An Anti-occlusion and Scale Adaptive Kernel Correlation Filter for Visual Object Tracking

Yingping Huang[1], Chao Ju[1]*, Xing Hu[1] and Wenyan Ci[2]*

[1]School of Optical-Electrical and Computer Engineering, University of Shanghai for Science & Technology, Shanghai 200093, China

[2]School of Electric Power Engineering, Nannjng Normal University, Taizhou College, Taizhou 225300, China

*Corresponding author: Chao Ju, Wenyan Ci

E-Mail: JuChao1992@163.com, 729076795@qq.com

Received May 3, 2018; revised June 25, 2018; accepted October 17, 2018; published April 30, 2019

Abstract

Focusing on the issue that the conventional Kernel Correlation Filter (KCF) algorithm has poor performance in handling scale change and obscured objects, this paper proposes an anti-occlusion and scale adaptive tracking algorithm in the basis of KCF. The average Peak-to Correlation Energy and the peak value of correlation filtering response are used as the confidence indexes to determine whether the target is obscured. In the case of non-occlusion, we modify the searching scheme of the KCF. Instead of searching for a target with a fixed sample size, we search for the target area with multiple scales and then resize it into the sample size to compare with the learnt model. The scale factor with the maximum filter response is the best target scaling and is updated as the optimal scale for the following tracking. Once occlusion is detected, the model updating and scale updating are stopped. Experiments have been conducted on the OTB benchmark video sequences for compassion with other state-of-the-art tracking methods. The results demonstrate the proposed method can effectively improve the tracking success rate and the accuracy in the cases of scale change and occlusion, and meanwhile ensure a real-time performance.

Keywords: Object tracking, kernel correlation filter, scaling invariance, occlusion

http://doi.org/10.3837/tiis.2019.04.020 ISSN : 1976-7277

1. Introduction

Visual tracking is to locate a moving object in a video sequence and has been a core problem in computer vision with wide-ranging applications in video surveillance, robot perception, intelligent vehicles and human-machine interfaces. In recent years, with the introduction of Kernel Correlation Filter and machine learning technology, the performance of visual object tracking has been greatly improved. However, the problems regarding to multi-scales and obscured objects have not been solved well. This paper addresses on these issues and exploits the properties of the conventional Kernel Correlation Filter to achieve an anti-occlusion and scale adaptive object tracker.

Visual object tracking can be divided into two main categories: generative model methods and discriminant model methods. The generative model methods model the target region in the current frame and search for the region similar to the model in the next frame to determine the predicted position. The methods include Kalman Filtering, Particle Filter, Mean Shift and so on. Discriminant model methods take the target area as the positive sample in the current frame and the background area as the negative sample to train a classifier, and search for the optimal area with the trained classifier in the next frame. Classical discriminant model methods include structured output tracking with kernels (STRUCK) [1] and tracking-learning-detection (TLD) [2]. Different from generative model methods, discriminative model methods use machine learning to train a classifier that can effectively distinguish between foreground and background. In general, discriminant model methods have a better tracking performance than generating model methods.

Bolme et al. [3] proposed a Minimum Output Sum of Squared Error algorithm (MOSSE) and firstly introduced a correlation filter for tracking. Correlation is a measure of the similarity between two signals. The higher the correlation, the more similar the two signals are. For tracking purpose, it is to design such a filter whose response reaches its maximum when it acts on the tracking target. Henriques et al. [4] proposed a Circulant Structure of Tracking-by-detection with kernels algorithm (CSK) based on the MOSSE algorithm. The CSK used a cyclic matrix to conduct dense sampling, therefore, the characteristics of the entire picture was used. In the meantime, the CSK formed a non-linear classifier by introducing a kernel function so that the classifier worked in a high dimensional feature space. This mechanism well solved the problems in the cases of low-dimensional linear inseparable or non-linear separable. However, the CSK does not solve the problems of gray scale features and boundary effects caused by the circular matrix. Furthermore, they proposed Kernel Correlation Filter algorithm (KCF) [5] to optimize for multichannel features and kernel methods. They used the multi-channel HOG feature to replace the single-channel gray feature and a Gaussian kernel function to optimize the operation. However, since the bounding boxes in the KCF were fixed and the target size may change from time to time, the bounding boxes may drift during the tracking process, resulting in a failure of tracking. Also, the KCF did not work in the case that the target is obscured during a tracking process.

In order to solve the multi-scales problem, Li et al. [6] adopted the KCF algorithm to train a filter with HOG and CN features, and used a multi-scaled template to search for the target by shifting the filter around the image. Danelljan et al. [7] used only HOG features and proposed the Discriminative Scale Space Tracking (DSST) method. Discriminative Correlation Filter (DCF) was used to detect the target translational position. They also learned a correlation filter based on a scale pyramid representation to detect scale changes, which was the first time that translation filtering is combined with scale filtering. And later, they put forward a series of accelerated DSST [8] methods and achieved a better real-time performance. Similar to DSST method, Pan et al. [9] exploited the scale pyramid strategy to map the image pyramids into a one-dimensional eigenvector that is used as input of a scale correlation filter. The target scale is estimated from the highest filter response.

Occlusion problem can be considered as a long-term tracking problem. The key is to determine whether the target is obscured. In recent years, some methods [10,11] have been proposed for this issue, such as the use of the partitioning method to track the target. The basic idea of the partitioning method is to divide the target into different parts to learn the features, track them separately, and finally combine the tracked parts together to get the position of the target. In addition, Ma et al. [12] proposed a long-term correlation tracking algorithm. In the basis of the translation-correlation filtering and a scale-correlation filtering of DSST, they also added the third filter to measure the tracking confidence. The measuring module was the Random Ferns Classifier used in TLD algorithm, and was further adjusted to Support Vector Machine classifier. The third confidence filter resembled a MOSSE without dense sampling and a cosine window on the features, and was conducted after translational detection. Yang et al. [13] proposed a long-tern tracking method. They introduced a spatial regularization component for learning of the classifier and used the Newton iteration to get maximizing response score. The confidence of the target with maximum response score was compared to train an on-line support vector machine classifier so that the target can be re-detected in the case of tracking failure.

