# Drifting resistant algorithm for video target tracking based on Kernelized Correlation Filters framework

Zili Shan[a], Xuan Zhang[*b], Pengfei Zhai[a], Shuo Liang[a], Jinyong Chen[a]
[a]The 54th Research Institute of China Electronics Technology Group Corporation Shijiazhuang, China; [b]Hebei Open University, Shijiazhuang, China
*Email: makerel998@163.com

## ABSTRACT

Video target tracking has a wide range of application value in the field of automatic driving, UAV target tracking, security monitoring, etc. How to maintain stable tracking of the target among video data frames is the focus of the research. A robust tracking algorithm that effectively solves the target drift problem is proposed for the problem of target loss due to image perturbation, scale change, target occlusion and other disturbances when the KCF algorithm is used for video target tracking. The algorithm is based on the KCF algorithm framework, which proposes a multi-scale sampling strategy and designs a multiple classifier screening algorithm to ensure the accuracy of the target template. Through experimental verification, the algorithm can effectively solve the drift problem in the tracking process and realize the continuous accurate tracking of the target. The algorithm provides a real-world application reference for engineering applications of real-time video data processing.

**Keywords:** DR-KCF, video target tracking, multi-scale sampling, target drift

## 1. INTRODUCTION

Intelligent transportation, intelligent security, intelligent manufacturing and other fields all have a rigid demand for the robustness of video target tracking, through the analysis and processing of video data, to obtain the position of the moving target in the video between consecutive frames, to form a complete motion trajectory of the target in the video. The core requirement of robust video target tracking, the changing illumination, complex background, object occlusion, and target scale change of the video scene pose a challenge to the stable tracking of video targets. The design of tracking algorithm, feature extraction algorithm affect the robustness of video target tracking. The feature extraction algorithm extracts the unique features that can distinguish the target from other objects from the rich information of the target, and the tracking algorithm takes the features of the target as the processing object, and searches for the target with the maximal similarity feature according to a specific strategy between consecutive frames of the video, and obtains the position of the target. Therefore, designing appropriate feature extraction algorithms and tracking algorithms is the basis for realizing accurate and stable tracking of video targets. In terms of tracking algorithms, according to the different theoretical basis of their algorithms, they can be mainly divided into the following categories:

### 1.1 Tracking algorithm based on kernel density estimation theory

Among such algorithms, the Mean Shift (MS) algorithm is the most representative, which belongs to a probabilistic density gradient function non-parametric estimation method to find the target position through iterative operations, and was first applied to video target tracking by Comaniciu et al. [1] Its most significant advantage is that it is small in computation and easy to implement, but the tracking effect is poor for small or fast-moving targets. In response to these problems, a Cam-Shift (continuous adaptive mean shift) algorithm was developed. Chen et al. [2] added Kalman filtering to the Cam Shift algorithm framework, which increased the accuracy while maintaining the tracking speed. Fiyad et al. [3] constructed a single-target tracking system for video data with low computational load by utilizing an advanced Kalman filter and Cam-Shift algorithm. Tracking algorithms based on kernel density estimation theory require the introduction of mean-drift variables from the analytic form, which restricts feature extraction and has some limitations.

### 1.2 Tracking algorithms based on probabilistic statistics

Tracking algorithms based on probabilistic statistics construct a state space model to transform the target tracking problem into an estimation problem of the target state. Among such tracking algorithms, the more widely used ones are Kalman filtering, extended Kalman filtering, traceless Kalman filtering, and particle filtering. Wang et al. [4] combined

International Conference on Automation and Intelligent Technology (ICAIT 2024), edited by R. Usubamatov, S. Feng, X. Mei, Proc. of SPIE Vol. 13401, 134010L · © The Authors. Published under a Creative Commons Attribution CC-BY 3.0 License
doi: 10.1117/12.3050251

extended Kalman filtering and Monte Carlo filtering to achieve target tracking. Amin et al. [5] used an extended Kalman Filter (EKF) to solve the target collision problem and Particle Swarm Optimization (PSO) to reduce the covariance of the measurement noise, resulting in a video multi-target tracking accuracy of 98%. The tracking algorithm based on probabilistic statistics has a good performance in solving the nonlinear problem, which brings complex computation and easily affects the tracking speed.

