Intelligent Automation And Soft Computing, 2020

Vol. 26, no. 2, 313–322
https://doi.org/10.31209/2019.100000105

Robust Visual Tracking Models Designs Through Kernelized Correlation Filters

Detian Huang[1], Peiting Gu[2], Hsuan-Ming Feng[3], Yanming Lin[1] and Lixin Zheng[1]

[1]Fujian Provincial Academic Engineering Research Centre in Industrial Intellectual Techniques and Systems, College of Engineering, Huaqiao University, Quanzhou 362021, China
[2]College of Mathematics and Computer Science, Quanzhou Normal University, Quanzhou 362000, China
[3]Department of Computer Science and Information Engineering, National Quemoy University, No.1 University Rd, Kin-Ning Vallage Kinmen 892, Taiwan

ABSTRACT

To tackle the problem of illumination sensitive, scale variation, and occlusion in the Kernelized Correlation Filters (KCF) tracker, an improved robust tracking algorithm based on KCF is proposed. Firstly, the color attribute was introduced to represent the target, and the dimension of target features was reduced adaptively to obtain low-dimensional and illumination-insensitive target features with the locally linear embedding approach. Secondly, an effective appearance model updating strategy is designed, and then the appearance model can be adaptively updated according to the Peak-to-Sidelobe Ratio value. Finally, the low-dimensional color features and the HOG features are utilized to determine the target state to further improve the robustness of the tracker. The experimental results on OTB-2015 benchmark validate that the proposed tracker can effectively solve the illumination variation, scale variation, partial occlusion and deformation in the complex background.

KEY WORDS: Tracking, Appearance model, Color attribute, Locally linear embedding, Multi-scale

1 INTRODUCTION

VISUAL tracking has very important practical applications in robotics (Chou and Nakajima, 2017), motion analysis (Syu, et. al., 2017), behavior recognition (Jo, et. al., 2017) and video surveillance (Yan, et. al., 2018), to name a few. Target tracking is essentially to obtain the location of interested target in the image through dealing with non-stationary, time-varying video stream containing the target and the background (Yin, et. al., 2013). Guo et. al. (2016) trained an adaptive mask by a combination of color distribution and weighting information, and proposed a robust vehicle tracker to effectively detect and track vehicles in night scenarios. Hsia et. al. (2016) utilized an adaptive search pattern to refine the central area search which adapted the optimal search pattern methods, and proposed a directional prediction CamShift tracker to improve the tracking accuracy and speed. Lee et. al. (2019) adopted analytical learning method of validity level to develop a robust target tracking approach consisted of object detection, tracking and learning. Despite that the visual tracking technology has achieved rapid development and progress, and a large number of trackers have been proposed, there are still a series of challenges, such as background clutters, illumination change, scale variation, motion blur, occlusion, etc. Therefore, it can be said that developing a robust tracking algorithm is still a very tough work (Lin, et. al. 2018; Lu, et. al., 2017; Zhang, et. al., 2014; Zin and Yamada, 2016).

Henriques et. al. (2015) improved the Circulant Structure of Tracking-by-detection with Kernels (CSK) algorithm proposed by Henriques et. al., (2012) through extending the single-channel kernel correlation filter to multi-channel to gain multi-dimensional image features, and applying the histogram of oriented gradient (HOG) feature instead of the original raw pixels, a Kernelized Correlation Filters (KCF) tracker is proposed. This tracker can not only solve the tracking problem in nonlinear condition, but also achieve the attractive results both

CONTACT Hsuan-Ming Feng hmfeng@nqu.edu.tw

in accuracy and efficiency. However, the KCF tracker still has the following problems: 1) the tracker is sensitive to illumination; 2) the tracker does not have the capability of dealing with the change of the target scale; 3) the tracker is not able to handle the situation where the target is occluded.

The main contributions of this paper are summarized as follows. To design a robust visual tracking model, the KCF tracker is improved from the following three aspects. The color attribute is applied to represent the target to overcome the shortcoming of illumination sensitivity, and then the low-dimensional and illumination-insensitive target features are adaptively selected with the locally linear embedding (LLE) method from the color feature space to preserve the target information. Considering the problem of target appearance updating in the process of occlusion, an effective appearance model updating strategy is proposed to avoid the tracking errors caused by partial or full occlusion. In addition, in order to further enhance the robustness of the proposed tracker, the previously obtained low-dimensional color features are combined with HOG features to serve as a basis for determining the position of the target. In recent years, smart video surveillance systems have been attracting increasing interests. In our follow-up work, the proposed tracker will be applied to the smart video system as it can provide a focus of attention for further investigation of this system.

The remainder of this paper is organized as follows. Section 2 reviews some related works. Section 3 describes the original KCF tracker. Section 4 describes the proposed algorithm in detail. Section 5 presents experimental results and analyses. Section 6 concludes this paper.

2 RELATED WORKS

DUE to the high computational efficiency of correlation filters, correlation filters have been widely used in target detection, target recognition and other fields. At present, some scholars have adopted correlation filters to the visual tracking community, and a series of correlation filter-based tracking algorithms have been presented. Such algorithms do not depend on the edge, texture and other features of the target, and has low computation complexity because fast Fourier transform is introduced in the computation, so these algorithms can run in real-time and achieve promising tracking capability.

