Object Tracking via Background-Aware Correlation Filter with Elliptical Search Area

Wencheng Yu

***Abstract*—The utilization of background-aware correlation filtering (BACF) has been a topic of significant research interest due to its ability to enhance both the accuracy and speed of target tracking algorithms. Despite its advantages, the BACF algorithm struggles to overcome tracking drift in complex scenarios, such as those involving fast motion and occlusion, since it does not take into account the motion state of the target. To address this limitation, we propose a novel algorithm called redesigning the search area of background-aware correlation filtering (RSABACF). This algorithm considers the complete motion state of the target and uses the Kalman filter to formulate an elliptical search area. Furthermore, the proposed algorithm can dynamically adjust the search range and angle based on the target's motion state to improve the background information in the motion direction and reduce it in the non-motion direction. As a result, the algorithm exhibits an improved ability to differentiate between target data and background data. Our proposed algorithm has been evaluated on the publicly available dataset OTB2015 and compared with mainstream tracking algorithms. The results show that our algorithm outperforms the BACF algorithm by 3.7% and 2.3% in terms of precision and success rates, respectively. Consequently, the proposed algorithm demonstrates its efficacy in handling complex tracking scenarios and exhibits high levels of robustness.**

***Index Terms*—correlation filtering, background-aware, target tracking, Kalman filter, dynamic adjustment, elliptical search area**

I. Introduction

TARGET tracking is a crucial area of computer vision research, with applications in video surveillance, drone detection and tracking, traffic control, pedestrian tracking, and other fields [1]-[4]. However, the tracking process presents several challenges, including target deformation, motion blur, fast motion, occlusion, target rotation, target and background similarity, background clutter, scale variation, illumination change, and target leaving the sensing range. Therefore, it is essential to develop tracking algorithms capable of addressing these challenges and improving accuracy.

Target tracking algorithms can be classified into two categories: generative methods [5], [6] and discriminative methods [7]-[9]. Generative methods extract features from a target and construct a model to predict the target's location, while discriminative methods collect both positive and negative samples to train a classifier to correctly identify the target's location. Within the discriminative methods, there are two main types of algorithms: correlation filtering algorithms and deep learning algorithms. Correlation filtering algorithms are known for their high speed and effectiveness in target tracking and continue to be actively researched. Deep learning algorithms, which use pre-trained convolutional neural networks (CNNs) applied to the correlation filter (CF) framework, have shown excellent tracking performance [10]-[12].

Currently, correlation filtering and deep learning methods are the two main research directions in the field of target tracking. While deep learning-based methods exhibit high accuracy in tracking, the complex feature calculation and model update of deep neural networks increase computational costs and reduce algorithm speed. Correlation filtering has been extensively studied due to its high accuracy and fast speed. Numerous improved methods based on correlation filtering have been proposed, significantly enhancing the accuracy of the algorithm. In 2010, Bolme et al. [13] proposed the Minimum Output Sum of Squared Error (MOSSE) tracking algorithm, which employs correlation filtering to determine the target's position by computing the correlation between the target and the detection area and identifying the maximum response value. The Fourier transform method is used to reduce computational complexity and improve tracking speed. In 2012, Henriques et al [14]. proposed the Exploiting the Circulant Structure of Tracking-by-detection with Kernels (CSK) algorithm to address problems with sample size in the MOSSE algorithm. The CSK algorithm calculates the correlation between two adjacent frames and uses the maximum corresponding value as the prediction target center to improve tracking accuracy. However, the algorithm's fixed target scale and lack of robustness in target scale transformation limits its efficacy. To address these

Manuscript received June 29, 2023; revised September 30, 2023.

Wencheng Yu is a postgraduate student at School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou, Guang Dong, 510000, China. (e-mail: 1358453643@qq.com).

limitations, Henriques et al [15]. improved the CSK algorithm in 2014 and proposed the Kernel Correlation Filter (KCF) algorithm. The KCF algorithm expands the number of negative samples by constructing a cyclic matrix and optimizes the algorithm's performance. However, the algorithm's tracking performance is not ideal under conditions of rapid target motion. In 2014, Danelljan et al [16]. proposed the Discriminative Scale Space Tracker (DSST), which employs the discriminant correlation filter and a scale estimation method to determine the target's position and scale data, respectively, effectively improving scale adaptation performance. Although correlation filtering is a template method, it exhibits poor tracking performance under fast motion or target deformation. To address this problem, Danelljan et al [17]. proposed Spatially Regularized Correlation Filters (SRDCF) in 2015. This algorithm can effectively suppress the response of the background area, expand the tracking range of the background data, and add spatial regularization constraints. Although it improves target tracking performance in complex backgrounds, its tracking speed is slow and unsuitable for real-time tracking. In 2016, Bertinetto et al [18]. proposed the Staple algorithm, which combines Histogram of Oriented Gradient (HOG) features and color name (CN) features to improve tracking performance. However, the algorithm lacks robustness when the target moves beyond the field of view at low resolution, and it cannot effectively track the target in real-time. In 2017, Galoogahi et al [19]. proposed the Background Aware Correlation Filters (BACF) algorithm, which densely extracts real negative examples from the background rather than from moving foreground to learn the filters. This background-aware correlation filter based on hand-crafted features can effectively simulate how the foreground and background of the target object change over time. Similar to traditional correlation filters, the BACF algorithm is computationally efficient, and extensive experiments on multiple datasets demonstrate its superior accuracy and real-time performance compared to advanced trackers based on deep learning.

To address the challenges of target and background similarity, fast motion, and occlusion encountered during the tracking process of the BACF algorithm, we propose the RSABACF algorithm by redesigning the search area and integrating a Kalman filter into the BACF tracking framework. The filter is trained by extracting background data, and the target position is determined using the maximum response value obtained through a linear interpolation method with the aid of an auxiliary factor g constructed in the algorithm. The augmented Lagrange method (ALM) [20] is employed to incorporate the constraint term into the optimization function, and the alternating direction method of multipliers (ADMM) [21] is used to minimize the filter and auxiliary factors, reducing computational complexity. A multi-resolution search method is utilized to estimate the scale of the target transformation.

Our contribution in this work includes the following:

(1) We revamp the BACF search area and add the Kalman filter to the tracking framework. By taking the motion state of the target into account, we dynamically change the search range and search angle of the elliptical search area, increasing the background data in the motion direction and decreasing it in the non-motion direction.

(2) Our new search area, unlike BACF's rectangular search area, completely considers the target's motion state. Moreover, the search region takes the form of a circle when the target's motion speed is too slow or immobile, satisfying the requirement that the target has an equal chance of moving in all directions.

(3) We compare our proposed algorithm with nine other mainstream tracking algorithms on the OTB2015 dataset [22]. The experimental results demonstrate strong tracking performance, achieving real-time tracking speeds of 38 FPS on CPU.

This paper is organized as follows: Section 2 introduces the tracking framework and principle of our proposed method. Section 3 presents the improved BACF tracker, and Section 4 discusses the experiments and presents the results. Finally, Section 5 provides a conclusion of this paper.

II. Design Elliptical Search Area

A. Whole frame

The Background Aware Correlation Filters (BACF) algorithm, based on the conventional correlation filtering method, enhances sample quality by multiplying them and applying clipping. However, in complicated scenes, the algorithm is susceptible to tracking drift, leading to tracking failure. As a solution, we propose the RSABACF algorithm by redesigning the search area to fully account for the target's motion state and extract additional background data in the target's motion direction. The algorithm is carried out in three steps, as depicted in Fig.1:

(1) Initial Phase: The first frame's target point is utilized as the origin to establish a plane rectangular coordinate system. We define a circular search area in this phase, ensuring that the algorithm's computational cost remains essentially unchanged, and the area matches that of the elliptical search area in the tracking phase.

(2) Designing the Elliptical Search Area: The Kalman filter is used to predict the target's motion state and direction to estimate its location in the next frame. The two focuses used to construct the elliptical search area are the target's location in the present frame and its expected position in the following frame.