### 1.3 Machine learning based tracking algorithms

Some researchers provided an overview of state-of-the-art applications in machine learning [6], for instance, Cheng et al. [7] proposed a novel digital twin fire model using ROM techniques and deep learning prediction networks to improve the efficiency of global wildfire predictions. Machine learning based tracking algorithms transform the tracking problem into a target classification problem, where the classifier learns the features of the target and the background, realizes the distinction between the target and the background, and obtains the position of the target in each frame of the image. In the early days, most of the classifiers used the offline learning mode, which requires a large number of training samples to support and has poor adaptability. The online learning mode utilizes the first frame of video data for training, the subsequent video frames extract a number of samples as candidate targets based on the target position in the previous frame, and then the classifier selects the most similar samples from the candidate targets as the target. Zhang et al. [8] utilized the theory of compressed perception, extracted the features with sparse matrices, and trained a plain Bayesian classifier to differentiate between the target and the background to achieve a fast tracking. Kalal et al. [9] proposed Tracking Learning Detection, which can track the target for a long time by constraining the positive and negative samples by an online structure. Henriques et al. [10] used correlation filters as a tracker based on the Fast Fourier Transform and proposed Kernelized Correlation Filters, which transform the convolution calculation in the time domain to the multiplication operation in the frequency domain, greatly improving the tracking speed. Sun et al. [11] proposed a target tracking algorithm for hyper-spectral low-altitude UAV video, using YOLOv5 to detect the coordinate information of the UAV target in the current frame, and using the KCF algorithm, using the kernel correlation filter for the target in the current frame. The KCF algorithm is used to track the target in the current frame using kernel correlation filtering, and more satisfactory results are obtained.

Overall, machine learning-based tracking algorithms consume less computational resources and are suitable for realizing video target tracking on end devices such as UAVs and robots. Among them, the KCF algorithm has excellent performance in coping with both illumination changes and scale changes, and can accurately track the target even under complex backgrounds, but it is prone to the problem of target position drift due to template updating in the tracking process. In this paper, to address the problem of target loss after the target encounters occlusion, we design a multi-scale sampling strategy and a multiple classifier filtering module, and propose a robust tracking algorithm capable of resisting drift (Drifting Resistant KCF, DR-KCF), which is able to ensure the accuracy of the target template, thus enhancing the stability and fault-tolerance of target tracking.

## 2. THE KCF ALGORITHMIC FRAMEWORK

The KCF algorithm is based on the iterative learning of ridge regression classifiers to achieve the extraction of targets in each frame. Using the cyclic matrix theory, the target region of the current frame of the video is densely sampled to obtain positive and negative samples, and its Histogram of Oriented Gradients (HOG) features are extracted, in which the information of the target region of the current frame of the video is determined by the classification result of the Ridge Regression Classifier on the previous frame of the video. If the input is the first frame of the video, the target region is sampled directly, and the HOG features are extracted, and then the parameters of the ridge regression classifier are trained as the ridge regression classifier for extracting the target location in the second frame of the video, and the specific flowchart is shown in Fig. 1.

### 2.1 Dense sampling

The thick sampling strategy for each frame of the video is shown in Fig. 2. The image surrounded by the yellow rectangular box is the target region, also referred to as the base sample, which can be denoted as $x = (x_i, x_{i+1} ... x_{i+n})$, The result of moving the samples from the base sample in the left and right directions, respectively, after the move $x = (x_{i-j}, x_{i-j+1} ... x_{i-j+n})$ and $x = (x_{i+j}, x_{i+j+1} ... x_{i+j+n})$ .Cycling in the horizontal and vertical directions in this way, sampling all possible image blocks around the base sample, a complete sample space covering the real target area can be obtained.

