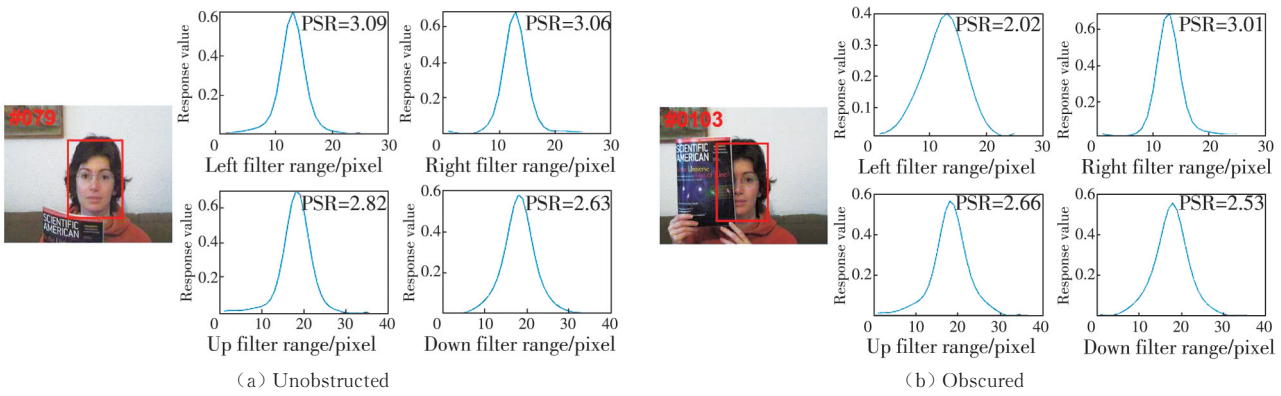
$$P_{t,\mathrm{left}}=\frac{\max(R_{t,\mathrm{left}})-\mathrm{mean}(R_{t,\mathrm{left}})}{o(R_{t,\mathrm{left}})}, \quad (25)$$

where $\boldsymbol{R}_{t,\mathrm{left}}$ is the response map of the left filter at frame t and $o(\,)$ denotes the standard deviation. The learning rate of the left filter at frame t is calculated by

$$\eta_{t,\mathrm{left}}=\begin{cases}\eta_1, & \dfrac{P_{t,\mathrm{left}}}{\bar{P}_{\mathrm{left}}}\geqslant\chi,\\ \eta_2, & \dfrac{P_{t,\mathrm{left}}}{\bar{P}_{\mathrm{left}}}<\chi,\end{cases} \quad (26)$$

where $\bar{P}_{\mathrm{left}}$ is the historical average. Empirically, $\eta_1=0.008$, $\eta_2=0.002$, $\chi=5/6$.

As shown in Fig.5(a), at frame 79 of the FaceOcc1, the response peaks of four boundary filters are in a sharp form, but due to the book being close to the bottom of the face, the below filter is slightly affected. According to the PSR score, all four boundary filters use a larger learning rate. As shown in Fig.5(b), at frame 103, the book is covering one side of the face, at which point the response peak of the left filter clearly widens, the PSR score drops significantly, and the up and down filters are also affected to some extent. Therefore, the left filter uses a smaller learning rate, while the other boundary filters still use a larger learning rate.



(a) Unobstructed (b) Obscured

Fig. 5 Response maps of boundary filters

2.4 Algorithm procedure

The flow chart of the proposed algorithm AAR-SRCF is shown in Fig.6. The input of the algorithm is the initial bounding box of the tracking target and the parameter settings of the model, and the output is the predicted target bounding box.