(3) Tracking Phase: The filter is trained, and the filter template is updated using background data extracted from the elliptical search region. The final target position is determined by enhancing the background data in the

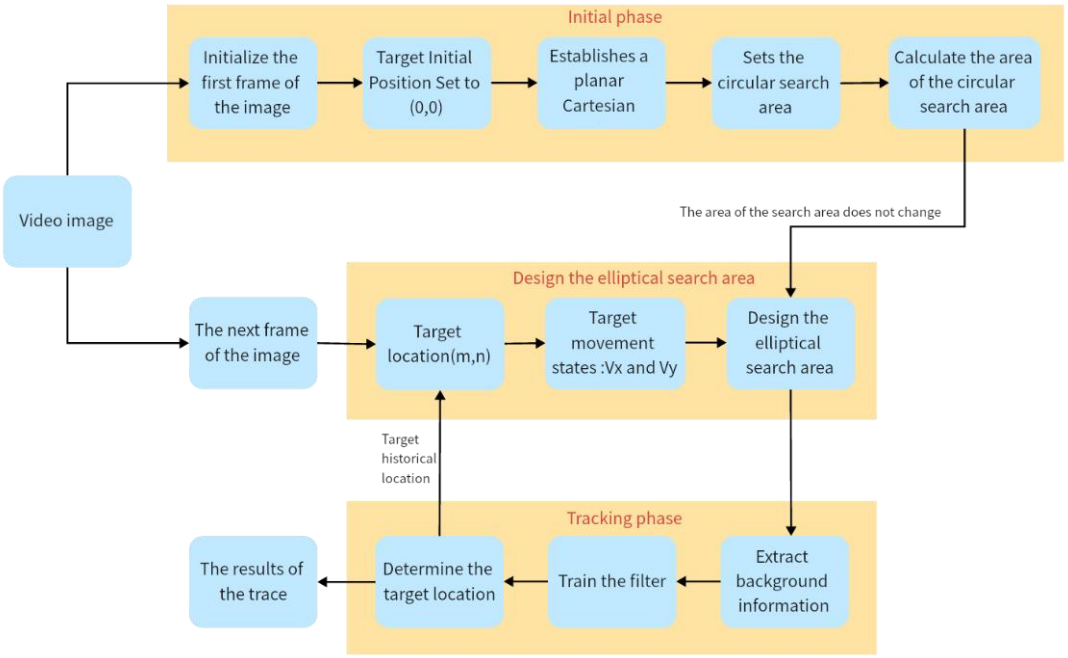


Fig. 1. Overall framework of algorithm

motion direction, improving the filter's ability to classify the target and background.

B. Kalman filtering prediction model

Kalman filtering [23] is a commonly used method for linear systems. The process of the target's transition from the previous state to the current state in the system is referred to as the state transition, which involves the transfer of the target's location, velocity, acceleration, etc. The state obtained at the current time is known as the observation state. This process is represented by a linear equation, assuming that the noise interference is independent and follows a Gaussian distribution [24]. The Kalman filter gain coefficient is used to weight the new state prediction value and the observation value obtained after the state transition to obtain the final target state value.

In this paper, we adopt the Kalman filter to predict the target's motion direction and state. The system's state equation is as follows:

$$\begin{cases} X_t = AX_{t-1} + BU_t + W \\ Z_t = HX_t + V \end{cases} \tag{1}$$

Equation (1) defines the variables in the Kalman filter model, including X_t as the system state matrix, A as the state transition matrix, B as the control input matrix, U_t as the control input, H as the state observation matrix, and A, B, and C as system parameters. W represents the process noise, Q is its covariance, Z_t is the observation of the state matrix, V represents the measurement noise, assumed to be Gaussian white noise, and R is its covariance. The Kalman filter is estimated using the following five iterative equations:

$$\hat{X}_{t/t-1} = A\hat{X}_{t-1} + BU_{t-1} \tag{2}$$

Here, $\hat{X}_{t-1}$ represents the state estimation at time t-1, $\hat{X}_{t/t-1}$ represents the predicted state at time t, and U_{t-1} represents the control quantity at time t-1. For our algorithm, since there is no control quantity, U_{t-1} is set to 0.

$$P_{t/t-1} = AP_{t-1}A^T + Q \tag{3}$$

In equation (3), P_t represents the covariance matrix, and $P_{t/t-1}$ represents the covariance matrix of the prediction for the next frame.

$$K_g = \hat{X}_{t/t-1}H^T/(HP_{t/t-1}H^T + R) \tag{4}$$

In equation (4), K_g denotes the Kalman gain.

$$\hat{X}_t = \hat{X}_{t/t-1} + K_g(Z_t - H\hat{X}_{t/t-1}) \tag{5}$$

In equation (5), $\hat{X}_t$ is the estimated state at time t.

$$P_a = (I - K_gH)\,P_{t/t-1} \tag{6}$$

In equation (6), I represents the unit matrix.

C. Definition of elliptical search area

Assuming that motion occurs in the positive direction of the x-axis, the origin of the coordinate axis is set to be the target's position in the current frame, denoted as (0, 0), and the mathematical model of the elliptical search region is specified as:

$$\frac{(x-c)^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{7}$$

In equation (7), A, B, and C represent the long axis, short axis, and half of the ellipse's focal length, respectively.

The size of the displacement s, which indicates the motion displacement of the target for each frame, is equal

to the elliptical focal length and is expressed as follows:

$$s = 2c \tag{8}$$

To ensure algorithm stability and execution speed, the search area should remain unchanged. As the motion speed becomes zero, i.e., c = 0, the present search area takes the form of a circle having a radius r, expressed as:

$$\pi r^2 = \pi ab \tag{9}$$

$$a^2 - b^2 = c^2 \tag{10}$$

Three times the length of the initial rectangular tracking box, h, is chosen as the size of r, yielding R=3h. This allows for determination of the ellipse's parameters a and b.

D. The principle of elliptical search area

When the initial position is located at (0, 0), the search point can be determined by utilizing the area encompassed by Equation (7) as the search region, which is expressed as:

$$x_i = c - a + i \cdot k \qquad (i = 0,1,2,\ldots,\tfrac{2a}{k}) \tag{11}$$

$$y_j = (b - j \cdot k)\sqrt{1 - \tfrac{(x-c)^2}{a^2}} \quad (j = 0,1,2,\ldots,\tfrac{2b}{k}) \tag{12}$$

The search point $F(x_i, y_j)$ can be determined using equations (11) and (12).

This approach aims to address the single target tracking problem by extracting the motion vector from the expected value of the Kalman filter to obtain the displacement size and angle change w of the target. The initial state transition matrix A, state observation matrix H, covariance matrix Q, and R matrix in the Kalman filter prediction model are presented as follows:

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$Q = \begin{bmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix}$$

$$R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Finally, the state space vector of the target is obtained by employing the Kalman filter, expressed as:

$$X = [O_x, O_y, V_x, V_y] \tag{13}$$

In equation (13), V_x and V_y represent the target's velocities in the horizontal and vertical directions, respectively. By utilizing the velocity components V_x and V_y of the object, its horizontal displacement m and vertical displacement n during each video frame can be accurately computed. A positive value of m indicates that the object is moving rightward, whereas a negative value signifies a leftward movement. Similarly, a positive n denotes an upward movement of the object, while a negative value represents a downward movement. The value of s, which measures the target's motion in each frame, can be calculated as:

$$s = \sqrt{m^2 + n^2} \tag{14}$$

As the target's direction changes constantly during movement, displacement can be used to determine the following angle ω:

$$\omega = arctan\frac{n}{m} \tag{15}$$

To rotate the search point $F(x_i, y_j)$ counterclockwise with the angle ω, we use the following equations:

$$x^* = x_i cos(\omega) - y_j sin(\omega) \tag{16}$$

$$y^* = y_j cos(\omega) + x_i sin(\omega) \tag{17}$$

Assuming the current position to be $C(P, Q)$, the search point will rotate and move towards the current position.

$$x = x^* + p \tag{18}$$

$$y = y^* + q \tag{19}$$

The elliptical search area is ultimately determined by all points (x, y).

