

# Target Tracking based on KCF Combining with Spatio-Temporal Context Learning

Aili Wang<sup>a,\*</sup>, Zhennan Yang<sup>a</sup>, Yushi Chen<sup>b</sup>, Yuji Iwahori<sup>c</sup>

<sup>a</sup>*Higher Education Key Lab for Measure & Control Technology and Instrumentations of Heilongjiang,  
Harbin University of Science and Technology, Harbin, 150080, China*

<sup>b</sup>*School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin, 150001, China*

<sup>c</sup>*Dept of Computer Science, Chubu University, Aichi, Japan*

---

## Abstract

Most target tracking is based on a lot of samples training to build the model of the target, which is then carried on the tracking processing. This will need to choose a lot of tracked target samples for learning and training. However, there are all kinds of deformation of the training samples, including variety of light and scale, and so on, causing the long computation time, high computational complexity, and less robustness. The traditional kernel correlation filtering (KCF) tracking is through online learning of the first frame in the target video. It then uses cyclic matrix to strengthen samples robustness, reducing the complexity of the calculation and time. But, the traditional KCF nuclear is unsatisfactory used for complex scenarios and quick treatment. In this paper, under the framework of the KCF, the target context information is introduced to make the tracking have better robustness and a better effect to deal with complex scenarios.

**Keywords:** target tracking; kernel correlation filtering; spatio-temporal context learning; online learning; cyclic matrix

(Submitted on November 26, 2017; Revised on December 29, 2017; Accepted on January 29, 2018)

© 2018 Totem Publisher, Inc. All rights reserved.

---

## 1. Introduction

Object tracking is a prevalent considerable topic; it is widely used in many image processing fields, such as surveillance, motion analysis, activity recognition, and so on. The main challenge for object tracking is to handle large appearance changes of the target object and the background over time due to occlusion, illumination changes, and pose variation [2,5,6,12]. The time tracking robustness and tracking effect remain to be improved; how to achieve a good tracking algorithm is always a problem worth studying.

Wang's [12] tracker is divided into five parts: the motion model, feature extraction, target model, model updating and overall effect's post-processing. The motion model is the target estimation of the following video frames or target selection in a certain range based on the information of the prior frame. Feature extraction is the assurance of subsequent sequence confirmed target location [1]. Target model's update is particularly important for controlling the adaptability and drift of tracking [2]. The post-processing of overall effect can fine-tune the parameters of the tracker to achieve a better result.

Most trackers are established on detection. The selection of training samples is a positive sample extraction at the centre of the target samples, whereas negative samples are based on the surrounding image [13]. Then, the positive sample is labeled 1, and the negative sample is labeled 0. A disadvantage of this tag sample method is not to express the weight of each negative sample to target; that is to say, the weights are the same for the samples far from the centre and the ones near the centre [14]. Reference [3] gives the evaluation and comparison of the 29 tracking algorithms. 2015 assesses the tracking performance of 100 video sequences. CSK uses cyclic shift to carry out dense sampling, combined with FFT in the frequency domain to replace convolution in the time domain for reducing the amount of computation [3]. CSK can extract all the information of the target, but only uses the gray feature. Thus, the description of the target feature is insufficient. The

---

\* Corresponding author.

E-mail address: aili925@hrbust.edu.cn

author also proposed KCF [4] (Kernel Correlation Filters) algorithm to extract the feature, with HOG (Histogram of Oriented Gradients) instead of the natural characteristics of simple pixel value. KCF algorithm lacks robustness for the video sequence of the background and the target's large change. This paper strengthens to optimize the algorithm based on the KCF framework. After the HOG feature extraction combined with shape context information to obtain good tracking effect [8], 1.5 times the area around the target is the tracking area.

## 2. KCF Principles

In 2010, reference [15] put forward relevant filter application in object tracking application. The main principle of filtering is: the correlation between the two signals  $f$  and  $g$  is associated with:

$$(f \otimes g) = \int_{-\infty}^{\infty} f^*(t)g(t + \tau)dt \quad (1)$$

Where  $f^*$  is the conjugate of  $f$  shows in Equation (1). In simple terms, correlation is the similarity between two functions at a certain time. As a theoretical basis applied to tracking, the idea is that the higher the similarity of two signals, the higher the relative value, so the question becomes finding the largest term of target response.