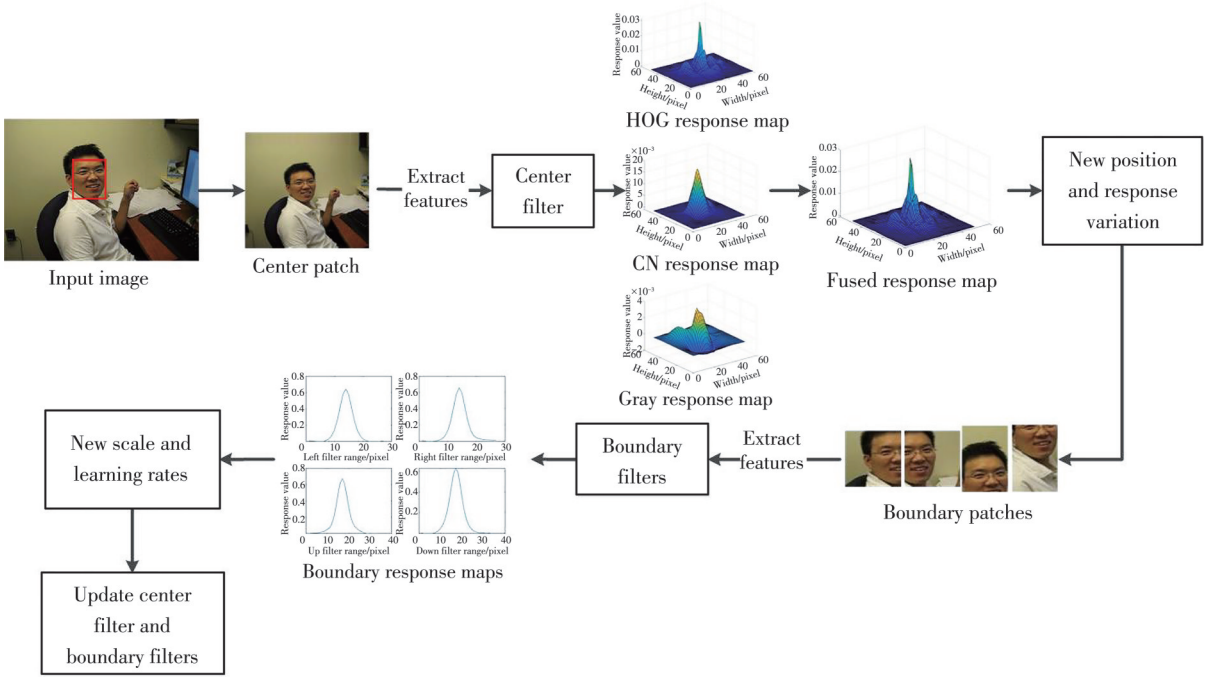


Fig. 6 Flow chart of AAR-SRCF

In the first frame, the algorithm skips the tracking steps and initializes the model by solving Eq.(10). In the subsequent frames, first, the center patch is obtained and features are extracted from the predicted position of the previous frame, the response map of each feature is obtained via Eq. (23), the fused response map is obtained via Eq. (7), and finally the position of the tracking target is obtained via Newton's iteration method, besides, the response variations are updated according to the fused response map.

Then, the boundary patches are obtained and features are extracted according to the current bounding box, the response map of each boundary filter is obtained via Eq.(24), and finally the boundary positions are obtained according to the peak of the response maps, besides, the learning rate of the boundary filter is decided according to the PSR of the response map. After the prediction is finished, the center filter and boundary filters are

updated alternately by solving Eq. (10). The algorithm cycles through the above steps until the last frame. The pseudo-code of the proposed algorithm is as follows.

Algorithm 1: Spatial-temporal regularized correlation filtering algorithm with adaptive aspect ratio

Input: initial target bounding box and other initialization parameters
Output: predicted target bounding box

1. **For** frame 1 to n **do**
2. **If** frame = 1 **then**
3. Initialize the center filter and boundary filters;
4. **Else**
5. Obtain center patch based on the bounding box obtained from the previous frame and extract features;
6. Obtain the response maps of each feature via Eq.(23);
7. Calculate weights of each response map via Eq.(5)(6);
8. Obtain the fused response map via Eq.(7);
9. Determine the target position;
10. Update the response variations based on Section 1.1;
11. Obtain boundary patches based on the bounding box obtained from the previous steps and extract features;
12. Obtain the response maps of each boundary filter via Eq.(24);
13. Determine the boundary positions;
14. **If** PSR of the boundary filter > threshold **then**
15. Learning rate of the boundary filter = η_1;
16. **Else**
17. Learning rate of the boundary filter = η_2;
18. **End if**
19. **While** ADMM iterative **do**
20. Update center filter and boundary filters alternately by solving Eq.(10);
21. **End while**
22. **End if**
23. **End for**

3 Experiments

3.1 Implementation details

Experiments of tracking performance evaluation were conducted using MATLAB R2018a on a PC with an Intel(R) Xeon(R) E5-2695 v3 CPU, 2.3 GHz processor, and 128G RAM.

For the center filter, we set the spatial-temporal regularization parameters as $\delta = 0.2$, $\zeta = 13$, $\nu = 2 \times 10^{-5}$, response variation threshold as $\phi = 3\ 000$, and ADMM iterations as 3. For the boundary filters, to get the suitable learning rate parameters for the proposed algorithm, four different sets of experiments were set up for performance comparison, distance precision (DP) and area under the curve (AUC) were chosen as evaluation metrics, and the larger the metric value, the better the performance. As shown in Table 1, the final learning rate parameters are $\eta_1 = 0.008$, $\eta_2 = 0.002$, and the threshold $\chi = 5/6$, ADMM iterations is 2. Additional parameters refer to AutoTrack and IBCCF.

Table 1 Performance comparison of different learning rate parameters

η_1	η_2	χ	DP/%	AUC/%
0.01	0.008	5/6	0.857	0.640
0.01	0.005	5/6	0.861	0.643
0.01	0.001	5/6	0.859	0.641
0.008	0.002	5/6	0.883	0.661

3.2 Dataset and evaluation metrics

In this study, experiments were conducted on OTB-2013[13], OTB-100[14], and OTB-50 datasets. OTB-50 is selected from OTB-100 and consists of 50 video sequences with only a small overlap with OTB-2013. The OTB dataset contains 11 different tracking attributes, including fast motion (FM), background clutter (BC), motion blur (MB), deformation (DEF), illumination variation (IV), in-plane rotation (IPR), low resolution (LR), occlusion (OCC), out-of-plane rotation (OPR), out of view (OV), and scale variation (SV), which are complex and authoritative.

