Journal of Computers Vol. 33 No. 2, April 2022, pp. 69-81
doi: 10.53106/199115992022043302006

# An Improved Kernel Correlation Filter Tracking Combined with Mobilenet SSD

Chang-Jian Wang, Yong Ding*, Ye Ji

School of Automation, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, China
{cjwang, dingyong, nuaaconjy}@nuaa.edu.cn

Received 6 June 2021; Revised 1 October 2021; Accepted 1 November 2021

**Abstract.** This article mainly solves the problems that exist when using the Kernel Correlation Filter (KCF) for tracking in complex scenarios. To make the algorithm suitable for target tracking under complex conditions such as scale changes, similar interference, and occlusion, a MobileNet SSD (Single Shot Detection) target detection combined with an improved KCF target tracking algorithm is proposed. Firstly, the MobileNet SSD is used to locate the target in the initial frame, and the location is sent to KCF for training. Secondly, aiming at the problem of scale changes, a Binary-Tree scale search strategy is proposed. In this strategy, the scale value is searched in a tree shape according to the response size, which reduces the number of scale searches. Finally, the average peak correlation energy is used for occlusion determination, and the model update strategy is improved, thereby enhancing the algorithm's ability to track occluded targets. The results of experimental evaluation and comparison on the OTB100 and UAV123 data sets show that when the target has complex conditions such as scale changes, similar interference, occlusion, etc., the proposed algorithm performs well in mainstream related filtering algorithms. Through the quantitative and qualitative analysis of the experimental results, the effectiveness of the proposed algorithm is verified.

**Keywords**: object tracking, KCF, object detection, Binary-Tree, model update strategy

## 1 Introduction

The detection and tracking of moving targets is an important research content in the field of computer vision field and has important applications in civilian and military fields such as intelligent surveillance, military reconnaissance, and human-computer interaction [1, 2]. Target tracking technology can be divided into two categories: generative model and discriminative model [3]. Early target tracking technology mainly used the generative model. Target tracking relied on mathematical modeling of the target object, or extracting manual features of the target, and then searching for similar features in the video frame image to achieve target positioning, such as Optical flow Method [4], Particle filter algorithm [5], Camshift algorithm [6], etc. These generative algorithms usually do not fully utilize the image background information, and a single mathematical model is difficult to fully characterize the constantly changing target in a complex environment. The algorithm has great limitations. Since the discriminative model considers both the target information and the background information, the background information is introduced into the tracking model, and the method of correlation filtering or deep learning is used to distinguish the target and the background, which can better realize the tracking of the target.

KCF tracking algorithm is a typical discriminant method [7]. It extends the features of fHOG [8] based on CSK and uses the diagonalization of the circulant matrix in the frequency domain to greatly reduce the computational complexity of the algorithm and improve the tracking speed. Although the target tracking algorithm based on kernel correlation filtering performs well in terms of speed, the effect of directly processing the video sequence shot under the UAV's perspective is not satisfactory. Especially when there are complex situations such as scale change, deformation, occlusion, etc. of the target in the video taken by the drone, it is easy to fail to track.

In this article, we propose a target tracking algorithm combining MobileNet SSD with the improved KCF, and verify the feasibility of the algorithm through experiments. The experimental results show that the algorithm in this paper has a better effect on target tracking from the perspective of UAVs. Finally, we summarized the technical achievements of this work as follows.

(1) A MobileNet SSD target detection algorithm is used to determine the position of the first frame of the target, and pass it to the tracking network for initialization, which improves the time loss caused by manual labeling.

* Corresponding Author

(2) A binary tree scale search strategy is proposed to select the optimal scale of the target which improves the scale problem in the target tracking process.

(3) A model update method named APCE (Average Peak-to-Correlation Energy) is used to determine the template update when the target is occluded or similar background interference.

## 2. Related Work

Nowadays, the related filtering target tracking technology is a hot research direction in the target tracking field. Most of the initial frame target location is still manually selected by the program executor. The operation of the tracking algorithm needs to rely on the pre-provided initial target position. In response to this problem, Ding [9] et al. proposed to use the SSD target detection algorithm to initialize the target to be tracked in the initial frame; Kwan [10] et al. used the YOLO framework during the tracking. The target detection technology using the deep learning method can accurately locate the initial frame target, but the complex network structure will slow down the computer processing information, and waste too much time in the process of generating the initial frame target. Secondly, when the target has a scale change in the process of being tracked, Martin [11] proposed the DSST algorithm, based on the two-dimensional position filter, the scale filter is introduced to collect 33 scales at the center of the target for similarity comparison, and the maximum response is selected as the best matching scale. The algorithm realizes the self-adaptation to the target scale, but it needs to calculate the response value 33 times for each frame target, which requires a large amount of calculation. Aiming at the occlusion problem in target tracking, Jia-Jia and others [12] combined the mean shift algorithm with the Kalman filter algorithm to effectively predict the target position in the presence of occlusion. Z. Y. Chen et al. [13] built a deep context model to reveal the spatial correlation between the expected surrounding areas of the target to improve the algorithm's ability to track occluded targets. The above algorithms solve the occlusion problem to a certain extent, but they all need to introduce some new modules, which increases the complexity of the network structure.