### 2.2 Ridge regression classifier

A linear ridge regression classifier is trained with the goal of finding a function $\min_w \sum_i (f(x_i)-y_i)^2+\lambda\|w\|^2$ that minimizes a loss function:

$$\min_w \sum_i (f(x_i)-y_i)^2+\lambda \| w \|^2 \tag{1}$$

where $x=(x_1,x_2...x_n)$ denotes the sample, $\lambda$ is used as a regular term to prevent over-fitting, and $w$ is the desired parameter. A closed solution can be found from equation (1):

$$w=(X^T X+\lambda I)^{-1} X^T y \tag{2}$$

where the $X$ matrix is the cyclic matrix of the combination of base samples and y denotes the set of labeled values for each sample. For those samples that cannot be classified in the original space, it is necessary to introduce the kernel function, which maps the linearly indivisible patterns in the low-dimensional space to the high-dimensional space to achieve linear separability through the kernel function, which is of the following form:

$$k(x,z)=\varphi(x)\varphi(z) \tag{3}$$

where $k(x,z)$ is the kernel function and $\varphi(x)$ and $\varphi(z)$ are the mapping functions from the low-dimensional space to the high-dimensional space.

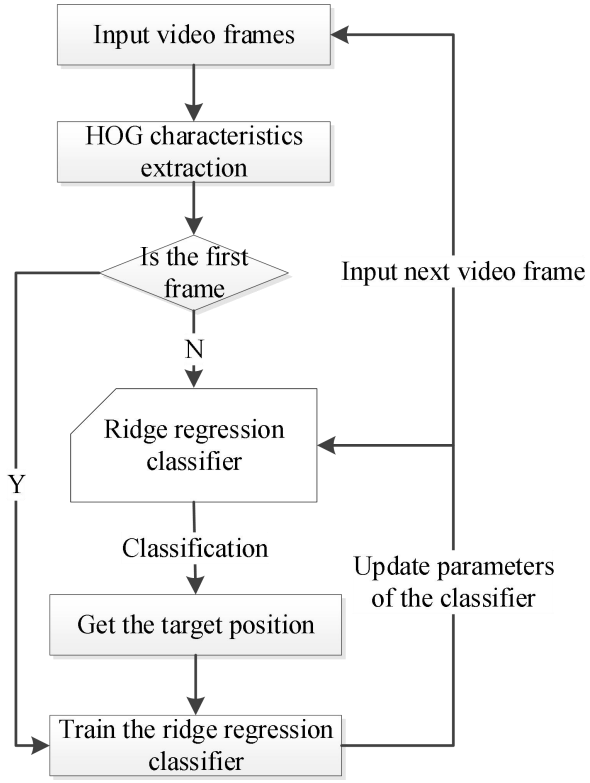


Figure 1. Flowchart of KCF algorithm

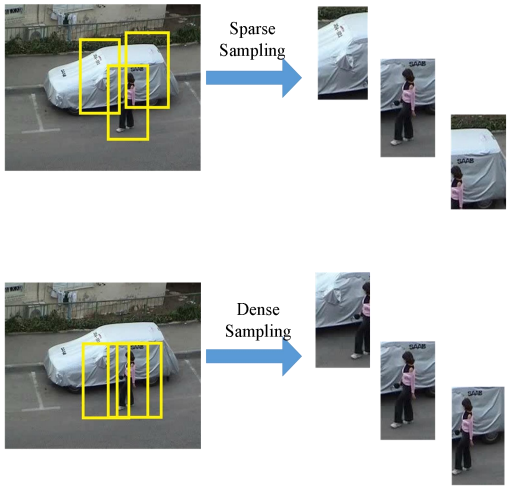


Figure 2. An example of dense sampling

When the kernel function is used to map the samples $x=(x_1,x_2...x_n)$ to $\varphi(x)$, the coefficients $w$ in the desired classifier $f(x)=w^Tx$ are transformed into $a$ in the pairwise space. From the Representer Theorem, the coefficients $w$ are linear combinations of the samples $x$.

$$w=\sum_i a_i\varphi(x_x) \tag{4}$$

$$f(x)=\sum_i a_i k(x,x_i) \tag{5}$$

It is obtained jointly from Eq. (1) and Eq. (4):

$$a=(K+\lambda I)^{-1}y \tag{6}$$

where $K$ is the kernel matrix after mapping $K_{i,j}=k(x_i,x_j)$. In the detection phase, the input video is classified by the ridge regression classifier to determine the final target location information, and the obtained target location information is continued to be used for the training of the ridge regression classifier, and so on, to realize the continuous tracking of the target.