To analyze the performance of the algorithm, the one-pass evaluation[13] was chosen as the evaluation metric, which includes two contents: precision plot and success plot.

Center location error (E_{CL}) was calculated by the Euclidean distance between the predicted position (m_t, n_t) and the ground-truth position $(m_t^{\mathrm{truth}}, n_t^{\mathrm{truth}})$ as

$$E_{\mathrm{CL}} = \sqrt{(m_t - m_t^{\mathrm{truth}})^2 + (n_t - n_t^{\mathrm{truth}})^2}. \tag{27}$$

The precision plot can be drawn by calculating the percentage of frames with an E_{CL} less than the threshold, which reflects the localization performance of the tracking algorithm. We defined the precision with a center location error threshold of 20 pixels as distance precision (DP), which was used to rank the algorithms in the precision plot.

Overlap rate (O_{R}) is calculated by the ratio of the intersection of the predicted bounding box r_t and the ground-truth bounding box r_t^{truth} to their union as

$$O_{\mathrm{R}} = \frac{\left| r_t \cap r_t^{\mathrm{truth}} \right|}{\left| r_t \cup r_t^{\mathrm{truth}} \right|}. \tag{28}$$

The success plot was drawn by calculating the percentage of frames with O_{R} greater than the threshold, which reflects the ability of the tracking algorithm to maintain the target's appearance. We used the AUC to rank the algorithms in the success plot.

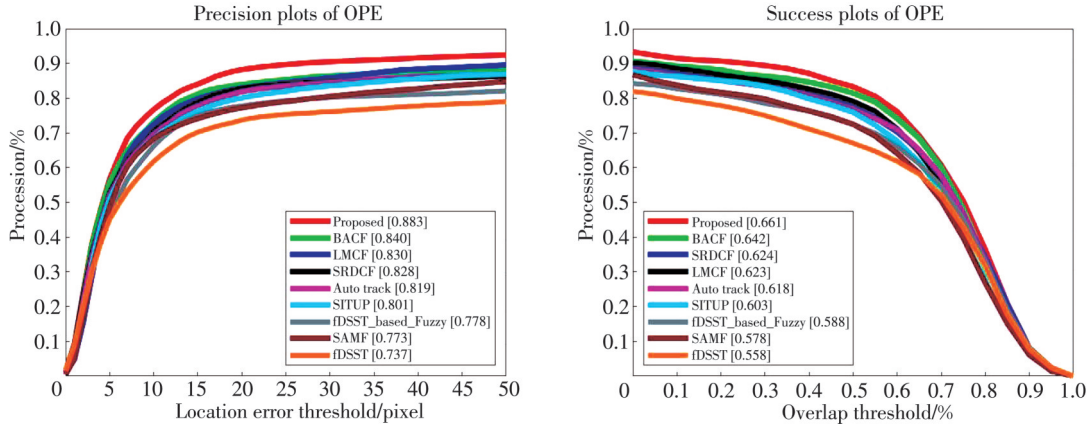
3.3 Evaluation

3.3.1 Quantitative analysis with correlation filtering trackers

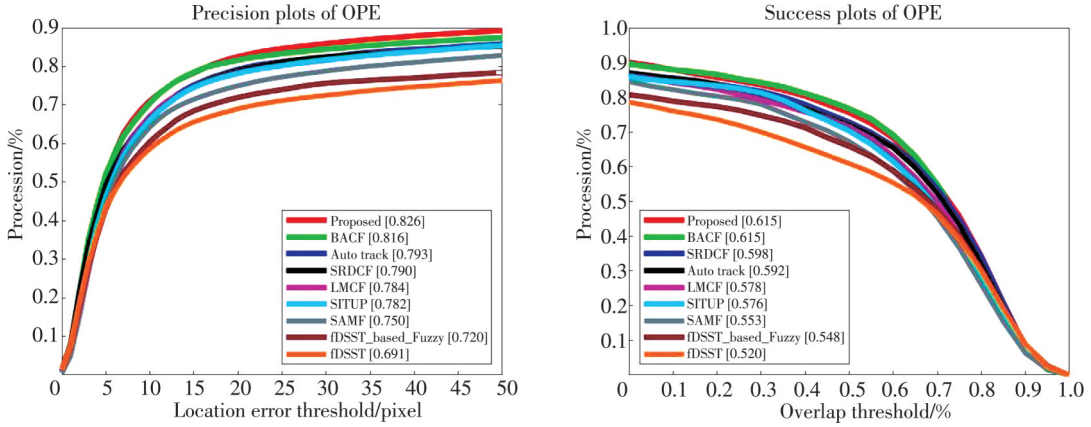
We compared and analyzed the proposed algorithm with

the state-of-the-art correlation filtering-based trackers, i.e., AutoTrack, fDSST_based_Fuzzy[15], SITUP[16], LMCF[17], BACF[18], SRDCF, fDSST[19], and SAMF[20].