Inspired by the above, for target tracking in complex background, we propose a target tracking algorithm: the MobileNet SSD combined with the improved KCF(MS-KCF). The main work of this paper includes: 1) Use a lightweight SSD target detection algorithm to determine the initial position of the target in the first frame; 2) Proposes a scaled Binary Tree. The scale search strategy uses the binary tree structure to select the best scale of the target to improve the scale problem in the target tracking process; 3) For the occlusion problem in the target tracking, the tracking result is first determined by the multi-peak value. When there are multiple peaks, using APCE (Average Peak-to-Correlation Energy) to determine the template update when the target is occluded or similar background interference, which improves the tracking accuracy of the algorithm when the target is occluded.

## 3 An Improved KCF Combined with Mobilenet SSD Tracking Algorithm

In this section, we first introduce the overall framework of the algorithm, and then, based on the whole tracking process, divide it into target detection part, scale change part, and occlusion repeat detection part to describe respectively.

### 3.1 An Improved KCF Combined with Mobilenet SSD Tracking Framework

Based on the original KCF algorithm, the MS-KCF algorithm adds the target detection initialization module, the scale prediction part, and the occlusion re-detection mechanism. The overall network framework is shown in Fig. 1. The algorithm is composed of two parts. The first part is the target detection module. The initial frame is input into the mobilenet SSD network for feature extraction, and the initial position of target $(x, y, w, h)$ is passed to KCF for initialization operation. The initialization operation mainly includes adding a cosine window to reduce the image smoothness caused by the boundary effect and adding padding to the target window. Then constructing the circulant matrix and using position filter to forecast the best location of the target. The scale filter is used near the location to select the optimal scale. Finally, through the mechanism of the model, judge whether meet the update condition. If not, the target detection is initialized again till the end of the tracking.

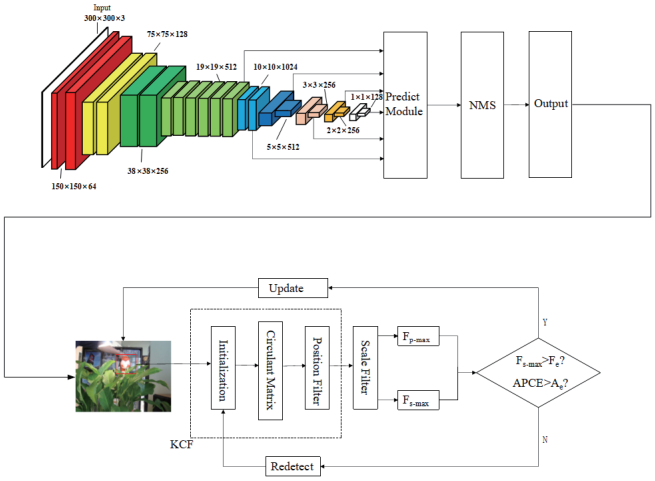



**Fig. 1.** MS-KCF algorithm flow chart

### 3.2 Mobilenet SSD Target Detection

Aiming at the problem of obtaining the initial target frame of the related filtering algorithm, consider using the target detection algorithm to detect the initial video frame object, and input the coordinate information into the KCF as the initial frame training sample. In order to be effectively combined with the KCF algorithm framework, we select the MobileNet SSD [14] as the target detection framework. The MobileNet SSD simplifies the structure of SSD [15] to reduce the number of training parameters.

#### 3.2.1 SSD Object Detection

The SSD target detection algorithm is based on the VGG16 [16]. The last three fully connected layers are replaced by the convolutional layers. The candidate frame mechanism of the Faster R-CNN [17] target detection algorithm is used for reference. SSD carries out predictive regression by using candidate boxes with different aspect ratios on different feature maps to ensure the features that participate in the final classification come from the feature maps of multiple scales. The algorithm can effectively predict targets at different scales. Assume that feature maps from m convolutional layers are used for prediction, then the size of the *k*-th feature graph is:

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m-1}(k-1)\ k \in \{1,2,...,m\}. \tag{1}$$

In the formula, $s_{\min}$, $s_{\max}$ represent the minimum and maximum proportions of candidate boxes in the feature map. The height and width of each candidate frame are:

$$w_k^a = s_k\sqrt{a_t}, \tag{2}$$

$$h_k^a = s_k / \sqrt{a_t}, \tag{3}$$

where, $s_k$ represents the size of the kth feature map, $a_t$ represents the aspect ratio of candidate frame, $a_t \in \{1,2,\frac{1}{2},3,\frac{1}{3}\}$. Subsequently, the SSD selects six feature layer images of conv4_3, conv7, conv8_2, conv9_2, conv10_2, and conv11_2 for the final detection and classification. However, the SSD algorithm using the VGG16 has a large amount of data calculation, and it is more difficult to apply it to the target tracking algorithm.

### 3.2.2 MobileNet SSD

Andrew et al. used mobilenet as the basic network architecture to replace the VGG16 in the traditional SSD network. This paper uses this lightweight target detection network to detect and provide more accurate initial target information for KCF initialization training.

The depth separable convolution block is used for feature extraction, and the process is shown as Fig. 2.