E. Example Test

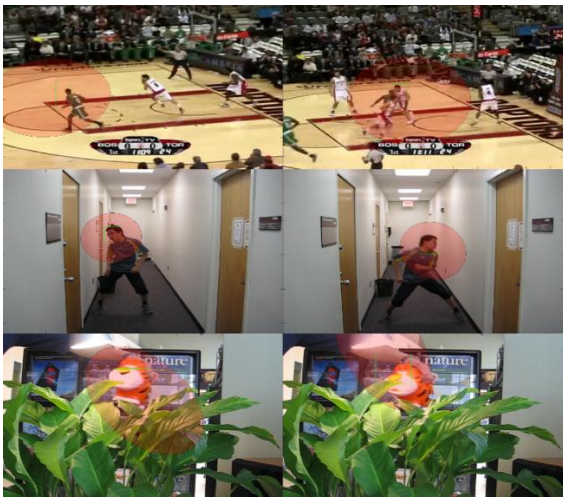


Fig. 2. Example test renderings

III. Redesigned the Search Area of BACF

A. BACF tracker

Target tracking is a challenging task in computer vision due to the effects of rapid motion, scale change, rotation and boundary effect. The boundary effect problem arises when the target moves out of the tracking window, leading to incomplete target information and inaccurate tracking. To address this problem, Galoogahi et al. proposed Correlation Filters with Limited Boundaries (CFLB) in 2015. CFLB utilizes ridge regression to minimize the difference between the predicted and actual target locations, which is defined by the following equation (20):

$$E(h) = \tfrac{1}{2}\textstyle\sum_{j=1}^{D}\left\| y(j) - \sum_{k=1}^{K} h_k^T x_k[\Delta\tau_j] \right\|_2^2 + \tfrac{\lambda}{2}\sum_{k=1}^{K}\|h_k\|_2^2 \tag{20}$$

In equation (20), N represents the number of training pictures, T represents the size of the picture, y is the goal of

regression, h represents the training filter, H is a transposition, P is a two-dimensional matrix of D × T, which is used to extract D elements in signal x. D is the target sample size, and typically D<<T. Additionally, P is a constant matrix that can be calculated, and λ is the regularization coefficient. The CFLB algorithm effectively solves the boundary effect problem and achieves good tracking performance. Experimental results demonstrate its effectiveness in challenging situations such as deformation and occlusion. The proposed method provides a new solution to the challenging problem of target tracking.

Background-aware correlation filtering is an improved method that enhances the traditional correlation filtering algorithm by incorporating multi-channel features, namely the Histogram of Oriented Gradients (HOG), as defined by equation (20):

$$E(h) = \frac{1}{2}\sum_{v=1}^{T}\left\|y(v) - \sum_{k=1}^{K} h_k^H P_{x_k}[\Delta\tau_v]\right\|_2^2 + \frac{\lambda}{2}\sum_{k=1}^{K}\|h_k\|_2^2 \tag{21}$$

In equation (21), N is equal to 1, and the HOG feature is added to the algorithm. Here, y(v) represents the v element of y, K represents the number of characteristic channels, h_K is a multi-channel filter, $x_k \in R^T$(k = 1, 2... K) denotes the extracted feature samples, and $y \in R^T$ and $h \in R^D$. One notable improvement of this algorithm lies in expanding the sampling area of the circular matrix while cutting the sample simultaneously. To ensure sample quality, an effective area of the sample is extracted. Compared to the traditional correlation filtering algorithm, background-aware correlation filtering demonstrates superior performance in target tracking by effectively incorporating multi-channel features.

B. RSABACF algorithm model

This paper proposes an algorithm for enhancing the filter's ability to distinguish targets from the background by extracting background data using an elliptical search area for training the filter. Equation (21) can be reduced in this context to the following equation (22):

$$E(h) = \frac{1}{2}\sum_{v=1}^{T}\|y(v) - \sum_{k=1}^{K} h_k^H P x_k[\Delta\tau_v]\|_2^2 + \frac{\lambda}{2}\sum_{k=1}^{K}\|h_k\|_2^2 + \frac{\gamma}{2}\sum_{k=1}^{K}\left\|O_\delta^{x_2}\varphi\right\|_2^2 \tag{22}$$

In equation (22), γ is the regularization coefficient, and $O_\delta^{x_2}(\delta \in [1,\partial])$ represents target background data extracted from the elliptical search area. ∂ represents the number of background blocks extracted, and φ is the weight coefficient ranging from 0 to 1, which is related to the number of extracted background blocks. As the number of background blocks selected increases, φ approaches 1. The equation (22) is converted to the frequency domain, resulting in the following equation (23):

$$E(h,\hat{g}) = \frac{1}{2}\|\hat{y} - \hat{x}\hat{g}\|_2^2 + \frac{\lambda}{2}\|h\|_2^2 + \|\hat{g}(t)\varphi\|_2^2 \tag{23}$$

$$s.t.\ \hat{g} = \sqrt{T}(FP^H \otimes I_k)h$$

Subject to the constraint that $\hat{g} = [\hat{g}_1{}^H, \ldots, \hat{g}_K{}^H]^H$. In equation (23), $\hat{g}$ is an auxiliary variable represented by a K×T dimensional column vector of K vector channels, g_k, and ∧ is the Fourier transform of the signal. $\hat{g}(t)$ is the target state at time t. F is the orthogonal matrix of $T \times T$ and maps the basis of T-dimensional vector signal in Fourier domain. ⊗ is the Kronecker product, and I_K is a unit matrix of order K. h is a K×D dimensional column vector, h_k, representing a cascade of K vector channels. The proposed algorithm extracts background data using an elliptical search area and utilizes this data for training the filter, effectively enhancing its ability to distinguish targets from the background.

C. Model solving

To solve Equation (23), the augmented Lagrange multiplier method (ALM) is used to add constraints to the optimization function, expressed as follows:

$$\zeta(\hat{g},h,\hat{\xi}) = \frac{1}{2}\|\hat{y} - \hat{x}\hat{g}\|_2^2 + \frac{\lambda}{2}\|h\|_2^2 + \hat{\xi}\left(\hat{g} - \sqrt{T}(FP^H \otimes I_k)h\right) + \frac{\mu}{2}\left\|\left(\hat{g} - \sqrt{T}(FP^H \otimes I_k)h\right)\right\|_2^2 + \frac{\gamma}{2}\|\hat{g}(t)\varphi\|_2^2 \tag{24}$$

In equation (24), $\hat{\xi} = [\hat{\xi}_1{}^H, \ldots, \hat{\xi}_K{}^H]^H$ is the Lagrange vector, and μ is the penalty factor. The alternate direction method of multipliers (ADMM) is used to solve (24). The target problem is transformed into two subproblems, g^* and h^* for solving. Both subproblems are convex, smooth differentiable functions with closed-form solutions and are globally optimal.