In most cases, motion tracking is to determine target motion trajectory or direction through judging the relationship between adjacent frames, continue the iterations, and complete the target tracking [7,10]. KCF algorithm applies kernel function to fuse HOG features in multi-channel and make the training of the classifier deal with the detection of the target with explanatory power. KCF tracker can deal with many situations including illumination change, occlusion, non-rigid deformation, motion blur, and background clutter and rotation. Tracking effect is not good for scale change, fast motion, and rigid deformation.

First, calculate  $k(x, x') = g(x^T x')$  for some function  $g$ . Then,  $k^{xx'}$  has elements [16]:

$$k_i^{xx'} = k(x', P^{i-1}x) = g(x^T P^{i-1}x). \quad (2)$$

$g(x)$  is kernel function Equation (2). It will be written in vector form Equation (3):

$$k^{xx'} = g(C(x)x'). \quad (3)$$

where  $\mathcal{F}^{-1}$  expresses IDFT.

A polynomial kernel  $k(x, x') = (x^T x' + a)^b$  is shown in Equation (4):

$$k^{xx'} = \left( \mathcal{F}^{-1}(\hat{x} \odot \hat{x}') \right)^b. \quad (4)$$

The KCF tracker learns the object information through a separate image patch and uses circulant matrix and kernel function to solve the KCF filter. It tracks the object via the maximum correlation function of filter response calculation [9].

In order to explain the simplicity, only conduct analysis on single channel information dimension. That is to say,  $x, x'$  is extended to one-dimensional vector. It is also applicable to two-dimensional programming code. One-dimensional vector translations can be processed by a cyclic shift operator.

The product  $P_x = [x_n, x_1, x_2, \dots, x_{n-1}]^T$  shifts  $x$  by one element, modeling a small translation. In order to achieve a larger translation, the matrix power  $P_x^u$  is used. A negative  $u$  will shift in the reverse direction. The same signal  $x$  periodically every  $n$  shifts can be obtained.

The matrix elements of the first row are shifted by the first element, and so on. So the matrix of line  $x$  is  $P_x = [x_n, x_1, x_2, \dots, x_{n-1}]^T$ . Then, the  $N$  cyclic shift vectors are obtained by multiplying the permutation matrix, and arranged in a

matrix in order to generate circle matrix.  $C(x)$  is final representation and is a stationary stochastic process. The stationary process of  $P$  is as in Equation (5):

$$\{P^u = u = 0, \dots, n-1\}. \quad (5)$$

It can be seen from the above formula, in the period of Fourier transformation, the structure of cyclic matrix will greatly reduce the amount of calculation [8].

$$X = F \text{diag}(\hat{x}) F^H \quad (6)$$

Where  $\hat{x}$  is a discrete Fourier transformation of  $x$  and  $F$  is a diagonal matrix of discrete Fourier transformation in Equation (6). Thus, the process of matrix inversion can be change to calcite the eigenvalue of the matrix. According to  $\mathcal{F}(z) = \sqrt{n}Fz$ , the unique map can be calculated in the vector space.

### 3. Spatio-temporal Context Learning

The main basis of the context of time and space: the appearance of the target tracking process, shape and random changes of factors such as shade, tracking target alone will make it difficult to track, reduce robustness, and can't make good use of the target environment information. This article accepts the target on the feature extraction of certain areas around to correlate the relationship between spatial context; this will reduce the proportion of tracking failure to raise the success rate of tracking. For example, in Figure 1, only consider the target itself (green area), so when the light changes, motion blur, scale changes, tracking success rate is difficult to guarantee. But, surrounding pixels (red area) will greatly increase the success rate. Under the condition of multiframe, shape context information is available to use [10].

