



Improved Camshift object tracking algorithm in occluded scenes based on AKAZE and Kalman

Lili Pei¹ · He Zhang² · Bo Yang³

Received: 14 November 2020 / Revised: 16 September 2021 / Accepted: 7 October 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

Camshift algorithm tracking is susceptible to interference when a tracking object is occluded or when its hue is similar to the background. An improved Camshift object-tracking algorithm combining AKAZE (Accelerated-KAZE) feature matching and Kalman filtering is proposed. First, the video channel is converted for processing. Second, AKAZE is used to match the object feature points and Kalman filtering is used to predict the next position. Then different scenes are judged by the threshold and the Camshift and Kalman tracking algorithms are used for object tracking, respectively. Finally, the improved Camshift algorithm is used to test the moving object in a variety of situations and compared with the traditional Camshift algorithm and the Kalman filter improved Camshift algorithm. Experimental results show that the improved joint tracking algorithm can continue tracking under full occlusion. The effective frame rate of recognition is increased by about 20%, and the single-frame image processing time is less than 35 ms, which can meet the real-time tracking requirements.

Keywords AKAZE algorithm · Camshift algorithm · Feature matching · Kalman filtering · Object tracking · Video processing

1 Introduction

In the last two years, the COVID-19 has been raging around the world. There is an urgent need to improve the ability of monitoring public travel and trajectories while using surveillance video object tracking to find close contacts to reduce the risk of large-scale infections. Improving the accuracy of the object-tracking algorithm plays a key role in completing these tasks. Simultaneously, the application of the object-tracking algorithm in the security

✉ Lili Pei
peilili@chd.edu.cn

¹ School of Information Engineering, Chang'an University, Xi'an 710064, Shaanxi, China

² Xi'an Xiangteng Micro-Electronics Technology Co., Ltd., Xi'an 710068, Shaanxi, China

³ Datang Mobile Communication Equipment Co., Ltd. Xi'an Branch, Xi'an 710061, Shaanxi, China

field can monitor areas with large traffic and complex public security environments, which is of great significance for fighting crime and preventing emergencies.

Object tracking algorithms include two categories. The first involves generative methods and the second involves discriminative methods [36]. The main idea of the generative method is to select the moving object to be tracked at the initial moment and obtain its characteristic parameters. Then, in the subsequent image sequence, an area similar to the initial object is found as the moving object area, so as to achieve continuous and effective tracking of the object. Representative methods in generative methods include the particle filter algorithm [27], the Kalman filter algorithm [1], and the mean shift algorithm [16]. These algorithms perform well in object tracking in front of simple backgrounds. However, when the colors are similar or the scene is changed or occluded, the complete and accurate tracking of the moving object cannot be completed. Typical discriminant methods include TLD (Tracking Learning Detection) [15], CNN (Convolutional Neural Network) [29], and KCF (Kernel Correlation Filter) [6]. Some of these methods are based on deep learning, which require model training on a large number of samples. In actual object detection and tracking, the object is often uncertain. Therefore, it is difficult to use the pre-trained network of deep learning to continuously track it under full occlusion. And lots of KCF methods are based on hand-craft features, which will waste time.

Many scholars have improved traditional algorithms to improve the adaptive ability of parameters and make the algorithms adaptive in complex environments. Cheng [31] uses the Kalman filter to predict the position of moving objects and the Camshift algorithm is used to adjust the position and size of the search window through prediction, which effectively overcomes the detection interference caused by color and improves the accuracy and robustness of the object tracking. Vasif [20] uses the AKAZE algorithm to reconstruct banknote fragments, effectively solving the problem of fragment matching and fragment splicing. Banknote fragments are successfully synthesized. Zhang [2] uses the adaptive Gaussian mixture model as the background model and combines the Kalman filter with the Camshift algorithm to effectively retrieve useful video information. Gao [5] obtains color and texture histograms through multi-feature fusion in object tracking and proposes a new algorithm based on the Camshift algorithm, which effectively overcomes object interference in complex environments to achieve object tracking. Zhang [23] proposed a tracking method based on the Camshift algorithm related to the probability map. Through the circular arrangement of pixels in the inverse probability projection map, the object location and tracking are realized.

In terms of moving object tracking technology, great achievements have been made after decades of rapid development [22, 30, 34]. In order to combat crime and terrorist attacks and protect the personal and property safety of citizens, the US Defense Advanced Research Projects Agency invited Carnegie Mellon University and Massachusetts Institute of Technology to jointly develop the VSAM (Visual Surveillance and Monitor) system [11]. Xu [28] et al. conducted in-depth research on particle filter algorithms and proposed an improved particle filter algorithm, which combines the reversible jumping mechanism with the Markov chain Monte Carlo (MCMC) and uses a global estimation algorithm to track. The model parameters of the object are modified to achieve accurate object tracking. Zhang [35] proposed a moving object-tracking algorithm using spatio-temporal context information. The algorithm needs to train the tracked object in advance to obtain a object model, which is used in subsequent object tracking to determine the location of the object in the scene. Cao [4] et al. proposed an improved adaptive particle filter algorithm, which can have high robustness. Li [14] used convolutional neural networks and support vector machines to combine the external contours of moving objects and various background

modeling methods. The algorithm has an online structured output function. Lee [13] et al. applied infrared sensing technology to object tracking. This method has good performance in tracking small objects.

There have also been breakthroughs in some two-frame image processing technologies in recent years [24, 25]. Liao [26] et al. proposed a new method based on CNN for forensic detection of a chain made of two image operators. This method can automatically learn operation detection features directly from image data. The robustness of the proposed method is studied in two scenarios.

