ISSN 2616-5775 Vol. 6, Issue 11: 1-8, DOI: 10.25236/AJCIS.2023.061101

Target Tracking Based on Improved Kernel Correlation Filter

Xiaorong Qiu[1,2,a,*], Md Gapar Md Johar[1,b], Jacquline Tham[1,c], Lilysuriazna Raya[1,d]

[1]*Management & Science University, Selangor, Malaysia*
[2]*Wuxi Institute of Technology, Wuxi, China*
[a]*qiuxr@wxit.edu.cn,* [b]*mdgapar@msu.edu.my,* [c]*jacquline@msu.edu.my,* [d]*lilysuriazna@msu.edu.my*
[*]*Corresponding author*

Abstract: *In response to the shortcomings of using a single feature to describe targets and the limitations of target scale invariance in traditional kernel correlation filter tracking algorithms, this paper proposes a scale adaptive kernel correlation filter algorithm that integrates color attribute features. Using principal component analysis to obtain color attribute features with strong discrimination, in order to reduce computational complexity and achieve color adaptation, and designing scale adaptive filters to dynamically adjust the target scale. Compared with the original kernel correlation filter and its improved algorithm, this method has good adaptability in situations such as occlusion, deformation, rotation, and background cluster interferences.*

Keywords: *Computer Vision, Target Tracking, Kernel Correlation Filter, Color Adaptation, Scale Adaptation*

1. Introduction

Target tracking is a hot topic in the field of computer vision [1]. Its main purpose is to calculate the position of the given target in each frame of the image through appropriate methods in a given video sequence. Target tracking is widely used in fields such as video surveillance [2], human-computer interaction [3], and robotics [4]. Despite the continuous research on target tracking in recent years, various target tracking algorithms have been proposed and have good robustness. When the tracking target background is complex, the target is easily affected by various factors, such as changes in lighting, target deformation, target occlusion, similar targets, etc.

The accuracy of tracking results is an important measure of the excellence of target tracking algorithms. With the development of technology and continuous practice, target tracking algorithms based on Tracking-Detection theory have achieved great success. This type of method considers the process of selecting the target location as a classification problem between the target and the background. Train an online classifier through the tracked sequence, which determines the boundary of the target by distinguishing it from the background. This type of method widely uses machine learning algorithms, such as Boosting [5], Random Forests [6], Support Vector Machine [7], and other technologies, and has achieved good tracking results.

In recent years, Correlation Filter (CF) has made significant progress in the field of target tracking. Bolme et al. first proposed the Minimum Output Sum of Squared Error (MOSSE) [8], which applies correlation filter to target tracking for the first time, greatly improving tracking speed. Henriques et al. introduced Circular Structure with Kernel (CSK) [9] to greatly expand training samples without losing computational speed. In subsequent work, they proposed Kernelized Correlation Filters (KCF) [10], which improved tracking accuracy by utilizing Felzenszwalb's Fast Histogram of Oriented Gradient (FHOG) [11] based on CSK. Danelljan et al. [12] proposed the idea of color adaptation based on the Color Name (CN) [13] proposed by Weijer, selecting the most discriminative color features for tracking. Danelljan et al. proposed Discriminative Scale Space Tracker (DSST) [14] for the first time to address the issue of scale changes caused by the movement of targets before and after, using the idea of shift filter and scale filter. The two filters are trained separately and locally optimized to accurately estimate the target scale. At the same time, deep learning has also gained an increasingly important position in the field of target tracking. Li et al. combined CF and end-to-end methods to train in Convolutional Neural Networks (CNN) [15], which can achieve good tracking results.

ISSN 2616-5775 Vol. 6, Issue 11: 1-8, DOI: 10.25236/AJCIS.2023.061101

In order to improve the robustness of the tracking algorithm, this paper combines FHOG features and CN features as translation filters to obtain the target's translation position based on the traditional KCF framework. Simultaneously utilizing Principal Component Analysis (PCA) to obtain the most discriminative features and achieve color adaptation to reduce computational time. At the same time, the idea of shift filter and scale filter is adopted, and a 33 scale filter is added to accurately estimate the scale change of the target. Comparative experiments were conducted on standard datasets [16], and the experimental data showed that compared with classic tracking algorithms in recent years, our algorithm outperformed the comparison method on average in terms of appearance deformation, scale changes, lighting changes, background similar interference, and can meet real-time requirements.

