Scale Adaptive Block Target Tracking Based on Multi-layer Convolution Features and Kernel Correlation Filter

Ting Zhang¹, Dong Hu¹,²,³*, and Jing Zhang¹

¹ Jiangsu Province’s Key Lab of Image Procession and Image Communications, Nanjing, 210003, China
² Education Ministry’s Key Lab of Broadband Wireless Communication and Sensor Network Technology, Nanjing, 210003, China
³ Education Ministry’s Engineering Research Center of Ubiquitous Network and Health Service, Nanjing University of Posts and Telecommunications, Nanjing, 210003, China

*Email: 1218012403@njupt.edu.cn; hud@njupt.edu.cn; 1217012312@njupt.edu.cn

Abstract. Target tracking is currently a hot research topic in Computer Vision and has a wide range of use in many research fields. However, due to factors such as occlusion, fast motion, blur and scale variation, tracking method still needs to be deeply studied. In this paper, we propose a block target tracking method based on multi-convolutional layer features and Kernel correlation filter. Our method divides the tracking process into two parts: target position estimation and target scale estimation. First, we block the target frame based on the condition number. Second, we extract the features by the convolutional layer and apply it to the kernel correlation filter to get the center position of different block targets. With the reliability of different blocks measured by the Barker coefficient, the overall target position center is obtained. Then, the affine transformation is adopted to achieve the scale adaptation. The algorithm in this paper is evaluated by the public video sequences in OTB-2013. Numerous experimental results demonstrate that the proposed tracking method can achieve target scale adaptation and effectively improve the tracking accuracy.

1. Introduction

Target tracking is a basic computer vision task in Computer Vision and pattern recognition. Due to the complex scenarios in tracking, it is still a great challenge to achieve efficient and accurate target tracking. With the development of computer technology, researchers have proposed many excellent theories and tracking algorithms.

The target tracking algorithm has developed rapidly in recent years and has mainly gone through three periods. The first period is target tracking using generative methods, such as Mean Shift [1], Kalman filter [2], particle filter [3] and so on. The second period is applying the correlation filter to target tracking. The earliest squared error minimum output sum (MOSSE) proposed by Bolme et al. [4] in 2010 is the simplest object tracking method for correlation filter ideas. Two years later, Henriques introduced the Kernel Method based on MOSSE and proposed the CSK algorithm [5]. Henriques also proposed the KCF algorithm [6], which has an amazing tracking speed. Aiming at scale adaptation, the researchers proposed SAMF [7] method using multi-scale spatial sampling based on KCF. At the same time, Martin Danelljan et al. pioneered the DSST [8] method using a combination of translational filter...
and scale filter, which achieved scale adaptation. The third period is the rise of deep learning. Meanwhile, the correlation filter is also constantly developing and integrating into deep learning method. Many scholars have applied convolutional neural network (CNN) \cite{9} to target tracking. For example, Hong et al. used pre-trained CNN for discriminative saliency learning, and used online SVM \cite{10} to train to generate the target appearance model and separate the target from the background. Ma et al. \cite{11} used pre-trained VGGNet \cite{12} to extract the features of different convolutional layers and fused them into correlation filters, respectively, to obtain the final target position with weighted fusion.

In this paper, we divide the target into multiple blocks and select different convolutional layer features using VGGNet. Based on multi-convolution layer features and correlation filter, the target tracking method is discussed in the rest of the paper to improve the tracking accuracy. The second section describes the principle of kernel correlation filter applied on target tracking and the introduction of the convolutional neural network. In the third section, we introduced a target tracking method that uses affine transformation to achieve scale adaptation in details. The forth section provides the experimental results and comparisons with other advanced algorithms. Finally, our conclusions are given in the fifth section.

2. Related Work

2.1. Tracking by Kernel Correlation Filter

Target tracking based on the kernel correlation filter is to use a certain feature of the image to train the filter, and then model the target as a whole. First, the target selected by the specified target center tracking frame, and the subsequent frame search box associated filter performs a convolution operation. The maximum value of the output convolution result is the new position of the target, and the filter is continuously updated according to the tracking result.