Bolme et. al. (2010) designed a minimum output sum of squared error (MOSSE) correlation filter used to model the target appearance, and it has the ability to deal with the change of the appearance. Henriques et. al. (2012) proposed CSK tracker which employs correlation filter in the kernel space. Henriques et. al. (2015) improved the CSK tracker and proposed the KCF tracker. The latter not only has significant tracking capability, but also has high efficiency. However, it is often sensitive to illumination, scale variation, occlusion and other disturbing factors. To solve the problem of the change of illumination and scale in the KCF algorithm, Li and Zhu (2014) proposed an effective scale adaptive tracking algorithm, which uses the features composed HOG and color-naming features to boost its performance. It can solve the problem of illumination and scale in the KCF tracker, but its computational complexity is too high. To handle the scale variation problem of the KCF tracker, Zhang, et. al. (2016) proposed an adaptive scale tracker based on KCF through designing a scale estimation strategy, and Ding, et. al. (2018) proposed a quadrangle KCF through estimating the scale of the object based on the positions of its four corners, which can deal with the scale variation that occurs when the targeted target is moving. To deal with the severe occlusion problem of the KCF tracker, Yang et. al. (2016) compared the confidence of the target with the maximum response score and trained an online support vector machine classifier. Yang, et. al. (2018) proposed a joint multi-feature correlation filter tracking algorithm based on HOG and color-naming features to improve the performance of the KCF algorithm in terms of occlusion and fast motion.

However, most of the above-mentioned and other existing tracking methods are only limited to predicting the location of the target, without considering the effects of illumination change, scale variation, and occlusion at the same time, which limits the robustness of the tracking algorithms in the complex background to a certain extent.

3 KCF TRACKER

APPLYING the property of circulant matrix, the KCF tracker trains a classifier with a positive example and negative examples obtained by translating it (Henriques, et. al., 2015). For the sake of simplicity, let an $n\times 1$ vector $\mathbf{x}$ denote the positive example. Assuming $\mathbf{X}$ is an $n\times n$ circulant matrix, and it can be constructed by all possible circularly shifts to $\mathbf{x}$,

$$\mathbf{X}=\mathrm{C}(\mathbf{x})=\begin{bmatrix} x_0 & x_1 & \cdots & x_n \\ x_n & x_1 & \cdots & x_{n-1} \\ \cdots & \cdots & \cdots & \cdots \\ x_1 & x_2 & \cdots & x_0 \end{bmatrix}, \quad (1)$$

where, the first row of $\mathbf{X}$ is the vector $\mathbf{x}$, and the second row is the result of shift $\mathbf{x}$ by one element. In this way, all the other rows of $\mathbf{X}$ can be derived.

Since ridge regression has a simple closed-form solution for any input and does not require complex iterations, it is used in the training process of the KCF tracker. The purpose of the KCF tracker training is to find a function $f(z)=\mathbf{w}^T z$ to minimize the squared

error over the samples x_i and their regression targets y_i,

$$\min_{\mathbf{w}} \sum_i \left(f(x_i) - y_i \right)^2 + \lambda \|\mathbf{w}\|^2 , \quad (2)$$

where, $\lambda > 0$, is a regularization parameter used to prevent over-fitting. According to the Representer theorem, we can get the optimal solution $\mathbf{w} = \sum_i \alpha_i \varphi(x_i)$ of Eq.(2). The vector $\boldsymbol{\alpha}$ is composed of all the elements α_i. So the parameter of the solution we need to solve becomes $\boldsymbol{\alpha}$ from $\mathbf{w}$. Finally, the simple closed-form solution of Eq.(2) is obtained by the Kernelized Regularized Least Square (KRLS) (Henriques, et. al., 2012),

$$\boldsymbol{\alpha} = \left(K + \lambda \mathbf{I} \right)^{-1} \mathbf{y} , \quad (3)$$

where, $\mathbf{I}$ is the identity matrix, $\mathbf{K}$ is a Kernel function matrix, and each element of the vector $\mathbf{y}$ is y_i. Since $\mathbf{K}$ is a circulant matrix, the following expression can be derived according to the property of the circulant matrix:

$$\hat{\boldsymbol{\alpha}} = \frac{\hat{\mathbf{y}}}{\hat{\mathbf{k}}^{xx} + \lambda} , \quad (4)$$

where, $\mathbf{k}^{xx}$ is the first row of the circulant matrix $K = \mathrm{C}(\mathbf{k}^{xx})$, and the hat ^ denotes the DFT of a vector.

4 PROPOSED ALGORITHM

4.1 Low-dimensional color features

IN recent years, the color attribute plays an important role in the target recognition, target detection and some other communities. In this paper, the color attribute is utilized to represent the target to solve the illumination sensitive problem in the visual tracking. Berlin and Kay (1969) divided the colors into 11 categories: black, blue, brown, gray, green, orange, pink, purple, red white and yellow. In our tracker, these 11-dimensional color features are used to represent the target to solve the illumination sensitivity problem. Considering that the high dimensional feature representation will increase the computational complexity, the LLE algorithm (Roweis and Saul, 2000) is employed to reduce the dimension of the color features; on the other hand, it also plays an important role in preserving useful target information.

The LLE algorithm utilizes linear reconstruction to reflect the non-linear structure in the high-dimensional data space, which enables the reduced dimension data maintain the original topology (Liu, et. al., 2016). Therefore, in our tracker, the LLE algorithm is adopted to reduce the dimension of the color features, which can effectively reduce the loss of target information caused by the dimensionality reduction. In this work, the principle of reducing the dimension of the color features is as follows:

(1) Find β nearest neighbour samples for each given sample. For the point x_i of the sample in the high-dimensional space, the distances between the point x_i and the other N-1 sample points are calculated and sorted, and the top β points are selected as nearest neighbour points of x_i.

(2) The original high-dimensional data is linearly expressed by the nearest neighbour points, and then the low-dimensional spatial data is obtained. An error function is defined as below:

$$\min \varepsilon(\mathbf{W}) = \sum_{i=1}^{N} \left| x_i - \sum_{j=1}^{\beta} \omega_{ij}^{'} x_{ij} \right|^2 , \quad (5)$$

where, $x_{ij} \left(j = 1,2,\cdots \beta \right)$ is the j[th] nearest neighbour point of x_i. $\omega_{ij}^{'}$ is the weight between x_i and x_{ij}, and it needs to satisfy two constraints: (1) each data x_i must be reconstructed by its nearest neighbour points; otherwise, $\omega_{ij}^{'} = 0$; (2) each row of the weights is summed to1, that is $\sum_{j=1}^{\beta} \omega_{ij}^{'} = 1$. To solve the matrix $\mathbf{W}$, we need to minimize the Eq.(5), and the local optimal reconstruction weight matrix is constructed.

$$\omega_{ij}^{'} = \frac{\sum_{m=1}^{k} \left(\mathbf{Q}^{\mathbf{i}} \right)_{jm}^{-1}}{\sum_{p=1}^{k} \sum_{q=1}^{k} \left(\mathbf{Q}^{\mathbf{i}} \right)_{pq}^{-1}} , \quad (6)$$

In general, $\mathbf{Q}^i$ is a singular matrix.