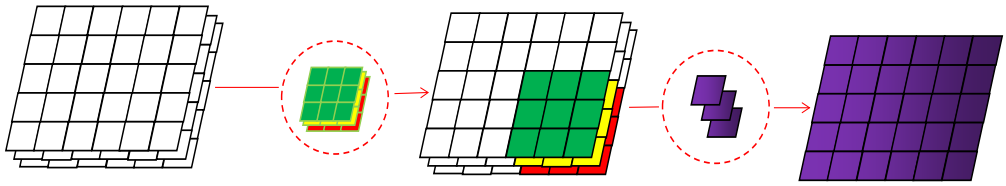


**Fig. 2.** Depth-wise separable convolutions

The input image is first extracted with a 3*3 filter, and the information between different channels is merged through 1*1 convolution on the feature map generated after extraction. Standard convolutions have the computational cost of:

$$D_K \times D_K \times M \times N \times D_F \times D_F, \tag{4}$$

where the computational cost depends multiplicatively on the number of input channels $M$, the output channels $N$, the kernel size $Dk \times Dk$ and the feature map size $D_F \times D_F$. Depthwise convolution has a computational cost of:

$$D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F. \tag{5}$$

The ratio of calculation amount is:

$$\frac{D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F}{D_K \times D_K \times M \times N \times D_F \times D_F} = \frac{1}{N} + \frac{1}{D_K^2}. \tag{6}$$

When the 3*3 convolution kernel is used, the calculation amount of the depth separable convolution is about 11.1% of the standard convolution. Through the classification and regression of the effective feature layer information, the position of the target in the first frame of the KCF algorithm is determined.

## 3.3 Binary Tree Scale Search Strategy

### 3.3.1 KCF Object Tracking Algorithm

The KCF algorithm trains the classifier in the process of tracking the target and uses the classifier to identify the specific position of the target in the next frame. The algorithm introduces multi-channel HOG features that are more robust to illumination changes based on CSK. KCF uses the kernel function to map the linear space ridge regression problem to the nonlinear space, which solves the dual problem in the nonlinear space and some Common constraints. KCF utilizes the diagonalization property of the circulant matrix in Fourier space, transforms the matrix operation into the point multiplication operation of the elements. It reduces the amount of calculation and improves the real-time performance of the algorithm. The working principle of the KCF algorithm is as follows.

Let the training set be $\{(x_i, x_y), i = 1, 2, ..., n\}$, the corresponding linear regression function is $f(x_i) = w^T x_i$, $w$ represents the weight, solved by the least square method and expressed as a matrix form:

$$\min \left\| Xw - y \right\|^2 + \lambda \left\| w \right\|^2, \tag{7}$$

$X = [x_1, x_2, ..., x_n]^T$, each row is represented as a sample vector, $y$ is the label vector, the minimizer has a closed-form:

$$w = (X^H X + \lambda I)^{-1} X^H y. \tag{8}$$

Since the corresponding solution formula of Equation (8) includes inverse matrix operation, to simplify the operation process, KCF adopts a circulant matrix for sample collection. The circulant matrix C(X) is shown as Fig. 3.

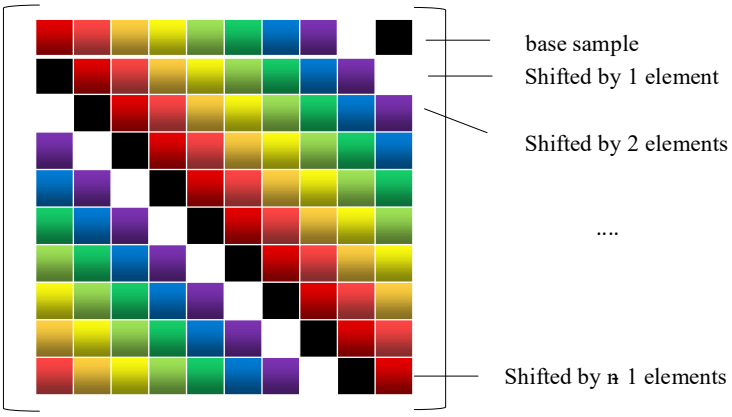



**Fig. 3.** Circulant matrix

All circulant matrices can be diagonalized in Fourier space:

$$X = F diag(\hat{x}) F^H, \tag{9}$$

$\hat{x}$ denotes the DFT of the generating vector, $\hat{x}=F(x)=\sqrt{n}Fx$, $F$ is the constant matrix.

Then the formula (8) could be:

$$\begin{aligned} w &= (F diag(\hat{x}^*) F^H F diag(\hat{x}) F^H + \lambda F^H)^{-1} F diag(\hat{x}^*) F^H y \\ &= (F diag(\hat{x}^* \odot \hat{x} + \lambda) F^H)^{-1} F diag(\hat{x}^*) F^H y \\ &= F diag(\frac{\hat{x}^*}{\hat{x}^* \odot \hat{x} + \lambda}) F^H y \end{aligned}. \tag{10}$$

Taking the Fourier transform on both sides, the result is:

$$\hat{w} = \frac{\hat{x} \odot \hat{y}}{\hat{x}^* \odot \hat{x} + \lambda}. \tag{11}$$