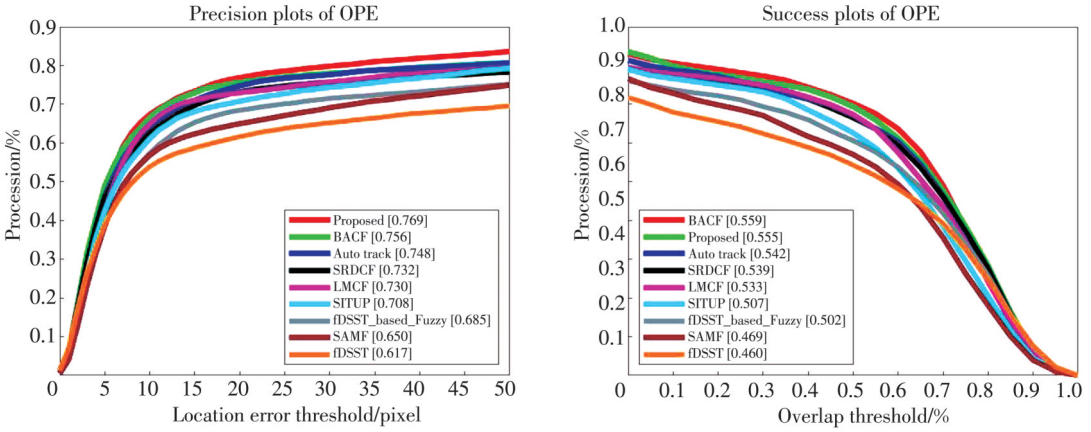
Fig. 7 shows the precision and success plots of the compared algorithms on three datasets, where Fig.7(a) is OTB-2013, Fig.7(b) is OTB-100, and Fig.7(c) is OTB-50. The precision and success rates of the proposed algorithm are 88.3% and 66.1% on OTB-2013, an increase of 6.4% and 4.3% compared to the baseline algorithm, respectively; on OTB-100, 82.6% and 61.5%, an increase of 3.3% and 2.3%, respectively; and on OTB-50, 76.9% and 55.5%, an increase of 2.1% and 1.3%, respectively. Overall, the tracking performance of the proposed algorithm outperforms the other algorithms compared and shows a significant improvement compared to AutoTrack.



(a) OTB-2013



(b) OTB-100



(c) OTB-50

Fig. 7 Experimental result curves of correlation filtering algorithms

Table 2 and Table 3 present the DP and AUC scores achieved by the comparison algorithms on 11 challenge attributes of OTB-2013, with bolded being the best and underlined being the second best. It can be seen that the algorithm in this paper shows significant performance on most of the attributes. In terms of precision, it achieves first

place in all scores except for OV, and in terms of success rate, it also achieves first place in all scores except for IPR and OV. Notably, our algorithm's DP scores increase by 11.4%, 8%, 7.1%, and 7.6% on LR, OCC, OPR, and SV, respectively, compared to those of AutoTrack, and the AUC scores on LR, SV, and OPR increase by 16.5%, 4.8%, and 6.2%, respectively. It indicates that the proposed algorithm can accurately sense the target appearance changes by introducing a stable adaptive aspect ratio mechanism, and can better cope with target occlusion and rotation to alleviate the tracking failure and improve the robustness of the tracker.

Table 2 DP scores of each correlation filtering algorithm on 11 attributes of OTB-2013

Algorithm	DP/%										
	FM	BC	MB	DEF	IV	IPR	LR	OCC	OPR	OV	SV
Proposed AAR-SRCF	**0.787**	**0.857**	**0.803**	**0.884**	**0.812**	**0.847**	**0.783**	**0.877**	**0.891**	0802	**0.825**
AutoTrack	0.732	0.796	0.770	0.839	0.743	0.786	0.669	0.797	0.820	0.785	0.749
fDSST_based_Fuzzy	0.681	0.842	0.732	0.684	0.752	0.757	0.546	0.718	0.757	0.685	0.742
SITUP	0.673	0.736	0.646	0.781	0.709	0.727	0.501	0.805	0.790	0.714	0.750
LMCF	0.730	0.848	0.714	0.837	0.762	0.779	0.485	0.824	0.826	0.695	0.757
BACF	0.771	0.792	0.746	0.790	0.760	0.840	0.768	0.806	0.842	**0.810**	0.789
SRDCF	0.742	0.803	0.791	0.827	0.743	0.767	0.767	0.825	0.818	0.680	0.761
fDSST	0.536	0.694	0.576	0.656	0.726	0.780	0.682	0.703	0.745	0.511	0.721
SAMF	0.599	0.676	0.552	0.777	0.662	0.709	0.709	0.816	0.763	0.636	0.708

Table 3 AUC scores of each correlation filtering algorithm on 11 attributes of OTB-2013

