Scale-adaptive Correlation Filter Tracking Algorithm Combined with Statistical Color Histogram Feature

WANG Xinying, WEI Haiping and YU Hongfei *

School of Computer and Communication Engineering, Liaoning Shihua University, Fushun 113001, Liaoning, China

*Corresponding author Email: xy.w_05@foxmail.com

Abstract. The algorithm based on kernel correlation filtering shows excellent performance. Aiming at the problem that traditional correlation filtering algorithms are sensitive to deformation and cannot solve the tracking failure caused by scale changes, this paper proposes a scale-adaptive correlation filtering algorithm combining statistical color feature. Firstly, predict and train the HOG features of the correlation filter and the color histogram features to obtain the respective response values, namely the template response and the histogram response. Then adopt the block method, and perform scale prediction based on the position of each response point. Finally, the final fusion response is calculated. The experimental results demonstrate that compared with traditional KCF, when the target scale has large scale change and deformation, the improved algorithm in this paper performs better, and can still track stably.

1. Introduction
Tracking technology has always been a hot topic and also broad application prospects and been applied in many aspects[1]. However, due to the complexity and variability of the actual application scene, there are often various complex interference factors, such as target scale changes, rapid lighting and attitude changes. The tracking algorithm must be able to overcome these interference factors and remain higher real-time. Kernel-correlation filtering-based algorithms, have recently shown excellent performance and robustness against motion blur and lighting changes. However, they are very sensitive to deformation.

In this paper, we propose a new scale-adaptive kernelized correlation filter that combines statistical color histograms. It simply combines a kernelized correlation filter with HOG features and color histogram, and is robust to color changes and distortions. Aiming at the problem that KCF cannot change as the target changing, we divide the target into four rectangular blocks with the center as the origin, and the distance between the maximum response positions of the four blocks in the initial frame is calculated. Then continue tracking the position of the maximum response value in four blocks in subsequent frames. By comparing the size of the position change to achieve scale adaptation. Experiments show that our algorithm performs better than the original one.

2. Related Work
Because calculation is in frequency domain[2], the speed is faster. Early correlation filtering algorithms used only gray features, such as MOSSE[3] and CSK[4]; Then Henriques et al.[5] replaced the original gray features with HOG features, and proposed KCF, which improves tracking accuracy while
ensuring tracking speed. However, for those algorithms, it is a great challenge[6] when the target deforms during tracking. The easiest way is to use a representation that is not sensitive to deformation. The color histogram has property because it is invariant for cyclic shift. However, a histogram alone is usually insufficient to distinguish objects from background[7]. Recently this class of algorithms, DAT [8](Distractor-Aware Tracker), has been proven to be competitive, which processes and explicitly suppresses areas with similar colors[9] by using adaptive threshold. The above research shows that the accurate characterization of the target and the adaptation of the scale are still issues that require further research.

In KCF, the derivation of the closed-form solution of the correlation filter uses the ridge regression method, and the idea of using a cyclic matrix effectively reduces the calculation amount, thereby quickly obtaining the kernelized correlation filter.

2.1. KCF

In this paper, KCF is used as the framework. In the calculation of the classifier, the introduction of a Gaussian kernel function makes the algorithm more general. KCF is mainly composed of model building, online matching, and template updating.

(a) Model establishment

KCF trains filters through ridge regression. The aim of the training is to find a linearized objective function \( f(z) = w^T z \) to obtain the minimum mean square error between sample \( x_i \) and the regression targets \( y_i \) as

\[
\min_w \sum_i(f(x_i) - y_i)^2 + \lambda \|w\|^2
\]  
(1)

where \( W \) can be used to represent the relationship between samples \( x \) and \( f(x) \). By transforming it into frequency domain calculations and introducing high-dimensional kernel functions to transform nonlinear into linear problems, the solution of kernel ridge regression can also be obtained from equation (2):

\[
\alpha = (K + \lambda I)^{-1} y,
\]  
(2)

where \( \alpha \) represents the solution of dual space.

(b) Matching online

When detecting the target of a new frame, a cyclic matrix is used to construct a candidate image block \( z \). And use the properties of the cyclic matrix to simplify the matrix operation, and finally get the response of each test sample

\[
f(z) = \hat{K} \hat{x} \odot \hat{a}.
\]  
(3)

(c) Template updating

After the target position is determined, relevant parameters need to be updated. The update strategy is:

\[
\hat{a}_t = (1 - \eta)\hat{a}_{t-1} + \eta \hat{\alpha}_t,
\]

\[
\hat{x}_t = (1 - \eta)\hat{x}_{t-1} + \eta \hat{x}_t,
\]  
(4)

where \( \hat{a}_{t-1} \) and \( \hat{a}_t \) are the coefficient matrices of the previous frame and the t-th frame, respectively.