For non-linear situations, KCF introduces a kernel function to perform ridge regression on the sample in high-dimensional kernel space and solves the following objective function:

$$\alpha = (K + \lambda I)^{-1} y. \tag{12}$$

Defined $K=\phi(X)\phi(X)^T$ as the kernel matrix of the kernel space, $\phi(X)$ denotes non-linear mapping function. Using the properties of the circulant matrix, the equation (12) can be solved to obtain:

$$\alpha = \frac{\hat{y}}{(\hat{K}^{xx} + \lambda)^*}, \tag{13}$$

where $\hat{K}^{xx}$ denotes the first row of the kernel matrix $K=\phi(X)\phi(X)^T$ . The optimal position of the target is shown by the following formula:

$$\hat{f}(z) = (\hat{K}^{XZ}) \bullet \hat{\alpha}. \tag{14}$$

KCF algorithm introduces circulant matrix and Gaussian kernel function, it optimizes the calculation problem in target tracking, transforms matrix inversion operation into numerical multiplication, and uses multi-channel HOG feature as tracking feature for the first time, which performs well in tracking. However, the KCF algorithm does not have a good solution to the problem of scale change in the target tracking process.

### 3.3.2 Scale Binary- Tree

Aiming at the scale problem of the kernel-related filtering tracking algorithm, this paper uses the sample response value as the criterion to classify the scale in a tree, as shown in Fig. 4. It can be seen from the figure that the scale is scaled at the locked target position in the previous frame, and the target in the next frame is cropped according to the specified scale ratio. The scale of the child node with a larger Gaussian response is selected as the scale to be selected, and the scale is selected layer by layer. And finally, select the node with the largest response as the best scale.

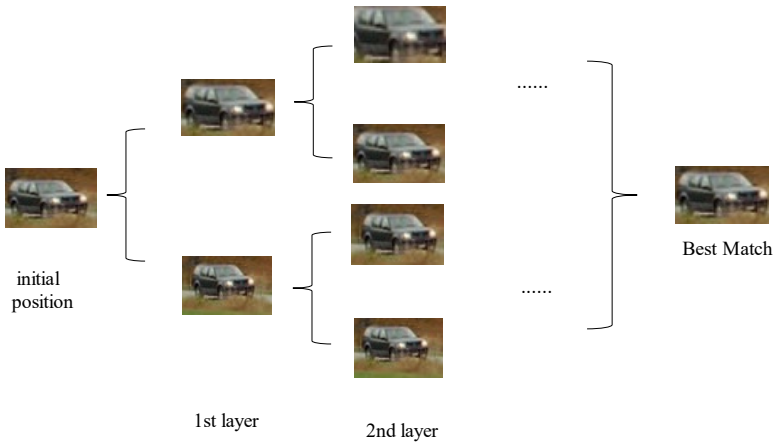



**Fig. 4.** Scale Binary-Tree

Assuming that the initial size of the tracking target in the t-1 frame is *S*, the maximum value of the target response $\max \hat{f}_t(z)$ in the *t*-th frame is calculated by the formula (14); then, select $S_t = \sigma_{i,j} \bullet S$ , where i represents the *i*-th level of the scale tree, $j \in \{a,b\}$ ,where a and b respectively represent the two sub-branches of the *i*-th level of the scale tree. The selected scale values meet the following conditions:

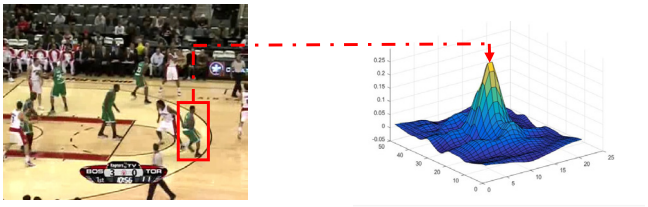
$$\sigma_{i,j} = \max \hat{f}(z)_S \; i = \{1,2,3,4,5\}; j = \{a,b\}. \tag{15}$$

When expanding (shrinking) the candidate area, the maximum target response value is greater than the original response value, and the larger-scale branch is selected; if the maximum target response value is less than the original response value, the smaller-scale branch is selected until the best scale is selected. Through the binary tree of scales, the comparison can be performed at most 5 times, which reduces the number of operations and improves the efficiency of the algorithm.

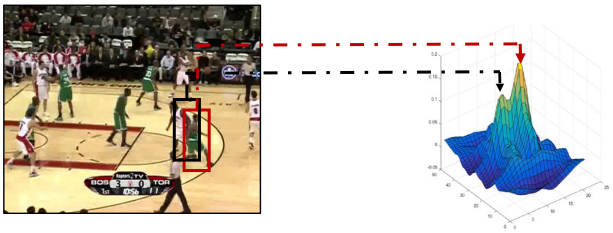
### 3.4 Occlusion Re-detection Based on High Confidence

During the target tracking process, when the target is occluded, the position filter is susceptible to interference from similar backgrounds and thus learns the wrong target information. At this time, updating the template will lead to the failure of tracking. The algorithm in this paper does not use the conventional template update strategy per frame but adds a model update mechanism.

Fig. 5 is a schematic diagram of Gaussian response with multiple peaks. It can be seen from Fig 5(a) that when the target is successfully tracked, the Gaussian response graph shows a single peak; when the target has interference, it can be seen from Fig 5(b) As a result, the Gaussian response graph will have multiple peaks.