(3) The output vector of the sample point is calculated by $\omega_{ij}^{'}$ in Eq.(6) and its nearest neighbour point x_i. To map all the sample data into the low-dimensional space, and output the low-dimensional data under the premise of preserving the sample topology, we need to construct a cost function and minimize it in the mapping process:

$$\min \varepsilon(\mathbf{U}) = \sum_{i=1}^{N} \left| u_i - \sum_{j=1}^{\beta} \omega_{ij}^{'} u_{ij} \right|^2 , \quad (7)$$

where, $\varepsilon(\mathbf{U})$ denotes the value of the cost function, u_i is the output vector of x_i. $u_{ij} \left(1,2,\cdots,\beta \right)$ is the j[th]

nearest neighbour point of u_i, and it needs to meet two conditions, $\sum_{i=1}^{N} u_i = 0$, and $\sum_{i=1}^{N} u_i u_i^T = \mathbf{I}$. By minimizing the cost function $\varepsilon(\mathbf{U})$ to seek the optimal solution u_i, then $\varepsilon(\mathbf{U})$ can be written as,

$$\min \varepsilon(\mathbf{U}) = \sum_{i=1}^{N} \sum_{j=1}^{N} \mathbf{M}_{ij} u_i^T u_j , \tag{8}$$

where, $\mathbf{M}$ is an $N \times N$ symmetric matrix, and its expression is $\mathbf{M} = (\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W})$. From Eq.(8), to minimize the above cost function, $\mathbf{U}$ can be the constructed by the smallest d non-zero eigenvalues of $\mathbf{M}$.

The number β of the nearest neighbour points and the feature dimension d determine the performance of the LLE algorithm (Roweis and Saul, 2000). If β is too large, it will not represent the local features, and its performance is similar to the principal component analysis (PCA) algorithm; otherwise, it is incapable of reflecting the topological structure of the low-dimensional sample data. In this paper, through a large number of experimental verification, it can get better results when β is set to 7. If d is too large, the selected samples are usually affected by noise; otherwise, it is incapable of reflecting the feature of the samples. After a large number of experiments, we choose d to be 3. In the proposed tracker, the LLE algorithm is used to reduce the dimension of the color features, which can effectively reduce the loss of target information caused by the dimension reduction process.

4.2 Fast scale detection

In the actual scene, the scale of the target usually varies indefinitely. An effective target scale detection strategy can give an important boost to the tracking accuracy of the algorithm. As the KCF tracker uses the fixed template size, and does not have the ability to adapt to scale varies, it is easy to be influenced by scale variation. To handle this problem, a multi-scale filter is introduced to predict the target scale of the current frame through the previous frame (Li and Zhu, 2014). The calculation of the scale filter and the location filter is performed in parallel, which effectively improved the tracking efficiency. The steps of the scale detection are as follows.

Firstly, a series of multi-scale image blocks z_s^j are acquired at and around the target location p_{t-1} of the previous frame. Suppose $l * h$ denotes the size of the target at the current frame t, and S_t denotes the size of the scale filter. The size of the image blocks z_s^j is $a^j l \times a^j h$, where, a represents the scale factor, $j \in \left\{ \left\lfloor -\frac{S_t - 1}{2} \right\rfloor, \ldots, \left\lfloor \frac{S_t - 1}{2} \right\rfloor \right\}$.

Then, the HOG features are extracted from the acquired multi-scale image blocks z_s^j, and the Hanning Window is used to eliminate the edge interference, then the training samples are obtained.

Finally, the sample set is trained by the KRLS classifier, and the scale detection is performed according to the maximum response of the scale filter. Let f_j represent the HOG features extracted from the image blocks z_s^j, then the expression of the scale filter can be expressed,

$$G_s = \frac{\sum_{j=1}^{t} \hat{y}_s^j \odot F_j^*}{\sum_{j=1}^{t} F_j \odot F_j^*} , \tag{9}$$

where, y_s^j is the Gaussian label output of the sample g_s^j, F_j is the Fourier transform of the feature f_j, F_j^* is the conjugate of F_j, and the symbol $\odot$ represents element-wise multiplication in the Fourier domain.

$$y_s = F^{-1}(\alpha_s R_s), \tag{10}$$

where, $R_s = F(\mathbf{r}_s)$, the elements of the vector $\mathbf{r}_s$ are $r_{s_i} = k(\mathbf{x}_s, \mathbf{z}_{s_i})$. $\mathbf{x}_s$ denotes the scale model obtained in the previous frame, and $\mathbf{z}_{s_i}$ denotes the acquired sample in the current frame. F_j is the Fourier transform of the feature f_j, and F_j^* is the conjugate of F_j. The scale filter's response value G_s can be calculated by Eq.(9), and the target scale of the current frame is obtained by calculating the scale filter's maximum response $G_{\max}$. Then, the scale model updating is carried out according to Eq.(11) and Eq.(12).

$$\boldsymbol{\alpha}_{s_{t+1}} = (1-\gamma)\boldsymbol{\alpha}_{s_t} + \gamma \hat{\boldsymbol{\alpha}}_{s_t} , \tag{11}$$

$$\mathbf{x}_{s_{t+1}} = (1-\gamma)\mathbf{x}_{s_t} + \gamma \mathbf{x}_{s_{t-1}} , \tag{12}$$

where, the hat ^ also denotes the DFT of a vector, $\boldsymbol{\alpha}_{s_t}$ and $\boldsymbol{\alpha}_{s_{t+1}}$ are the updated coefficient vectors of the current frame and the next frame respectively. $\mathbf{x}_{s_{t-1}}$ denotes the appearance model learned from the previous frame, $\mathbf{x}_{s_t}$ and $\mathbf{x}_{s_{t+1}}$ represent the updated

model of the current frame and the next frame, respectively.

