Scale Adaptive Kernel Correlation Filter Tracking Algorithm Combined with Learning Rate Adjustment

Di Wu[1,a] , Li Peng[2]

[1]School of Internet of Things, Jiangnan University, Wuxi, 214122, China
[2]Jiangsu Key Laboratory of IOT Application Technology, Taihu University of Wuxi， Wuxi, 214122, China

Abstract. Aiming at the problem that the traditional correlation filter tracking algorithm is prone to tracking failure under the target's scale change and occlusion environment, we propose a scale-adaptive Kernel Correlation Filter (KCF) target tracking algorithm combined with the learning rate adjustment. Firstly, we use the KCF to obtain the initial position of the target, and then adopt a low-complexity scale estimation scheme to get the target's scale, which improves the ability of the proposed algorithm to adapt to the change of the target's scale, and the tracking speed is also ensured. Finally, we use the average difference between two adjacent images to analyze the change of the image, and adjust the learning rate of the target model in segments according to the average difference to solve the tracking failure problem when the target is severely obstructed. Compared the proposed algorithm with other five classic target tracking algorithms, the experimental results show that the proposed algorithm is well adapted to the complex environment such as target's scale change, severe occlusion and background interference. At the same time, it has a real-time tracking speed of 231 frame/s.

1 Introduction

Target tracking technology is an important part of computer vision. It has been widely used in human-computer interaction [1-2], intelligent transportation [3], unmanned driving, etc. But it also faces many difficulties [2], including target tracking failure caused by changes in the target apparent model (such as target scale changes, rotation, deformation, etc.) and changes in surrounding environment (such as occlusion, illumination changes, slow background movement, etc.). Aiming at these problems, many excellent target tracking algorithms [4-12] have been proposed. Among them, the method based on correlation filtering [8-12] has attracted much attention due to its good tracking effect and computational efficiency.

Henriques et al.[8] proposes a Circulant Structure with Kernels (CSK) correlation filter tracking algorithm, the algorithm intensively samples the training samples through the cyclic shift strategy, which can extend the sample data set without affecting the running speed of the tracking algorithm, and achieves good results. Danelljan et al. [9] design a CSK-based tracking algorithm that incorporates color attribute features (Color Name, CN), and adaptively selects more obvious color attribute features to adapt to scene changes, which improve the accuracy of target tracking. Henriques et al. [10] design a Kernelized Correlation Filter (KCF) tracking algorithm, which replaces the original grayscale features with the Histogram of Oriented Gradient (HOG) feature on the basis of CSK. The feature channel has been extended to improve tracking performance. However, the above algorithms [8-10] use fixed-scale training samples, and when the scale of the target changes, it is easy to generate tracking drift. In order to solve the problems caused by the change of target scale, Danelljan et al. [11] proposes a Discriminative Scale Space Tracker (DSST), and Li et al. [12] design a Scale-Adaptive and Multi Feature Integration Tracker (SAMF), but they all use the method of estimating the scale of each frame of image. The algorithm takes a long time and the tracking speed is limited. At the same time, since most of the existing correlation filter tracking algorithms [8-12] adopt a fixed learning rate, the error accumulation will be caused after the target occlusion, which causes a large tracking deviation or even tracking failure.

In view of the above analysis, in order to enhance the robustness of the correlation filter tracking algorithm to target scale variation and occlusion, and to ensure the tracking speed of the algorithm, we propose a scale adaptive correlation filter tracking algorithm combined with the learning rate adjustment based on the KCF model. In the process of scale estimation, since

* Corresponding author: [a]Di Wu:1934169204@qq.com

the target scale between two adjacent frames does not change much, it takes more time to perform detection per frame. Therefore, a scale response detection is performed every M frames to reduce the consumption of time, thereby improving tracking speed. In addition, the average difference between two adjacent frames is used to analyze the change of the image, and the learning rate of the target apparent model is adjusted according to the average difference's size to solve the tracking failure problem when the target is severely occluded.