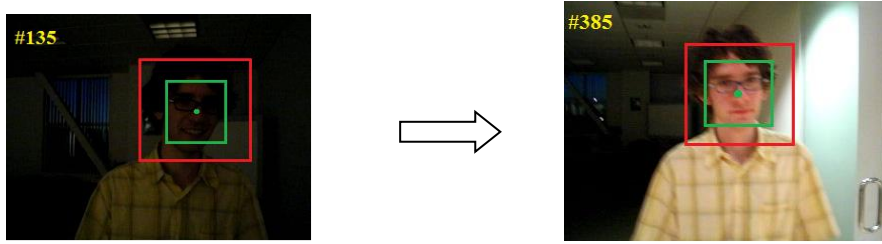


Figure 1. The Context Region of Object

Because the target frame slightly changes, Spatio-Temporal Context (STC) is a spatial context model based on the space of the target and its local region. Use this model to update the next frame. By combining the information of space to complete convolution of regions, we can get the confidence map and find the maximum likelihood probability as the best position [19].

#### 3.1. Context Prior Model

According to the prior probability of  $P(v(z)|o)$ , the spatial context model and the final model parameters are expressed as

$$P(v(z)|o) = I(z)\omega_\sigma(z - x^*) \quad (7)$$

and

$$\omega_\sigma(d) = \alpha \cdot \exp\left(-\frac{d^2}{\alpha^2}\right) \quad (8)$$

$I(z)$  is the gray process at the  $z$  point, mainly for the description of the target contour and texture,  $\omega$  is the weighted process of  $z$ . The greater is the weighted coefficient, said the distance between the  $z$  from  $x$  is farther away which is shown in Equation (7) and Equation (8).

### 3.2. Confidence Map

The confidence map of the object centre of STC is defined as Equation (9)

$$c(x) = b \cdot \exp\left(-\left|\frac{x-x^*}{\alpha}\right|^\beta\right) \quad (9)$$

Where  $\alpha$  express the changes of scale. The key parameters  $\beta$  is essential for the final decision of model.  $\beta$  will have a great effect on the confidence map. The analysis of the impact of the confidence of each  $\beta$  can be seen in Figure 2. From the figure,  $\beta = 1$  is the value we need to select [18].

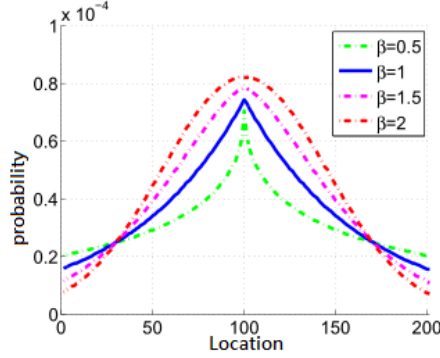


Figure 2. Analysis of the influence of  $\beta$  parameter on the confidence value

### 3.3. Spatial Context Model

$P(x, v(z)|o)$  is defined as Equation (10):

$$P(x, v(z)|o) = h^{sc}(x - z) \quad (10)$$

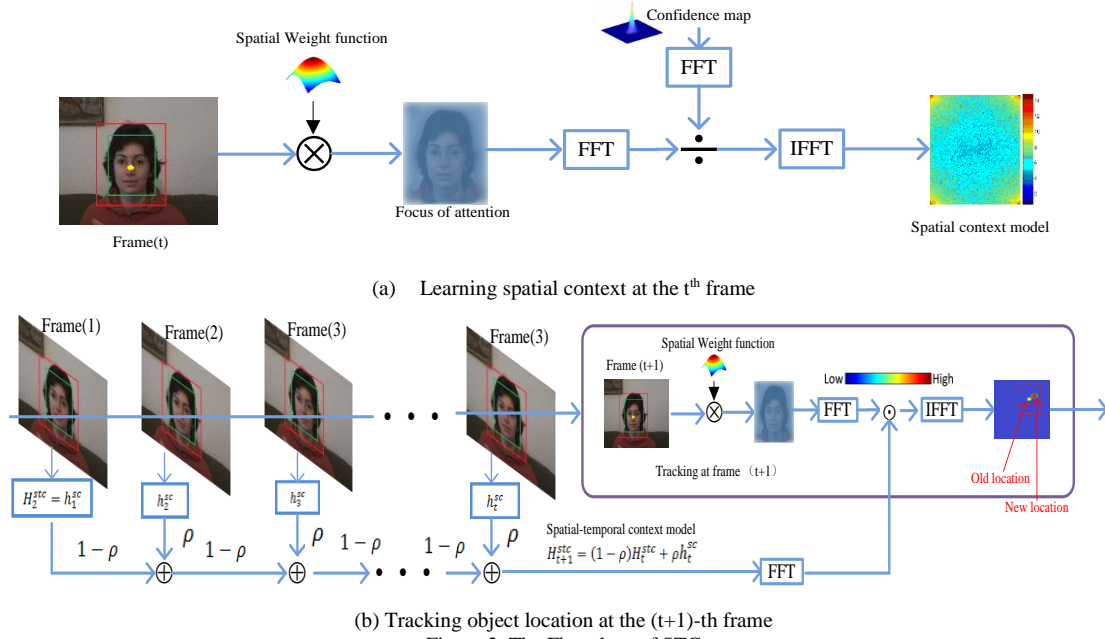
$c(x)$  can be expressed in the format as Equation (11):