Tracking confidence reflects the tracking reliability and can be used to judge whether the target is lost. Generative model methods use similarity measure function as tracking confidence measure while discriminant model methods use the classification probability. Two indexes including the maximum response value and the response mode reflect the tracking confidence in correlation filtering methods. Wang et al. [14] used multimodal target detection with high-confidence updating strategy. High-confidence updating means that the model should be updated only when the tracking confidence is relatively high, which avoid contamination of the model and increase the running speed. The high-confidence measure is defined as the combination of the response peak and the average Peak-to-Correlation Energy. In addition, the tracking confidence index includes the Peak to Sidelobe Ratio [3] in MOSSE. It is calculated from the correlation filter peak, the mean and standard deviation of the sidelobes outside the peak window. Lukezic et al. [15] introduced the channel and spatial reliability concepts in DCF tracking and combined the maximum response peak with the ratio between the second and first primary modes in the response graph to reflect the tracking reliability.

This paper proposes an occlusion detection scheme and a scaling pool searching method to achieve anti-occlusion and scale adaptive object tracking in the basis of KCF. The main contributions of the work can be found: 1) Both object scale changes and occlusion are taken into considerations to extend the KCF to an anti-occlusion and scale adaptive tracker. 2) The proposed method achieves an appealing performance with regarding to the tracking success rate and accuracy in comparison with the start-of-the-art tracking algorithms.

2. KCF tracking algorithm fundamentals

2.1 Overview of KCF

The basic idea of the KCF is to train a discriminant classifier in the current frame, and then uses it to detect the target in the next frame. The detected region and its derivations are then taken as new training samples to update the discriminant classifier. When training the discriminant classifier, the target region is selected as the positive sample, the regions around the target obtained by the cyclic shifts are taken as the negative samples. The multi-channel HOG feature is extracted from the samples for the classifier training. A Gaussian Kernel is used for the kernel correlation trick to solve for the ridge regression classifier. **Fig. 1** shows the process of the classifier training and the target tracking of the KCF algorithm. The red dashed box in the left image is the initially detected target. The red solid box is the base sample generated from the target added with padding. The other boxes are the samples generated with the padding window cyclically shifted around. A classifier can be trained by using these samples. The trained classifier is then used to detect target position in the next frame (right image). Taking the predicted area, i.e., the red solid box (the same position as in the previous frame) as the base patch, cyclically shifting the base patch and using the classifier to calculate the responses for each shift, the box with the maximum response is taken as the new target position, i.e., the yellow box. The relative translation between the yellow box and the red box is the translation of the target. Repeat the training process to update the classifier in the new frame for the subsequent detection. With the continuous updating mechanism, the background changes have little effect on the tracking results.

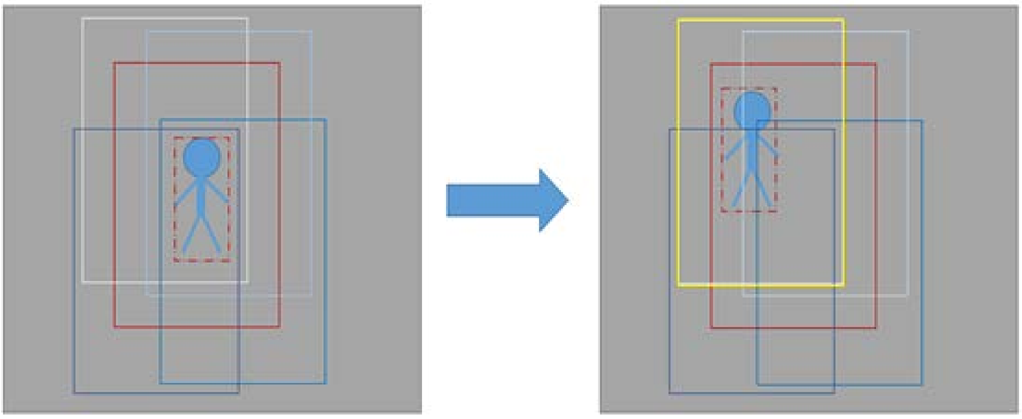


Fig. 1. Process of classifier training and target tracking of the KCF algorithm

2.2 Sample representation

The training samples in the KCF are obtained by cyclically shifting the base sample. For notational simplicity, we consider single-channel one-dimensional signals. The results can be generalized to multichannel two-dimensional images in a straightforward way. Consider an one-dimension vector $x = [x_1, x_2, \cdots\cdots x_n]^T$ as the base sample representing a patch of the target. Use a cyclic shift operator $P = \begin{bmatrix} 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$ to model one-dimensional translation of this vector. The product $Px = [x_n, x_1, \cdots\cdots x_{n-1}]^T$ shifts x by one element. Cyclic shift of x for n times, i.e. $\{P^i x \quad i = 1, \ldots n\}$, generates n one-dimension vectors. Concatenating these vectors yields a matrix, called circulant matrix, which can be expressed as

$$X = C(x) = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_n & x_1 & x_2 & \cdots & x_{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_2 & x_3 & x_4 & \cdots & x_1 \end{bmatrix} \tag{1}$$

The circulant matrix can be made diagonal by the Discrete Fourier Transfer (DFT) of all generated vectors.

$$\mathrm{X} = \mathrm{Fdiag}(\hat{\mathrm{x}}\,)\ \mathrm{F}^{\mathrm{H}} \tag{2}$$

where F is the constant DFT matrix, $\hat{x}$ denotes the DFT of the generated vectors, the hat ^ is the shorthand notation for the DFT of a vector. We have $\hat{x} = F(x) = \sqrt{n}\, Fx$. F^H denotes the complex conjugate transpose of F. By using the diagonal expressions with the circulant matrices, we also have

$$X^H = Fdiag(\hat{x}^*)F^H$$

$$X^H X = Fdiag(\hat{x}^* \odot \hat{x})F^H \tag{3}$$

where $\odot$ denotes the element-wise product, and $\hat{x}^*$ denotes the complex-conjugate of $\hat{x}$, the star * is the operator for the complex-conjugate. The above steps summarize the general approach taken in diagonalizing expressions with circulant matrices. Operations on diagonal matrices are element-wise.