## 3. DRIFTING RESISTANT KCF ALGORITHM (DR-KCF)

The KCF video target tracking algorithm, although more accurate and faster in processing, is prone to lose the target when the target is in the video if there is an occlusion. When the occlusion is removed, it is also impossible to retrieve the target again to continue tracking. To address this problem, this paper designs a screening module and fuses it with the position information of the target before it is occluded for fusion processing, which is used to improve the ability of the KCF algorithm to resist target occlusion, and its flow is shown in Figure 3. The input video frame is first sampled at multiple scales, and the sampled samples are fed into the screening module, which consists of three classifiers, namely, variance screening, cascade decision tree screening and template screening, and the samples of the input video frames may become the target region only through the three screenings of the classifiers. The specific process is shown in Figure 4.

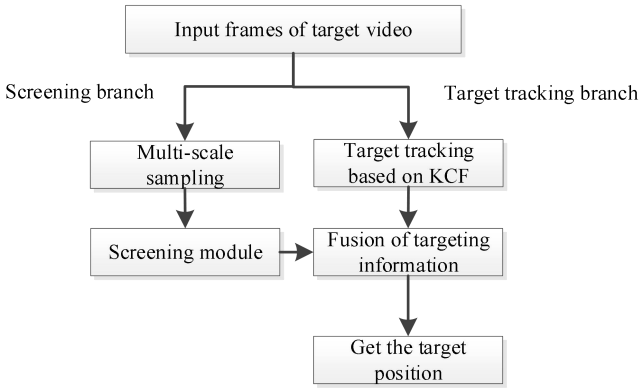


Figure 3. Flowchart of improved KCF based target tracking algorithm

### 3.1 Multi-scale sampling strategy

Triple filtering is performed to filter the image blocks in a video frame to remove those that do not contain the target. To make the sampling contain as much information as possible, multi-scale sampling is performed. In this paper, the initial target bracket box is used as the base, and 10 scale windows are enlarged and shrunk respectively with a scaling ratio of 1.2 times each time to form 21 samples in the scale space (including 1 initial bracket box scale and scaled 20 scales), and the scanning window is smoothly moved in the video image with a step size of 10% of the width or height to generate rectangular boxes of different scales. For an image of size 240*320, after sampling with the above strategy, about 50,000 image block samples can be generated, containing almost all the information of the image. For all the samples of

different scales, the positive and negative samples are determined and labeled by the size of the overlap between each rectangular box and the target rectangular box, using the target enclosing box as a benchmark.

### 3.2 Variance screening

If the gray value variance of the image block to be detected is greater than 50% of the gray value variance in the target region to be tracked, the image block $I$ passes the screening. The gray value variance of all pixel points in the image block can be represented by equation $E(I^2)-E^2(I)$ , wherein a gray scale histogram is utilized to obtain $E(I)$ . After screening the variance of the image block to be detected, about half of the image blocks that are not related to the target are discarded.

### 3.3 Cascade decision tree screening

The image blocks filtered by variance are further filtered by a cascading decision tree, consisting of 10 decision trees connected in series. Each decision tree $i$ with a decision cascade of $n$ compares pairs of pixel points at $n$ specific locations in the input image block, and the result of the comparison is converted into an $n$ -bit binary code $x$ , as shown in Fig. 5. Each binary code x corresponds to a posterior probability $P_i(y|x)$ , where $y \in (0,1)$ . Since the decision trees are independent of each other, the average of the posterior probabilities of the individual decision trees is representative of the filtering results of the cascade decision tree. A threshold is set for this, if the average of the calculated the posterior probability is greater than this threshold, the image block is considered to contain the target and passes the screening, otherwise, it does not pass the screening.