Subproblem 1: Solving h^*. Given $\hat{y}$, $\hat{g}$, λ, μ, γ, φ, and P as known quantities in (24), subproblem 1 can be transformed into a subproblem with a closed-form solution, as follows:

$$h^* = arg\ min\left\{\frac{\lambda}{2}\|h\|_2^2 + \hat{\xi}^H\left(\hat{g} - \sqrt{T}(FP^H \otimes I_k)h\right) + \frac{\mu}{2}\left\|\hat{g} - \sqrt{T}(FP^H \otimes I_k)h\right\|_2^2\right\} = T\frac{(\mu g + \xi)}{\lambda + \mu T} \tag{25}$$

Subproblem 2: Solving g^*. Given the known quantities $\hat{y}$, $\hat{g}$, λ, μ, γ, and P in equation (24), subproblem 2 can also be transformed into a subproblem with closed-form solutions, as follows:

$$g^* = arg\ min\left\{\frac{1}{2}\|\hat{y} - \hat{x}\hat{g}\|_2^2 + \hat{\xi}^H\left(\hat{g} - \sqrt{T}(FP^H \otimes I_k)h\right) + \frac{\mu}{2}\left\|\hat{g} - \sqrt{T}(FP^H \otimes I_k)h\right\|_2^2 + \frac{\gamma}{2}\|\hat{g}(t)\varphi\|_2^2\right\} \tag{26}$$

As solving g^* directly in equation (26) is computationally large, it cannot achieve real-time tracking. Therefore, the problem of solving g^* is divided into L independent objective functions to solve, and the equation (26) can be decomposed into L linear subproblems of K×K

to solve separately. The value of $\hat{y}$, denoted as $\hat{y}(t)$, only depends on the K value of $\hat{x}(\mathrm{t}) = [\hat{x}_1(t), \ldots, \hat{x}_K(t)]^H$ and $\hat{g}(\mathrm{t}) = [conj(\hat{g}_1(t)), \ldots, conj(\hat{g}_K(t))]^H$, where conj(.) represents a conjugate operation. According to the Parseval theorem, equation (26) can be further optimized to obtain:

$$g^*(t) = arg\,min\{\frac{1}{2}\|\hat{y}(t) - \hat{x}(t)^H\hat{g}(t)\|_2^2 + \hat{\xi}(t)^H\big(\hat{g}(t) - \hat{h}(t)\big) + \frac{\mu}{2}\big\|\hat{g}(t) - \hat{h}(t)\big\|_2^2 + \frac{\gamma}{2}\|\hat{g}(t)\varphi\|_2^2\} \quad (27)$$

In equation (27), $\hat{h}(t) = [\hat{h}_1(t), \ldots, \hat{h}_K(t)]$, and $\hat{h}_K = \sqrt{D}FP^Hh_K$. Once again, according to the Parseval theorem, equation (27) can be further optimized, resulting in the following equation (28):

$$\hat{g}(t)^* = (\hat{x}(t)\hat{x}(t)^H + T(\mu + \gamma\varphi)I_K)^{-1}\big(\hat{y}(t)\hat{x}(t) - T\hat{\xi}(t) + T\mu\hat{h}(t)\big) \quad (28)$$

The time complexity of equation (28) is O(TK^3), which involves a large amount of calculation. To solve $(\hat{x}(\mathrm{t})\hat{x}(\mathrm{t})^H + T(\mu + \gamma\varphi)I_K)^{-1}$, the Sherman-Morrison theorem can be applied, given by $(A + uv^H)^{-1}$. Letting $u = v = \hat{x}(\mathrm{t})$, we derive the following equation(29):

$$(\hat{x}(t)\hat{x}(t)^H + T(\mu + \gamma\varphi)I_K)^{-1} = \frac{1}{T(\mu+\gamma\varphi)}I_K - \frac{1}{T(\mu+\gamma\varphi)} \times \frac{\hat{x}(t)\hat{x}(t)^H}{T(\mu+\gamma\varphi)+\hat{x}(t)^H\hat{x}(t)} \quad (29)$$

Substituting equation (29) into equation (28), we obtain:

$$\hat{g}(t)^* = \frac{1}{(\mu+\gamma\varphi)}(T\hat{y}(t)\hat{x}(t) - \hat{\xi}(t) + \mu\hat{h}(t)) - \frac{\hat{x}(t)}{(\mu+\gamma\varphi)b}(\frac{\hat{y}(t)\hat{s}_x(t)}{T} - \hat{s}_\xi(t) + \mu\hat{s}_h(t)) \quad (30)$$

In equation (30), $\hat{s}_x(\mathrm{t}) = \hat{x}(\mathrm{t})^H\hat{x}$, $\hat{s}_\xi(\mathrm{t}) = \hat{x}(\mathrm{t})^H\hat{\xi}$, $\hat{s}_h(\mathrm{t}) = \hat{x}(\mathrm{t})^H\hat{h}$, and $b = \hat{s}_x(\mathrm{t}) + T(\mu + \gamma\varphi)$. The time complexity of (30), which is O(TK), can meet the real-time tracking requirements.

Subproblem 3: Solving ξ. The Lagrange multiplier vector is updated as follows:

$$\hat{\xi}^{(i+1)} \leftarrow \hat{\xi}^{(i)} + \mu(\hat{g}^{(i+1)} - \hat{h}^{(i+1)}) \quad (31)$$

where $\hat{h}^{(i+1)}$ and $\hat{g}^{(i+1)}$ are the results of solving n+1 ADMM iterations in $\hat{h}$ and $\hat{g}$. Here, $\hat{h}^{(i+1)} = (FP^H \otimes I_k)h^{(i+1)}$. The parameter μ is updated by $\mu^{(i+1)} = min(\mu_{max}, \vartheta\mu^{(i)})$, where ϑ is a constant and μ_{max} is the maximum allowable value of μ.

D. RSABACF algorithm model update

In this paper, we adopt an online adaptive strategy to enhance tracking robustness in cases where the target undergoes deformation or changes scale, as given by:

$$\hat{x}_{model}^{(f)} = (1 - \eta)\hat{x}_{model}^{(f-1)} + \eta\hat{x}^{(f)} \quad (32)$$

In equation (32), η is the adaptive rate, and $\hat{x}_{model}^{(f)}$ is used to replace $\hat{x}(\mathrm{t})$ in equation (30) for solving $\hat{g}(\mathrm{t})^*$, $\hat{s}_x(\mathrm{t})$, $\hat{s}_\xi(\mathrm{t})$ and $\hat{s}_h(\mathrm{t})$.

IV. Experimental and Result

To facilitate comparison with current mainstream algorithms, we evaluated our proposed method on the standard tracking datasets OTB2013 [26] and OTB2015 [22]. OTB2013 is a widely used benchmark for tracking performance evaluation. OTB2015 extends OTB2013 by adding 50 video sequences and annotation of 11 attributes, including fast motion, background clutter, motion blur, deformation, illumination variation, in-plane rotation, low resolution, occlusion, out-of-plane rotation, out-of-view, and scale variation.

A. Evaluating indicator

The OTB dataset evaluates the robustness and quantitative analysis of tracking algorithms from two perspectives: tracking distance precision and tracking success rate. Typically, the mainstream method employs a single evaluation called One Pass Evaluation (OPE). Initially, the ground truth value of the first frame of the video sequence is used as the starting point for the evaluated algorithm. Then, the algorithm calculates and predicts the position of each frame. Finally, the average position precision and tracking success rate of all video sequences are outputted.

1) Precision Plots:

For quantitative analysis of precision, it is a common practice in image tracking to calculate the Center Location Error (CLE), which is the Euclidean distance between a target's real position and its predicted position. Assuming the coordinate position of the real pixel in each frame to be (m_a, n_a) and the coordinate of the tracking position to be (m_b, n_b), the CLE is calculated as follows:

$$CLE = \sqrt{(m_a - m_b)^2 + (n_a - n_b)^2} \quad (33)$$

During OTB evaluation, a distance precision evaluation can be conducted by counting the number of frames with center position errors less than 20 pixels, and the percentage of frames with errors less than 20 pixels to the total frames is visually displayed by the distance precision map.

2) Success Plots:

Another important evaluation index is the success rate (SR). The intersection ratio of the pixel area of the predicted target frame and the real target frame is used to calculate the tracking score. A count is made of the ratio of frames with overlap rates greater than a threshold to the total number of frames. Assuming the real target tracking box is marked as s_a and the target box obtained by the tracking algorithm is marked as s_b, the overlap rate is defined as follows:

$$Overlap\ Rate = \frac{s_a \cap s_b}{s_a \cup s_b} \quad (34)$$

As the overlap rate increases, the tracking effect becomes more accurate. The success rate diagram sets an overlap rate threshold of 0.5.