2.3 Classifier determination

The KCF algorithm uses a kernelized ridge regression classifier. Supposing that the training samples are x_i and, the goal of classifier training is to find a linear ridge regression function $f(z) = w^T z$ to minimize the squared error over samples x_i and their regression targetsy_i,

$$\min \textstyle\sum_{i=0}^{n-1} (f(x_i) - y_i)^2 + \lambda||w||^2 \tag{4}$$

where λ is the regularization parameter to ensure the generalization performance of the classifier. The solution can be obtained in the following complex form

$$w = (X^H X + \lambda I)^{-1} X^H y \tag{5}$$

where the data matrix X has one sample per row x_i, each element of y is a regression target y_i, and I= $F^H F$ is an identity matrix. X^Hrepresents the Hermitian transpose of complex conjugate of X.

Applying Eq.3 recursively into Eq.5 generates Eq.6 as follows

$$\widehat{w}^* = \frac{\hat{x} \odot \hat{y}}{\hat{x}^* \odot \hat{x} + \lambda} \tag{6}$$

where $\widehat{w}^*$ is the complex-conjugate of the Fourier transform of w. The physical meaning of the operation is that $\widehat{w}$ can be solved in Fourier domain where the equation becomes element-wise division. We can easily recover w in the spatial domain with the Inverse DFT. The element-wise product operation instead of matrix operation greatly improves the calculation efficiency.

Kernel trick is then applied to map the inputs of a linear problem to the non-linear kernel spaces so that inseparable in the low-dimensional linear space becomes separable in the kernel space. The kernelized version of the ridge regression function can be written as

$$f(z) = w^T z = \sum_{i=1}^{n} a_i k(z, x_i) \tag{7}$$

where a_i is called the dual space coefficient. So the problem of solving for *w* is converted to find the optimal solution of a_i. For most commonly used kernel function, the circulant matrix trick an also be used, so

$$\hat{a} = \frac{\hat{y}}{\hat{k}^{xx} + \lambda} \tag{8}$$

where k^{xx} is defined as kernel correlation of *x* with itself in the Fourier domain in ref. [5], a is the vector of coefficients a_i.

The KCF adopts the Gaussian kernel which can be applied the circulant matrix trick as below:

$$K^{xx'} = exp\left(-\frac{1}{\sigma^2}\left(\left||x|\right|^2 + \left||x'|\right|^2 - 2F^{-1}(\hat{x}^* \odot \hat{x'})\right)\right) \tag{9}$$

where σ is Gaussian kernel standard deviation.

2.4 Fast detection

The patch *z* at the same location in the next frame is treated as the base patch. The candidate patches to be tested are cyclic shifts of the base patch, i.e. $\mathrm{z_i} = \mathrm{P^i z}$. The kernel matrix of the detection is

$$K^z = C(k^{xz}) \tag{10}$$

where k^{xz} is the kernel correlation of *z* and the base sample *x*. Each element of K^z is given by $k(P^{i-1}z, P^{j-1}x)$.

The regression function, i.e. response, for all candidate patches is calculated from Eq.8, that is.

$$f(z) = (K^z)^T a \tag{11}$$

Diagonalzing it yields $\hat{f}\,(z) = \hat{k}^{\,xz} \odot \hat{a}$, which contains responses for all candidate patches. The patch with the maximum response is the target area.

3. Improved Kernel Correlation Filter Tracker

3.1 Scale invariance

The scale change occurs in the process of object tracking. The traditional KCF algorithm uses a fixed model size for the classifier and does not have adaptive model scale updating mechanism. When the target shrinks, the classifier will learn a large number of background information. When the target expands (bigger than the model), the classifier will only trace the partial target. Both cases can cause the model drift and result in a track loss.

A modification of the searching strategy has been made in this paper. Instead of searching for a target with a fixed sample size, we sample the target area with multiple scales, called scaling pool, and then resize it into the sample size to compare with the learnt model. The scale factor with the maximum filter response is the best target scaling and will be updated as the optimal scale to adjust the model size for the following tracking. By this scheme, scale invariance can be achieved in the tracking process**.** **Fig. 2** shows the diagram of the method.