3. Proposed Approach

Although the HOG feature of KCF is extremely robust to motion blur and lighting changes, the tracking effect is poor when the target is deformed or moves fast. In view of the shortcomings of KCF, an improved kernel-correlation filtering algorithm combined with statistical color features is proposed.

3.1. Integration model

Quoting the idea of [10], the HOG model and the statistical histogram model in this paper, are learned in a common framework, which we use to linearly combine two scores.

\[
\text{score} = (1 - \alpha) \text{score}_\text{hog} + \alpha \cdot \text{score}_\text{hist}.
\]  
(5)
Where $score_{hog}$ and $score_{hist}$ are the score of the correlation filter and the score of the histogram template respectively, which are linear functions of two different K-channel feature maps $\phi_x$ and $\psi_x$ respectively; $\alpha$ is a parameter selected on the validation set. The hog template score and histogram score are calculated separately, where the HOG template score is shown in formula (6):

$$score_{hog}(x; h) = \sum_{u \in T} h[u]T^{\phi_x[u]},$$  \hspace{1cm} (6)$$

where the weight template $h[u]$ is learned by the HOG feature. $T$ is a finite grid, which is an image block in the image set $x$. $u$ is the position of a pixel in the image block. According to [8], due to the spatial invariance of the histogram score, we can express $score_{hist}(x; \beta) = g(\psi_x; \beta)$ as the average value of the scalar value score map $\zeta(\beta, \psi_x)[u] = \beta^T \psi[u]$ as

$$score_{hist}(x; \beta) = \frac{1}{|H|} \sum_{u \in H} \zeta(\beta, \psi_x)[u].$$  \hspace{1cm} (7)$$

Where $\beta$ is a model parameter representing the probability that the point belongs to the foreground. $H$ is a finite grid.

To accelerate the calculation, it is calculated by the features shared by the overlapping windows. The histogram score can be calculated by the integral graph. The overall model can be expressed as $\theta = (h, \beta)$. According to [9], using a convolution filter to find the training loss by least-squares, and by taking the cyclically shifted feature map as a sample and using a high-dimensional representation, a large number of samples can be learned. Because the overall model score does not change for the cyclic shift, the overall model is divided into two independent ridge regression, so we can separately calculate $h_t$ and $\beta_t$ as

$$h_t = \arg \min_h \{L_{hog}(h; X_t) + \frac{1}{2} \lambda_{hog}||h||^2\},$$  \hspace{1cm} (8)$$

$$\beta_t = \arg \min_{\beta} \{L_{hist}(\beta; X_t) + \frac{1}{2} \lambda_{hist}||\beta||^2\}.$$  \hspace{1cm} (9)$$

$X_t$ is the training sample and its regression value; $\lambda$ is the weight, which is used to avoid overfitting. The parameter $h_t$ can be quickly solved by the formula of KCF, while $\beta_t$ can only be calculated by transposing the ordinary matrix, so it cannot be solved by cyclic shift. According to the recursive formula of the least square loss function, the template parameter $h$ of the convolution filter can be obtained, that is, the minimum value of $\ell_{hog}(x, p, h) + \lambda_{hog}||h||^2$.

$$\hat{h}[u] = \frac{1}{(d[u] + \lambda)\sigma[u]}$$  \hspace{1cm} (10)$$

Where $L_{hog}(x, p, h)$ is the loss function of each image; $s'[u]$ and $\hat{r}'[u]$ are a K-dimensional square matrix and a K-dimensional vector, respectively; $d[u]$ is the trace of the matrix $s'[u]$. The HOG template update formula is as follows:

$$\hat{d}_t = (1 - \eta_{hog})\hat{d}_{t-1} + \eta_{hog}d_t$$

$$\hat{r}_t = (1 - \eta_{hog})\hat{r}_{t-1} + \eta_{hog}r_t.$$  \hspace{1cm} (11)$$

Where $\eta_{hog}$ is the learning rate of the HOG template; the parameters $d_t$ and $r_t$ of the template are calculated from the Fourier transform of the HOG feature map to obtain new template.