B. Experiment setting

MATLAB 2016a was used for programming in this study. The processor used was i5-12600KF, and the system memory was 16 GB. The operating system used was Windows 11. The following Table I is the experimental parameter settings.

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT TRACKERS

Parameter	Value
Number of HOG feature channels K	31
Regularization factor λ	0.01
Regularization factor β	1
Scale S	5
Scale step α	1.01
Number of iterations L	2
Penalty factor μ	1
Maximum penalty factor μ_{max}	1000
Value of ϑ for penalty factor update	10
Number of blocks ∂	12

C. Experimental result analysis

To verify the effectiveness of our proposed algorithm under different factors, we conducted tests on the OTB2015 dataset and compared it with 9 other algorithms. This section presents experimental analyses using both quantitative and qualitative approaches.

1) Quantitative Analysis

In the quantitative analysis, we selected BACF [19], DeepSRDCF [27], DSST [16], ECO-HC [28], KCF [29], SAMF [30], SRDCF [31], Staple [18], and STRCF [32] as the best algorithms in recent years. Fig.3 includes precision and success rate plots of the proposed algorithm and the nine tracking algorithms evaluated on the OTB2015 dataset. The success rate plots of each algorithm on 11 video attributes are shown in Fig.4. Tables II and III present the success rate and accuracy of each algorithm on the OTB2015 dataset and its 11 video attributes, respectively. Table IV shows the tracking speed of each algorithm.

Table II shows that our proposed algorithm achieved an accuracy of 84.7% and a success rate of 63.1%, ranking second to DeepSRDCF and outperforming traditional algorithms such as STRCF, ECO-HC, and SRDCF. Our algorithm designs the elliptical search area based on the target's motion state, which contributed to the improved performance. Compared to the BACF algorithm, the proposed algorithm exhibits an improvement of 3.7% in accuracy and 2.3% in success rate.

Fig.4 displays the success rate curves of each algorithm on the OTB2015 dataset under various video attributes. Our proposed algorithm achieved higher success rates than other algorithms under FM, BC, IV, and other attributes, while also demonstrating good performance in complex scenarios. Table III presents the tracking success rates of the proposed algorithm under FM, BC, and IV attributes, which are 0.629, 0.645, and 0.651, respectively.

Table IV presents the average number of frames that each algorithm processes on the OTB2015 dataset. Our proposed algorithm extracts background information from the elliptical search area, leading to a smaller number of background data compared to the BACF algorithm. This reduction in computation effectively improves the algorithm's comprehensive performance while maintaining real-time tracking speed of 31 FPS. Compared with the BACF algorithm, our proposed algorithm improves 3.5 FPS, further contributing to real-time tracking capabilities.

1) Qualitative Analysis

To provide a more intuitive representation of the tracking performance of various algorithms, we tracked and displayed each method on a few test sequences, as shown in Fig5.

In the deer sequence, the algorithm performance is tested against challenges such as motion blur (#11 and #25), fast motion (#11, #25, and #70), and background clutter (#25, #41, and #56). The target animal runs quickly in the water, resulting in a blurred image between frames 11 and 25. DSST and KCF fail to respond to these changes, resulting in tracking frame removal of the location of the target object. In frame 56, background data appears around the target that is similar to the target, leading to interference with target tracking. SAMF and KCF use background data similar to the target from the initial frame, resulting in tracking drift.

In the fleetface sequence, the target undergoes challenges such as scale variation (#433 and #690), fast motion (#494 and #591), and in-plane rotation (#494, #591, and #658). In frame 494, most trackers can detect the target when it is just rotated. However, in frame 658, SRDCF, KCF, SAMF, DSST, ECO-HC, and Staple lose some of the target data due to slight tracking drift caused by rotation. Our proposed algorithm copes well with the rotation of the target plane on a certain scale by increasing the background data in the direction of motion.

In the football sequence, the target faces challenges such as fast motion (#263 and #267), occlusion (#285 and #297), and background clutter (#277 and #362). In frame 285, the target is occluded by similar targets, which seriously interferes with the performance of the tracking algorithm. In frame 362, the ECO-HC, STRCF, and DSST algorithms lose the target due to complex background data, where the background response is greater than the actual target response, leading to tracking drift to a similar background.

In the lemming sequence, the scene features challenges such as fast motion (#250), background clutter (#303 and #381), and occlusion (#332 and #352). Before frame 332, each tracker can effectively track the target. Due to target occlusion, the ECO-HC, STRCF, Staple, and BACF

algorithms drift until tracking failure, while other algorithms can still track the target successfully.

In the singer2 sequence, deformation (#18 and #24) and illumination changes (#10, #18, and #24) appear in the background as the target moves, leading to deformation of the target. In frame 24, the SRDCF and BACF algorithms cannot adapt to the scale and appearance change of the target, resulting in tracking drift.

In the tiger1 sequence, the scene presents challenges such as background clutter (#48 and #59), occlusion (#59 and #312), and fast motion (#48, #68, and #317). In frame 59, the target is obscured by green plants, causing the BACF algorithm to lose the target and resulting in tracking failure. Due to the rapid motion of the target, in frame 317, the DSST and BACF algorithms drift during the tracking process. Our proposed algorithm addresses these challenges by increasing the proportion of background data in the motion direction by redesigning the elliptical search area, enhancing the robustness of the target under the challenges of fast motion and motion blur.

V. Conclusion

In this research, we proposed a novel approach for object tracking using the Kalman filter and a redesigned search area in the background-aware correlation filter tracking algorithm. Our proposed method provides two key advantages over current mainstream algorithms. Firstly, the elliptical search area improves the filter's classification capacity and enhances background data in the motion direction while decreasing background data in the non-motion direction. Secondly, our system can dynamically adjust the range and angle of the elliptical search area based on the target's motion status to accurately extract background data. The suggested approach outperforms the BACF algorithm in terms of tracking speed, with a high accuracy rate of 0.847 and a success rate of 0.641, while maintaining a fast frame rate of 38 FPS. Moreover, our experiments demonstrate that our proposed algorithm can handle complex scenarios effectively, including fast motion, motion blur, background clutter, and in-plane rotation. Future work may involve extending this method to other tracking algorithms and exploring ways to further improve the accuracy and robustness of object tracking. Overall, our findings demonstrate the potential effectiveness of using the Kalman filter and adjusted search area for object tracking in a wider range of applications.