The following contents of this paper are arranged as follows. Section 2 introduces the principle of kernel correlation filter. Section 3 introduces the optimization strategy of color adaptation. Section 4 introduces the optimization strategy of scale adaptation. Section 5 is experimental results and performance analyses. Section 6 is a summary and outlook.

2. Principle of Kernel Correlation Filter

2.1. Kernel Correlation Filter

The establishment of traditional kernel correlation filter tracking models is achieved by solving the minimum mean square error between the expected output and the actual output response. Find the appropriate filter w. Assuming a total of n training samples $X=[x_1,x_2,\cdots,x_n]$, the expected output $y=[y_1,y_2,\cdots,y_n]$, and the response function $f(z)=w^Tz$, the minimum mean square error between the expected output and the actual response output is shown in equation (1):

$$\min_w \sum_i (f(x_i)-y_i)^2+\lambda_t \|w\|^2 \tag{1}$$

In equation (1), λ_t is the coefficient of the regularization term to prevent overfitting. The solution of filter w is:

$$w=\left(X^TX+\lambda_t I\right)^{-1}X^Ty \tag{2}$$

In equation (2), I represents the identity matrix, and the superscript T represents the transpose operation. The inverse operation of equation (2) requires a large amount of computation, and utilizing the relevant properties of the complex frequency domain can reduce the computational complexity of the inverse operation. The expression of equation (3) in the complex frequency domain is

$$w=\left(X^HX+\lambda_t I\right)^{-1}X^Hy \tag{3}$$

X^H is the Hermitian transformation of X. That is, $X^H=\left(X^*\right)^T$, and X^* is the complex co-occurrence of X, which is equivalent to equation (2) and equation (3) in the real field.

2.2. Cyclic Shift

The cyclic shift operation can greatly increase the training sample set, thereby improving the robustness of the filter. Similar to the fundamental and harmonic waves of signals, based on their correlation properties [17], it can be quickly solved by projecting them into the complex frequency domain. The form of diagonalization of a cyclic shift matrix in Fourier space is as follows:

$$V=Fdiag(\hat{v})F^H \tag{4}$$

In equation (4), V is the cyclic shift matrix, v is the basis vector, F is the Fourier transform matrix, the superscript ^ represents Fourier transform, and the superscript H represents the conjugate transpose. If the training sample X is generated by the cyclic shift of the base sample x, then combine equation (3) and equation (4) to obtain equation (5):

ISSN 2616-5775 Vol. 6, Issue 11: 1-8, DOI: 10.25236/AJCIS.2023.061101

$$\hat{w} = \frac{\hat{x} \odot \hat{y}}{x^* \odot \hat{x} + \lambda_t} \tag{5}$$

In equation (5), the operator ^ represents the Fourier transform, the operator $\odot$ represents dot multiplication, and the operator * represents complex conjugation.

2.3. Kernel Trick

Equation (5) provides an analytical solution for linearly separable cases. The linear indivisibility issue can be resolved by defining kernel function κ and mapping input x to high-dimensional space $\varphi(x)$. A linear combination of $w = \sum_i a_i \varphi(x_i)$ of $\varphi(x)$ can serve as a representation for the filter w. In this way, the expression of response function $f(z) = w^T z$ in high-dimensional space is $f(z) = \sum_i a_i \varphi(x_i)\varphi(z)$. The expression of kernel function κ is $\kappa(x, x') = \left\langle \varphi(x), \varphi(x') \right\rangle$, and $\langle , \rangle$ is the dot multiplication operator. So the kernel transformation form of response function $f(z)$ is:

$$f(z) = \sum_i a_i \kappa(z, x_j) \tag{6}$$

Combining Equation (6) and Equation (1), the following can be obtained:

$$\alpha = (K + \lambda_t I)^{-1} y \tag{7}$$

In equation (7), K is a $n \times n$ kernel matrix, and the expression is $K_{ij} = \kappa(x_i, \mathrm{x}_j)$. When K is a cyclic shift matrix, combined with equation (4), the fast solution form of equation (7) can be obtained as:

$$\hat{\alpha} = \frac{\hat{y}}{\hat{k}^{xx'} + \lambda_t} \tag{8}$$

In equation (8), the operator ^ represents the Fourier transform, and $k^{xx'}$ represents the first row of the kernel matrix K.

By defining K^z as the kernel dependent cyclic shift matrix of training sample x and candidate region z, K^{xz} is the first row of K^z, and equation (6) can be represented as $f(z) = (K^z)^T a$. CombinE equation (4), the output response can be obtained as

$$\hat{f}(z) = \hat{k}^{xz} \odot \hat{a} \tag{9}$$

In equation (9), the operator ^ represents the Fourier transform, the operator $\odot$ represents dot multiplication.

2.4. Template Updating

In order to reduce computational complexity, the following update strategy is adopted:

$$\begin{aligned} x_t &= (1 - \eta_t) x_t + \eta_t x_{t-1} \\ a_t &= (1 - \eta_t) a_t + \eta_t a_{t-1} \end{aligned} \tag{10}$$

Equation (10) x_t represents the training sample of current frame, x_{t-1} represents the training sample from the previous frame, a_t represents the current frame weight coefficient, and a_{t-1} represents the number of weight coefficients from the previous frame, η_t is the template update rate.

3. Color Adaptation

The traditional KCF algorithm uses grayscale color features during the tracking process. This paper utilizes the advantages of color attributes that are insensitive to changes in light and partial occlusion as target features. At the same time, in order to avoid the time complexity caused by high-dimensional color

ISSN 2616-5775 Vol. 6, Issue 11: 1-8, DOI: 10.25236/AJCIS.2023.061101

features, PCA is used to adaptively reduce the dimensionality of high-dimensional color features, thereby improving the efficiency of the algorithm.

Berlin et al. [18] classified 11 basic colors by studying color attributes and target features: black, blue, brown, gray, green, orange, pink, purple, red, white, and yellow. They referred to this color naming method as CN, which performs better than other feature spaces related to color attributes. Therefore, in this paper, RGB space image features are mapped to CN space in order to improve tracking accuracy.

However, when using 11 dimensional color features, the tracking speed will be significantly slowed down. Therefore, an adaptive dimensionality reduction strategy is adopted to reduce the dimensions of other redundant color features while retaining useful feature information so as to improve tracking speed.

By minimizing the cost function (11), a suitable dimensionality reduction strategy can be found for the current frame p.

$$\eta^{p} = \alpha_{p}\eta_{data}^{p} + \sum_{j=1}^{p-1}\alpha_{j}\eta_{smooth}^{j} \tag{11}$$

In this formula, η_{data} represents the data item, determined by the current frame, while η_{smooth} represents the smooth item, related to frame j. The weights of data items and smooth items are α.

Then, by using $\hat{x}^{p}$ to represent the D1 dimensional target model, the mapping matrix B_{p}, with the size of $D_{1}\times D_{2}$, composed of orthogonal vectors can be found to achieve dimensionality reduction. B_{p} calculates the D2 dimensional feature map $\tilde{x}^{p}$ through linear mapping by using formula $\tilde{x}^{p}(m,n) = B_{p}^{T}\hat{x}^{p}(m,n), \forall m,n$. From this, it can be concluded that the data items including the current target feature reconstruction error are:

$$\eta_{data}^{p} = \frac{1}{MN}\sum_{m,n}\left\| \hat{x}^{p}(m,n) - B_{p}B_{p}^{T}\hat{x}^{p}(m,n)\right\|^{2} \tag{12}$$

The method of minimizing (12) is to perform PCA calculation on the current target feature $\hat{x}^{p}$. But updating the data items using this method alone will reduce the overall quality of the algorithm, as the previously trained classifier coefficients have not been updated accordingly.