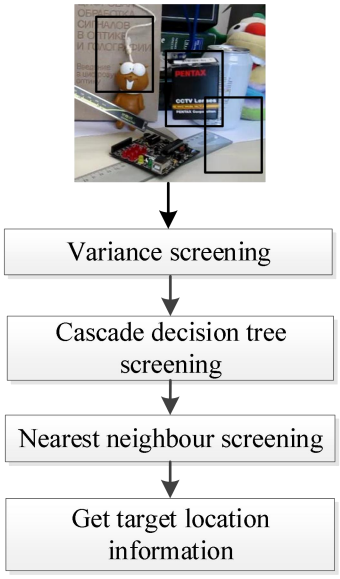


Figure 4. Flowchart of the detecting module

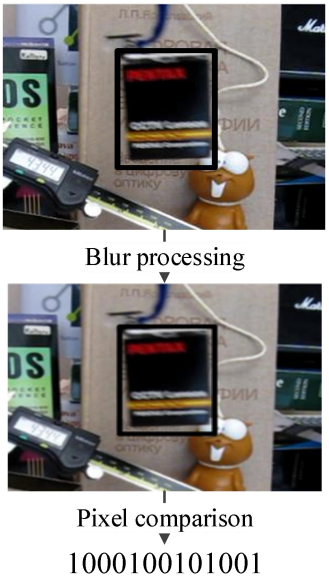


Figure 5. Conversion from picture block to binary code

In this stage, the positions of the pixel points to be compared for each decision tree are randomly determined before the tracking starts and remain constant during the tracking process. In order to ensure the stability of the image pixels so that accurate comparisons can be made, the image to be tracked is first pre-processed with Gaussian convolution to reduce the noise in the image and increase the robustness of the image. Then the differences of the pixel pairs are compared in the image block according to the pre-determined pixel positions, and each pixel pair returns 0 or 1 for the difference of the comparison, and the sequential concatenation of these return values completes the binary coding.

The mutual independence of the decision trees in this process is a very critical factor, therefore, in order to ensure the independence of the decision trees, the following measures are taken: firstly, the image is normalized, then the pixel point positions of the image are discretized, based on which all the pixel point pairs that can be compared are determined in the horizontal and vertical directions, and finally, the comparisons of these pixel pairs are rearranged and assigned to

the individual decision tree. In this way, each decision tree is able to ensure that the comparisons of pixel pairs are not duplicated, thus ensuring the independence of the decision trees. At the same time, all these pairs of pixels cover the entire image block, and the result of the comparison is able to represent a feature of the image block.

Each decision tree corresponds to a posterior probability $P_i(y \mid x)$. Since there are $n$ layers of decision trees, then each decision tree generates an eigenvalue ($n$-bit binary encoding) corresponding to $2^n$ posterior probabilities. The posterior probability is calculated as follows:

$$P_i(y \mid x) = \frac{P}{P+N} \tag{7}$$

Where $P$ and $N$ denote the number of positive and negative samples for the same feature value. In the initial stage, each a posterior probability value is 0. During the tracking process, each decision tree is trained by the labeled positive and negative samples, and then the posterior probabilities are constantly updated. It is experimentally verified that when $n$=13, all the pixel points can cover the whole image with good results.

**3.4 Nearest neighbour screening**

After the first two stages of screening, there will be few remaining image blocks. In this stage, the similarity between the remaining image block $I$ and the image block in the existing target template library $T$ is calculated, and the similarity is determined by the following normalized correlation coefficient.

$$S(I_i, I_j) = 0.5(NCC(I_i, I_j) + 1) \tag{8}$$

Only misclassified samples are added to the template library $T$, which represents the latest target samples obtained so far as well as the background samples $T = \{I_1^+, I_2^+ \cdots I_1^-, I_2^- \cdots\}$ .

Define $S^+(I,T) = \max_{I_i^+ \in T} S(I, I_i^+)$ , $S^-(I,T) = \max_{I_i^- \in T} S(I, I_i^-)$ . The relevant similarity between the candidate image block and the template library is:

$$S^r = \frac{S^+}{S^+ + S^-} \tag{9}$$

Where $S^r \in (0,1)$ , the closer the correlation similarity is to 1, the more likely it is that the image block contains the target. If the correlation similarity between an image block and an existing template is greater than a given threshold, i.e., $S^r(I,T) > \theta_{th}$ , the image block passes the screening and is considered to contain the target.