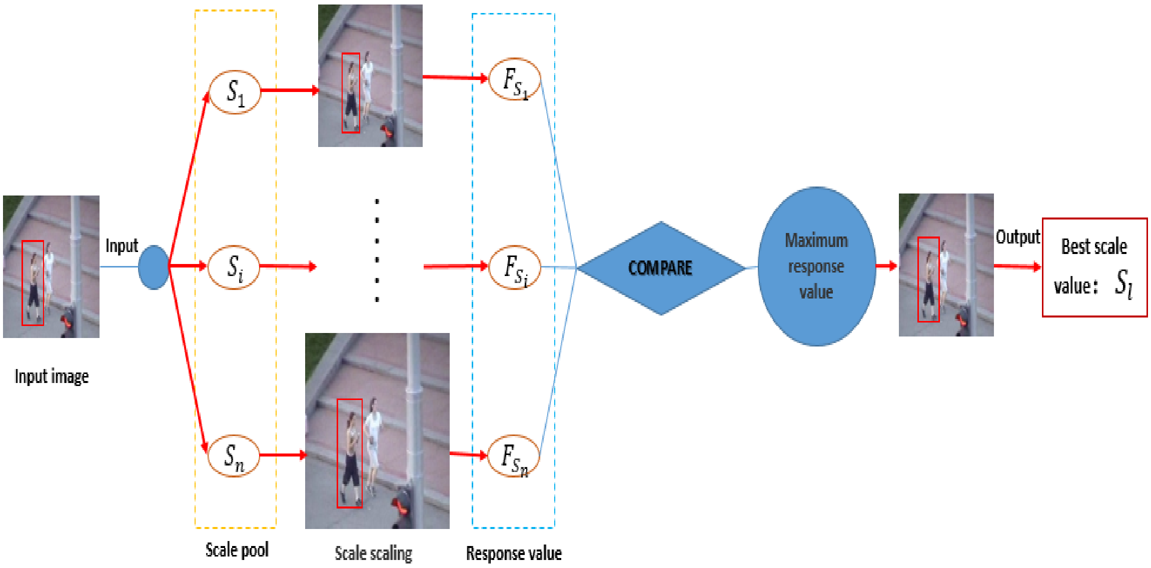


Fig. 2. Target searching scheme with multiple scales

The scaling pool method has also been used in Ref. [6] where seven scales with steps $S = \{0.980, 0.985, 0.99, 1.0, 1.005, 1.01, 1.015\}$ were adopted for target searching. In order to maintain the tracking robustness without significant reduction in processing speed, we reduce the scaling number and increase the step size. Our experiments show that the scaling pool of $S = \{0.90, 0.95, 1.0, 1.05, 1.10\}$ gives a compromise between the tracking performance and the running speed.

3.2 Occlusion detection

The correlation filter response peak ($\mathrm{F_{max}}$) is normally used as the tracking confidence measure, which can be obtained by taking the maximum in Eq.11. The larger the value of $\mathrm{F_{max}}$, the better the tracking effect. F_{max} can somehow indicate an occlusion case, but does not fully reflect the response oscillation since it is not always lower than the historical average peak when the target is obscured. Therefore, only using $\mathrm{F_{max}}$ as judgement does not fully reflect whether the tracking target is occluded or not. In this work, we take both the correlation filter response peak and the average *Peak-to-Correlation Energy* (*APCE*) into consideration for determining whether the target is occluded. The *APCE* can be calculated from the response-map, detailed in Ref. [14], as follow,

$$\mathrm{APCE} = \frac{|\mathrm{F_{max}} - \mathrm{F_{min}}|^2}{\mathrm{mean}(\sum_{\mathrm{w,h}}(\mathrm{F_{w,h}} - \mathrm{F_{min}})^2)} \tag{12}$$

where F_{max} is the response peak, $\mathrm{F_{min}}$ is the response minimum, $\mathrm{F_{w,h}}$ is the response of the sample at position *(w, h)*. This index can fully reflect the response oscillation. A sudden and sharp drop of the *APCE* indicates that the target is occluded.

Fig. 3 is one of video sequences presented in the Benchmark50 database [16], showing a process of a target from partial occlusion to complete occlusion and then unobstructed. **Fig. 4** shows the corresponding $\mathrm{F_{max}}$ and the *APCE* values of each frame. It can be seen that both $\mathrm{F_{max}}$ and *APCE* has a sharp drop near the 74th frame where the target is obscured. At the 100th frame where the target appears again, the both values get recovered to the average. In this work, only when both indexes are greater than a certain proportion of the mean response, the model will be updated. That is, when one of the two indexes drops below the threshold, model updating will be stopped. The proportion thresholds are set as 0.5 and 0.4 for F_{max} and the *APCE* respectively.