Therefore, a smoothing term has been added in (11) as follows:

$$\eta_{smooth}^{j} = \sum_{k=1}^{D2}\lambda_{j}^{(k)}\left\| b_{j}^{(k)} - B_{p}B_{p}^{T}b_{j}^{(k)2}\right\| \tag{13}$$

By using formulas (12) and (13), the cost function (11) is minimized by decomposing the eigenvalues of matrix $R_{p} = \alpha_{p}C_{p} + \sum_{j=1}^{p-1}\alpha_{j}B_{j}\Lambda_{j}B_{j}^{T}$ under the constraint of $B_{p}B_{p}^{T} = I$. Among them, C_{p} is the covariance matrix of the current target feature, and Λ_{j} is the diagonal matrix with weight value $\lambda_{j}^{(k)}$ and size $D_{2}\times D_{2}$, respectively. In equation (11), α_{j} is the preset learning factor which ensures that the feature of the target in the previous frame can be effectively preserved during dimensionality reduction.

4. Scale Adaptation

When the target scale changes significantly, the traditional KCF algorithm often introduces too much background information or overemphasizes the local details of the target due to the incomplete description of the current target's features by the detected samples, leading to errors in the filtering response peak and resulting in tracking drift or even loss of the target. To improve this issue, a classifier for estimating the target scale is trained through a scale pyramid. This classifier can independently calculate the scale of the target after the classifier based on the target color attribute estimates the target position, thereby improving the accuracy of the algorithm. In order to reduce time complexity, target scale evaluation and target position tracking are performed separately. After the target position is

ISSN 2616-5775 Vol. 6, Issue 11: 1-8, DOI: 10.25236/AJCIS.2023.061101

preliminarily calculated using an adaptive color filter, the target scale size is estimated at that position using an adaptive target scale filter.

This paper introduces a one-dimensional correlation filter for the algorithm to estimate the scale of the target. $P*R$ represents the target size of the current frame, and S represents the size of the scale filter. Set a as the scaling factor, and for each $n \in \{-(S-1)/2, \cdots, (S-1)/2\}$, extract the image frame J_n with the size $a^n P * a^n R$ centered on the target. The learning scaling coefficient B is calculated using equation (14), and the scaling filter response $\hat{y}$ is calculated using equation (15). Update the corresponding parameters according to equation (16) and equation (17).

$$B_s = \frac{F(y_s)}{F(k)+\lambda} \tag{14}$$

The element of k is $k_i = k(x_s, x_{si})$, and y_s is the expected output of the target scale classifier.

$$\hat{y}_s = F^{-1}(B_s H_s) \tag{15}$$

$H_s = F(h_s)$, $h_s = k(x_s, z_{si})$, x_s is the scale model learned from the previous image frame, and z_s is the samples obtained from the new image frame.

$$\begin{aligned} \hat{B}_{s_t} &= (1-\beta)\hat{B}_{s_{t-1}} + \beta B_{s_t} \\ \hat{x}_{s_t} &= (1-\beta)\hat{x}_{s_{t-1}} + \beta x_{s_t} \end{aligned} \tag{16}$$

In formula (16), β is the scale learning factor, where $\hat{B}_{s_t}$ and $\hat{B}_{s_{t-1}}$ are the updated scale kernel coefficients B for the current and previous image frames, respectively. $\hat{x}_{s_t}$ and $\hat{x}_{s_{t-1}}$ represent the updated scale model x_s for the current and previous image frames, respectively.

5. Performance Analysis

The following experiment is conducted using MATLAB programming environment, with computer configured as Inter Core i5-7500CPU@3.40GHz and 16G DDR4 2666 RAM. In order to verify the effectiveness of the designed algorithm, three image frame sequences in OTB50 provided by OTB1.0 [19], namely BlurFace, Bolt and Crowds, are selected for testing and analyzing. At the same time, it is compared with CN, CSK, KCF, SAMF and other algorithms.