4.3 Adaptive appearance model updating

The appearance model updating strategy is critical for tracking algorithm. If the appearance model is not updated, the original appearance model cannot meet the tracking requirements due to the appearance change caused by illumination, scale, occlusion, etc. To solve this problem, a novel effective appearance model updating strategy based on Peak-to-Sidelobe Ratio (PSR) (Li, et. al., 2015) is proposed. According to the PSR value at the target location to determine whether the target is occluded, and the weight ω_t of the appearance model is adaptively updated, which makes the proposed algorithm have the anti-occlusion ability. Firstly, the initial frame x_0 is regarded as the training sample, and the initial classifier coefficients $\boldsymbol{\alpha}$ are obtained through training. Then, the weight of the appearance model is calculated from the PSR values of each frame. If the PSR value is less than the set threshold, it indicates that the target is occluded, and the template weight of the current frame is assigned as 0 and the appearance model will not be updated. Otherwise, the template weight is calculated according to the PSR value of the current frame, and then the new appearance model is updated. The proposed appearance model updating strategy is as follows,

$$\boldsymbol{\alpha}_{t+1} = (1-\gamma\omega_t)\boldsymbol{\alpha}_t + \gamma\omega_t\hat{\boldsymbol{\alpha}}_t, \tag{13}$$

$$\mathbf{x}_{t+1} = (1-\gamma\omega_t)\mathbf{x}_t + \gamma\omega_t\mathbf{x}_{t-1}, \tag{14}$$

$$\omega_t = \begin{cases} 0, & \mathrm{P}_t < P_{thresh} \\ 1, & \mathrm{P}_t \geq P_{thresh} \end{cases}, \tag{15}$$

where, P_t denotes the PSR value of the t^{th} frame, and $P_t = \dfrac{\max[f(z)] - \mu}{\sigma}$. μ and σ denote the mean and standard deviation of the target box of the current frame, respectively.

4.4 Target status determined

Although the color features are somewhat robust to illumination variation, they are difficult to adapt to severe illumination variation. Considering that the HOG feature also has good invariance to illumination, we utilize the previously obtained low-dimensional color features and the HOG features to serve as the basis for determining the target state (the position of the target).

It is assumed that the thresholds corresponding to the color feature and the HOG features are $P_{\text{thresh}}^{\text{color}}$ and $P_{\text{thresh}}^{\text{HOG}}$, respectively. If the PSR value of the current frame, including P_t^{color} and $\mathrm{P}_t^{\text{HOG}}$ corresponding to these two kinds of features, satisfies $P_{\text{t}}^{color} \geq P_{\text{thresh}}^{\text{color}}$ and $P_{\text{t}}^{HOG} \geq P_{\text{thresh}}^{\text{HOG}}$, which indicates that the current tracking result is very reliable, the state S_t of the target is determined by these two kinds of features,

$$S_t = \frac{2}{3}S_{\text{t}}^{color} + \frac{1}{3}S_{\text{t}}^{HOG}, \tag{16}$$

where, S_{t}^{color} denotes the target state of the current frame determined by the color feature, and S_{t}^{HOG} denotes the target state of the current frame determined by the HOG features.

Secondly, if only one of the features has a PSR value greater than the corresponding threshold, the target state of the current frame is determined by the feature alone,

$$\mathrm{S}_t = \begin{cases} S_t^{color}, & \mathrm{P}_{\text{t}}^{color} \geq P_{\text{thresh}}^{color} \quad and \quad \mathrm{P}_{\text{t}}^{HOG} < P_{\text{thresh}}^{HOG} \\ S_t^{HOG}, & \mathrm{P}_{\text{t}}^{HOG} \geq P_{\text{thresh}}^{HOG} \quad and \quad \mathrm{P}_{\text{t}}^{color} < P_{\text{thresh}}^{color} \end{cases}, \tag{17}$$

Finally, if the PSRs of both kinds of features are less than their corresponding thresholds, which indicates that the reliability of the tracking results is low. For this case, according to the target states respectively obtained by these two kinds of features, the state closer to the target of the previous frame is selected as the target state of the current frame,

$$\mathrm{S}_t = \underset{S_t^{color}, S_t^{HOG}}{\arg\min}\left(\left(\left(S_{t-1} - \mathrm{S}_t^{color}\right)^2, \left(S_{t-1} - \mathrm{S}_t^{HOG}\right)^2\right)\right), \tag{18}$$

where S_{t-1} stands for the target state of the previous frame.

5 EXPERIMENTS AND RESULTS

IN order to verify the effectiveness of our algorithm, one hundred video sequences of the OTB-2015 benchmark (Wu, et. al., 2015) are tested, and obtained results are compared with KCF (Henriques, et. al., 2015) tracker, discriminative scale space tracker (DSST) (Danelljan, et. al., 2014), and discriminant correlation filters network (DCFNet) (Wang, et. al., 2017) tracker. The experiment is implemented in MATLAB software, which runs on an Intel (R) Core (TM) i5-4590M CPU @ 3.30GHz with 8G of memory. In the experiment, the parameters of the other three algorithms are kept their original settings. The parameters of our algorithm are as follows: σ is set to 0.2, the learning factor γ is set to 0.075, the number β of nearest neighbours is 7, the feature dimension d is set to 3, and the number of the scale of the filter is set to 33. The thresholds $P_{\text{thresh}}^{\text{color}}$ and $P_{\text{thresh}}^{\text{HOG}}$ corresponding to the color feature and the HOG features are set to 4/5 of the PSR value corresponding to the second frame, respectively.