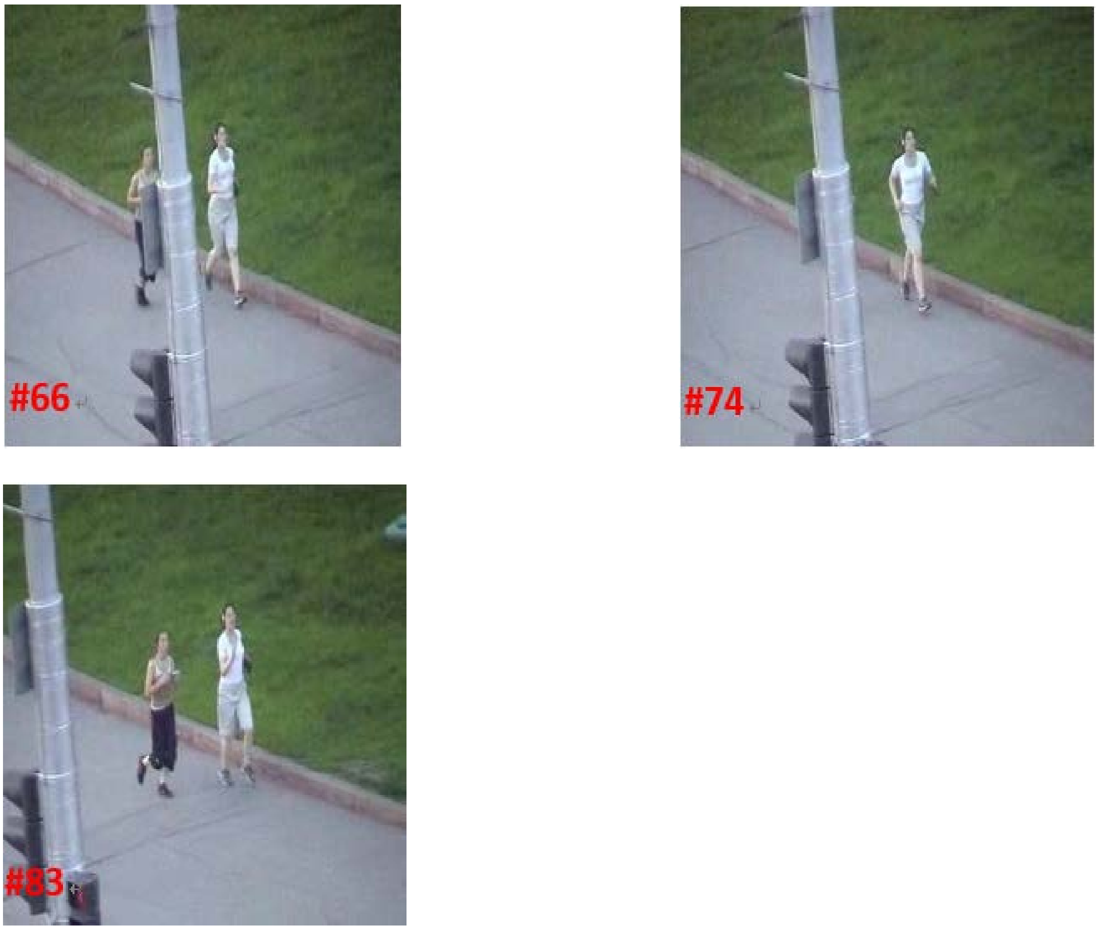


Fig. 3. 66th, 74th and 83th frame of an occlusion case

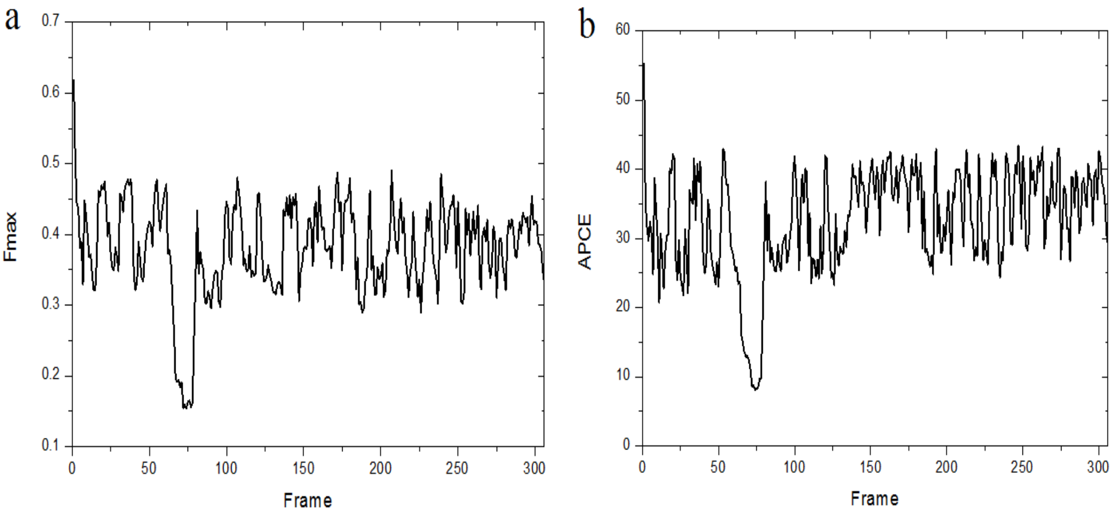


Fig. 4. $F_{max}$and *APCE* values of each frame in the video equence of **Fig. 3**

3.3 Model updating

In the KCF algorithm, the model coefficient is linearly interpolated with

$$\alpha = (1-\beta)\alpha_{pre} + \beta\alpha_{x'} \tag{13}$$

where α, α_{pre}, and $\alpha_{x'}$are the model coefficients obtained from the training samples in the next, current and previous frame respectively, β is a constant. This interpolation is to avoid the model from changing drastically. However, once the target is occluded for several frames, the model may be completely contaminated and therefore may not be able to recover. For anti- occlusion tracker, β should not be a fixed value. Instead, we take

$$\beta = \begin{cases} 0.012, & in\ the\ case\ of\ non-occlusion \\ 0, & in\ the\ case\ of\ occlusion \end{cases} \tag{14}$$

This setting is to stop the model updating when the target is obscured.

Occlusion is caused by horizontal movement. In the cases of occlusion, the target only moves in horizontal direction and does not move in the longitudinal direction, thus the target size will not change. Therefore, once occlusion is detected, scaling updating must also be stopped. This will avoid model drift and track loss drift caused by scale updating.

3.4 Algorithm flow

By incorporating the scaling pool searching and occlusion detection mechanism into the traditional KCF tracking algorithm, the improved kernel-related filter algorithm is as follows:

Anti-occlusion and Scale Adaptive Kernel Correlation Filter Tracking Algorithm

```
Parameter initialization
Read the $i$ th frame of the image sequence
if ($i=0$) Use the first frame and the bounding box to initialize the tracker
else
    The filter response values $f$ for each sample are calculated based on the current frame
    Calculate the tracking confidence index $F_{max}$ and the APCE, and the mean response $\bar{f}$
    if $F_{max}>0.5*\bar{f}$ && the APCE> $0.4*\bar{f}$
```

```
8.              Use the scale pool searching to find the best target scale value $S_l$
9.              Set β= 0.012 to update the classifier model in terms of the size and the coefficients
                    $\alpha = (1-\beta)\alpha_{pre} + \beta\alpha_{x'}$
10.             Follow steps 8 and 9 to update the tracker and the target area
11.         else (i.e. occlusion is detected)
12.         Stop the scale updating, set $\beta$ as zero, stop updating the classifier model
13.     Return to step 2
```

4 . Experiments and Results

4.1 Experimental conditions and evaluation index

Experiments are conducted in a PC with 2.90GHZ Intel® Dual Core i5 processor and 8GB of RAM. The traditional KCF parameters remain unchanged. The padding window is 2.5 times of the target. The Gaussian kernel standard deviation σ is 0.6. The linear interpolation factor β is 0.012. The regularization parameter λ is 0.0001. The scaling pool is $\mathrm{S} = \{0.90, 0.95, 1.0, 1.05, 1.10\}$.

The evaluation index, including the center location error (CLE), overlap precision (OP) and Frame Per Second (FPS), are used to evaluate the tracking performance of the proposed method. CLE is the Euclidean distance between the centers of tracking result and the manually marked ground truth. Tracking is considered successful when CLE is less than a threshold (20 pixels). OP, also called tracking score, is defined as the ratio of the intersection and the union of the ground truth bounding box B_t and the tracked bounding box B_a.

$$\mathrm{score} = \frac{\mathrm{area}(\mathrm{B_t} \cap \mathrm{B_a})}{\mathrm{area}(\mathrm{B_t} \cup \mathrm{B_a})} \in [0，1] \tag{15}$$

When the score is greater than a given threshold (0.5), the tracking is regarded as successful. OP actually reflects the extent of the overlap between the tracked results and the ground truth, i.e. tracking accuracy.