2 Kernel correlation filtering target tracking algorithm

In the KCF tracking algorithm, the rectangular region image block $\boldsymbol{x}$ of size $M \times N$ is selected to train the linear classifier $f(\boldsymbol{x}) = \langle \boldsymbol{\omega}, \varphi(\boldsymbol{x}) \rangle$ centering on the target center point detected by the current frame. Different from the traditional moving window [4-7] method of acquiring samples, the KCF tracking algorithm intensively samples $\boldsymbol{x}$ by cyclic shifting, and uses sampled image blocks $\boldsymbol{x}_i (i \in \{0,...,M-1\} \times \{0,...,N-1\})$ as training samples of the classifier. The feature is extracted from the training samples to obtain the corresponding tag data y_i, and a Gaussian function is used to describe y_i. First, establish a minimum objective function model

$$\min_{\boldsymbol{\omega}} \sum_i (f(\boldsymbol{x}_i) - y_i)^2 + \lambda \|\boldsymbol{\omega}\|^2 \tag{1}$$

By introducing a kernel function, the model in equation (1) becomes

$$\min_{\boldsymbol{\omega}} \sum_{i=1} \left[\langle \boldsymbol{\omega}, \varphi(\boldsymbol{x}) \rangle - y_i \right]^2 + \lambda \|\boldsymbol{\omega}\|^2 \tag{2}$$

Where $\varphi(\boldsymbol{x})$ represents the mapping of the original input space to the Hilbert feature space. At this time, the target solution can be represented as $\boldsymbol{\omega} = \sum_i \boldsymbol{\alpha}_i \varphi(\boldsymbol{x}_i)$ and can be solved using the kernel function $\kappa(\boldsymbol{x}, \boldsymbol{x}') = \langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{x}') \rangle$ [13].

Using the properties of circulant matrix and Discrete Fourier Transform (DFT), the coefficient $\boldsymbol{\alpha}$ of the classifier weight $\boldsymbol{\omega}$ is obtained [13]

$$\boldsymbol{\alpha} = \mathrm{F}^{-1}\left[\frac{\mathrm{F}(\boldsymbol{y})}{\mathrm{F}(\boldsymbol{k}^{\boldsymbol{xx}}) + \lambda} \right] \tag{3}$$

Where F represents a DFT and $\boldsymbol{k}^{\boldsymbol{xx}} = \kappa(\boldsymbol{x}, \boldsymbol{x})$ represents a kernel function. KCF uses Gaussian kernel function for calculation. For more details about the Gaussian kernel function please see the literature [10].

In a new frame, the position of the target is detected by acquiring the candidate image blocks $\boldsymbol{z}$, and the output response of the classifier is [13]

$$\hat{\boldsymbol{y}} = \mathrm{F}^{-1}\left[\mathrm{F}(\boldsymbol{k}^{\hat{\boldsymbol{x}}\boldsymbol{z}}) \odot \mathrm{F}(\hat{\boldsymbol{\alpha}}) \right] \tag{4}$$

Where $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{\alpha}}$ represent the target apparent model and classifier parameters obtained by learning, and the update method is

$$\hat{\boldsymbol{x}}_f = (1-\eta)\hat{\boldsymbol{x}}_{f-1} + \eta \hat{\boldsymbol{x}} \tag{5}$$

$$\hat{\boldsymbol{\alpha}}_f = (1-\eta)\hat{\boldsymbol{\alpha}}_{f-1} + \eta \hat{\boldsymbol{\alpha}} \tag{6}$$

The position where the output response $\hat{\boldsymbol{y}}$ takes the maximum value is the position of the target in a new frame. The main workflow of KCF is shown in Figure 1:

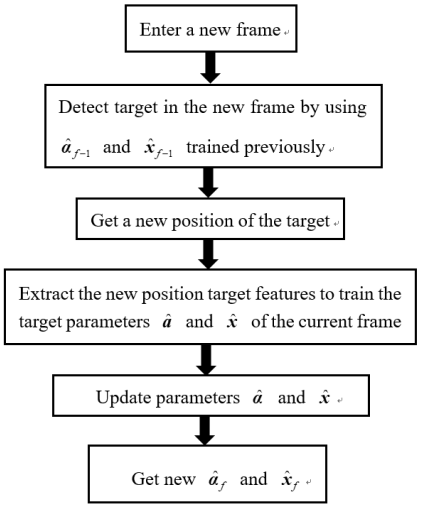


Figure1. Correlation filter workflow

3 Scale adaptive kernel correlation filter tracking algorithm combined with learning rate adjustment

3.1 Low complexity scale estimation method

When the target at the maximum value of the classifier output response is obtained, the scale detection of the target can further reduce the influence of noise. However, scale detection is a very time-consuming task, which will reduce the tracking speed multiple. Therefore, we propose a low-complexity scale estimation method for the time-consuming problem of scale detection. Specific steps are as follows:

Step1. Set a scale update interval T .Since the scale change of the target between two adjacent frames is very small, it is mostly ineffective to perform scale detection for each frame. In this paper, we change the scale comparison from two adjacent frames to two frames with interval T .When there is a relatively obvious change in the target scale, we then scale and update it, which increases the tracking accuracy and reduces the number of scale updates, and it can effectively reduce the loss of speed by scale update.