$$\begin{aligned} c(x) &= \sum_{v(z) \in x^c} P(x|v(z), o) P(v(z)|o) \\ &= \sum_{v(z) \in x^c} h^{sc}(x - z) I(z) \omega_\sigma(z - x^*) \\ &= h^{sc}(x) \otimes (I(z) \omega_\sigma(z - x^*)) \end{aligned} \quad (11)$$

Where  $\otimes$  denotes the convolution operator. Due to the large amount of computation and time convolution operation takes, the fast Fourier transform is introduced to improve the computation speed. Thus, Equation (12) can be transformed into frequency domain to use fast calculation of Fast Fourier Transform (FFT), i.e.

$$F(c(x)) = F(h^{sc}(x)) \odot F(I(z) \omega_\sigma(z - x^*)) \quad (12)$$

Figure 3 shows the flowchart of STC. The object tracking can be regarded as detection in each image frame. The true value of target position of the first frame in video sequence can be obtained directly from the data set. Use the model to extract the feature of the target region in the current frame; do the same feature extraction for each next frame. Among the process of detection, represent the detection range of  $t$ -th frame model containing the target area.  $h_t^{sc}(x)$  is the parameter representation of the target model when the  $t+1$  frame is detected. When the detection task is completed, plan to track according to  $X_{t+1}^c = \{c(z) = (I_{t+1}(z), z) | z \in \Omega_c(x_t^*)\}$ . Use the feature set of the local region of the target as a two-dimensional set. After completing the context model update task, accelerate the computation speed with fast Fourier transform. Finally, select the target area to track according to the best value of the latest confidence map [17].



$$x_{t+1}^* = \underset{x \in \Omega_c(x_t^*)}{\operatorname{argmax}} c_{t+1}(x) \quad (13)$$

The spatio-temporal context model is updated by

$$H_{t+1}^{stc} = (1 - \rho)H_t^{stc} + \rho h_t^{sc} \quad (14)$$

$\rho$  is the percentage of model in the  $t$  frame feature of target area to the study, and Equation (13) and Equation (14) can be used to calculate the target location of the  $t+1$  frame in the local range of the target, which is shown in Equation (15).

$$H_{\omega}^{stc} = F_{\omega} h_{\omega}^{sc} \quad (15)$$

Where  $H_{\omega}^{stc} \triangleq \int H_{\omega}^{stc} e^{-j\omega t} dt$  is the Euler representation of the Fourier transform in the frequency domain of  $H_t^{stc}$ , and  $F_{\omega}$  computational procedure is as Equation (16):

$$F_{\omega} = \frac{\rho}{e^{j\omega} - (1-\rho)} \quad (16)$$

where  $j$  denotes imaginary unit. The proposed spatiotemporal context model can commendably filter non-target information, so as to complete the task of tracking.