To train the tracker quickly, it is necessary to simplify the calculation by fast Fourier transform FFT. Image input: $F = \mathcal{F}(f)$, Filter: $H = \mathcal{F}(h)$. In the convolution theorem, the correlation is elementary multiplication in the Fourier domain, and the correlation can be expressed as:

$$G = F \odot H^*$$

(1)

where $\odot$ indicates that the corresponding elements are multiplied and $^*$ indicates complex conjugation. Therefore, the correlation filter can be expressed as:

$$H_i = \frac{G_i}{F_i^*}$$

(2)

In order to obtain the ideal output corresponding to the input image and find the optimal filter by minimizing the error function, designed as the square difference between the actual output and the ideal output obtained by the filter. The form of this minimization problem is:

$$\min \sum_{i} |F_i \odot H^* - G_i|^2$$

(3)

By solving the correlation filter, we can get equation (4):

$$H^* = \frac{\sum_i G_i \odot F_i^*}{\sum_i F_i \odot F_i^*}$$

(4)

Through the correlation filter and the image input convolution, we can get the correlation response map. The position of the largest value of the response map is the target position.

The kernel correlation filter has ridge regression of cyclically shifted samples and can use fast logarithmic Fourier transform instead of matrix algebra to perform fast operations. The purpose of the algorithm training is to find an objective function minimizing the error function (5):

$$\min \sum_i (f(x_i) - y_i)^2 + \lambda \|w\|^2$$

(5)
where $X_i$ represents the image sample, $Y_i$ represents the regression result of the sample, and $\lambda \|w\|^2$ is a type of regularization term.

The kernel-based regularized least squares method gives the closed-form result of ridge regression

$$\hat{\alpha} = \frac{\hat{y}}{k^{sv}} + \lambda,$$

where $k^{sv}$ is the kernel correlation of vector $X$ and vector $X'$. Here, the Gaussian kernel correlation is $k^{sv} = \exp(-\frac{1}{2\sigma^2}||x||^2 + ||x'||^2 - 2F^{-1}(\tilde{z} \odot \tilde{z}'))$. Therefore, the target probability distribution of the image in all candidate regions can be calculated: $f(z) = F^{-1}(\tilde{k}^sv \odot \tilde{a})$. The tracking area corresponding to the maximum probability value is the tracking target area.

2.2. Convolutional Neural Network

Convolutional neural network is a typical deep learning architecture [13]. It is end-to-end learning, which can be regarded as a type of feedforward neural network with multiple layers of results. It can extract robust features with invariance of image translation, rotation, and deformation. A complete convolutional neural network includes many convolutional layers, pooling layers and fully connected layers. Related features are automatically extracted from the original image data layer by layer, and the extracted features are more effective and accurate. Since AlexNet[14], complex CNNs supported by GPU have repeatedly become the winning algorithms for large-scale visual recognition competitions [15], including VGGNet and ResNet [16]. Therefore, the application of convolutional neural networks in target tracking has also increased and become more flexible recently.

In this paper, we use VGGNet-19 convolutional feature map to achieve the model of the target appearance. Because VGGNet-19 is used to extract features[17], each set of convolutional layers will output a set of feature data in the entire process. It is easy to select the appropriate multi-convolutional layer features according to the different conditions of the picture described, which can efficiently increase the accuracy of the object tracking algorithm.

3. Scale Adaptive Block Target Tracking

3.1. Target Position Estimation

Most of the traditional target tracking algorithms use the overall appearance model to determine the target position, but once the target is occluded, it is difficult to directly respond to the sudden change of the scene or the target, which will result in poor target tracking. If the target area is divided into several blocks, the external view of the target is a different combination of target blocks. The different blocks are tracked and positioned, and then the tracking results are combined, so that once a partial block occurs, the remaining blocks still can continue accurate tracking.