Step 2. Use the average dichotomy to detect the largest response scale. Set the scale pool N contains 7 scales, the number of scales is too many can easily affect the tracking speed, and too little will reduce the

tracking accuracy:

$$N=\{0.85,0.90,0.95,1,1.05,1.10,1.15\}$$

The elements in N are sequentially multiplied by the width and height of the previous tracking frame to obtain a new scale detection range. Most of the existing algorithms are to find the maximum value in all responses as a new scale after obtaining the classifier response for each new detection range. This calculation method does not first effectively evaluate the increase or decrease of the target size, and then update the scale in a targeted manner, which increases the calculation amount and reduces the efficiency. In this paper, we use the average dichotomy to find the maximum response. First, compare down, let $N(i)$ be the maximum response value of the search range corresponding to the $i-th$ element in the scale pool, first calculate the most intermediate $N(4)$, $N(3)$ and $N(5)$, if $N(4)$ maximum, the scale is unchanged. If $N(3)$ is the largest, $N(2)$ and $N(3)$ are compared downwards. If $N(5)$ is the largest, $N(5)$ and $N(6)$ are compared upwards. Until the maximum value of the response down or up is found, we update the scale and set it as the initial scale of the next frame. Such a scale detection strategy can effectively reduce the number of calculation scale responses, save computation time, reduce speed penalty while ensuring solution to scale problems, and make the algorithm meet real-time requirements.

3.2 Online learning rate adjustment algorithm

The classifier and the target apparent model of the correlation filter tracking algorithm are updated by equations (5) and (6), wherein the online learning rate η usually takes a fixed empirical value and cannot reflect the change of the scene in the video in real time.

The online learning rate η indicates the learning ability for the change of the target appearance. The smaller the value of η, the slower the learning rate is, for scenes with small changes in the target apparent model caused by changes in the surrounding environment (such as lighting changes, occlusion, rapid background movement, etc.), the tracking effect is better; The larger the value of η, the faster the learning rate is, for scenes with large changes in the target apparent model caused by the change of the target itself (such as non-rigid deformation of the target, scale change, rotation, etc.), the tracking effect is better.

In this paper, we propose a method of segmentally adjusting the online learning rate η by using the average difference between two adjacent frames. The specific steps are as follows:

Step1. Calculation of the average difference between two adjacent frames. For the image $\boldsymbol{W}$ of $M\times N$, the pixel value is represented by $\boldsymbol{W}_{ij}$, and the average difference between the $f-th$ frame and the previous frame is

$$e=\frac{\sum_{i,j}^{M,N}\left|(\boldsymbol{W}_{ij}^{f}-\boldsymbol{W}_{ij}^{f-1})\right|}{MN} \tag{7}$$

Step 2. Adjust the online learning rate. When $e<e_{\min}$, it means that the average difference between two adjacent frames is less than the set lower threshold. At this time, the target apparent model changes little, and the online learning rate can be set relatively low. When $e_{\min}\le e\le e_{\max}$, it means that the average difference between two adjacent frames is within the allowable threshold range. At this time, the target apparent model changes moderately, and the online learning rate can take the commonly used empirical value. When $e>e_{\max}$, it means that the average difference between two adjacent frames is greater than the set upper threshold. At this time, the target apparent model changes greatly, and the online learning rate should be set to a larger value. Through experimental tests, the values of $e_{\min}$ and $e_{\max}$ in this paper are $e_{\min}=2.5$ and $e_{\max}=8$ respectively. The following segmentation strategy is used to adjust the online learning rate:

$$\eta=\begin{cases}0.0025, e<2.5\\ 0.0125, 2.5\le e\le 8\\ 0.1, e>8\end{cases} \tag{8}$$