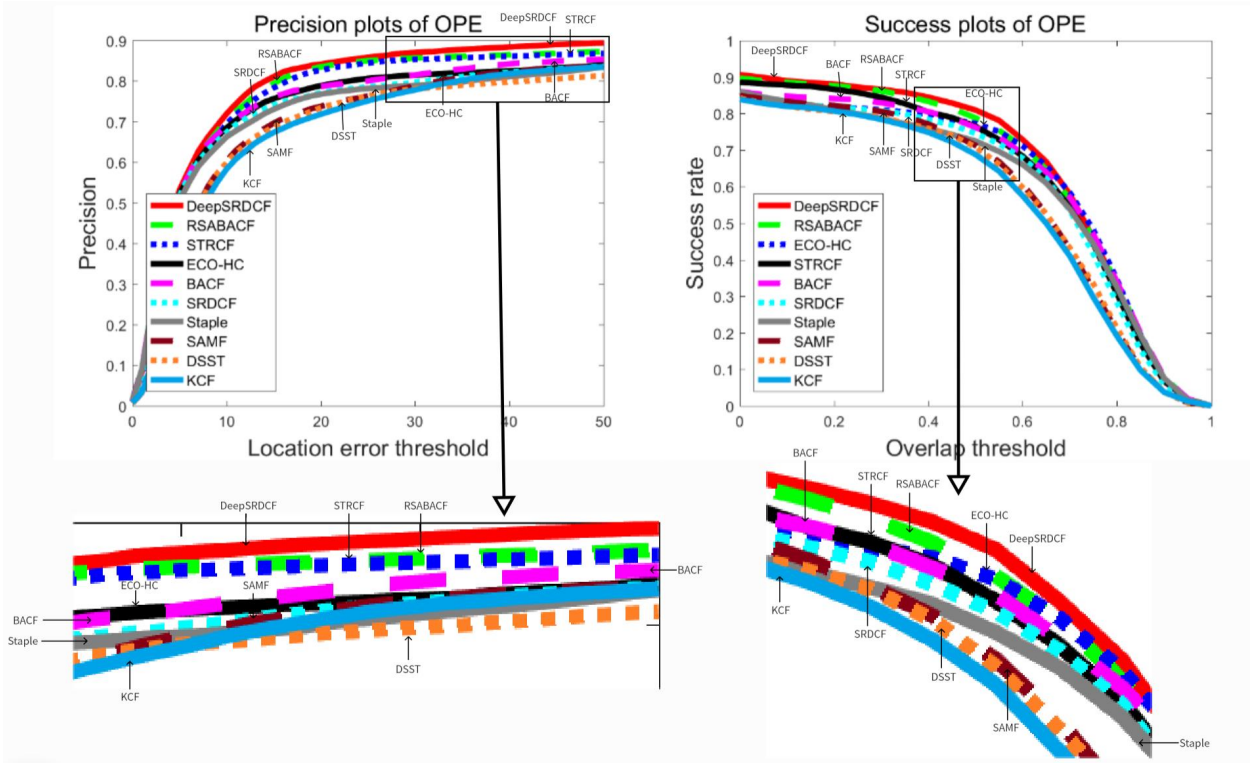
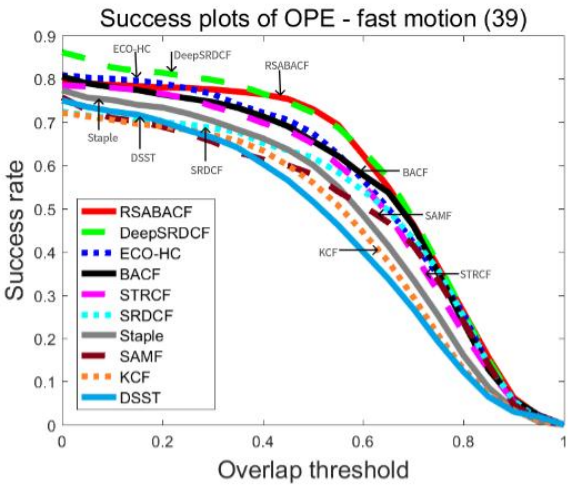
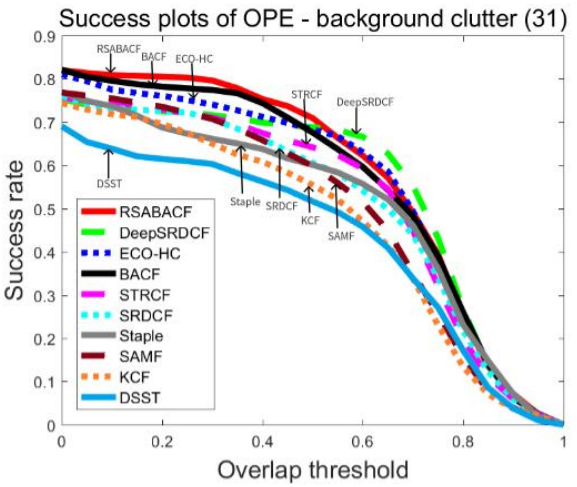
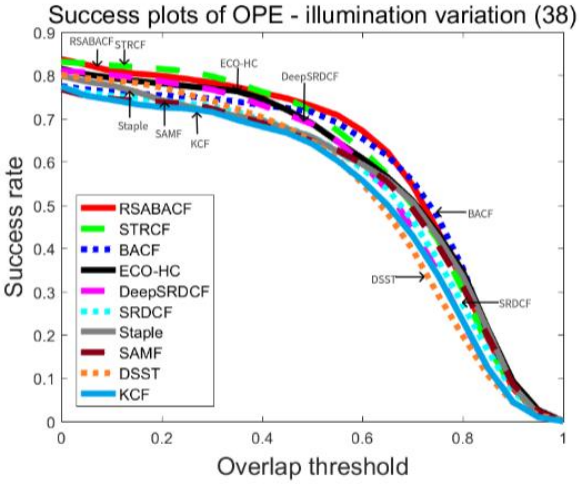
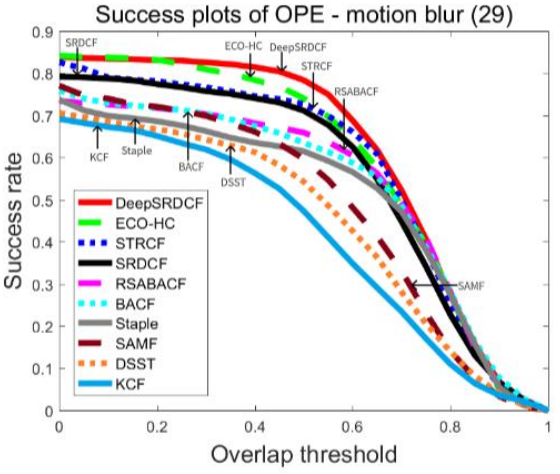


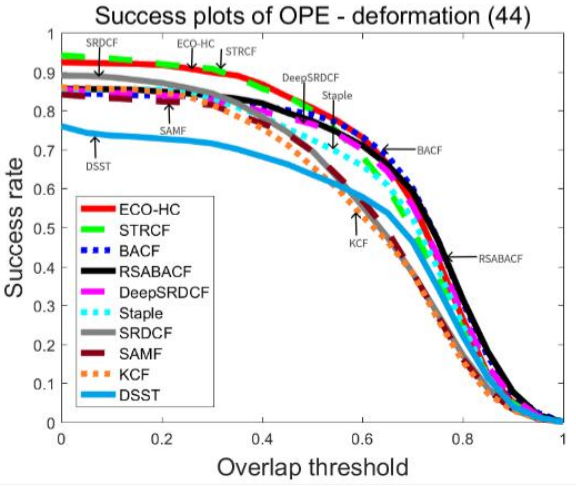
Fig. 3. Precision plots and successs rate plots of the proposed algorithm with 9 trackers on OTB2015 dataset

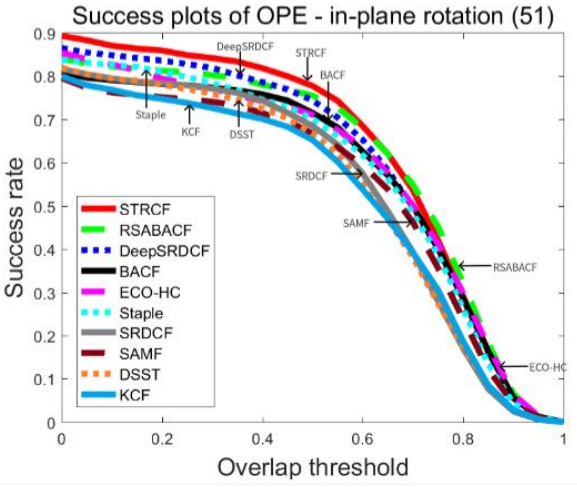

Success plots of OPE - fast motion (39)
Success rate
Overlap threshold
RSABACF
DeepSRDCF
ECO-HC
BACF
STRCF
SRDCF
Staple
SAMF
KCF
DSST


Success plots of OPE - background clutter (31)
Success rate
Overlap threshold
RSABACF
DeepSRDCF
ECO-HC
BACF
STRCF
SRDCF
Staple
SAMF
KCF
DSST