(a) Without occlusion



(b) Under occlusion

**Fig. 5.** Schematic diagram of Gaussian response peak

To ensure that the template is updated under the premise of successful tracking, this paper adopts the high-confidence update-index APCE as the judgment condition, and the template is updated only when the high-confidence is satisfied. This indicator can reflect the peak level and fluctuation degree of the response graph. The specific calculation method is:

$$APCE = \frac{\left|F_{\max} - F_{\min}\right|^2}{mean\left[\sum_{W,H}\left(F_{W,H} - F_{\min}\right)^2\right]}, \tag{16}$$

where, $F_{\max}$, $F_{\min}$ respectively indicate the maximum and minimum values on the response graph, $F_{W,H}$ denotes the response value at the position $(W,H)$, $mean[\bullet]$ denotes an average value operation. It can be seen that when the peak is sharper and the response fluctuation is smaller, the APCE is larger.When interference or occlusion occurs, the APCE value will decrease sharply. To improve the accuracy of the template, the following update methods are used:

$$\hat{\alpha}_t = \theta[(1-\eta S_{\max})\hat{\alpha}_{t-1} + \eta S_{\max}\hat{\alpha}], \tag{17}$$

where , $\hat{\alpha}_{t-1}$ denotes the filter model in time t-1 , $\eta$ is the model update rate, $\theta$ is the discriminant factor, and the value of $\theta$ is as follows:

$$\theta=\begin{cases}1 & APCE>\lambda_1\overline{APCE},\ F_{\max}>\lambda_2\overline{F_{\max}} \\ 0 & else\end{cases}, \tag{18}$$

where $\overline{APCE}$ denotes the historical average of the average peak energy ratio, $\overline{F_{\max}}$ is the historical average of the maximum value of the response, $\lambda_1$ , $\lambda_2$ denotes the preset ratio factor.

### 3.5 The MS-KCF Algorithm

**Algorithm 1.** Target tracking algorithm based on MS-KCF

Step 1. Input the first image of the sequence to be tracked and initialize the parameters;

Step 2. Use the Mobiletnet SSD target detection algorithm to perform target detection on the input image, and pass the detected image location information $(x, y, w, h)$ to the KCF algorithm for filter training;

Step 3. Extract HOG features, train position filter;

Step 4. Select the best-matched scale at the position of the maximum response in the next frame;

$$\sigma_{i,j}=\max \hat{f}(z)_S \ i=\{1,2,3,4,5\};\ j=\{a,b\}.$$

Step 5. Calculate the APCE value of the target response curve, and save the APCE value and maximum response value of the response;

$$APCE=\frac{\left|F_{\max}-F_{\min}\right|^2}{mean\left[\sum_{W,H}\left(F_{W,H}-F_{\min}\right)^2\right]}.$$

Step 6. Determine whether the current frame response APCE value sum and $F_{\max}$ is greater than its historical average in the ratio of $\lambda_1$=0.45 , $\lambda_2$=0.35 , if it is greater than the historical average value, the template will be updated;

Step 7. If the template is not updated for 5 consecutive frames, return to Step1 to re-detect and initialize the target;

Step 8. Repeat Step 2~Step 7 until the trace is all over.

## 4 Experimental Results and Analysis

In order to effectively verify the performance of the algorithm proposed in this paper, the algorithm is simulated and verified. The comparison algorithms selected in the experiment all come from the codes published by the corresponding authors. The experiment was completed on Intel(R)Core(TM)i5-9400F CPU@2.9GHZ, 6GB memory, NVIDIA GeForce GTX 1660 desktop computer, and the algorithm was implemented by Pycharm, CUDA9.0, CUDNN7.0. The experiment is described from three aspects: tracking effect evaluation, center position error, and overall tracking performance verification of the algorithm.

### 4.1 Experimental Parameters

The data sets used in this experiment are all from OTB100 and UAV123, and the algorithm initialization parameter configuration is shown in Table 1.

**Table 1.** Algorithm initialization parameter configuration

| Parameter | Value |
|---|---|
| Regularization factor $\lambda$ | 0.01 |
| Minimum ratio of candidate frames $s_{\min}$ | 0.2 |
| Maximum ratio of candidate frames $s_{\max}$ | 0.9 |
| Model update rate $\eta$ | 0.01 |
| Scale step | 0.01 |
| Area expansion factor *padding* | 1.5 |

### 4.2 Related Trackers Comparison

The proposed tracker algorithm was compared with some state-of-the-art approaches:

**DSST.** This approach proposed a tracking method that accurately measures the target scale in tracking. This precise scale estimation method can be combined with any other tracking algorithm without scale estimation.

**KCF.** This approach introduced the kernel matrix and multi-channel feature processing. It speeded up the algorithm operation speed and the robustness of extracting features.

**SAMF.** This approach used the fusion of HOG, CN, and Gray features, and utilized multi-scale technology.

**LMCF.** This approach proposed a structured SVM with stronger judgment ability as the classifier and introduced the multi-peak forward detection technology to solve the situation of similar object interference during the tracking process.

Fig. 6. shows the tracking block diagram results of the algorithm in this paper, MS-KCF, KCF, DSST, SAMF, and LMCF. Different color image boxes represent different algorithms. We will compare and analyze the algorithm from four aspects: target scale change, target occlusion, in-plane rotation, and small target.