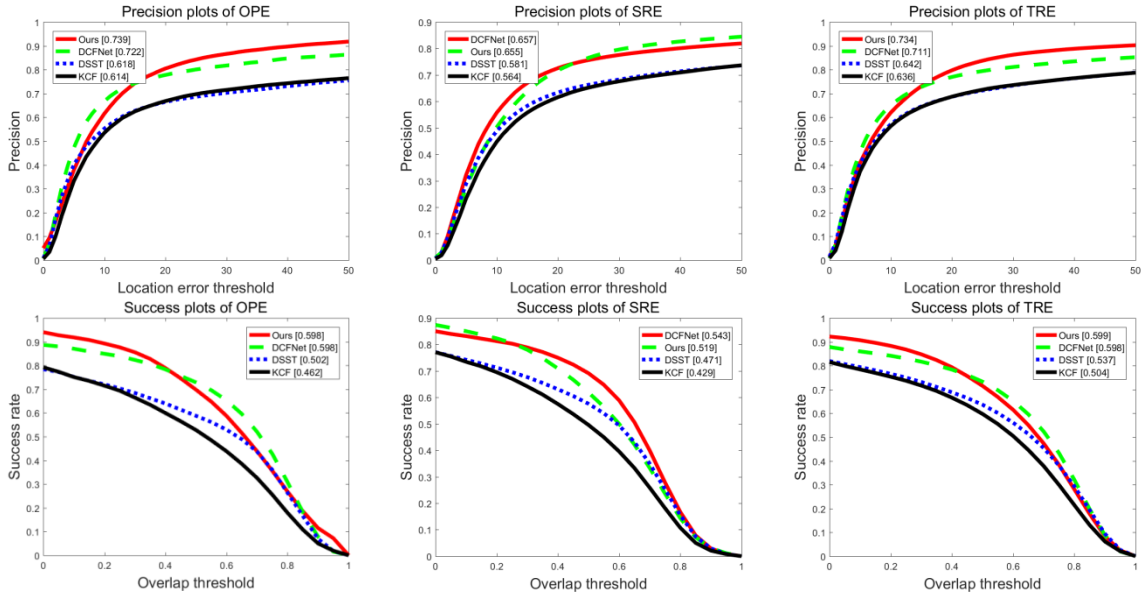

Figure 1. Precision and Success Plots for OPE, SRE and TRE. The Legend contains the ACU score for each tracker.

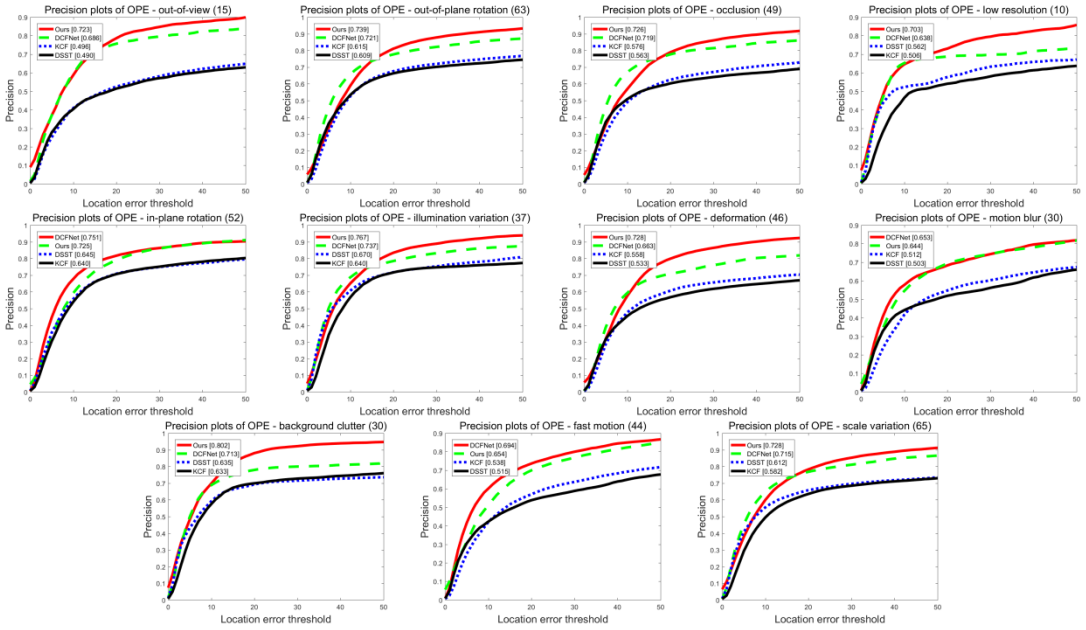

Figure 2. ACU results of each tracker on sequences with different challenge for OPE about precision