Success plots of OPE - illumination variation (38)
Success rate
Overlap threshold
RSABACF
STRCF
BACF
ECO-HC
DeepSRDCF
SRDCF
Staple
SAMF
DSST
KCF


Success plots of OPE - motion blur (29)
Success rate
Overlap threshold
DeepSRDCF
ECO-HC
STRCF
SRDCF
RSABACF
BACF
Staple
SAMF
DSST
KCF


Success plots of OPE - deformation (44)
Success rate
Overlap threshold
ECO-HC
STRCF
BACF
RSABACF
DeepSRDCF
Staple
SRDCF
SAMF
KCF
DSST


Success plots of OPE - in-plane rotation (51)
Success rate
Overlap threshold
STRCF
RSABACF
DeepSRDCF
BACF
ECO-HC
Staple
SRDCF
SAMF
DSST
KCF

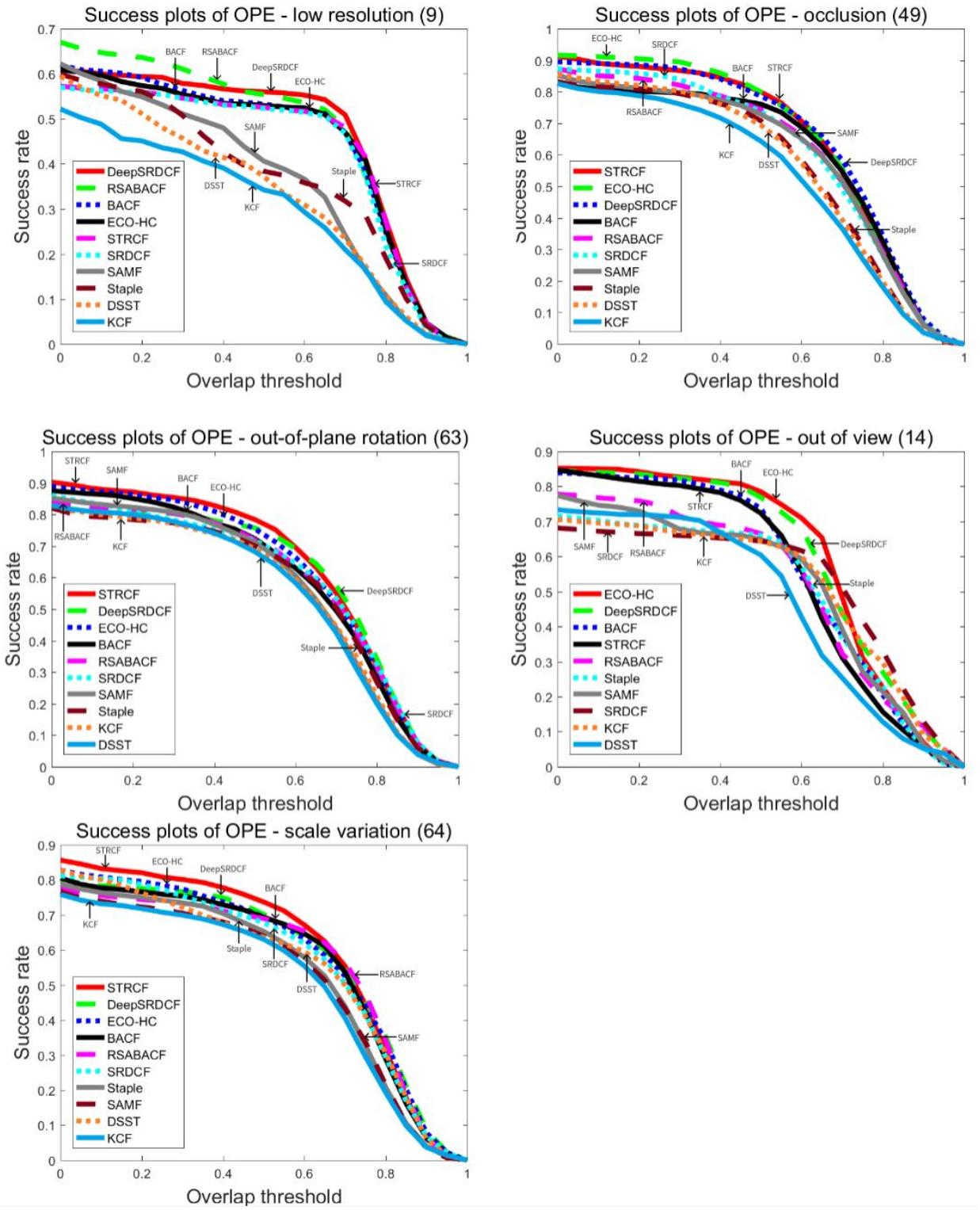


Fig. 4. Success rate plots of the proposed algorithm with 9 trackers on OTB2015 dataset

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT TRACKERS

	RSABACF	BACF	DeepSRDCF	DSST	ECO-HC	KCF	SAMF	SRDCF	Staple	STRCF
precision	84.7%	81.7%	85.1%	69.7%	84.0%	69.3%	75.4%	79.0%	78.4%	84.5%
Success rate	63.1%	61.7%	63.5%	49.2%	62.2%	47.7%	55.4%	59.9%	57.8%	62.9%

TABLE III
SUCCESS RATE OF THE PROPOSED ALGORITHM WITH 9 TRACKERS UNDER DIFFERENT CHALLENGE FACTORS

	FM	BC	IV	MB	DEF	IPR	LR	OCC	OPR	OV	SV
RSABACF	**0.629**	**0.645**	**0.651**	0.588	0.571	0.612	0.556	0.556	0.564	0.513	0.562
BACF	0.600	0.605	0.638	0.574	0.580	0.588	0.534	0.570	0.578	0.547	0.571
DeepSRDCF	0.627	0.627	0.621	**0.642**	0.566	0.589	**0.561**	0.601	0.607	0.553	0.607
DSST	0.403	0.487	0.497	0.469	0.410	0.493	0.336	0.480	0.453	0.393	0.436
ECO-HC	0.625	0.625	0.629	0.615	**0.590**	0.557	0.529	0.607	0.598	**0.588**	0.597
KCF	0.411	0.498	0.451	0.460	0.436	0.466	0.304	0.449	0.455	0.401	0.395
SAMF	0.501	0.535	0.534	0.525	0.512	0.518	0.410	0.542	0.537	0.484	0.501
SRDCF	0.593	0.583	0.605	0.594	0.544	0.529	0.507	0.549	0.547	0.457	0.560
Staple	0.508	0.569	0.590	0.543	0.550	0.535	0.389	0.527	0.534	0.505	0.523
STRCF	0.595	0.594	0.640	0.604	0.588	**0.621**	0.514	**0.627**	**0.611**	0.532	**0.628**