**3.5 Fusion of target information**

The number of image blocks that completely pass the triple screening may be $\{0,1,n\}, n > 1$ . Different strategies are designed for each of these three different cases, and this result is fused with the tracking result to obtain the latest position of the target, and the position is used as a benchmark for training, and the position of the target is constantly updated to realize the continuous and accurate tracking of the target. As shown in Fig. 3.

(a) When both the tracking branch and the screening branch can get the position of the target (the screening branch may get more than one target position) and the results of these two target positions are relatively close to each other (the overlap rate of the target rectangle box is large), then the results obtained by the screening branch will be averaged first, and the multiple target positions will be combined into a single target position, which will then be compared with the target position obtained by the tracking branch and that obtained by the screening branch by a weight of 10:1 weighting for weighted averaging, and finally get a new target position.

(b) When the difference between the target position obtained by the tracking branch and the screening branch is large, the confidence judgment is performed on the results of the screening branch, and when the results obtained after

screening are more accurate, the results of the screening branch are used as the basis for re-initializing the target position. Otherwise, the tracking branch results prevail to continue tracking.

(c) When the target position cannot be obtained after screening, take the result of the tracking branch as the final position of the target.

## 4. EXPERIMENTAL RESULTS AND ANALYSES

The experiment compares the tracking results of the DR-KCF algorithm of this paper with the KCF algorithm under a variety of conditions, and the yellow and red boxes correspond to the tracking results of the KCF algorithm and the DR-KCF algorithm, respectively.

Fig. 6 shows the tracking results for the video without interference. At frame 70, both algorithms track the target accurately. At frame 71, both algorithms show different degrees of target drift. At frames 74 through 90, the KCF algorithm still tracked the wrong target, while the DR-KCF algorithm recaptured the target automatically and continued to track it accurately due to the assistance of the screening algorithm.

Figure 7 shows the tracking results in the case of a slight disturbance in the video. At frame 10, both algorithms track accurately. At frame 15, the video picture quality is not clear, causing both algorithms to lose the target. At frame 17, the KCF algorithm drifts and tracks the wrong target, while the DR-KCF algorithm recaptures the correct target. Until frame 40, the KCF algorithm still tracks the wrong target and the DR-KCF algorithm is able to track the target accurately.

Figure 8 shows the video of a rigid body target in the case of encountering a stationary object occlusion. At frame 424, both algorithms lose track of the book because the object occludes the book and the complete book features cannot be acquired. At frame 426, the KCF algorithm then suffers from a drift problem, locking the target to the background, while the DR-KCF algorithm recaptures the target. Until frame 450, the KCF algorithm still fails to track the target correctly, while the DR-KCF algorithm is able to track the target accurately.

Figure 9 shows the video in the case where a non-rigid target encounters occlusion by a moving object. At frame 91, both algorithms are able to track the target accurately before the pedestrian occludes the target. At frames 95 and 96, the pedestrians pass by the target and occlude the target to some extent, the KCF algorithm is disturbed to some extent, and the tracking rectangle box becomes larger, while the DR-KCF undergoes a brief offset due to the occlusion of the pedestrians, but quickly recaptures the target again. At the 120th frame, as the video progresses deeper, the tracking rectangle box of the KCF algorithm becomes larger and larger, containing a large amount of non-target information, while the DR-KCF algorithm, which recaptures the target, is able to track the target accurately.

Fig. 10 shows the video of the case where a non-rigid target encounters occlusion by a stationary object. At frame 373, the rectangular frames of the two algorithms show differences, but both are able to track the target accurately. At frames 380 to 395, the tracking result of the KCF algorithm is shifted to the occluded object, while the DR-KCF algorithm is still able to track the target accurately.