4.2 Comparison with the KCF tracker

Experiments have been conducted on OTB Benchmark50 image sequences, a public video database presented in Ref. [16]. The OTB database, containing the image sequences of 50 scenarios, provides a benchmark platform for evaluation and comparison of different tracking algorithms. The images in the OTB database are annotated with the ground truth of the target position. **Table 1** lists the tracking success rate in terms of CLE and OP with a comparison of the proposed method and the traditional KCF algorithm [5]. Six scenarios are selected from the OTB video database, including obscured objects (Jogging, Women), scale changes (CarScale), fast-moving (Car4), light changes (BlueCar2), and rotate (Dog).

It can be seen that our method has a significant improvement with regard to CLE and OP in Jogging and Women scenarios where the target is sometimes occluded. For Jogging scenario, the CLE is increased from 23.13% to 99.02%; OP from 22.48% to 82.08%. For Women scenario, the CLE is increased from 47.91% to 92.29%; OP from 22.48% to 82.08%.

In the cases of scale change (CarScale) and fast moving (Car4), our method is improved significantly with regard to OP. For CarScale scenario, the OP is increased from 44.84% to 83.73%. For Car4 scenario, the OP is increased from 38.24% to 92.97%. These results indicate that our scale invariance and the classifier model updating mechanism take significant effect in the tracking process.

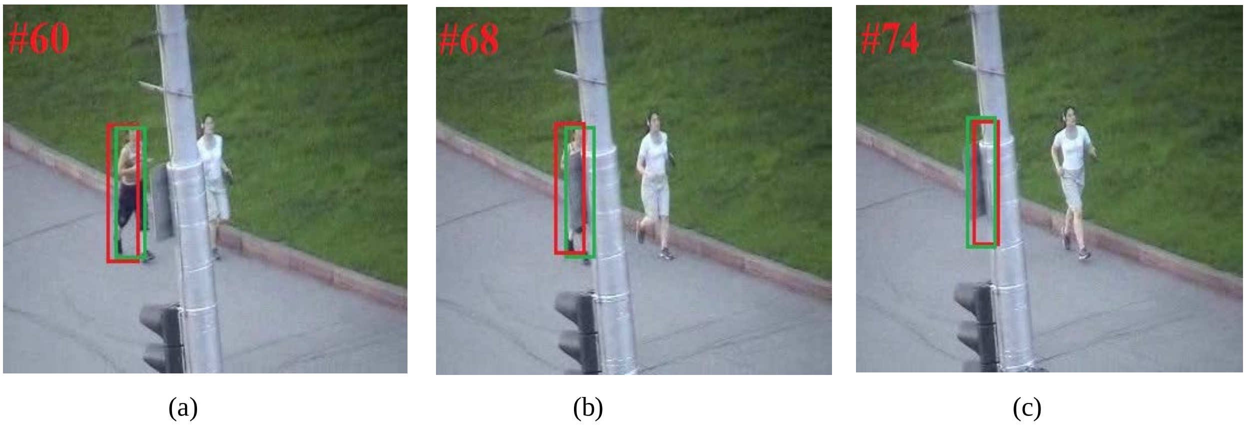
In the cases of light changes (BlueCar2) and rotate (Dog), our algorithm still retains the efficiency of the KCF algorithm. For all scenarios, the FPS has dropped slightly due to the increased complexity of the algorithm but it still guarantees a real-time traceability.

Table 1. Comparison of tracking rate of our method and the KCF using different evaluation indexes for multiple scenarios

Video sequences (scenarios)	Evaluation index	KCF[5]	Ours
Jogging (obscured objects)	CLE	23.13%	99.02%
	OP	22.48%	82.08%
	FPS	32	21
Women (obscured objects)	CLE	47.91%	92.29%
	OP	37.52%	61.14%
	FPS	35	28
CarScale (scale changes)	CLE	72.6%	76.98%
	OP	44.84%	83.73%
	FPS	32	20
Car4 (fast-moving)	CLE	97.27%	100%
	OP	38.24%	92.97%
	FPS	32	21
BlueCar2 (light changes)	CLE	99.65%	99.83%
	OP	94.70%	92.99%
	FPS	21	16
Dog (rotation)	CLE	100%	97.64%
	OP	13.39%	29.13%
	FPS	32	21

Fig. 5 shows an example for the comparison of the tracking process of our tracker (red) and the KCF tracker (green) in the case of occlusion (Jogging, frame 60-100). Two trackers work well until 68th frame where the left pedestrian is partially obscured. At 74th frame where the left pedestrian is completely obscured, the KCF classifier model is contaminated, resulting in the track loss from 74th to 100th frame. Our tracker stops the model updating at 74th frame once the occlusion has been detected by using the method presented in section 3.2. The classifier model retains until the target occurs again at 87th frame. The tracking on the left pedestrian get recovered from the 87th frame to 100th frame. These results indicate that our occlusion detection method takes significant effect in the tracking process.

Fig. 6 shows an example for the comparison of the tracking process of our tracker (red) and the KCF tracker (green) in the case of scale change (CarScale, frame 80-184). In this scenario, the vehicle moves close to the camera with a significant scale change. The traditional KCF algorithm does not have scale updating, thus the size of the green box (KCF tracker) does not change as the target is getting bigger. The KCF tracking will fail from 161th frame in terms of overlap precision (OP score > 0.5). Compared with the KCF tracker, our method has an adaptive scale updating mechanism. Therefore, it can be seen from the figure that the red box (our tracker) is getting bigger as the target is getting bigger. Our tracker can successfully track the target until frame 184th, indicating the better tracking accuracy (OP). These results indicate that our scale pooling searching strategy and model updating mechanism take significant effect in the tracking process.