TABLE IV
SPEED (IN FRAMES PER SECOND (FPS)) ON THE OTB-2015 DATABASE OF 9 EXCELLENT TRACKERS AND OUR TRACKER

	RSABACF	BACF	DeepSRDCF	DSST	ECO-HC	KCF	SAMF	SRDCF	Staple	STRCF
FPS	38.0	34.5	0.2	87.1	27.6	424.2	19.4	6.5	59.1	29.7

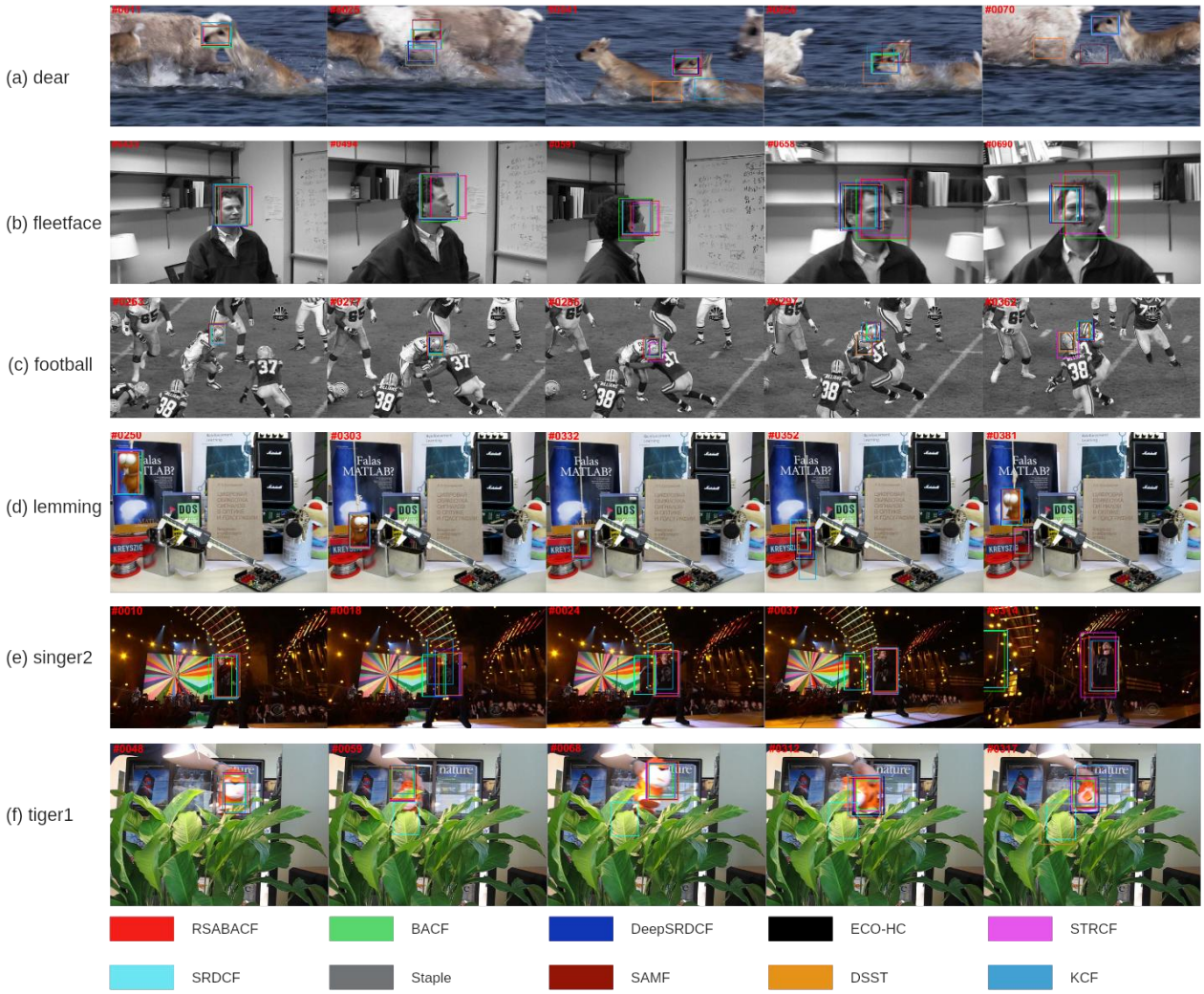


Fig. 5. Tracking results of each algorithm on OTB2015

REFERENCES

[1] Chen, R., V.G. Ferreira and X. Li, Detecting Moving Vehicles from Satellite-Based Videos by Tracklet Feature Classification. Remote Sensing, 2022. 15(1).

[2] Nurman, S., et al., Attitude Tracking Control on SO(3) Group with Linearization on Moving Operating Point for Transporting Quadrotor. IAENG International Journal of Computer Science, vol.48, no.4, pp940-951, 2021.

[3] Ghazaleh, K., K. Walid and H.S. Fouad, Link traffic speed forecasting using convolutional attention-based gated recurrent unit. Applied Intelligence, 2020. 51(4).

[4] Yuanbin, W., L. Yujie and H. Qian, "Vehicle-Mounted Infrared Pedestrian Tracking Based on Scale Adaptive Kernel Correlation Filter." IAENG International Journal of Computer Science, vol.49, no.2, pp349-356, 2022.

[5] Benying, T., et al., A novel dictionary learning method for sparse representation with nonconvex regularizations. Neurocomputing, 2020. 417.

[6] H., F. and X. J., Robust Visual Tracking with Multitask Joint Dictionary Learning. IEEE Transactions on Circuits and Systems for Video Technology, 2017. 27(5): p. 1018-1030.

[7] Nai, K., et al., Robust Object Tracking via Local Sparse Appearance Model. IEEE transactions on image processing: a publication of the IEEE Signal Processing Society, 2018. 27(10).

[8] M., W., L. Y. and H. Z. Large Margin Object Tracking with Circulant Feature Maps. in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017.

[9] T., Z., X. C. and H.Y. M. Multi-task Correlation Particle Filter for Robust Object Tracking. in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017.

[10] Jianming, Z., et al., Visual object tracking based on residual network and cascaded correlation filters. Journal of Ambient Intelligence and Humanized Computing, 2020. 12(8).

[11] B., H., S. J. and A. H. BranchOut: Regularization for Online Ensemble Tracking with Convolutional Neural Networks. in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017.

[12] M., D., et al. ECO: Efficient Convolution Operators for Tracking. in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017.

[13] D., S.B., et al. Visual object tracking using adaptive correlation filters. in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2010.

[14] HENRIQUES J F, C.R.M.P., Exploiting the Circulant Structure of Tracking-by-detection with Kernels. Lecture Notes in Computer Science, 2012. 7575.

[15] J., F.H., et al., High-Speed Tracking with Kernelized Correlation Filters. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015. 37(3): p. 583-596.

[16] M., D., et al., Discriminative Scale Space Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017. 39(8): p. 1561-1575.

[17] M., D., et al. Learning Spatially Regularized Correlation Filters for Visual Tracking. in 2015 IEEE International Conference on Computer Vision (ICCV). 2015.

[18] L., B., et al. Staple: Complementary Learners for Real-Time Tracking. in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016.

[19] H., K.G., F. A. and L. S. Learning Background-Aware Correlation Filters for Visual Tracking. in 2017 IEEE International Conference on Computer Vision (ICCV). 2017.

[20] Karl, V. and D. Wachsmuth, an augmented Lagrange method for elliptic state constrained optimal control problems. Computational Optimization and Applications, 2018. 69(3).

[21] Junli, L., et al., Robust ellipse fitting via alternating direction method of multipliers. Signal Processing, 2019. 164.

[22] Y., W., L. J. and H.Y. M., Object Tracking Benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015. 37(9): p. 1834-1848.

[23] Gang, W., X. Rui and W. Jinxin, A distributed maximum correntropy Kalman filter. Signal Processing, 2019. 160.

[24] Paine, P.J., et al., An elliptically symmetric angular Gaussian distribution. Statistics and Computing, 2018. 28(3).

[25] H., K.G., S. T. and L. S. Correlation filters with limited boundaries. in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015.

[26] Y., W., L. J. and H.Y. M. Online Object Tracking: A Benchmark. in 2013 IEEE Conference on Computer Vision and Pattern Recognition. 2013.

[27] M., D., et al. Convolutional Features for Correlation Filter Based Visual Tracking. in 2015 IEEE International Conference on Computer Vision Workshop (ICCVW). 2015.

[28] M., D., et al. ECO: Efficient Convolution Operators for Tracking. in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017.

[29] J., F.H., et al., High-Speed Tracking with Kernelized Correlation Filters. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015. 37(3): p. 583-596.

[30] Li Y, Z.J., A scale adaptive kernel correlation filter tracker with feature integration. European Conference on Computer Vision, 2014: p. 254-265.

[31] M., D., et al. Learning Spatially Regularized Correlation Filters for Visual Tracking. in 2015 IEEE International Conference on Computer Vision (ICCV). 2015.

[32] F., L., et al. Learning Spatial-Temporal Regularized Correlation Filters for Visual Tracking. in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2018.