The predicted position of the target can be obtained according to Equation (19). The tracking target scale may alter at different times, so add the scale parameter to scale  $\omega_{\sigma}$ , fitting the change of scale parameter when  $\sigma$  changes. Specific program updates are shown in Equation (17):

$$\begin{cases} s_t' = \sqrt{\frac{c_t(x_t^*)}{c_{t-1}(x_{t-1}^*)}}, \\ \bar{s}_t = \frac{1}{n} \sum_{i=1}^n s_{t-i}', \\ s_{t+1} = (1 - \lambda)s_t + \lambda \bar{s}_t, \\ \sigma_{t+1} = s_t \sigma_t, \end{cases} \quad (17)$$

where  $c_t(\cdot)$  is the confidence map, and  $s_t'$  is the estimated scale between two consecutive frames. The estimated target scale  $s_{t+1}$  is the size estimation of a target in next frame, and  $\bar{s}_t$  is representative of the average scale parameters of  $n$  continuous frames.

#### 4. KCF Combining with Spatio-Temporal Context Learning

KCF has inherited the sampling method of CSK cycle matrix, which greatly reduces the amount of computation and has a great advantage in tracking speed. For the appearance of occlusion, sudden change and other factors, the target tracking effect is not always good. This is because the sampling method of the KCF algorithm only uses the colour feature of the object itself, and does not have the appropriate use of the background information around the target. The spatio-temporal context algorithm proceeds sampling the target itself. The background of the area around the target learns the relationship between the regional contexts. Therefore, this paper proposes a method to extract the feature of the object by using the temporal context and cyclic matrix. The characteristics of time and space context can make the tracking more robust, and the advantages of KCF make tracking faster. The main flow of the algorithm is shown in Figure 4.

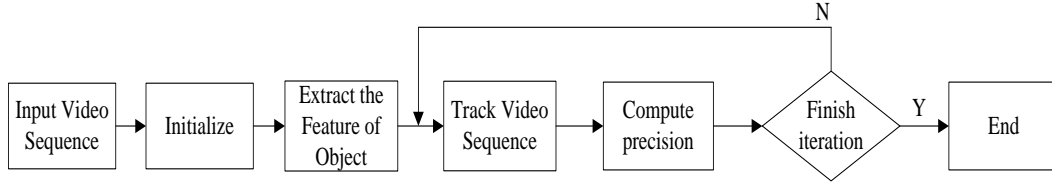


Figure 4. The Flowchart of Object Tracking

Assume that  $x \in R^2$  is a particular location;  $o$  is the target that needs to be tracked. Define belief space for object  $x$  to measure the presence possibility, which is shown in Equation (18).

$$m(x) = P(x|o) \quad (18)$$

Then, define  $X^c = \{c(z) = (I(z), z) | z \in \Omega_c(x^*)\}$  as the context characteristic collection,  $x^*$  is the location of object,  $\Omega_c(x^*)$  is the neighborhood of  $x^*$  (1.5 times),  $I(z)$  is the image gray of  $z$ . That means  $x^*$  is the center, take the target frame around the size of the image as features, as shown in the red box of Figure 1. Thus, the above formula is expanded using full probability in Equation (19):

$$m(x) = P(x|o) = \sum_{c(z) \in X^c} P(x|c(z), o) P(c(z)|o) \quad (19)$$

In addition, the image obtained by cyclic shift at the boundary is not very smooth; the basic way to eliminate this phenomenon is using base image multiplied by a Hanning window to reduce the weight of the edge image. However, the use of the same window function to reduce the weight of the image edge will cause sampling error of target feature in video frames when the target is close to the image edge. So, in this paper, the spatial weighting function is used to deal with the image edge effect, in Equation (20) and Equation (21):

$$S_f(x) = \sum_{l=1}^d x^l * f^l \quad (20)$$

$$\varepsilon_t(f) = \sum_{k=1}^t \alpha_k \|S_f(x_k)\|^2 + \lambda \sum_{l=1}^d \|f^l\|^2 \quad (21)$$

Where  $S$  is response value,  $X$  is the feature in the search area,  $f$  is the learned filtering coefficients and  $*$  expresses circular convolution, Eq.9 is loss function which uses cyclic matrix to solve. Its shortcomings are that circular convolution brings boundary effects, for the convolution kernel is difficult for reducing the boundary's adaptability of model. So this paper adds regularization term and space weighting function (penalty term  $w$ ) to KCF, namely the Equation (21) is amended as Equation (22):

$$\varepsilon_t(f) = \sum_{k=1}^t \alpha_k \|S_f(x_k) - y_k\|^2 + \lambda \sum_{l=1}^d \|w \cdot f^l\|^2 \quad (22)$$

Where the punishment weight  $w$  satisfies gauss distribution and solves the boundary's effect to a certain extent.