We propose the target block based on the condition number. The superiority of the block is measured by calculating the Hessen matrix $H$ and the condition number:

$$K = \frac{\sigma_{max}(H)}{\sigma_{min}(H)} \tag{6}$$

where $\sigma_{min}(H)$ and $\sigma_{max}(H)$ are the minimum singular value and maximum singular value of the Heisen matrix $H$. It can be seen from the formula (6) that the smaller the value of $K$, the more stable the matrix is. Then the sorting of $K$ will be based on the condition number from small to large.

Then, we select the size and number of blocks through experimental verification of multiple video sequences. Experiments show that when the size of the block object is between one-quarter and one-sixth of the target, the result will not change much, but when the size of the block is larger or smaller, the result will be worse. When the number of blocks is greater than five, the number of blocks will not have much influence on the result. Therefore, the optimal number of blocks is four. The method of block operation is as follows. First, take the pixel corresponding to the minimum condition value as the first
block target center, and the size of the block is a quarter of the size of the initial target frame. Then select the remaining condition numbers in turn. The smallest point, if the point is already in the first block, it is discarded, otherwise the point is taken as the center of the second target block, the size is still a quarter of the initial target frame, and so on, until the completion of the four blocks.

The first and fourth convolutional layers of VGGNet-19 as multi-channel features chosen in our method. We use $X$ expressing the extracted feature vector on the lth layer, which size is $M \times N \times D$.

Learn the correlation filter by solving the following minimization problem:

$$W^* = \arg \min \sum_{m,n} \| W \cdot X_{m,n} - y(m,n) \|^2 + \lambda \| W \|_2^2$$

(7)

where linear product is defined as $\langle \cdot, \cdot \rangle = \sum_{i,j} W_{ij} X_{i,j}$. The filter learned in the frequency domain on the $d$-th ($d \in \{1, ..., D\}$) channel is:

$$W^d = \frac{Y \odot \hat{X}^d}{\sum_{n=1}^{D} \hat{X}^d \odot Z^d}$$

(8)

Given a block of an image, the correlation response map of the layer is:

$$f(z) = F^{-1} \left( \sum_{d=1}^{D} W^d \odot Z^d \right)$$

(9)

where $F^{-1}$ represents inverse Fourier transform. Let $(\hat{m}, \hat{n}) = \arg \max_{m,n} f_i(m,n)$ denote the position of the maximum value on the lth layer, then the best position of the target in the l-1th layer is:

$$\arg \max_{m,n} f_{l-1}(m,n) + \gamma f_i(m,n)$$

$$s.t. |m - \hat{m}| + |n - \hat{n}| \leq r$$

(10)

The regularization term $\gamma$ is used to weight the last layer of the response graph, and then backpropagation. By maximizing the formula (10), the target center position of each block is estimated.

At the same time, the reliability of different blocks is measured, and the similarity between the tracking result and the original tracking frame is calculated as follows:

$$C = g(p, p^\wedge)$$

(11)

where $g(\cdot)$ represents the Barker coefficient, $p$ represents the distribution probability of the correlation filter in the current frame target, and $p^\wedge$ represents the distribution probability of the ideal response. With the weights to each block and the center position of each block, we can finally obtain the center position of the overall target.

3.2. Target Scale Adaptation

The center position of the different blocks is estimated in Section 3.1, and the association of the center points of these blocks in different frames reflect the change ratio of the target frame. In this part, we use the affine matrix to estimate the scale transformation of the target frame.