4 Experimental results and analy-sis

In order to verify the effectiveness of our algorithm, we select eight representative video sequences (Car4, Basketball, Singer1, Girl, Box, Jogging, Lemming, Subway) to test. These videos cover interference factors such as scale changes, occlusion, deformation, background interference, fast motion, illumination changes, and rotation [14].At the same time, we compare our algorithm with other five classical tracking algorithms such as CSK [8], CN [9], KCF [10], SAMF [11] and DSST [12]

4.1 Experimental environment and evaluation Indicators

The experiments in this paper are completed on Matlab R2013b, Window 10 system, Intel Core i7-4790 CPU, 4GHz, 4GB memory configuration. In order to facilitate quantitative analysis, three performance evaluation indicators are used in this paper: Center Location Error

(CLE), Distance Precision (DP) and Overlap Precision (OP). Where CLE represents the average Euclidean distance between the detected target center position and the target true center position [15]; DP represents the ratio of the number of frames whose CLE is less than a certain threshold (20 pixels in the experiment) to the total number of frames of the video; OP indicates the ratio of the number of frames in which the overlap of the tracking frame exceeds a threshold (0.5 in the experiment) to the total number of frames of the video. The average CLE, DP, and OP of the six algorithms on the eight sets of test videos are shown in Table 1, Table 2, and Table 3. For each set of videos, the results of the best performing algorithm are expressed in bold.

Table 1. Center location error comparision of different methods (pixel)

Video sequence	CSK	CN	KCF	SAMF	DSST	OURS
Car4	18.6	12.7	6.5	8.4	10.8	**5.4**
Basketball	164.0	153.0	90.7	4.3	72.6	**3.8**
Singer1	90.2	87.2	80.3	13.6	34.2	**7.7**
Girl	32.6	27.3	16.5	13.2	13.4	**9.5**
Box	17.6	13.2	6.3	4.2	3.8	**3.2**
Jogging	113.0	90.7	100.0	77.4	77.8	**11.2**
Lemming	137.0	129.8	98.6	92.3	92.4	**12.5**
Subway	162.3	152.6	3.4	**2.6**	78.5	3.9
Average	91.91	83.32	50.29	27.0	47.94	**7.15**

Table 2. Distance precision comparision of different methods (%)

Video sequence	CSK	CN	KCF	SAMF	DSST	OURS
Car4	37.2	46.8	94.6	86.3	84.2	**96.7**
Basketball	22.6	27.8	32.4	**100.0**	53.7	**100.0**
Singer1	25.8	27.4	32.2	97.4	72.3	**100.0**
Girl	72.3	73.5	91.4	93.7	92.0	**100.0**
Box	34.2	37.8	75.3	97.4	96.8	**98.2**
Jogging	30.8	43.6	33.1	52.3	51.7	**83.7**
Lemming	24.2	24.6	38.3	42.5	40.6	**82.9**
Subway	23.0	23.2	**100.0**	**100.0**	52.6	**100.0**
Average	33.76	38.09	62.16	83.70	67.99	**95.19**

Table 3. Overlap precision comparision of different methods (%)