Algorithm	AUC/%										
	FM	BC	MB	DEF	IV	IPR	LR	OCC	OPR	OV	SV
Proposed AAR-SRCF	**0.595**	**0.630**	**0.610**	**0.674**	**0.617**	0.617	**0.573**	**0.648**	**0.645**	0.641	**0.626**
AutoTrack	0.561	0.591	0.602	0.653	0.576	0.577	0.408	0.605	0.597	0.613	0.564
fDSST_based_Fuzzy	0.537	0.618	0.568	0.529	0.571	0.571	0.322	0.538	0.561	0.565	0.564
SITUP	0.533	0.553	0.528	0.610	0.548	0.548	0.300	0.613	0.586	0.619	0.559
LMCF	0.555	0.625	0.543	0.636	0.581	0.579	0.289	0.626	0.611	0.598	0.569
BACF	0.593	0.598	0.578	0.614	0.586	**0.630**	0.453	0.619	0.630	0.632	0.606
SRDCF	0.571	0.587	0.604	0.629	0.574	0.567	0.471	0.623	0.600	0.555	0.583
fDSST	0.444	0.517	0.477	0.517	0.568	0.572	0.378	0.539	0.542	0.462	0.543
SAMF	0.482	0.520	0.459	0.619	0.513	0.524	0.376	0.608	0.558	0.555	0.578

Table 4 presents the tracking efficiency comparison results of nine algorithms on OTB-2013. The proposed algorithm achieves a tracking speed of 8.28 fps, which falls within the lower-middle range. However, it is comparable to most algorithms, and more importantly, the algorithm demonstrates notable advantages in tracking performance. While LMCF and fDSST exhibit outstanding speed performance, their tracking accuracy is only average. In addition, the other eight compared algorithms are unable to predict aspect ratio variations.

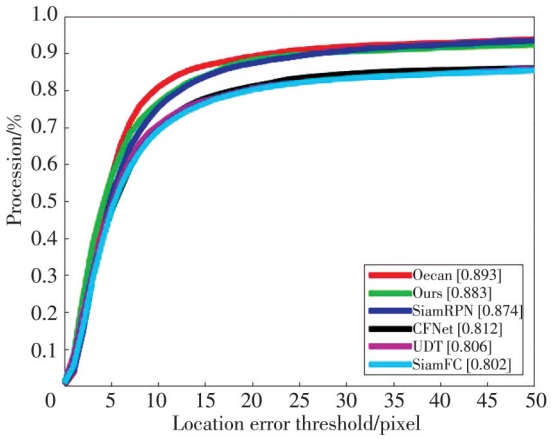
Table 4 Speed comparison of correlation filtering algorithms

	Proposed AAR-SRCF	AutoTrack	fDSST_based_Fuzzy	SITUP	LMCF	BACF	SRDCF	fDSST	SAMF
Speed/fps	8.28	12.15	4.96	18.06	85.23	18.21	4.01	63.43	15.85

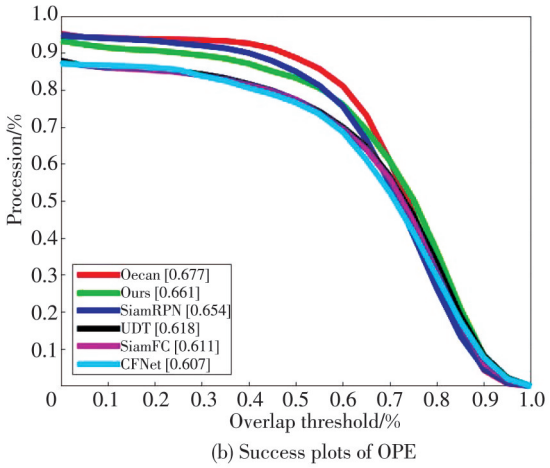
3.3.2 Quantitative analysis with deep learning trackers

To comprehensively verify the tracking effectiveness of the algorithm in this paper, five deep learning-based target tracking algorithms are selected for comparison experiments, namely Ocean[21], UDT[22], SiamRPN, CFNet[23], and SiamFC[24], which use a variety of representative deep learning methods, such as deep feature, anchor-free, unsupervised, and Siamese network.

Fig. 8 shows the precision plot and success plot. Table 5 and Table 6 show the DP and AUC scores on 11 attributes for the comparison algorithms on OTB-2013, respectively. Although the expression of manual features such as HOG is far inferior to those extracted by deep neural networks, it can be seen that the proposed algorithm still performs well overall, and the tracking performance is better than some state-of-the-art algorithms, especially on OCC, MB, and DEF.



(a) Precision plots of OPE



(b) Success plots of OPE

Fig. 8 Experimental result curves of algorithms on OTB-2013

Table 7 illustrates the tracking efficiency of the proposed algorithm and deep learning-based tracking algorithms tested on an NVIDIA Quadro K4200 GPU. It can be observed that the proposed algorithm is still at a moderate level, while high-performing deep learning algorithms rely on GPUs to enhance their efficiency, which imposes high hardware requirements.