Suppose that the target center position of different blocks in the first frame is $P^1_1 = (x^1_1, y^1_1)$, $P^1_2 = (x^1_2, y^1_2)$, ..., $P^1_m = (x^1_m, y^1_m)$ and the target center position point of each block in the t frame is $P^t_1 = (x^t_1, y^t_1)$, $P^t_2 = (x^t_2, y^t_2)$, ..., $P^t_m = (x^t_m, y^t_m)$. The affine matrix $A \in \mathbb{R}^{3 \times 3}$ is used to describe the transformation of the center point corresponding to the first frame and the t-th frame. Here, we select the first frame as the initial frame instead of the t-1 frame as the initial frame, because there are only small changes between adjacent frames; and relative to the first frame, other frames as the initial frame
are more prone to error accumulation. Taking the position center of the first frame and the t-th frame of block one as an example, the affine matrix is expressed as follows:

$$\begin{pmatrix} x^t_i \\ y^t_i \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & d_x \\ a_{21} & a_{22} & d_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x^1_i \\ y^1_i \\ 1 \end{pmatrix}$$

(12)

where $P^t_i = [x^t_i, y^t_i, 1]^T$ and $P^1_i = [x^1_i, y^1_i, 1]^T$. Considering that the positions of the target centers of different blocks have been obtained, only four parameters $a_{11}, a_{12}, a_{21}, a_{22}$ of the affine matrix $A$ need to be estimated. Then we express it as a two-dimensional affine matrix $G$:

$$\begin{pmatrix} x^t_i \\ y^t_i \\ \cdots \\ x^m_i \\ y^m_i \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x^1_i \\ y^1_i \\ \cdots \\ x^m_i \\ y^m_i \end{pmatrix}$$

(13)

The center position coordinates of the first frame and the t-th frame of each block are expressed as matrix $E$ and $T$ respectively. The matrix estimation problem matched by these four parameters is transformed into a linear equation solving system:

$$\min_G \| T^T - E^T G \|^2$$

(14)

The least-square solution of this equation is:

$$G = TE^T (EE^T)^{-1}$$

(15)

Then we can obtain the affine matrix $G$, which reflects changes in image rotation, scaling, etc.

The currently target center position points we have known are $P^t_i = (x^t_i, y^t_i)$, $P^1_i = (x^1_i, y^1_i)$, $P^m_i = (x^m_i, y^m_i)$. The combination samples based on these $m$ data points is from $C^1_m$ to $C^m_m$, each combination $C^k_m$ means to select $k$ points from all $m$ points for the estimation of the affine matrix. Different affine matrices can be estimated with different choices. Then $C^1_m + C^2_m + \ldots + C^m_m = 2^m - 1$ affine matrixes will be estimated.

Since the affine matrix is obtained from the center position of each block target in the first frame and the t-th frame, we multiply the initial frame image block by the affine matrix to obtain a series of target frames of different sizes, denoted as $B_1 = (w_1, h_1)$, $B_2 = (w_2, h_2)$, $\ldots$, $B_{2^m-1} = (w_{2^m-1}, h_{2^m-1})$. At the same time, the t-th frame image block is obtained, with the center of the estimated overall target position as the center, and $B_1, B_2, \ldots, B_{2^m-1}$ as the size of the target frame, a series of target candidate frames are obtained, and a total of $2^m - 1$ candidate frame sets can be obtained. The diversity of these candidate frames is diverse. Optimization is more conducive to selecting the optimal candidate frame. In our method, we take the average value of the tracking results from the previous five frames as the baseline sample. The optimal baseline sample criterion for obtaining the optimal candidate frame is to use the candidate frame closest to the baseline sample as the optimal candidate frame. Then, we take the size of the optimal candidate frame as the final size of the target.

### 3.3. Model Update

After the target frame size estimation is completed, the target state tracking of the current frame is realized. When a new image frame is an input, the update of the relevant filter model needs to be completed at the time, because in the continuous update of the video sequence, the target will be...
interfered by factors such as occlusion and scale changes. The update of the model is related to the robustness of target tracking. Due to a large amount of calculation to update the filter by minimizing the output error of the tracking result, we follow the new filter in a simpler way, the filter coefficients follow the new learning rate at a fixed learning rate \( r \), and the update strategy is:

\[
\begin{align*}
\alpha'_i &= (1 - \beta)\alpha_{i-1} + \beta\alpha'_i \\
x'_i &= (1 - \beta)x_{i-1} + \beta x'_i
\end{align*}
\]  

(16)

where \( \beta \in (0,1) \) is the learning rate, \( \alpha \) is the classifier coefficient and \( x \) is the input image block.