In the experiment, the algorithm parameters were fixed. The color dimension after dimensionality reduction was 2, the scale sample size S was 33, and the learning factor was 0.075. The conventional One Pass Evaluation (OPE) evaluation method was used to achieve the evaluation purpose.

5.1. Algorithm Performance Analysis

In order to quantitatively analyze the performance of tracking methods and the robustness of algorithms under different conditions, Center Location Error (CLE) and Overlap Precision (OP) were used as evaluation criteria. The calculation of these two evaluation criteria is shown in equations (17) and (18). The CLE and OP experimental data of the algorithm are shown in Table 1 and Table 2.

5.1.1. Center Location Error

The expression for CLE has been added in (17) as follows. n is the number of image frames in the tracking sequence, C_i is the target center position calculated by the algorithm, and C_i^{gt} is the standard center position.

$$\varepsilon = \frac{1}{n}\sum_{i-1}^{n}\left|C_i - C_i^{gt}\right| \tag{17}$$

ISSN 2616-5775 Vol. 6, Issue 11: 1-8, DOI: 10.25236/AJCIS.2023.061101

Table 1: Average Center Location Error

NO.	Sequence	Length	CN	CSK	KCF	SAMF	***Proposed Algorithm***
1	BlurFace	493	8.191	9.681	8.364	7.448	***4.525***
2	Bolt	350	5.849	415.635	6.365	5.701	***5.063***
3	Crowds	347	3.706	3.691	3.065	3.812	***2.994***
	Average		5.915	143.002	5.931	5.653	***4.194***

According to Table 1 and Figure 1, it can be seen that among the three tested image frame sequences, the proposed algorithm's average CLR is 4.194, which achieves optimal performance compared to the other four algorithms.

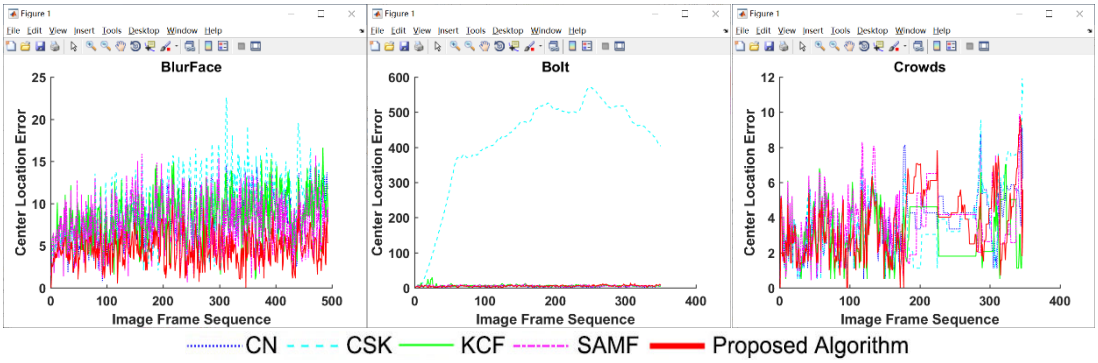

Figure 1: Average Center Location Error

5.1.2. Overlap Precision

The expression for OP is:

$$OP = \frac{R_{gt} \cap R_{tr}}{R_{gt} \cup R_{tr}} \tag{18}$$

In the above formula, R_{gt} stands in for the initial calibrated target region and R_{tr} stands for the target area that the algorithm tracks in real time. The $\cap$ and $\cup$ represent how R_{gt} and R_{tr} were intersected and united respectively.

Table 2: Average Overlap Precision

NO.	Sequence	Length	CN	CSK	KCF	SAMF	***Proposed Algorithm***
1	BlurFace	493	***1.000***	***1.000***	***1.000***	***1.000***	***1.000***
2	Bolt	350	***0.998***	0.017	0.943	0.974	0.986
3	Crowds	347	0.978	0.994	***0.999***	0.986	***0.999***
	Average		0.992	0.670	0.981	0.987	***0.995***

According to Table 2 and Figure 2, it can be seen that among the three tested image frame sequences, the proposed algorithm's average OP is 0.995, which achieves optimal performance compared to the other four algorithms.