3.3.3 Ablation study

The AAR-SRCF improves upon the AutoTrack by introducing an adaptive aspect ratio scale detection method (AAR) and by refining the boundary filter's template update strategy (TUS). Furthermore, the proposed algorithm uses a weighted fusion mechanism of multi-feature response maps (MFF). AAR has been improved compared to the boundary filter (IBC) in paper [7], and is related to TUS, while MFF is independent of other methods. In terms of computational complexity, the overall cost of AutoTrack is $O(KT_1\log(T_1)N_1)$, where K represents the number of channels, T_1 represents the number of elements in the center samples, and N_1 represents the number of ADMM iterations for AutoTrack. The complexity of the boundary filters in IBC and AAR methods is the same, both being $O(KT_2\log(T_2)N_2)$, where T_2 represents the number of elements in the boundary samples, and N_2 represents the number of ADMM iterations for boundary filters. The complexity of TUS and MMF methods is $O(1)$. Therefore, the overall cost of AAR-SRCF is $O\left(N_3K\left(T_1\log(T_1)N_1+T_2\log(T_2)N_2\right)\right)$, where N_3 represents the number of iterations to solve Eq. (10) using ADMM.

Table 5 DP scores of each algorithm on 11 attributes of OTB-2013

Algorithm	DP/%										
	FM	BC	MB	DEF	IV	IPR	LR	OCC	OPR	OV	SV
Proposed AAR-SRCF	0.787	0.857	0.803	0.884	0.812	0.847	0.617	**0.877**	0.891	0.802	0.825
Ocean	**0.895**	0.840	**0.856**	**0.891**	**0.880**	**0.928**	**0.702**	0.840	**0.923**	**0.868**	**0.913**
UDT	0.717	0.794	0.710	0.763	0.711	0.766	0.586	0.804	0.803	0.785	0.757
SiamRPN	0.790	**0.868**	0.783	0.874	0.828	0.866	0.643	0.822	0.889	0.755	0.861
CFNet	0.668	0.771	0.676	0.787	0.707	0.769	0.561	0.762	0.807	0.445	0.784
SiamFC	0.725	0.733	0.699	0.727	0.699	0.744	0.573	0.789	0.789	0.785	0.784

Table 6 AUC scores of algorithms on 11 attributes of OTB-2013

Algorithms	AUC/%										
	FM	BC	MB	DEF	IV	IPR	LR	OCC	OPR	OV	SV
Proposed AAR-SRCF	0.595	0.630	0.610	0.674	0.617	0.617	0.573	**0.648**	0.645	0.641	0.626
Ocean	**0.670**	0.630	**0.642**	0.666	**0.668**	**0.702**	**0.699**	0.635	**0.691**	**0.674**	**0.696**
UDT	0.570	0.590	0.573	0.601	0.558	0.586	0.437	0.614	0.605	0.645	0.592
SiamRPN	0.599	**0.642**	0.583	**0.675**	0.628	0.643	0.652	0.622	0.660	0.614	0.650
CFNet	0.521	0.568	0.536	0.587	0.530	0.561	0.579	0.567	0.579	0.422	0.583
SiamFC	0.547	0.551	0.517	0.552	0.537	0.573	0.550	0.601	0.591	0.637	0.603

Table 7 Speed comparison with deep learning algorithms

	Proposed	Ocean	UDT	SiamRPN	CFNet	SiamFC
Speed/fps	8.28	58	13.35	31.05	8.91	2.01

Two ablation experiments were designed and tested on OTB-2013. As shown in Table 8, through experiments, it was found that IBC generates too much tolerance, leading to tracking failure. Meanwhile, the improved method AAR in this paper has significantly better tracking performance than IBC, which demonstrates the effectiveness of spatial regularization.

As shown in Table 9, the algorithm is slightly improved after using AAR and MFF, and the improvement is more apparent after the integration of the two methods.

Table 8 Ablation experiments with IBC method

Algorithm	DP/%	AUC/%	Speed/fps
Baseline	0.819	0.618	12.15
Baseline+IBC	0.814	0.602	9.71
Baseline+AAR	0.834	0.623	9.49

Table 9 Ablation experiments of proposed algorithm

Algorithm	DP/%	AUC/%	Speed/fps
Baseline	0.819	0.618	12.15
Baseline+AAR	0.834	0.623	9.49
Baseline+MFF	0.823	0.613	9.16
Baseline+AAR+MFF	0.845	0.634	6.87
Baseline+AAR+TUS	0.86	0.65	9.22
Baseline+MFF+AAR+TUS	0.883	0.661	8.28

By improving AAR with TUS, a significant improvement in the tracking performance is achieved. Although the proposed methods increase the computational cost of the algorithm, it is worth noting that the tracking performance of each method in this paper is better than that of AutoTrack, and the integration of these methods demonstrates a significant advantage. The experimental data shows that the proposed improvement methods are feasible and effective.

3.3.4 Qualitative analysis

To demonstrate the performance of AAR-SRCF in real tracking situations, five video sequences in the dataset are selected for qualitative analysis as shown in Fig.9, from Fig. (a) to Fig. (e) for Basketball, DragonBaby, FaceOcc2, Lemming, and Freeman4, respectively.

Fig. 9 Comparison of algorithms on partial videos

In Fig. 9(a), for the Basketball sequence, starting from frame 14, when the target experiences deformation and the background becomes cluttered, algorithms such as SITUP and BACF drift consecutively, but the proposed algorithm still correctly tracks the target. From frame 161 to frame 599, as the target's appearance continuously changes, algorithms like SRDCF and SITUP, which use a fixed aspect ratio of the bounding box, always detect the incorrect target size. However, our proposed algorithm, by introducing a boundary filter that adapts to the aspect ratio, accurately detects the target size.