4. Experiments
In order to verify the effectiveness of the algorithm in this paper, the experiments are all implemented on a computer with MATLAB 2018b, Inter(R) Core(TM) i7-6700 CPU, NVIDIA GeForce GTX RTX 2080Ti GPU, 8G memory and win10 system. To fully verify the tracking robustness and scale estimation accuracy of the tracking algorithm in this paper, 37 groups of videos with scale changes in the OTB2013 test data set are tested and analyzed, and the algorithm in this paper is compared and analyzed with 4 current mainstream tracking algorithms. The selected comparison algorithms are: SAMF, STAPLE, DSST and HCFT.

4.1. Qualitative Experiment Analysis
In this section, four target video sequences with scale transformation are chosen from OTB2013 for qualitative analysis, and they are BlurBody, ClifBar, Human6 and Surfer.

![Comparison of tracking Results](image)

**Figure 1.** Comparison of tracking Results(The sequences from top to bottom is: BlurBody, ClifBar, Human6, Surfer).
The specific experimental results are shown in Figure 1, where red is the algorithm tracking box in this paper, green is the HCFT algorithm tracking box, blue is the SAMF algorithm tracking box, black is the STAPLE algorithm tracking box, and purple is the DSST algorithm tracking box. It can be seen that the scale changes of BlurBody, ClifBar, Human6, and Surfer sequence targets all occur, and the tracking results will be compared and analyzed in details. In BlurBody sequences, we can see that the selected four frames of images have obvious scale changes. At frame 215, SAMF and DSST drift. At frame 251, DSST obviously fails to track, and the algorithm scale estimation in this paper is more accurate. ClifBar, Human6, and Surfer sequences all have similar phenomena.

4.2. Quantitative Experiment Analysis

In this section, we compare our tracking algorithm with other four popular algorithms in Visual Tracker Benchmark, and they are HCFT, SAMF, STAPLE, DSST. The results are shown in Figure 2, Figure 3.

Figure 2 shows the average accuracy and success rate plots compared with HCFT, SAMF, STAPLE, DSST on 50 video sequences in OTB2013. Among them, Figure 2(a) shows the accuracy plots of scale transformation, the average accuracy of the algorithm in this paper reaches 80.2%, HCFT is 73.9%, SAMF is 72.2%, STAPLE is 66.1%, and DSST is 58.1%. Compared with the algorithm, improved by 7.8%, 9.9%, 17.5%, and 27.5% respectively. Figure 2(b) success rate curve, the success rate of this algorithm reaches 69.3%, HCFT is 61.6%, SAMF is 61.1%, and STAPLE is 57.0%, DSST is 49.9%, compared with these four algorithms, respectively improved by 11.1%, 11.8%, 17.7%, 27.9%.

Figure 3 is the average accuracy and success rate plots compared with HCFT, SAMF, STAPLE, DSST in the test of 37 scale transformation sequences in OTB2013. Among them, Figure 3(a) regards the accuracy plots of scale transformation. It can be seen from the figure that the average accuracy of our algorithm is 80.2%, HCFT is 73.9%, SAMF is 72.2%, STAPLE is 66.1%, DSST is 58.1%. Compared with these four algorithm, the algorithm we proposed has improved by 7.8%, 9.9%, 17.5%, and 27.5% respectively. The success rate plots of Figure 3(b) shows that the success rate of the algorithm in this paper is 69.3%, HCFT is 61.6%, SAMF is 61.1%, and STAPLE is 57.0%, DSST is 49.9%. Compared with these four algorithms, our method achieves an increase of 11.1%, 11.8%, 17.7%, 27.9%.