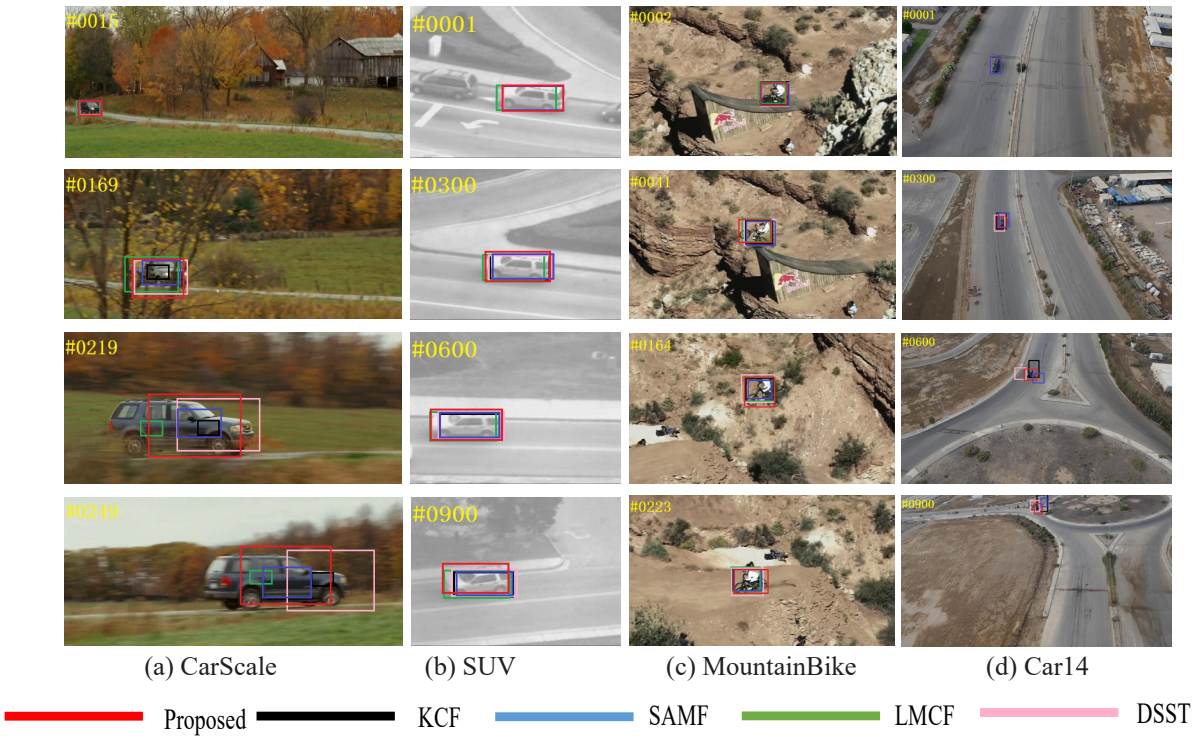



**Fig. 6.** Tracking contrast diagram

(1) **Change of target scale.** In the CARSCALE sequence in Fig. 6(a), the scale of the target in the image gradually increases from the far to the near of the black SUV car. The KCF algorithm does not include the scale change part, and the tracking box size remains unchanged, so the tracking effect is poor. The algorithm in this paper can adapt to the scale change well.

(2) **Target occlusion.** In the SUV sequence in Fig. 6(b), the target appears occlusion phenomenon. Due to the short occlusion time, all algorithms can track the target, but the target positioning is not completely accurate. The positioning range of the algorithm in this paper includes the white roof around frame 900, while the other algorithms do not.

(3) **Target rotation and similar background jamming.** In Fig. 6(c) MountainBike data set, the target appears in the air rotation action, and there is similar background interference, KCF, LMCF, DSST, SAMF all appear within a certain range of tracking fluctuations, the algorithm in this paper for the target location including the wheel and other details.

(4) **Small target tracking.** In the sequence of Fig. 6(d) CAR14, targets shot from the perspective of UAV account for a small proportion of the overall image, and the KCF and DSST algorithms learn less information, which is prone to tracking failure. The improved algorithm in this paper is able to locate targets more accurately.

To further analyze and explain the tracking effect, the center position error curve is drawn for the tracking effect in Fig. 6(a), Fig. 6(b), and Fig. 6(c). The center position error can reflect the error value between the predicted target position and the real target position, as shown in Fig. 7.