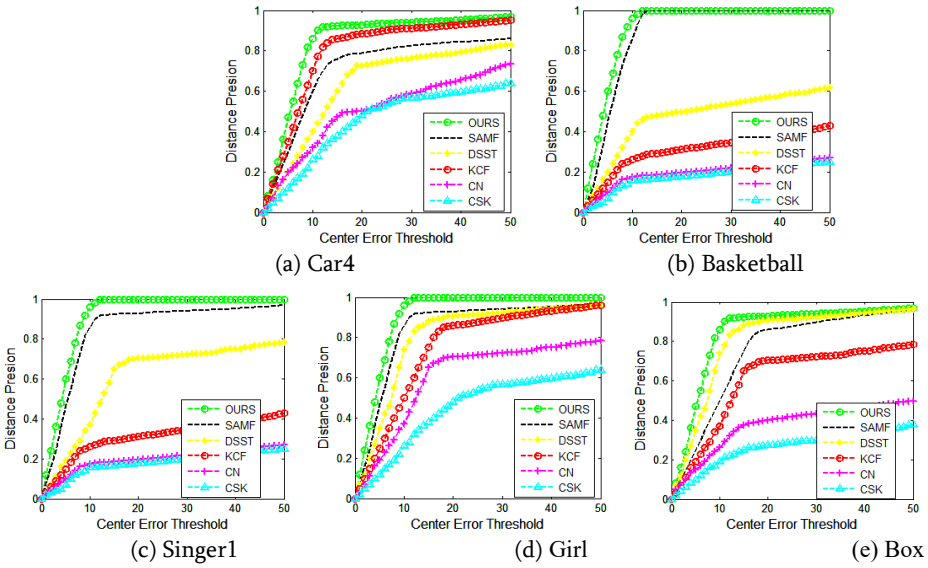
Video sequence	CSK	CN	KCF	SAMF	DSST	OURS
Car4	28.3	31.4	78.4	62.5	60.4	**81.3**
Basketball	22.5	22.5	23.7	70.2	42.3	**78.0**
Singer1	23.3	27.4	32.0	73.2	48.6	**82.6**
Girl	51.7	52.6	70.3	73.5	72.8	**78.3**
Box	30.3	46.4	66.3	76.4	92.3	**93.6**
Jogging	22.5	42.8	29.3	62.3	62.0	**80.3**
Lemming	23.2	23.5	32.6	37.3	37.7	**79.5**
Subway	22.3	22.3	**100.0**	**100.0**	63.2	98.8
Average	28.01	33.61	54.08	69.43	59.91	**84.05**

4.2 Algorithm performance comparison experiment

As can be seen from Table 1, Table 2, and Table 3, our algorithm obtained an average CLE of 7.15 pixels on 8 sets of videos, an average DP of 95.19%, and an average OP of 84.05%. Among the other five algorithms, SAMF performed best. Compared with SAMF, our algorithm reduced the average CLE by 19.85 pixels, the average DP increased by 11.49%, and the average OP increased by 14.62%. To facilitate visual comparison, we draw a DP curve as shown in Figure 2. As can be seen from Table 4，our algorithm has a much higher tracking speed than the DSST and SAMF tracking algorithms using the scale pyramid

Table 4. Speed comparison of 6 tracking algorithm frame/s

Tracking algorithm	CSK	CN	KCF	SAMF	DSST	OURS
Average	273	198	254	123	136	231

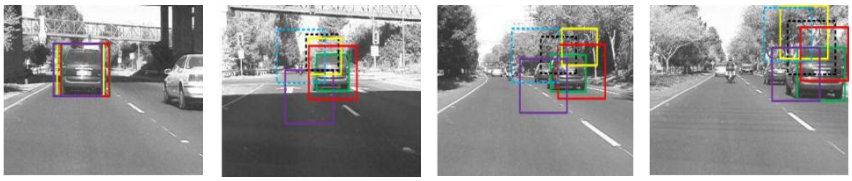


(a) Car4 (b) Basketball

(c) Singer1 (d) Girl (e) Box

Figure 2. Distance presicion curves of 6 methods for 5 test sequences

In figure 3(a), there are problems such as illumination changes, target scale changes and background interference in the Car4 video. When illumination and target scale change in the 238th frame, other five algorithms all have certain tracking deviations, only our algorithm can track the entire video column well. In figure 3(b), there are interference factors such as target scale change, deformation and fast motion in the Basketball video. When the target is deformed in the 54th frame, and the target moves rapidly in the 73rd frame, other algorithms appear larger tracking deviations and even tracking failures, only our algorithm is able to accurately track the entire video sequences. In figure 3(c), Singer1 video has a dramatic change in target scale and illumination. For the entire video sequences, only our algorithm and SAMF have better tracking effect. However, our algorithm's CLE, DP and OP performance indicators are better than SAMF. In figure 3(d), there are interference factors such as target scale change, rotation, attitude change and partial occlusion in the Girl video. When the target is partially occluded in the 437th frame, other algorithms have failed to track, only our algorithm can track the entire video column well. In Figure 3(e), the target has undergone scale change, fast moving and rotation in the Box video. For the whole video sequences, only our algorithm and DSST have better tracking effect, but the three performance indexes of CLE, DP and OP of our algorithm are better than DSST.