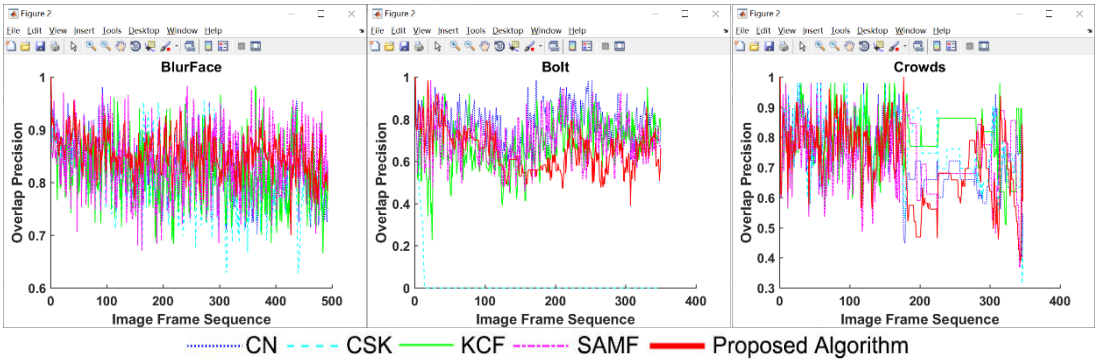

Figure 2: Overlap Precision

ISSN 2616-5775 Vol. 6, Issue 11: 1-8, DOI: 10.25236/AJCIS.2023.061101

5.2. Algorithm Result Analysis

5.2.1. Qualitative test of BlurFace image frame sequence

As shown in Figure 3, the target in this sequence is mainly affected by the factors such as motion blur, fast motion, and in-plane rotation when it is moving in indoor scenes. For example, at frame 40th, the camera begins to move rapidly. And at frames 155th and 230th, due to the rapid movement of the camera, the target generates some motion blur. At frame 440th, there is a certain in-plane rotation of the target. Through experiments, it can be seen that the proposed algorithm consistently tracks the target better than other four algorithms.

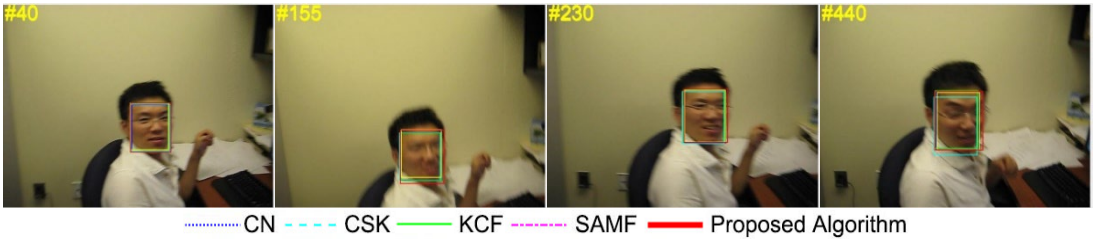

Figure 3: Qualitative test (BlurFace sequence)

5.2.2. Qualitative test of Bolt image frame sequence

As shown in Figure 4, the moving target in this sequence is mainly affected by factors such as occlusion, deformation, in-plane rotation, and out-of-plane rotation when it is moving in outdoor scenes. For example, from frame 150th to frame 220th, the target is moving in a straight line and is affected by the factors such as occlusion, deformation, and in-plane rotation. Afterwards, from frame 250th to frame 350th, because the target is running in a bend, it experienced a certain degree of out-of-plane rotation. Through experiments, it can be seen that the proposed algorithm consistently tracks the target well.

Figure 4: Qualitative test (Bolt sequence)

5.2.3. Qualitative test of Crowds image frame sequence

As shown in Figure 5, the moving target in this sequence is mainly affected by factors such as occlusion, deformation, and background cluster when it is moving in outdoor scenes. For example, from frame 40th to frame 170th, the target is moving in a straight line from the right to the left, with certain internal and external factors such as occlusion, deformation, and background cluster. From then on to frame 300th, the target has a certain turning motion until it finally leaves the screen. Through experiments, it can be seen that the proposed algorithm consistently tracks the target well.