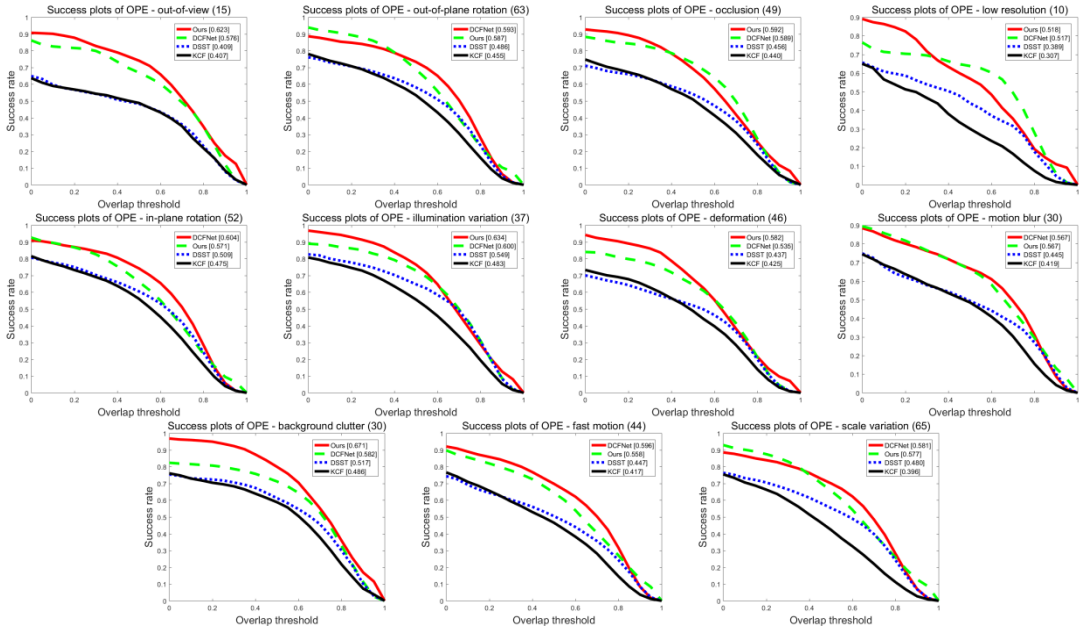

Figure 3. ACU results of each tracker on sequences with different challenge for OPE about success

5.1 Quantitative Analysis

To verify the accuracy and robustness of the proposed tracker, and considering that a tracker may be very sensitive to initialization, distance precision (DP), overlap success (OS), one-pass evaluation (OPE), temporal robustness evaluation (TRE), and spatial robustness evaluation (SRE) criteria (Wu, et. al., 2015), were selected to quantitatively evaluate these four algorithms in our experiment. The DP is defined as the percentage of frames whose centre location error (CLE) within a threshold of 20 pixels. The OS is defined as the percentage of frames where the bounding boxoverlap surpasses a threshold of 0.5.

Table 1. Quantitative comparison of 4 trackers on 100 sequences

Criteria	KCF	DSST	DCFNet	Ours
DP	0.668	0.665	0.778	***0.805***
OS	0.528	0.588	***0.729***	0.698

5.1.1 Overall Performance Analysis

The precision and success plots for OPE, TRE and SRE including the area-under-the curve (ACU) score over all 100 sequences are illustrated in Figure 1. As can be seen from Figure 1, although our algorithm is more sensitive to different bounding boxes than the DCFNet algorithm in terms of spatial robustness, it still has the best tracking performance among all four trackers in terms of time robustness. The DP and OS comparison of these four trackers is shown in Table 1. (Italic and bold format indicates the performance of the corresponding algorithm is optimal, underline indicates the performance of the corresponding algorithm is suboptimal.) From Table 1, we can see that the performance of the proposed tracker is optimal or suboptimal and our tracker performs well against the KCF (by 13.7%, 17%), DSST (by 14%, 11.4%) and DCFNet (by 2.7%, -3.1%) in terms of DP and OS, respectively.

5.1.2 Attribute-Based Performance Analysis

In order to fully evaluate the robustness of the proposed algorithm, we further evaluated the performance of the algorithm using 11 attributes on the OTB-100 video dataset as shown in Figure 2 and 3.

It can be seen from Figure 2 and 3, for the disturbing factors such as illumination (precision plots: 76.7% and success plots: 63.4%), scale changes (precision plots: 72.8% and success plots: 57.7%), occlusion (precision plots: 72.6% and success plots: 59.2%), the robustness of our tracker is better than that of the other three trackers.

5.2 Qualitative Analysis

In this paper, the comparison results of above four tracking algorithms for different benchmark video sequences are presented, and the tracking robustness of the algorithm is analysed in detail from four perspectives. In order to clearly illustrate the experimental results, only parts of the tracking results are shown.

Robustness to illumination variation. In the Singer1 and Singer2 video sequences shown in Figures 4(a) and 4(b), respectively, there is a challenge of illumination variation. The targets in

both videos are heavily affected by illumination variation, such as frames 120 and 339 in Singer1 sequences and frames 75 and 156 in Singer2 sequences. DCFNet drifted at frame 59 of Singer2, resulting in tracking failure in subsequent frames, but KCF, DSST and our trackers all track targets well. This is mainly because both color features and HOG features are robust to illumination variation. Among them, KCF and DSST adopt HOG features, and our tracker uses both color features and HOG features.

Robustness to scale variation. For the Car24 and Car1 video sequences shown in Figures 4(c) and 4(d), respectively, the main interference is scale variation. In the first few frames, the scale of the target changes little, and all trackers can better track the target, such as frames 434 in Car24 sequences and frames 31 in Car1 sequences. In the subsequent frames of the video, when the target scale changes, although all trackers can still track the target, KCF does not have the ability to scale estimation, while DSST, DCFNet and our algorithm can estimate the target scale, such as frames 624 and 2522 in Car24, and frames 191 and 193 in Car1.

Robustness to partial occlusion. In the Jogging2 and Subway video sequences shown in Figures 4(e) and 4(f), respectively, there is a challenge of occlusion. In the Jogging2 sequence, there are occlusions of the column, such as the 54th frame and the 57th frame, and in the Subway sequence, there are pedestrian occlusions, such as the 43rd frame and the 95th frame. After the target moves out of the occlusion, only DCFNet and our algorithm are able to reacquire the target in the Jogging2 sequence, such as the 65th frame; in the Subway sequence, DSST, DCFNet and our algorithm can recapture and track the target, such as the 145th frame. The main reason is that when occlusion occurs, it can be judged whether the target is occluded according to the PSR value, which largely avoids updating the background information to the template.