5. Experiments Results and Analysis

In order to verify the proposed algorithm’s effectiveness, 10 complex and representative video sequences benchmark are selected for the following experiments (as shown in Table 1). Computer configuration is with the frequency of 2.53 I3CPU. 6GB of memory, the operating system is 64-bit Windows7 and algorithm platform is MatlabR2014a. The scene needs to deal with the main factors including IV (Illumination Variation), SV (Scale Variation), OCC (occlusion), DEF (deformation), MB (Motion Blur), FM (Fast Motion), IPR (In-Plane Rotation), the OPR (Out-of-Plane Rotation), OV (Out-of-View), BC (Background Clutters), and LR (Low Resolution).

The mean value of the position error is a commonly used evaluation criterion [11]. But this rough average evaluation method will bring assessment errors when the video sequence has a small amount of frame loss; they are not being able to fully reflect the real performance of the tracking method. For example, the success rate uses the overlapping rate of tracked target window and the real target as evaluation criteria. If the threshold is higher than the given threshold, the tracking is regarded successful; otherwise, it is regarded as a failure. The random change of the boundary will have an influence on the evaluation of the effect. Even if the centre and the real target centre are near the boundary, the overlap rate cannot reach up to the threshold value of calibration. In this paper, the accuracy curve is used to evaluate the tracking algorithm.

Table 1. Information of the tested video sequences

Video sequence	Frame	Main factors
Basketball	725	IV, OCC, DEF, OPR, BC
BlurBody	334	SV, DEF, MB, FM, IPR
BlurCar2	585	SV, MB, FM
BlurFace	493	MB, FM, IPR
Bolt	350	OCC, DEF, IPR, OPR
Car1	1020	IV, SV, MB, FM, BC, LR
Car4	659	IV, SV
CarScale	252	SV, OCC, FM, IPR, OPR
David	770	IV, SV, OCC, DEF, MB, IPR, OPR
Diving	231	SV, DEF, IPR

Figure 5 is a comparison of the proposed algorithm and other excellent tracking algorithms (KCF, CSK, STRUCK and MOSSE) for different video sequences. From the video sequence tracking results, it can be seen that the proposed tracking algorithm in this paper has a better tracking effect when fuzzy, scale change or deformation occurs in the scene. Figure 6 gives the tracking results of different video sequences.