Figure 5: Qualitative test (Crowds sequence)

6. Conclusion

Under the traditional target tracking framework based on detection and tracking, this paper proposes an adaptive target tracking algorithm based on kernel correlation filter. The algorithm combines an adaptive color attribute target position filter and an adaptive target scale filter, which not only enhances

ISSN 2616-5775 Vol. 6, Issue 11: 1-8, DOI: 10.25236/AJCIS.2023.061101

the extraction of target features but also accurately calculates the target scale, thereby enabling the algorithm to adapt to changes in target deformation, scale changing, light variation, and background similar and other complex scenarios. Experimental data shows that compared to traditional kernel correlation filter algorithms, adaptive algorithms that combine color attributes and scale filtering have higher robustness.

References

[1] Kristan M, Leonardis A, Matas J, et al. The tenth visual object tracking VOT2022 challenge results [C]. European Conference on Computer Vision(ECCV), 2022: 431-460.
[2] Dilshad N, Hwang J Y, Song J S, et al. Applications and challenges in video surveillance via drone: A brief survey [C]. 2020 International Conference on Information and Communication Technology Convergence (ICTC), 2020: 728-732.
[3] Tsai T H, Huang C C, Zhang K L. Design of hand gesture recognition system for human-computer interaction [J]. Multimedia Tools and Applications, 2020, 79: 5989-6007.
[4] Tang Y, Chen M, Wang C, et al. Recognition and localization methods for vision-based fruit picking robots: A review [J]. Frontiers in Plant Science, 2020, 11: 510.
[5] Grabner H, Leistner C, Bischof H. Semi-supervised on-line boosting for robust tracking [C]. European Conference on Computer Vision(ECCV), 2008: 234-247.
[6] Saffari A, Leistner C, Santner J, et al. On-line random forests [C]. IEEE International Conference on Computer Vision Workshop(ICCVW), 2009: 1393-1400.
[7] Hare S, Golodetz S, Saffari A, et al. Struck: Structured output tracking with kernels [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(10): 2096-2109.
[8] Bolme D S, Beveridge J R, Draper B A, et al. Visual object tracking using adaptive correlation filters [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),2010: 2544-2550.
[9] Henriques J F, Caseiro R, Martins P, et al. Exploiting the circulant structure of tracking-by-detection with kernels [C]. European Conference on Computer Vision(ECCV), 2012: 702-715 .
[10] Henriques J F, Caseiro R, Martins P, et al. High-speed tracking with kernelized correlation filters [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(3):583-596.
[11] Felzenszwalb P F, Girshick R B, McAllester D, et al. Object detection with discriminatively trained part-based models [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010,32(9): 1627- 1645.
[12] Danelljan M, Shahbaz Khan F, Felsberg M, et al. Adaptive color attributes for real-time visual tracking [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014: 1090- 1097.
[13] Van De Weijer J, Schmid C, Verbeek J, et al. Learning color names for real-world applications [J]. IEEE Transactions on Image Processing, 2009, 18(7): 1512-1523.
[14] Danelljan M, Häger G, Khan F, et al. Accurate scale estimation for robust visual tracking [C]. British Machine Vision Conference, 2014.
[15] Li H, Shi L. Robust event-based object tracking combining correlation filter and CNN representation [J]. Frontiers in Neurorobotics, 2019, 13: 82.
[16] Wu Y, Lim J, Yang M H. Online object tracking: a benchmark [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, 24: 11-24.
[17] Gray R M. Toeplitz and circulant matrices: A review [J]. Foundations and Trends in Communications and Information Theory, 2006, 2(3): 155-239.
[18] Berlin B, Kay P. Basic color terms: their universality and evolution [M]. Univ of California Press, 1991.
[19] Wu Y, Lim J, Yang M H. Online object tracking: A bench-mark [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013: 2411-2418.