Histogram score solution. (c, b)EW represents a set of rectangular windows $c$ and their corresponding regression labels $b$, which include positive samples (p, 1). According to formula (7), a linear regression is applied to each feature pixel on the foreground O and the background B to simplify the calculation, that is, the pixel regression values of the foreground O and the background B are set to 1 and 0, respectively. For a single image, by looking up the value of $k$ in the lookup table, let $\beta^T \psi[u] = \beta^T [u]$, the loss function is decomposed into the sum of the loss functions of each histogram bin in the histogram as formula (12):

$$\ell_{hist}(x, p, \beta) = \sum_{j=1}^{K} |O|^{-1}N^j(O) \cdot (\beta^j - 1)^2 + |B|^{-1}N^j(B) \cdot (\beta^j)^2$$  \hspace{1cm} (12)$$

Where $K$ is the number of the histogram bin which pixel $u$ belongs to; $N^j(A)$ is the sum of the elements in the j-th histogram bin. The histogram ridge regression problem can be obtained, that is, the solution of formula (11) is
\[
\beta_t^j = \frac{p_t^j(o)}{p_t^j(o) + p_t^j(b) + \lambda}
\]  

(13)

where p is the ratio of the sum of the elements in the j-th histogram bin in area A to the total number of pixels in A. Through the above formula, we can get the histogram classifier response map by integrating the pixel probability map. The histogram model is updated as follows:

\[
p_t(A) = (1 - \eta_{hist})p_{t-1}(A) + \eta_{hist}p_t'(A)
\]

\[
p_t(B) = (1 - \eta_{hist})p_{t-1}(B) + \eta_{hist}p_t'(B)
\]

(14)

Where \( p_t(A) \) is a vector of \( p_t'(A) \), and \( \eta_{hist} \) is the learning rate of the histogram template.

3.2. Scale calculation

Because it is meaningless to add scale calculation to the histogram template, and has a bad influence on the result, so only scale calculation is performed on the HOG template. Apply the HOG template to the position of the maximum response obtained in the current frame. During tracking, at the \( t - 1 \) frame, a rectangular image block \( b_{t-1} \) is selected with the target centre point \( p_{t-1} \) as the center, let \( p_{t-1} \) be the origin of coordinates \((0,0)\), and divide \( b_{t-1} \) into four equal-sized rectangular blocks \( b_{l,t-1} \) with \( p_{t-1} \) as the centre, and the coordinates of their centre points are \((w_{l,t-1}^1, h_{l,t-1}^1)\) and \((w_{l,t-1}^2, h_{l,t-1}^2)\) and \((w_{l,t-1}^3, h_{l,t-1}^3)\) and \((w_{l,t-1}^4, h_{l,t-1}^4)\), where \( l = 1,2,3,4 \). Apply HOG classifier to \( b_{l,t-1} \) to get four new classifiers, and then train them separately. At the \( t \)-th frame, the target's location is at the position the maximum response value. A coordinate system is established with \( p_{t-1} \) as the centre, and a rectangular image block \( b_t \) is selected and divided into four rectangular blocks. The maximum response value of each image block is obtained by using the filter formula, and the position of the maximum response is the centre position of the block. The peak sidelobe ratio PSR[13] = \((\text{peak} - \mu)\sigma^{-1}\) is introduced to measure the peak intensity of the response value. By finding the PSR value of each image block, the scale change rate \( sc \) can be obtained:

\[
s c = \left\{ \frac{|w_{l,t}^1|}{|w_{l,t-1}^1|} \right\}^{\frac{1}{2}} \left\{ \frac{|h_{l,t}^1|}{|h_{l,t-1}^1|} \right\}^{\frac{1}{2}}
\]

(15)

Where \((w_{l,t}^k, h_{l,t}^k)\) and \((w_{l,t-1}^k, h_{l,t-1}^k)\) are the highest PSR values in the four image blocks at frame \( t \) and frame \( t - 1 \), respectively. Then the \( t \)-th frame target can be expressed as:

\[
w_t = scw_{t-1}, h_t = sc h_{t-1}
\]

(16)

Then in the \( t \)-th frame, the scale of \( b_t \) centering on \( p_t \) is \( w_t \times h_t \).

4. Experimental results and analysis

All experiments in this article use Matlab R2014b to run on Intel Core i5-7500K @ 3.4GHz computer. Because the validity of the improved algorithm in complex scenarios needs to be verified, this paper adopts publicly challenging tracking test videos and personal actual scene videos. All parameters of comparison algorithms are default.

4.1. Qualitative analysis

We selected the representative CarScale and Woman video sequences in the OTB[14] dataset and personal video FishEye for display. Figure 1 is a partial screenshot of the experimental results of each group of video sequences.

(a) Scale Variation

The tracking target in CarScale is a moving car. The target moves and gradually approaches the camera during driving, and its target size also increases. As shown in Figure 1 (a), at 164 frames, the target scale has changed significantly from the initial frame. At this time, although KCF and DAT can still track the target, the size of the tracking frames cannot change with the target scale. At 183 frames, the tracking frame of the improved algorithm can still increase as the target scale becomes larger.

(b) Fast Motion

In Woman, the lens has a fast moving speed, as shown in Figure 1 (b). During fast movement, at
562 frames, the KCF was lost, DAT is not susceptible to rapid movement due to its characteristics, so it can track normally, but cannot change with the target scale. The improved algorithm can track the fast-moving targets stably and change as the target scale changes.