Robustness to deformation. Both the Bird2 and Panda video sequences, shown in Figures 4(g) and 4(h), respectively, have the challenge of deformation. In the Birds sequence, the target is moving and flipping at the same time, such as the 48th frame and the 51st frame, and the same situation exists in the Panda sequence, such as the 171st frame and the 288th frame. Only DCFNet and our algorithm can complete the whole tracking, while DSST and KCF lose the target, such as the 82nd frame in Bird2 sequence and the 215th frame in Panda sequence. This is because the color features are robust to deformation, and DCFNet constructs a unique feature extractor whose trained features are also robust to deformation, while DSST and KCF only use HOG features which is not robust to deformation.

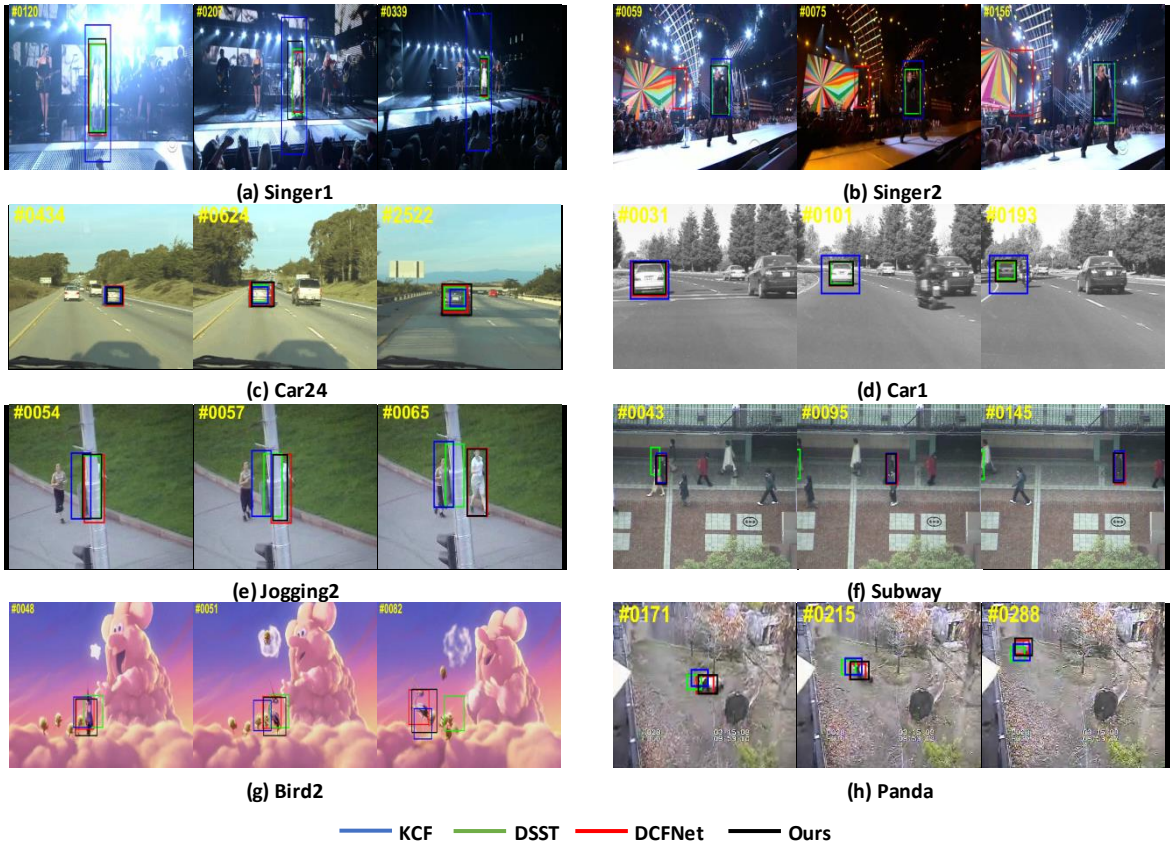

Figure 4. Different tracking results corresponding to different algorithms

6 CONCLUSION

TO overcome the shortcoming that the KCF tracker is not good at dealing with illumination variation, scale variation and occlusion, a robust tracking model based on KCF is proposed. In this work, the proposed model is improved from the following three aspects: the color feature is employed to represent the target, and the low-dimensional and illumination-insensitive target feature is obtained adaptively by the LLE algorithm. To make the appearance model adaptively update as the target appearance changes, an effective appearance model updating strategy is proposed. In addition, the obtained low-dimensional color features are combined with HOG features for determining the target state. One hundred video sequences of the OTB-2015 benchmark were selected for experiments, and the experimental results demonstrate that the performance of the proposed tracker is robust to illumination change, scale variation, partial occlusion and deformation.

Acknowledgment

This work was supported by the Nature Science Foundation of China (Grant No.61672335, No.61602191), by the Foundation of Fujian Education Department under Grand (No. JAT170053), by the Foundation of Quanzhou under Grand (No. 2017G046). The authors would like to thank the reviewers for their valuable suggestions and comments.

7 REFERENCES

Berlin, B. and Kay, P. (1969). Basic color terms: their universality and evolution, *Berkeley: University of California Press*. 11-25.

Bolme, D. S., Beveridge, J. R., Draper, B. A., and Lui, Y. M. (2010). Visual target tracking using adaptive correlation filters, *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 119, 2544-2550.

Chou, Y. C. and Nakajima, M. (2017). Particle filter planar target tracking with a monocular camera for mobile robots. *Intelligent Automation and Soft Computing*. 23(1), 1-9.

Danelljan, M., Häger, G., Khan, F. S. and Felsberg, M. (2014). Accurate Scale Estimation for Robust Visual Tracking, *In Proceedings of the British Machine Vision Conference*. 65, 1-11.

Ding, G., Chen, W., Zhao, S., Han, J., and Liu, Q. (2018). Real-time scalable visual tracking via Quadrangle Kernelized correlation filters, *IEEE Transactions on Intelligent Transportation Systems*. 19(1), 140-150.

Guo, J. M., Hsia, C. H., Wong, K., Wu, J. Y., Wu, Y. T., and Wang, N. J. (2016). Night time vehicle detection and tracking with adaptive mask training, *IEEE Transactions on Vehicular Technology*. 65(6), 4023-4032.