（a） Car4(#159,#238,#456,#641)

* Corresponding author: [a]Di Wu:1934169204@qq.com

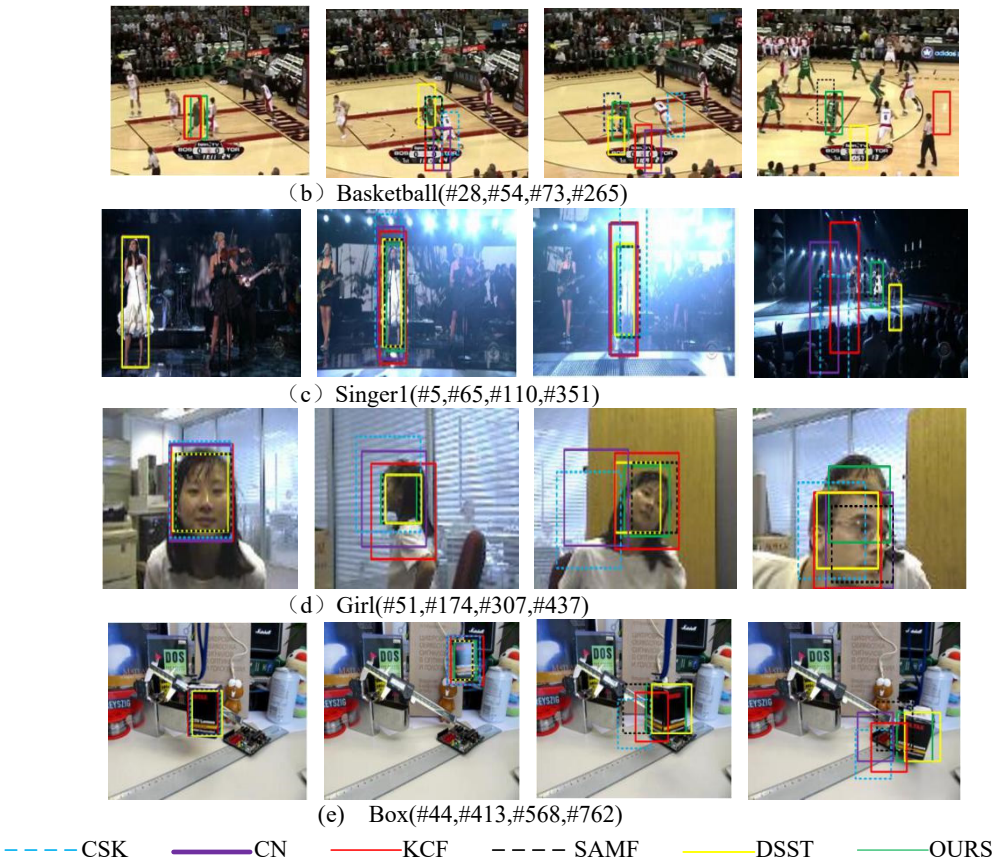


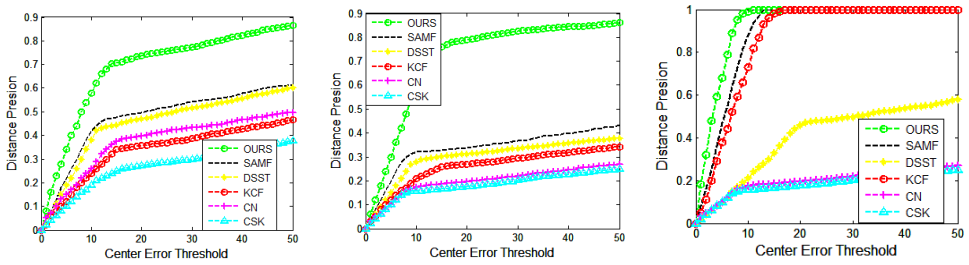
Figure 3. Tracking results of 6 algoriths

4.3 Tracking experiment when the target is severely occluded

In order to test the tracking effect of the six algorithms when the target is severely occluded, we select three sets of videos contain the target occlusion (Jogging, Lemming, and Subway) to test. Figure 4 shows the DP curves of the three sets of videos. It can be seen that compared with the other five algorithms, our algorithm can obtain more accurate and stable tracking results when the target is severely occluded.