In Fig. 9(b), For the DragonBaby sequence, the fDSST, with a poor discrimination capability, drifts at frame 9 due to interference from similar objects and loses the target in subsequent frames. During frames 22 to 33, as the target makes a turning motion, our proposed algorithm, which can stop filter learning when the tracking results become unreliable, accurately tracks the target, while other algorithms consecutively lose the target. In frame 81, the camera suddenly zooms in, causing all comparison algorithms to lose the target. After the camera gradually zooms out, the proposed algorithm regains the target.

In Fig. 9(c), for the FaceOcc2 sequence, when the target rotates at frame 343, all algorithms produce varying degrees of drift. After the target returns to its original angle at frame 556, the predicted bounding boxes of our proposed algorithm and fDSST_based_Fuzzy's are closer to the exact bounding box. At frame 593, when the target puts on a hat, our proposed algorithm accurately detects the target's face region, again demonstrating the superiority of our proposed adaptive aspect ratio method.

In Fig.9(d), in the Lemming sequence, starting from frame 342, the trackers degrade for AutoTrack, SRDCF, and fDSST, causing them to lose the target as it becomes occluded. Our proposed algorithm adjusts the tracking bounding box during the reappearance of the target, thus learning the correct information, and

achieving stable tracking. From frame 400 to frame 692, our proposed algorithm can continuously handle rapid target movements. During frames 968 to 1320, when the target rotates significantly, our proposed algorithm can keep on track normally because it reduces the penalty of temporal regularization during non-occlusion and non-loss situations and quickly learns changes in target appearance. Although LMCF briefly drifts during this process, it can recover the target through re-detection.

In Fig.9(e), freeman4 has attributes such as SV and LR and experiences occlusions at frames 46, 69, 144, and 240. All algorithms except ours fail to track consecutively. This demonstrates the effectiveness of our proposed algorithm in preventing model degradation. During frames 273 to 282, when part of the target crosses the boundary of the video, our proposed algorithm still succeeds in tracking.

4 Conclusions

To address the issue of AutoTrack being not in a position to perceive the aspect ratio of the target and poor performance in complex situations such as occlusion and shape variation, we proposd a spatial-temporal regularized correlation filtering algorithm with adaptive aspect ratio. Firstly, a weighted fusion mechanism is designed with multiple feature response maps to improve the accuracy of center filter tracking. Secondly, by combining near-orthogonality and spatial regularization, a boundary filter was implemented to detect changes in the size of the target, achieving aspect ratio adaptation. Finally, the boundary filter update strategy was improved to enhance the robustness and anti-occlusion ability of the algorithm.

The proposed algorithm was extensively compared with several state-of-the-art algorithms on challenging video sequences, and the validity and superiority of the algorithm were proved. In the following experiments, the model structure will be considered for improvement to reduce computational complexity and enhance algorithm performance.

Acknowledgement

This work was supported by National Natural Science Foundation of China (No.61806006); Jiangsu University Superior Discipline Construction Project; Talent Introduction Project (No. B12018)

Declaration of conflicting interests

The authors have no conflict of interests related to this publication.

References

[1] LI Y, LI M M, ZHENG Q B, et al. Survey on video object tracking algorithms. Journal of Frontiers of Computer Science and Technology, 2022, 16(7): 1504-1515.

[2] BOLME D S, BEVERIDGE J R, DRAPER B A, et al. Visual object tracking using adaptive correlation filters// 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 13-18, 2010, San Francisco, CA, USA. Piscataway, N. J.: IEEE, 2010: 2544-2550.

[3] HENRIQUES J F, CASEIRO R, MARTINS P, et al. High-speed tracking with kernelized correlation filters. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(3): 583-596.

[4] DANELLJAN M, HGER G, SHAHNAZ K F, et al. Accurate scale estimation for robust visual tracking//The British Machine Vision Conference, September 1-5, 2014, Nottingham, UK. Durham: BMVA Press, 2014: 1-11.

[5] LI F, YAO Y, LI P, et al. Integrating boundary and center correlation filters for visual tracking with aspect ratio variation//IEEE International Conference on Computer Vision, October 24-27, 2017, Venice, Italy. Piscataway, N. J.: IEEE, 2017: 2001-2009.

[6] DANELLJAN M, ROBINSON A, KHAN F S, et al. Beyond correlation filters: learning continuous convolution operators for visual tracking//European Conference on Computer Vision, October 8-16, 2016, Amsterdam, The Netherlands. Heidelberg: Springer, 2016: 472-488.

[7] LI B, YAN J, WU W, et al. High performance visual tracking with Siamese region proposal network//IEEE Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, USA. Piscataway, N. J.: IEEE, 2018: 8971-8980.

[8] DANELLJAN M, HGER G, SHAHNAZ K F, et al. Learning spatially regularized correlation filters for visual tracking//2015 IEEE International Conference on Computer Vision, June 5-12, 2015, Boston, MA, USA. Piscataway, N. J.: IEEE, 2015: 4310-4318.