Figure 6. Target tracking results without disruptions（$70^{th}$、$71^{th}$、$74^{th}$ and $90^{th}$ frame)

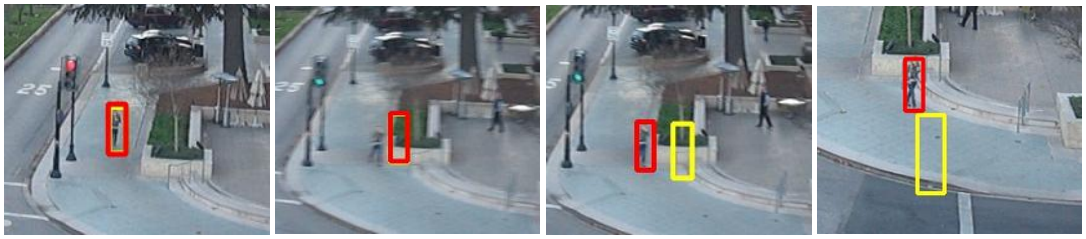

Figure 7. Target tracking results of slight perturbation（10th、15th、17th and 40th frame)

Figure 8. Tracking results of rigid target in occlusion（420th、424th、426th and 450th frame)

Figure 9. Tracking results of non-rigid target cross-talk（91th、95th、96th and 120th frame)

Figure 10. Tracking results of non-rigid target in occlusion（310th、373th、380th and 395th frame)

## 5. CONCLUSION

In this paper, on the basis of KCF algorithm, a robust algorithm that can effectively solve the drift problem in tracking is proposed. Firstly, the video is sampled at multiple scales, then three sets of classifiers are introduced to filter the collected samples, and finally fused with the tracking results to jointly obtain the position of the target. After experimental comparison and analysis, the DR-KCF algorithm is able to cope with the drifting problem occurring in the target, has the ability to recapture the target, and greatly improves the tracking accuracy. The DR-KCF algorithm improves the robustness through the multi-scale sampling and screening strategy, which leads to a rise in computation due to the need to process more samples, while UAV video data or autonomous vehicles have high requirements for real-time processing, and for this kind of data, the variance screening module and the cascade decision tree screening module can be handed over to the GPU for parallel processing, while the other modules are still in the CPU, which can realize the real-time processing of streaming video. Further research will concentrate on this topic.

## ACKNOWLEDGMENTS

Projects Financed by the National Ministries Fund(2022-JCJQ-ZD-124-00) and Hebei Science and Technology Innovation Project (SJMYF202313).

## REFERENCES

[1] Comaniciu D, Ramesh V, Meer P. Real-time tracking of non-rigid objects using mean shift[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2000:142-149.
[2] Chen X, Wu H, Li X. Real-time visual object tracking via camshift based robust framework[J]. International Journal of Fuzzy Systems, 2012,2(14): 264-271.
[3] Fiyad, Metwally, Hamid M. B, El-Hameed, Abozied, Mohammed. Improved real time target tracking system based on cam-shift and Kalman filtering techniques[J]. Journal of Applied Research and Technology, 2023,2(21):297-308.
[4] Wang H J, Jia Z R, Yan Y. Target tracking algorithm based on improved extend Kalman particle filter[J]. Application Research of Computers,2011,5(28):1634-1643.
[5] Amin Jahantighy, Hamed Torabi, Farahnaz Mohanna. Multiple targets video tracking based on extended kalman filter in combination with particle swarm optimization for intelligent applications[J]. SN Applied Sciences, 2023,3(5):1-14.
[6] Cheng S, et. Machine learning with data assimilation and uncertainty quantification for dynamical systems: a review[J]. IEEE/CAA Journal of AUTOMATICA SINICA, 2023,10(6):1361-1387.
[7] Zhong C, Cheng S, et. Reduced-order digital twin and latent data assimilation for global wildfire prediction[J]. Natural Hazards and Earth System Sciences, 2023,23(5):1755-1768.
[8] Zhang K, Zhang L, Yang M H. Real-time compressive tracking[C]. European Conference on Computer Vision,2012:866-879.
[9] Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012,34(7):1409-1422.
[10] Henriques J F, Caseiro R, Martins P, et al. High-Speed Tracking with Kernelized Correlation Filters[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015,37(3):583-596.
[11] Sun Haodong, Ma Pengge, Li Zhenghao, Ye Zhaoyi, Ma Yueran. Hyperspectral low altitude UAV target tracking algorithm based on deep learning and improved KCF[J]. Frontiers in Physics,2024,12(2):01-16.