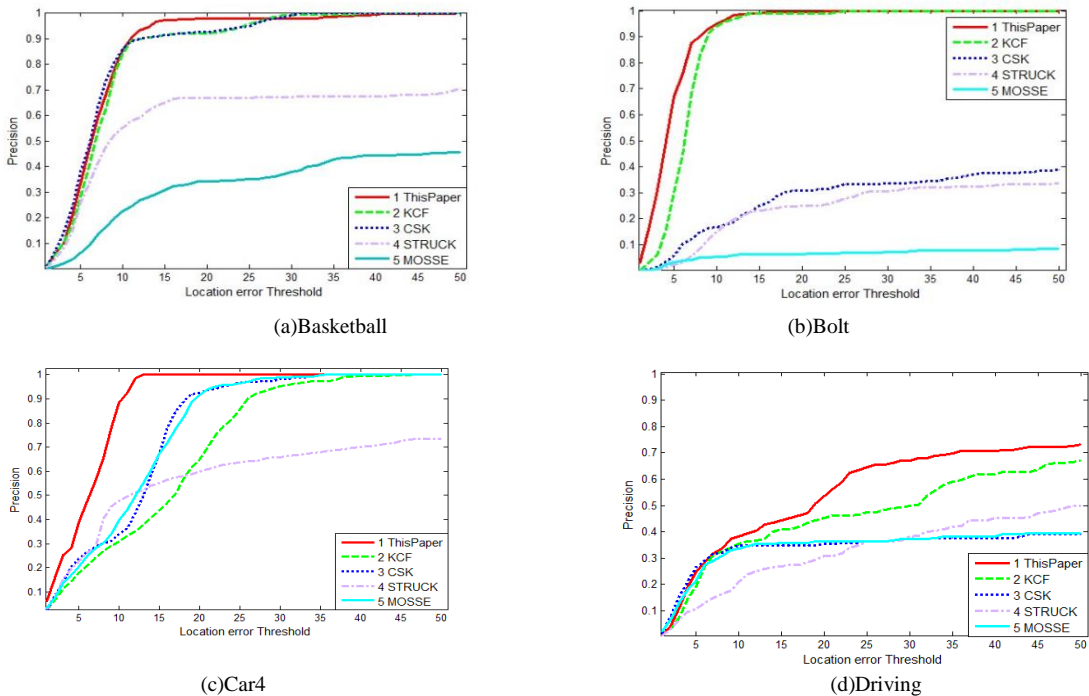


Figure 5. Tracking Results Comparison of Different Algorithms





Figure 6. Tracking Results of Different Video Frame (From top to bottom are Basketball, CarScale, Bolt, Car1, David, Diving, Car4, Deer)

In Table 2, the tracking accuracy and tracking speed of the selected 10 sets video sequences statistically show that the speed and accuracy of our algorithm are obviously superior to that of the traditional KCF algorithm.

It can be seen from the Table 2 that the proposed tracking algorithm has better robustness when the dimension changes or deformation happens. The shape context uses object as the main feature of target tracking; the negative samples around the goal are also used for feature tracking, greatly improving the robustness of tracking method. Sample circulation makes training samples diversification by avoiding the same kind of feature model affect other feature models. FFT accelerates the operation speed of the tracking algorithm. Sigma determines the weight of the virtual negative samples around the object.

When the sigma is small, the label value between the positive and negative samples becomes smaller, which leads to the difference of training results worse. So, it is easy to lose the target. Sigma is too large. Directly regard the negative sample value to be 0 and the positive sample value to be nearly 1. This will put too much emphasis on the importance of the intermediate base sample, so the object tracking results are not good.

In this paper, the size of cell block in sigma and HOG is adjusted to further improve the speed of tracking experiment. In addition, TLD and Struck all have good speed and effectiveness. The running speed of Struck is about 20f/s and TLD is about 28f/s, relatively fast. The speed is still not very fast. The tracking algorithm under the framework of KCF the speed has been greatly improved in the [9] based on the same testing data set; the average speed is about 200f/s. At the same time, the accuracy rate of the proposed tracking algorithm is higher than Struck and TLD.



Table 2. Comparison of tracking performance of different video sequences

Video sequence	Precision(KCF/ours)%	Speed(ours/KCF)FPS
Basketball	0.923/0.928	96.84/145.07
BlurBody	0.584/0.850	43.39/103.01
BlurCar2	0.916/0.938	69.48/146.84
BlurFace	0.765/0.842	80.50/113.32
Bolt	0.883/0.919	145.42/197.32
Car1	0.638/0.739	58.88/105.07
Car4	0.571/0.950	30.13/57.58
CarScale	0.662/0.806	199.80/246.03
David	0.819/0.848	57.70/88.36
Diving	0.463/0.535	85.30/130.30

## 6. Conclusions

In this paper, the feature extraction process under the framework of KCF, and the kernel correlation filter tracker based on shape context optimization is studied. In this method, feature, location and scale of the target are detected by using the improved kernel correlation filter. Then, the target tracking in video sequences is carried out. By adjusting the tracking area and processing the target boundary extraction effect produced in the process of processing, the proposed tracking algorithm solves the tracking for complex scene tracking instability problems, improves the anti-interference performance and has stronger adaptability. It can be seen from the 10 video sequence experiments that the proposed algorithm can handle the complex scene, especially since the robustness is better than that of the non-rigid deformation algorithm and the speed is higher than that of other algorithms.