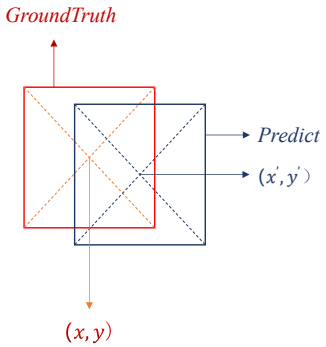
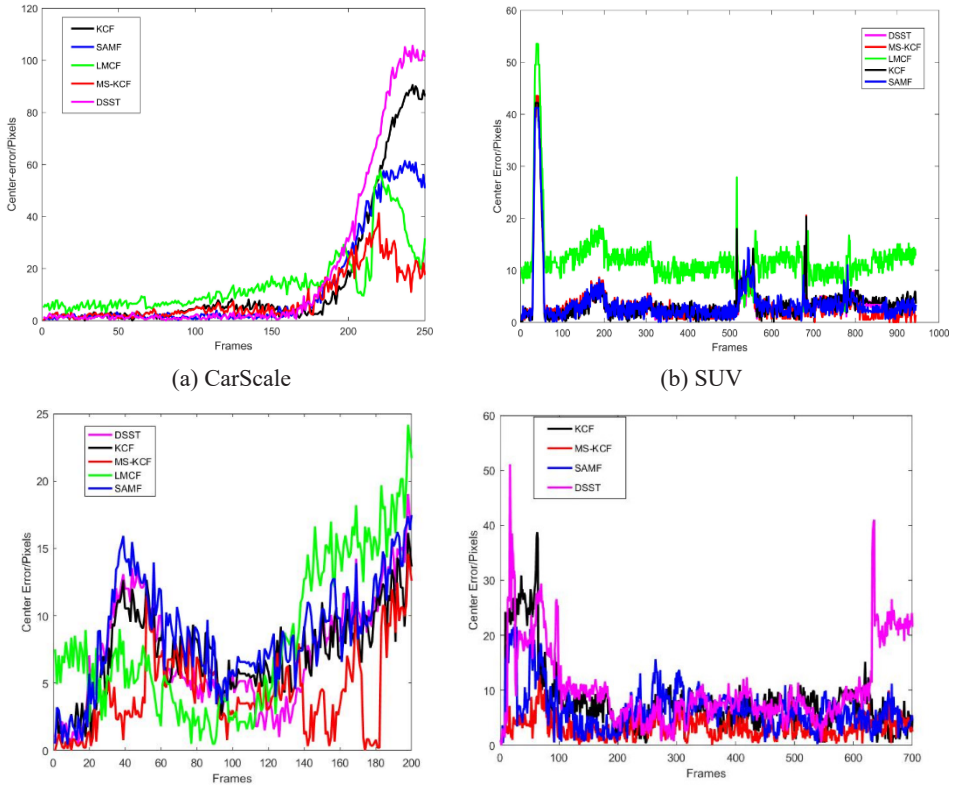



**Fig. 7.** Diagram of center position error

The center position error is defined as follows:

$$error = \sqrt{(x - x')^2 + (y - y')^2}. \tag{19}$$

Fig. 8 shows the comparison of center position errors of KCF, DSST, LMCF, SAMF, and MS-KCF algorithms. It can be seen from the figure that when the scale of the target changes greatly in the middle and late CARSCALE sequence, the algorithm in this paper can track the target well and its error is relatively low. For the SUV data set and CAR14 data set, the overall central error of the algorithm in this paper is relatively low, always below 20, which is more stable than other algorithms. On the MountainBike data set, the center position error of the proposed algorithm is significantly lower than that of other algorithms, which indicates that the proposed algorithm can achieve more accurate positioning of the target with tracking, and the algorithm has better performance.



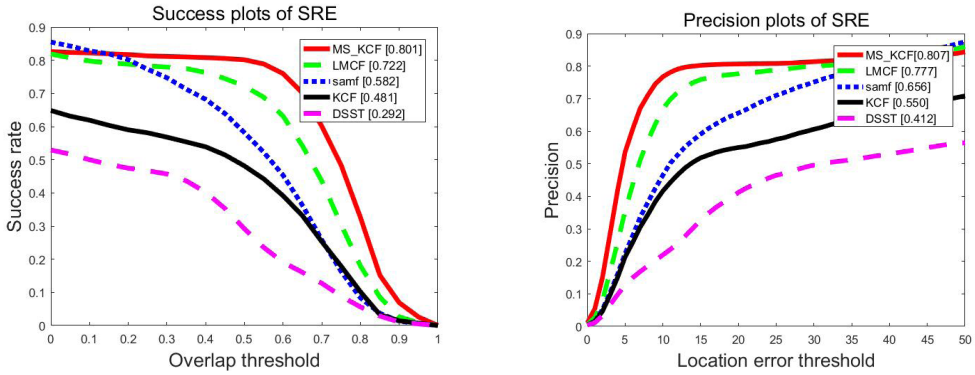

(a) CarScale (b) SUV

(c) MountainBike (d) Car14

**Fig. 8.** Center position error comparison

### 4.3 Verification of Overall Tracking Performance of MS-KCF in Complex Scenarios

To fully explain the tracking performance of the algorithm proposed in this paper, the accuracy curve and success rate curve of the OTB platform were used as evaluation indexes to carry out quantitative analysis of the algorithm proposed in this paper. The accuracy curve is the percentage of video frames that are more than a given threshold. Different thresholds allow different accuracy to be achieved to plot a curve. The Success rate curve reflects that the crossover ratio between the tracking box and the truth box obtained by the algorithm in the tracking process is greater than the percentage of the part of the set threshold value. Different Success rates can be obtained under different thresholds to draw a curve.




(a) Success (b) Precision

**Fig. 9.** The success and precision curves of the algorithm

In addition, in different situations, the tracking accuracy and success rate statistics of different algorithms are shown in Table 2 and Table 3 respectively.