[9] LI F, TIAN C, ZUO W, et al. Learning spatial-temporal regularized correlation filters for visual tracking//IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, USA. Piscataway, N. J.: IEEE, 2018: 4904-4913.

[10] LI Y, FU C, DING F, et al. AutoTrack: Towards high-performance visual tracking for UAV with automatic spatio-temporal regularization//IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 13-19, 2020, Seattle, WA, USA. Piscataway, N. J.: IEEE, 2020: 11923-11932.

[11] HUANG W, YUAN L, ZHANG J J, et al. An anti-occlusion and scale-adaptive long-time target tracking algorithm. Electronics Optics and Control, 2021, 28(10): 44-48.

[12] LIU S, YANG W, SHAO X L. A background-aware target tracking algorithm based on scale adaptation. Electronics Optics and Control, 2020, 27(9): 19-23.
[13] WU Y, LIM J, YANG M H. Online object tracking: a benchmark//IEEE Conference on Computer Vision and Pattern Recognition, June 23-28, 2013, Portland, OR, USA. Piscataway, N. J.: IEEE, 2013: 2411-2418.
[14] WU Y, LIM J, YANG M H. Object tracking benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1834-1848.
[15] LIU S, WANG S, LIU X, et al. Fuzzy detection aided real-time and robust visual tracking under complex environments. IEEE Transactions on Fuzzy Systems, 2021, 29(1): 90-102.
[16] MA H, ACTON S T, LIN Z. SITUP: Scale invariant tracking using average peak-to-correlation energy. IEEE Transactions on Image Processing, 2020, 29: 3546-3557.
[17] WANG M M, LIU Y, HUANG Z Y. Large margin object tracking with circulant feature maps//IEEE Conference on Computer Vision and Pattern Recognition, October 24-27, 2017, Venice, Italy. Piscataway, N. J.: IEEE, 2017: 4800-4808.
[18] GALOOGAHI H K, FAGG A, LUCEY S. Learning background-aware correlation filters for visual tracking// IEEE International Conference on Computer Vision and Pattern Recognition, October 24-27, 2017, Venice, Italy. Piscataway, N. J.: IEEE, 2017: 1144-1152.
[19] DANELLJAN M, HGER G, SHAHNAZ K F, et al. Discriminative scale space tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 39(8): 1561-1575.
[20] LI Y, ZHU J. A scale adaptive kernel correlation filter tracker with feature integration//European Conference on Computer Vision, September 6-12, Zurich, Switzerland. Heidelberg: Springer, 2014: 254-265.
[21] ZHANG Z, PENG H, FU J, et al. Ocean: Object-aware anchor-free tracking//The 16th European Conference Computer Vision, August 23-28, 2020, Glasgow, UK. Heidelberg: Springer, 2020: 771-787.
[22] WANG N, SONG Y, MA C, et al. Unsupervised deep tracking//IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 15- 21, 2019, Los Angeles CA, USA. Piscataway, N. J.: IEEE, 2019: 1308-1317.
[23] VALMADRE J, BERTINETTO L, HENRIQUES J, et al. End-to-end representation learning for correlation filter based tracking/IEEE Conference on Computer Vision and Pattern Recognition, October 24-27, 2017, Venice, Italy. Piscataway, N. J.: IEEE, 2017: 2805-2813.
[24] BERTINETTO L, VALMADRE J, HENRIQUES J F, et al. Fully-convolutional Siamese networks for object tracking//European Conference on Computer Vision, October 8-16, 2016, Amsterdam, The Netherlands. Heidelberg: Springer, 2016: 850-865.

纵横比自适应的时空正则化相关滤波算法

许 凯[1,2], 李 婷[1,2], 葛洪伟[1,2*]

1. 江南大学 人工智能与计算机学院, 江苏 无锡 214122;
2. 江苏省模式识别与计算智能工程实验室(江南大学), 江苏 无锡 214122

摘 要: 在目标跟踪中, 传统相关滤波算法无法感知运动目标尺度纵横比变化, 且易受复杂环境影响导致跟踪失败。为此, 提出了纵横比自适应的时空正则化相关滤波算法。首先, 参考平均峰值相关能量(Average peak-to-correlation energy, APCE)和响应峰值对每个特征的响应图进行加权融合, 以实现对目标的精确跟踪。其次, 结合近正交性和空间正则化提出一种新的一维边界滤波器, 通过定位目标包围框的四个边界位置实现对目标尺度和纵横比变化的自适应检测, 有效抑制了边界效应带来的负面影响。最后, 根据响应输出的峰值旁瓣比(Peak-to-sidelobe ratio, PSR)独立地调节各边界滤波器的学习率, 防止模型退化。在OTB数据集上进行了测试, 该算法表现出理想的跟踪效果, 在各个挑战属性上相较于其他优秀算法均取得了更优结果。

关键词: 目标跟踪; 相关滤波; 空间正则化; 自适应纵横比; 模板更新; 响应融合

引用格式: XU Kai, LI Ting, GE Hongwei. Spatial-temporal regularized correlation filtering algorithm with adaptive aspect ratio. Journal of Measurement Science and Instrumentation, 2024, 15(1): 9-22.