Henriques, J. F., Caseiro, R., Martins, P., and Batista, J. (2012). Exploiting the Circulant structure of Tracking-by-Detection with Kernels, *In Proceedings of the European Conference on Computer Vision*. 7575, 702-715.

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

Hsia, C. H., Liou, Y. J., and Chiang, J. S. (2016). Directional prediction CamShift algorithm based on adaptive search pattern for moving object tracking, *Journal of Real-Time Image Processing*. 12(1), 183-195.

Jo, K., Lee, M., Kim, J., and Sunwoo, M. (2017). Tracking and Behavior Reasoning of Moving Vehicles Based on Roadway Geometry Constraints, *IEEE Transactions on Intelligent Transportation Systems*. 18(2), 460-476.

Lee, Y. H., Ahn, H. C., Ahn, H. B., and Lee, S. Y. (2019). Visual object detection and tracking using analytical learning approach of validity level, *Intelligent Automation and Soft Computing*, 25(1), 205-215.

Li, Y. and Zhu, J. (2014). A scale adaptive kernel correlation filter tracker with feature integration, *In Proceedings of the European Conference on Computer Vision*. 8926, 254-265.

Li, Y., Zhu, J., and Hoi, S. C. H. (2015). Reliable patch trackers: robust visual tracking by exploiting reliable patches, *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 353-361.

Lin, Y., Huang, D., and Huang, W. (2018). Target tracking algorithm based on an adaptive feature and particle filter, *Information*. 9(140), 1-15.

Liu, Y., Yu, Z., Zeng, M., and Zhang, Y. (2016). An improved LLE algorithm based on iterative shrinkage for machinery fault diagnosis, *Measurement*. 77, 246-256.

Lu, K., Zhou, R., and Zhang, J. (2017). Approximate chemoff fusion of Gaussian mixtures for ballistic target tracking in the re-entry phase, *Aerospace Science and Technology*. 61, 21-28.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding, *Science*. 290(5500), 2323-2326.

Syu, J. L., Li, H. T., Chiang, J. S., Hsia, C. H., Wu, P. H., Hsieh, C. F., and Li, S. A. (2017). A computer vision assisted system for autonomous forklift vehicles in real factory environment, *Multimedia Tools and Applications*. 76(18), 18387-18407.

Wang, Q., Gao, J., Xing, J., Zhang, M., and Hu, W. (2017). DCFNet: Discriminant Correlation Filters Network for Visual Tracking, *In Proceedings of*

the IEEE Conference on Computer Vision and Pattern Recognition. arXiv:1704.04057.

Wu, Y., Lim, J., and Yang, M. H. (2015). Object tracking benchmark, *IEEE transactions on pattern analysis and machine intelligence*. 37(9),1834-1848.

Yan, Z., Xu, Z., and Dai, J. (2018). The big data analysis on the camera-based face image in surveillance cameras, *Intelligent Automation and Soft Computing*. 24(1), 123-131.

Yang, D. G., Cai, Y. Z., Mao, N., and Yang, F. C. (2016). Long-term object tracking based on kernelized correlation filters, *Optics and precision engineering*. 24(8), 2037-2049.

Yang, X., Zhang, H., Yang, L., Yang, C., and Liu, P. X. (2018). A joint multi-feature and scale-adaptive correlation filter tracker, *IEEE Access*. 6, 34246-34253.

Yin, H., Chao, P., Yi, C., and Qu, F. (2013). A robust object tracking algorithm based on surf and Kalman filter. *Intelligent Automation and Soft Computing*, 19(4), 567-579.

Zhang, K., Zhang, L., Liu, Q., Zhang, D., and Yang M. H. (2014). Fast visual tracking via dense spatio-temporal context learning, *In Proceedings of the European Conference on Computer Vision*. 8693, 127-141.

Zhang, L., Wang, Y., Sun, H., Yao, Z., and Wu, P. (2016). Adaptive scale object tracking with kernelized correlation filters, *Optics and precision engineering*. 24(2), 448-459.

Zin, T. T., and Yamada, K. (2016). An automatic target tracking system based on local and global features, *In Proceedings of the International Conference on Genetic and Evolutionary Computing*. 536, 255-262.

8 DISCLOSURE STATEMENT

NO potential conflict of interest was reported by the authors.

9 NOTES ON CONTRIBUTORS

D. Huang received the B.S degree from Xiamen University, Xiamen, Fujian, China, in 2008, received the Ph.D. Degree from University of Chinese Academy of Sciences, Beijing, China, in 2013. Currently, he is an assistant professor with the College of Engineering, Huaqiao University, Quanzhou, Fujian, China. His research interests are target tracking and image restoration.

P. Gu received the M.S. degree from Huaqiao University, Quanzhou, Fujian, China, in 2017. Currently, she is a teacher with the College of Mathematics and Computer Science, Quanzhou Normal University, Quanzhou, Fujian, China. Her research interests are image processing and target tracking.

H. M. Feng received the B.S. degree from Feng-Chia University, Taichung, Taiwan, in 1992, and received M.S. and Ph.D. degrees from Tamkang University, Taipei Hsien, Taiwan, in 1994 and 2000, respectively. Currently, he is the full professor with the Department of Computer Science and Information Engineering, National Quemoy University. His current research interests include fuzzy systems, neural networks, image processing and robot system.

Y. Lin is currently working toward the M.S. degree at the Huaqiao University, Quanzhou, Fujian, China. His research interest is target tracking.

L. Zheng received the B.S. and M.S. degree from Huaqiao University, Quanzhou, Fujian, China, in 1987 and 1990, respectively, and received the Ph.D. degrees from Tianjin University, Tianjin, China, in 2002. Currently, he is the professor with the College of Engineering, Huaqiao University, Quanzhou, Fujian, China. His current research interests include computer vision, and image processing.