**Table 2.** Precison of the performace

| Trackers | Background-cluster | Scale variation | Fast motion | Occlusion |
|---|---|---|---|---|
| Proposed | **0.718** | 0.683 | **0.653** | **0.705** |
| LMCF | 0.679 | 0.678 | 0.631 | 0.679 |
| SAMF | 0.686 | **0.701** | 0.626 | 0.704 |
| KCF | 0.611 | 0.634 | 0.579 | 0.626 |
| DSST | 0.629 | 0.605 | 0.538 | 0.609 |

**Table 3.** Success of the performace

| Trackers | Background-cluster | Scale variation | Fast motion | Occlusion |
|---|---|---|---|---|
| Proposed | **0.616** | **0.627** | **0.614** | 0.656 |
| LMCF | 0.605 | 0.612 | **0.614** | **0.659** |
| SAMF | 0.597 | 0.546 | 0.586 | 0.592 |
| KCF | 0.483 | 0.519 | 0.530 | 0.496 |
| DSST | 0.556 | 0.448 | 0.533 | 0.567 |

As can be seen from Table 2 and Table 3, when there are situations such as occlusion and scale changes in the tracking process, the precision and success rate of the algorithm in this paper are relatively excellent. The proposed algorithm can meet the needs of target tracking.

## 5 Conclusion

This paper presents a target tracking algorithm based on the Mobile-Net SSD combined with improved KCF. Firstly, the Mobile-Net SSD target detection algorithm is used to detect the image target, and the results are transmitted to the tracking algorithm for initialization. Secondly, a scale filter based on binary tree search strategy is added to the traditional position filter to estimate the target scale optimally. Finally, the APCE update strategy is adopted when the model is updated, and the target model is updated only when the maximum response peak value and APCE value are greater than their historical mean value in a certain proportion respectively. By comparing the proposed algorithm with some mainstream algorithms in the field of correlation filtering, the feasibility and effectiveness of the proposed algorithm are proved. In future work, the program can combine some hardware devices to perform some target detection tasks in the context of drones.

## References

[1] J. Janousek, P. Marcon; J. Pokorny; J. Mikulka, Detection and Tracking of Moving UAVs,in: Proc. 2019 PhotonIcs & Electromagnetics Research Symposium - Spring (PIERS-Spring), 2019.
[2] M. Archana, A. Ayyasamy, Tracking based Event Detection of Singles Broadcast Tennis Video, in: Proc. 2018 3rd International Conference on Communication and Electronics Systems (ICCES), 2018.
[3] Y. Ruan, Z. Wei, Discriminative descriptors for object tracking, Journal of Visual Communication & Image Representation 35(2016) 146-154.
[4] T. Brox, J. Malik, Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation, IEEE Transactions on Pattern Analysis & Machine Intelligence 33(3)(2011) 500-513.
[5] B.A. Delail, H. Bhaskar; M.J. Zemerly; M. Al-Mualla, Robust Likelihood Model for Illumination Invariance in Particle Filtering, IEEE Transactions on Circuits & Systems for Video Technology 28(10)(2017) 2836-2848.
[6] L. Li, Y. Luo, Improved Video Moving Target Tracking Based on Camshift, American Journal of Computational Mathematics 6(4)(2016) 357-364.
[7] J.F. Henriques, R. Caseiro, P. Martins, J. Batista, High-Speed Tracking with Kernelized Correlation Filters, IEEE Transactions on Pattern Analysis & Machine Intelligence 37(3)(2015) 583-596.
[8] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan, Object Detection with Discriminatively Trained Part-Based Models, IEEE Transactions on Pattern Analysis & Machine Intelligence 32(9)(2010) 1627-1645.
[9] L. Ding, X. Xu, Y. Cao, G. Zhai, F. Yang, L. Qian, Detection and tracking of infrared small target by jointly using SSD and pipeline filter, Digital Signal Processing 110(2021) 102949.
[10]C. Kwan, B. Chou, J. Yang, A. Rangamani, T. Tran, J. Zhang, R. Etienne-Cummings, Deep Learning-Based Target Tracking and Classification for Low Quality Videos Using Coded Aperture Cameras, Sensors 19(17)(2019) 3702.

[11]M. Danelljan, G. Hager, F.S. Khan, M. Felsberg, Accurate Scale Estimation for Robust Visual Tracking, in: Proc. 2014 British Machine Vision Conference, 2014.
[12]J.-J. Wu, J. Gao, M. Li, H.-H. Xu, Research on Target Tracking Algorithm Based on Mean Shift and Kalman Filter, in: Proc. 2014 Computer Technology and Development, 2014.
[13]Z.Y. Chen, L. Luo, D.F. Huang, M. Wen, C.Y. Zhang, Exploiting a depth context model in visual tracking with correlation filter, Frontiers of Information Technology & Electronic Engineering 18(5)(2017) 667-679.
[14]A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. <https://arxiv.org/abs/1704.04861>, 2017 (accessed 17.04.17).
[15]W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A.C. Berg, SSD: Single Shot MultiBox Detector, in: Proc. 2016 European Conference on Computer Vision, 2016.
[16]K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, in: Proc. 2015 Computer Science, 2015.
[17]S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, IEEE Transactions on Pattern Analysis & Machine Intelligence 39(6)(2017) 1137-1149.