(a) (b) (c)

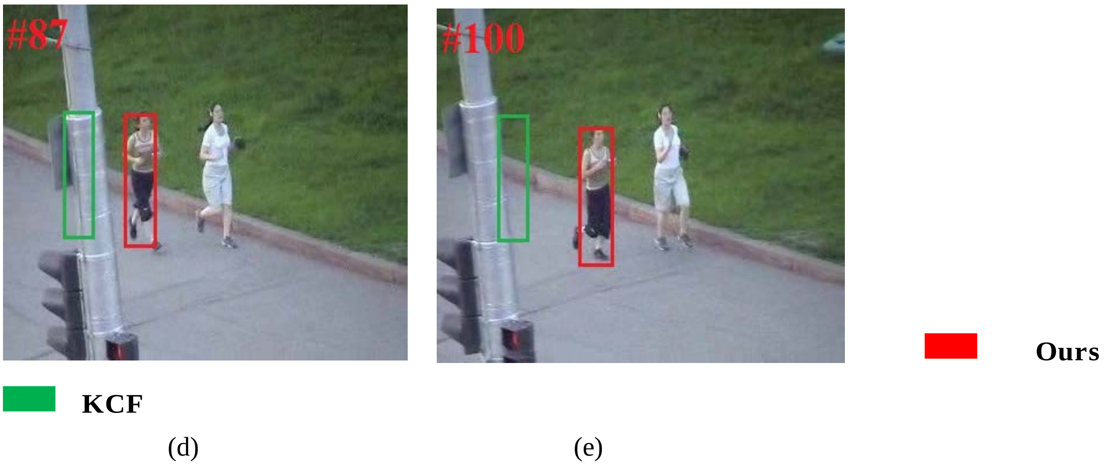


Fig. 5. Comparison of tracking results in the case of occlusion (Jogging)

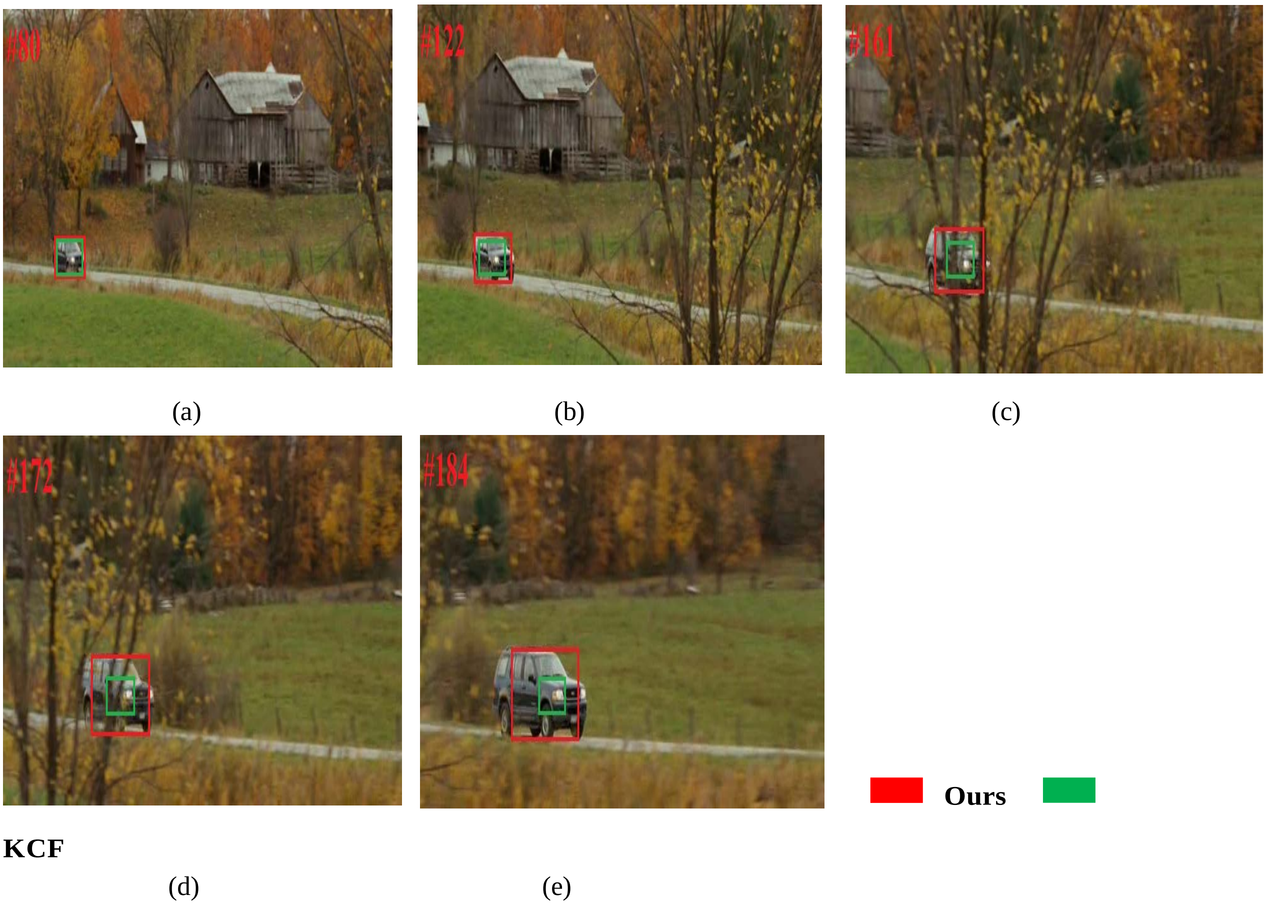


Fig. 6. Comparison of tracking results in the case of scale change (CarScale)

4.3 Comparison with other trackers

Our tracker is also compared with other tracking algorithms including SAMF [6], DSST [8], STRUCK [1] and TLD [2]. The experiments were conducted on all fifty image sequences provided in the OTB database by using the CLE as the evaluation index with a threshold of 20 pixels. These 50 sequences include scenarios such as occlusion, scale change, lighting changes, rotation and fast moving.

Table 2 shows the tracking successful rate of different trackers. SAMF and DSST are also based on KCF but they only handle scale changes and do not have occlusion detection mechanism. Their tracking rates are 77.4% and 74.3% respectively. Our method outperforms these two methods since our method exploits the properties of the conventional KCF and takes both scale change and occlusion into considerations. Compared with the traditional KCF algorithm, the tracking rate of our method is improved by 8.1%. Compared with the classical STRUCK and TLD methods, the tracking rate of our method is improved significantly.