Based on the experimental data, it can be found that compared with HCFT experiment results, the target positioning with the addition of the block method is more accurate, and the scale frame size based on the affine transformation is more successful than the scale frame size based on the edge frame algorithm. The algorithm in this paper is more in-depth than the previous research foundation, and the tracking results are more accurate and effective. In general, it demonstrates that the method we proposed in this paper can effectively deal with the scale change problem in object tracking and has higher accuracy.
5. Conclusions
We presented a novel tracking algorithm in this paper. The target is divided into four blocks, and different convolutional layer features of VGGNet-19 are selected. We calculate the center position of the different block target by combining with the Kernel correlation filter. Then the weight to the different blocks is assigned by the Barker coefficient, and the center of the overall target position is obtained. Meanwhile, we use the affine transformation matrix to get the set of candidate frames, and the closest to the baseline sample is optimal. The algorithm in this paper not only obtains the tracking results with high accuracy and good robustness, but also solves the problem of target scale change in the tracking, and effectively improves the success rate of target tracking.

References
[1] Vojir T, Noskova J, Matas J 2014 Robust scale-adaptive mean-shift for tracking Pattern Recognition Letters 2014 239-242
[2] G.A.Einicke and L.B.White. Robust Extended Kalman Filtering IEEE Transactions on Signal Processing 47(9) 2596-9
[3] Kwon J and Lee K M 2008 Tracking of abrupt motion using Wang-Landau Monte Carlo estimation Computer Vision European Conference on Computer Vision (France: Springer Berlin Heidelberg) pp 387-400
[4] Bolme D S, Beveridge J R, Draper B 2010 Visual object tracking using adaptive correlation filters IEEE Conference on Computer Vision and Pattern Recognition
[5] Henriques J F, Caseiro R, Martins P 2012 Exploiting the circulant structure of tracking-by-detection with kernels European Conference on Computer Vision
[6] Wen S P, Huang T W, Yu X H, Chen M Z Q and Zeng Z G 2016 Aperiodic sampled-data sliding-mode control of fuzzy systems with communication delays via the event-triggered method IEEE Transactions on Fuzzy Systems 24(5) 1048-57
[7] Li Y, Zhu J K. A 2014 Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration European Conference on Computer Vision
[8] Danelljan M, Häger G, Khan F 2014 Accurate Scale Estimation for Robust Visual Tracking British Machine Vision Conference
[9] Wen S P, Chen M Z Q, Yu X H, Zeng Z G and Huang T W 2017 Fuzzy control for uncertain vehicle active suspension systems via dynamic sliding-mode approach IEEE Transactions on Systems, Man and Cybernetics: Systems 47(1) 24-32
[10] Chen P H, Lin C J and Schölkopf B 2005 A Tutorial on ν-support vector machines *Stochastic Models in Business and Industry* 21 111-36
[11] Ma C, Huang J B, Yang X 2015 Hierarchical Convolutional Features for Visual Tracking *IEEE International Conference on Computer Vision* (ICCV 2015) pp 111-121
[12] Simonyan K and Zisserman A 2014 Very Deep Convolutional Networks for Large-Scale Image Recognition *Computer Science* pp 569-577
[13] Wen S P, Chen M Z Q, Zeng Z G, Huang T W and Li C J 2017 Adaptive Neural-Fuzzy Sliding-Mode Fault-Tolerant Control for Uncertain Nonlinear Systems *IEEE Transactions on Systems, Man and Cybernetics: Systems* 47(8) 2268-78
[14] Schmidhuber and Jürgen 2015 Deep learning in neural networks *Neural Networks* 61 85-117
[15] Wen S P, Zeng Z G, Chen M Z Q and Huang T W 2017 Synchronization of switched neural networks with communication delays via the event-triggered method *IEEE Transactions on Neural Networks and Learning Systems* 28(10) 2334-43
[16] He K, Zhang X, Ren S 2015 Deep Residual Learning for Image Recognition *IEEE Transactions on Systems* 1 267-75
[17] Wen S P, Yu X H, Zeng Z G and Wang J J 2016 Event-triggering load frequency control for multi-area power systems with communication delay *IEEE Transactions on Industrial Electronics* 63(2) 1308-17