## References

1. G. Adhikari, S. K. Sahani, M. S. Chauhan and B. K. Das, "Fast Real Time Object Tracking Based on Normalized Cross Correlation and Importance of Thresholding Segmentation," International Conference on Recent Trends in Information Technology, pp. 1-5, 2016.
2. S. D. Bolme, R. J. Beveridge and A. B. Draper, "Visual Object Tracking Using Adaptive Correlation Filters," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp.2544-2550, 2010.
3. R. R. Bharath, D. Bavya, and P. Bavani, "Moving Object Tracking Based on Gaussian Kernel and Template Modelling," International Journal of Signal Processing, Image Processing and Pattern Recognition,9(2), pp.203-210, 2016.
4. J. Choi and M. Maurer, "Local Volumetric Hybrid-Map-Based Simultaneous Localization and Mapping with Moving Object Tracking," IEEE Transactions on Intelligent Transportation Systems, pp.1-16, 2016.
5. F. J. Henriques, C. Rui and P. Martins, "Exploiting the Circulant Structure of Tracking-by-Detection with Kernels," Computer Vision – ECCV 2012. Springer Berlin Heidelberg, pp.702-715, 2012.
6. F. J. Henriques, C. Rui and P. Martins, "High-Speed Tracking with Kernelized Correlation Filters," IEEE Transactions on Pattern Analysis & Machine Intelligence, 37(3), pp.583-596, 2015.
7. L. Lei, "Online Mean Kernel Learning for Object Tracking," International Journal of Signal Processing, Image Processing and Pattern Recognition, 8(11), pp.273-282, 2015.
8. L. Meng, "A Survey of Video Object Tracking," International Journal of Control and Automation8 (9), pp.303-312, 2015.
9. A. Nussberger, H. Grabner and L. V. Gool, "Feature Article: Robust Aerial Object Tracking from an Airborne platform," IEEE Aerospace & Electronics Systems Magazine, pp.38-46, 2016.
10. C. Nyirarugira and T. Kim: "Adaptive Differential Evolution Algorithm for Real Time Object Tracking." IEEE Transactions on Consumer Electronics, pp. 833-838, 2013.
11. Ramasubramanian, M. Rangaswamy and G. MA. Reddy, "A Survey Study on Detecting and Tracking Objective Methods," IEEE National Conference on Emerging Trends in New and Renewable Energy Sources and Energy Management, pp.159-164, 2015.

12. K. R. Reddy, K. H. Priya and N. Neelima, "Object Detection and Tracking -- A Survey," International Conference on Computational Intelligence and Communication Networks (CICN), Jabalpur, India, pp. 418-421, 2015.
13. Ramasubramanian, M. Rangaswamy and G. MA. Reddy, "A Survey Study on Detecting and Tracking Objective Methods," IEEE National Conference on Emerging Trends in New and Renewable Energy Sources and Energy Management, pp.159-164, 2015.
14. S. S. Sengar and S. Mukhopadhyay, "Moving Object Tracking Using Laplacian-DCT Based Perceptual Hash". International Conference on Wireless Communications, Signal Processing and Networking, pp. 2345-2349, 2016.
15. D. Tung, and K. Y. Wu, "Wide Area Object Tracking Using Multiple Cameras with Mixed Configuration of Over-lapping and Non-Overlapping FOV," International Journal of Multimedia & Ubiquitous Engineering, 11(3), pp.239-248, 2016.
16. N. Wang, J. Shi and Y. D. Yeung, "Understanding and Diagnosing Visual Tracking System," IEEE International Conference on Computer Vision, pp.3101-3109, 2015.
17. Wilson and Kalisa, "Real-Time Tracking for Multiple Objects Based on Implementation of RGB Color Space in Video," International Journal of Signal Processing, pp.331-338, 2016.
18. Y. Wu, J. Lim, and H. M. Yang, "Online Object Tracking: A Benchmark," Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp.2411-2418, 2013.
19. K. L. Zhang and H. M. Yang, "Fast Tracking Via Spatio-Temporal Context Learning," Computer Science, pp.127-141, 2013.