Table 2. Comparison of tracking rate of different trackers using CLE as evaluation index for all OTB-50 image sequences

	SV(scale variation)	OCC(occlusion)	Mean OP
Ours	Yes	Yes	81.9%
SAMF[6]	Yes	No	77.4%
DSST[8]	Yes	No	74.3%
KCF[5]	No	No	73.2%
STRUCK[1]	Yes	Yes	65.6%
TLD[2]	Yes	Yes	60.8%

5．Conclusion

Aiming at the problems that the conventional KCF tracking algorithm has a poor performance in handling scale invariance and occlusion, this paper proposes an occlusion detection scheme and a scaling pool method to achieve anti-occlusion and scale adaptive object tracking. The searching scheme in the KCF is modified. Instead of searching for a target with a fixed sample size, the target area is searched with multiple scales and then resized into the sample size to compare with the learnt model. The scale factor with the maximum filter response is the best target scaling and is also updated as the optimal scale for the following tracking. The correlation filter response peak is combined with the average Peak-to- Correlation Energy (*APCE*) as an occlusion index to determine whether the target is occluded. Once occlusion is detected, the model updating and scale updating are stopped. The proposed method is compared with other state-of-the-art tracking algorithms by using the OTB benchmark video sequences. Experimental results show the proposed method outperforms other methods, can effectively improve the tracking success rate and tracking accuracy in the cases of scale changes and occlusion, and ensure a real-time performance.

Acknowledgments

The authors acknowledge funding support from National Natural Science Foundation of China (Project No. 61374197), and Specialized Research Fund for the Doctoral Program of Higher Education (Project No. 20133120110006).

References

[1] S. Hare, A. Saffari and P.H. Torr, "Struck: Structured output tracking with kernels," in *Proc. of IEEE International Conference on Computer Vision*, vol. 23, pp. 263-270, 2011. Article (CrossRef Link).

[2] Z. Kalal, K. Mikolajczyk and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012. Article (CrossRef Link).

[3] D. S Bolme, J.R Beveridge, B.A Draper and Y.M Lui, "Visual object tracking using adaptive correlation filters," in *Proc. of IEEE Computer Vision and Pattern Recognition*, vol. 119, pp. 2544-2550, 2010. Article (CrossRef Link).

[4] J. F. Henriques, C. Rui, P. Martins and J. Batista, "Exploiting the Circulant Structure of Tracking-by-Detection with Kernels," in *Proc. of European Conference on Computer Vision*, vol. 7575, pp. 702-715, 2012. Article (CrossRef Link).

[5] J. F. Henriques, C. Rui, P. Martins and J. Batista, "High-Speed Tracking with Kernelized Correlation Filters," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 37, no. 3, pp. 583-596, 2015. Article (CrossRef Link).

[6] Y. Li, and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proc. of European Conference on Computer Vision*, vol. 8926, pp. 254-265, 2014. Article (CrossRef Link).

[7] M. Danelljan, G. Häger, F. S. Khan and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. of The British Machine Vision Conference*, pp. 65.1-65.11, 2014. Article (CrossRef Link).

[8] M. Danelljan, G. Häger, F.S. Khan and M. Felsberg, "Discriminative Scale Space Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1561, 2017. Article (CrossRef Link).

[9] Z. F. Pan, Y. L. Zhu, "Kernelized Correlation Filters Object Tracking Method with Multi-Scale Estimation," *Laser&Optoelectronics Progress*, vol. 53, no. 10, pp. 101501-1, 2016. Article (CrossRef Link).

[10] O. Akin, K. Mikolajczyk, "Online Learning and Detection with Part-Based, Circulant Structure," in *Proc. of IEEE International Conference on Pattern Recognition*, pp. 4229-4233, 2014. Article (CrossRef Link).

[11] T. Liu, G. Wang and Q. Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 4902-4912, 2015. Article (CrossRef Link).

[12] C. Ma, X. Yang, C. Zhang and M.H. Yang, "Long-term correlation tracking," in *Proc. of Computer Vision and Pattern Recognition. IEEE*, pp. 5388-5396, 2015. Article (CrossRef Link).

[13] D. D. Yang, Y. Z. Cai, N. Mao and F. C. Yang, "Long-term object tracking based on kernelized correlation filters," *Optucs and Precision Engineering*, vol. 24, no. 8, pp. 2037-2049, 2016. Article (CrossRef Link).

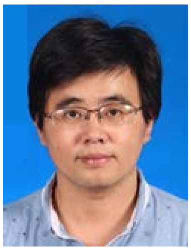
[14] M. Wang, Y. Liu, Z. Huang. "Large Margin Object Tracking with Circulant Feature Maps," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4800-4808**,** 2017. Article (CrossRef Link).

[15] A. Lukezic, T. Vojir, L.C. Zajc, J. Matas and M. Kristan, "Discriminative Correlation Filter with Channel and Spatial Reliability," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp.4847-4856, 2017. Article (CrossRef Link).

[16] Y. Wu, J. Lim, M.H. Yang, "Online object tracking: A benchmark," in *Proc. of IEEE Computer vision and pattern recognition*, vol. 9, no. 4, pp. 2411-2418, 2013. Article (CrossRef Link).

Yingping Huang is a professor at University of Shanghai for Science and Technology, Shanghai, China. His research interests involve Image Processing/Computer Vision, Vehicle Electronics and Intelligent Vehicles.

Chao Ju is a master student at University of Shanghai for Science and Technology, Shanghai, China. His research interests involve Image Processing and Computer Vision.



Xing Hu is a lecturer at University of Shanghai for Science and Technology, Shanghai, China. He received his PhD degree from Shanghai Jiaotong University in 2017. His research interests involve Machine Learning and Computer Vision.

Wenyan Ci is a lecturer at Nanjing Normal University Taizhou Colledge, Taizhou, China. He is also a PhD student at University of Shanghai for Science and Technology, Shanghai, China. His research interests involve Pattern Recognition, Computer Vision and Intelligent Vehicles.