(c) Deformation

As shown in Figure 1 (c), in FishEye captured by the fisheye camera, the target is greatly deformed. At 242 frames, the target is stretched. KCF can track it, but it cannot adapt to the change in scale. DAT has a slight drift. The improved algorithm is stable tracking and adaptive scale changes. At 394 frames, the target bends and turns to produce deformation, and both KCF and DAT drift. The improved algorithm stably tracks the deformed target while adaptively tracking the window as the target scale changes.

Figure 1.

Figure 1. Qualitative comparison part of the tracking algorithm (yellow: OURS green: KCF blue: DAT)

4.2. Quantitative analysis

Use Distance Precision (DP)[15] to show tracking results on the dataset. We set the threshold to 20 pixels. Table 1 is the distance accuracy test results of the three algorithms. It is obviously that the distance accuracy of the algorithm in this paper is the highest. In CarScale, the distance accuracy is improved by about 10.7%. In Woman and FishEye, besides the scale change, the target has sideways and rapid movements, which leads to a large deviation in the KCF tracking.

| Video sequence | KCF   | DAT   | OURS  |
|----------------|-------|-------|-------|
| CarScale       | 0.816 | 0.807 | 0.903 |
| Woman          | 0.384 | 0.967 | 1.000 |
| FishEye        | 0.593 | 0.652 | 0.977 |
5. Conclusion
Based on KCF, this paper linearly combines the HOG template and the statistical color histogram template to enhance the robustness of the algorithm to fast motion and deformation. In addition, based on the introduction of block idea, an adaptive scale processing method is proposed to solve the problem that KCF is insensitive to the change of target scale to some extent. The experiments show that the optimization results of this algorithm are obvious, and it is also robust to the problems of occlusion, lighting, deformation, and fast motion encountered in daily tracking. At the same time, it also found that this algorithm has certain defects, that is, when the target drift or loss, there is not a re-detected mechanism. The next research direction is also to consider the robustness of tracking to achieve long-term tracking.

Acknowledgement
*This research was supported by the National Natural Science Foundation of China (61702247).
*This research was supported by the National Science Foundation of Liaoning Province (2019-ZD-0052).

References
[1] LIU Q Y, WANG Y R, ZHANG J L. Research progress of video tracking methods based on correlation filter[J]. ACTA AUTOMATICA SINICA, 2019,45(02):265-275.
[2] MENG Lu, YANG Xu. A Survey of Object Tracking Algorithms. ACTA AUTOMATICA SINICA, 2019, 45(7): 1244-1260. doi: 10.16383/j.aas.c180277
[3] Bolme D S, Beveridge J R, Draper B A, et al. Visual object tracking using adaptive correlation filters [C]// CVPR, 2010.
[4] Henriques J F, Rui C, Martins P, et al. Exploiting the Circulant Structure of Tracking-by-Detection with Kernels, In Proc. of the ECCV,2012:702-715.
[5] Henriques J F, Caseiro R, Martins P, et al. High-Speed Tracking with Kernelized Correlation Filters[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 37(3): 583-596.
[6] GUO W, XING Y Z, QU H C. Real-time object tracking based on high-confidence complementary learning [J]. Journal of Image and Graphics, 2019, 24(08): 1315-1326.
[7] Danelljan M, Khan F S, Felsberg M, et al. Adaptive Color Attributes for Real-Time Visual Tracking, In Proc. of the CVPR, 2014:1090-1097.
[8] Possegger H, Mauthner T, Bischof H. In Defense of Color-based Model-free Tracking[C]// IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2015.
[9] Xu Y L, Wang J B, Li H, Li Y, Miao Z, Zhang Y F. Patch-based scale calculation for real-time visual tracking. IEEE Signal Processing Letters, 2016, 23(1): 40–44
[10] Bertinetto L, Valmadre J, Golodetz S, et al. Staple: Complementary Learners for Real-Time Tracking [C]// CVPR, 2016.
[11] XIONG X X, WANG W W. Kernelized correlation filtering method based on fast discriminative scale estimation [J]. Journal of Computer Applications, 2019, 39(02): 546-550.
[12] Liu T, Wang G, Yang Q X. Real-time part-based visual tracking via adaptive correlation filters. In: Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition. Boston, USA: IEEE, 2015. 4902–4912
[13] Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(7): 1409–1422.
[14] Wu Y, Lim J, Yang M H. Online object tracking: a benchmark[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Portland, USA: IEEE, 2013, 9(4): 2411-2418. [DOI: 10.1109/CVPR.2013.312]
[15] Wu Y, Lim J, Yang MH. Object tracking benchmark. IEEE Translactionson Pattern Analysis and Machine Intelligence 2015, 37(9): 1834-1848. [doi: 10.1109/TPAMI.2014.2388226]