Figure 5 shows the tracking results of the three groups of videos from the beginning of occlusion to the end of occlusion. It can be seen that when the target encounters severe occlusion in the Jogging video and the Lemming video, only our algorithm can still track the target effectively, and other five algorithms all have the failure of tracking; In the Subway video, the target of the 41st frame is severely occluded. By the end of the 51st frame occlusion, our algorithm, KCF and SAMF algorithms can still accurately track the target. The other three algorithms have failed to track the target, but the CLE, DP, and OP indicators of our algorithm are superior to KCF and SAMF.

The above experimental results show that compared with the other five tracking algorithms, our algorithm is more robust and can effectively track the target in the case of severe occlusion.



(a)Jogging (b)Lemming (c)Subway

Figure 4. 3 groups of severely blocked video's DP curves

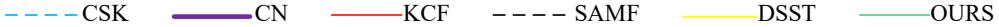


Figure 5. 6 algorithms' tracking result on 3 videos under serious occlusion

5 Conclusion

In the framework of kernel correlation filtering, we propose a scale adaptive correlation filtering target tracking algorithm combined with learning rate adjustment. Our algorithm achieves the scale estimation of the target by using a small number of scale samples, which not only improves the adaptability of our algorithm to the scale change of the target, but also improves the tracking speed compared to the scale estimation strategy using the scale pyramid. In addition, by using the strategy of segmentally adjusting the online learning rate to update the apparent model of the target in real time, the problem of tracking failure when the target is severely occluded is solved. Compared with other five classic target tracking algorithms, our algorithm is more robust in complex environments such as target scale variation, severe occlusion and background interference. At the same time, the average tracking speed of 231 frame/s can meet the requirements of real-time.

References:

1. G.P. Zhao, Y.P. Shen, J.Y. Wang. Adaptive Feature Fusion Object Tracking Based on Circulant Structure with Kernel[J]. Acta Optica Sinica**37**,201-210(2017).
2. Y.C. Wang, H. Huang, S.M. Li, et al. Correlation Filter Tracking Based on Online Detection and Scale-Adaption[J].Acta Optica Sinica**38**,02150-02(2018).
3. W. Gao, M. Zhu, B.G. He, et al. Overview of Target Tracking Technology[J]. Chinese Optics**7**, 365-375(2014).
4. S. Hare, S. Golodetz, A. Saffari, et al. Struck: Structured Output Tracking with Kernels[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence**38**, 2096(2015).
5. K. Zhang, L. Zhang, M.H. Yang. Real-Time Compressive Tracking[C].European Conference on Computer Vision. Springer-Verlag**2**, 864-877(2012).
6. J. Kwon, K.M. Lee. Visual Tracking Decomposition[C].Computer Vision and Pattern Recognition. IEEE, 1269-1276(2010).
7. T. Vojir, J. Noskova, J. Matas. Robust Scale-Adaptive Mean-Shift for Tracking[C]. Scandinavian Conference on Image Analysis. Springer, Berlin, Heidelberg, 652-663(2013).
8. C. Rui, P. Martins, J. Batista. Exploiting the Circulant Structure of Tracking-by-Detection with Kernels[C]. European Conference on Computer Vision. Springer-Verlag, 702-715(2012).
9. M. Danelljan, F.S. Khan, M. Felsberg, et al. Adaptive Color Attributes for Real-Time Visual Tracking[C]. Computer Vision and Pattern Recognition. IEEE, 1090-1097(2014).
10. J.F. Henriques, C. Rui, P. Martins, et al. High-Speed Tracking with Kernelized Correlation Filters[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence**37**, 583-596(2015).
11. M. Danelljan, M. Häger, K.S. Fahad, et al. Accurate Scale Estimation for Robust Visual Tracking[C]. British Machine Vision Conference**65**, 1-11(2014).
12. Y. Li, J. Zhu. A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration[J] **8926**,254-265(2014).
13. Q. Shen, X.L. Yan, L.F. Liu, et al.Multi-Scale Correlation Filtering Tracker Based on Adaptive Feature Selection. Acta Optica Sinica**5**,166-175(2017).
14. C.Z. Xiong, L.L. Zhao, F.H. Guo. Kernelized Correlation Filters Tracking Based on Adaptive Feature Fusion. Journal of Computer-Aided Design & Computer Graphics**29**,1068-1074(2017).
15. L. Zhang, Y.J. Wang, H.H. Sun, et al.Adaptive Scale Object Tracking with Kernelized Correlation Filters.Optics and Precision Engineering**24**,448-459(2016).