According to the inspiration of the above algorithm improvement, this paper analyzes the performance of the Camshift algorithm, the Kalman filter algorithm, and the AKAZE feature-matching algorithm. On the basis of the original Camshift algorithm, the improved Camshift object-tracking algorithm in complex backgrounds based on AKAZE and Kalman is proposed. The main contributions are as follows:

- 1) Through the AKAZE feature-matching algorithm, the position and size of the tracking object in the current image sequence can be obtained, which ensures that the algorithm can relocate the moving object when the tracked object is lost. At the same time, when the moving object is completely occluded, the Kalman algorithm can realize the prediction of the foreground object position, thereby obtaining the repositioning of the object. In general, the proposed algorithm can continuously track a completely occluded moving object in a complex background.
- 2) Through testing the improved algorithm and comparing with the original algorithm, it is found that the improved joint-tracking algorithm detects an effective frame rate increase of about 30%. The single-frame image processing time is less than 35 ms, which meets the real-time tracking requirements.

2 Methodology

2.1 Camshift object tracking algorithm

The mean shift tracking algorithm was first proposed by Fukunaga in the 1970s [21]. It is a kind of density clustering, which uses continuous iterative operations to search for the position of the sample point with the largest probability density in the feature space. The search direction always shifts toward the direction where the density of sample points increases the most and the trend of the moving object is analyzed and judged according to this data, so as to track the moving object. The above process is shown in Fig. 1.

The disadvantage of the mean shift algorithm is that the detection rectangle cannot be changed adaptively with the size of the object. When the distance between the object and the camera changes or there is a slight disturbance from the outside, it is easy to lose track of or mistrack the object. The Camshift algorithm was created in order to resolve the shortcomings of the mean shift algorithm [3]. The Camshift tracking algorithm uses the mean shift object tracking algorithm for each frame of the video sequence. The tracking result of each image determines the initial value of the tracking iteration in the next image and the tracking window will automatically change as the object changes.

The specific algorithm implementation steps are as follows:

- 1) Initialize the search window.

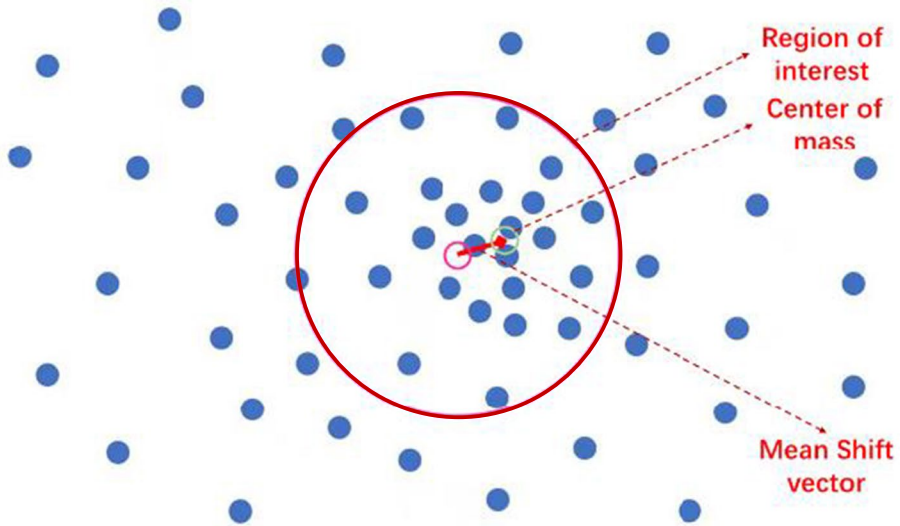


Fig. 1 Mean shift tracking algorithm

Select the moving object to be tracked and calibrate the initial search window, which solely contains the entire tracking object. Simultaneously, the image must be converted to the HSV (Hue, Saturation, Value) color space because the RGB (Red, Green, Blue) model is more sensitive to lighting changes, which will reduce the impact of lighting.

2) Determine the color probability distribution of the window.

Extract the hue channel in the video image sequence from the selected window to obtain the object object histogram model and the color probability lookup table. Inquire the object color histogram model for each pixel in the subsequent video sequence to determine the probability that the pixel is the object pixel. Finally, back project the image [12].

- 3) Run the mean shift algorithm to get the size, position and angle of the new search window.
- 4) Use the value in step 3 to reinitialize the size and position of the search window in the next frame of the video. Use the new search window and repeat starting from step 2.

2.2 AKAZE Feature matching algorithm

In 2012, French scholars proposed a new KAZE algorithm for feature point detection and description at the ECCV conference. The feature detection of KAZE is similar to Scale Invariant Feature Transform [7] (SIFT), but it is more stable in terms of feature detection. The feature point detection is realized by constructing a nonlinear scale space and more details in the image are preserved. The KAZE algorithm mainly includes the following four steps:

- 1) Construct the KAZE algorithm scale space.

Constructing the scale space is a necessary basic link to realize the KAZE algorithm, which is realized by combining the additive splitting algorithm (ASO) with the nonlinear spreading filter.

2) Detection and precise positioning of feature points.

(1) Feature point detection

The KAZE feature detection algorithm needs to construct the Hessian matrix and find the extreme points of the Hessian determinant to determine the feature points in the image [19].

The determination of extreme points is shown in Fig. 2. By comparing successively adjacent scale windows with the current scale window to find the pixels in each window and all its neighboring pixels, the maximum value is the extreme point, and a 3×3 window is used.

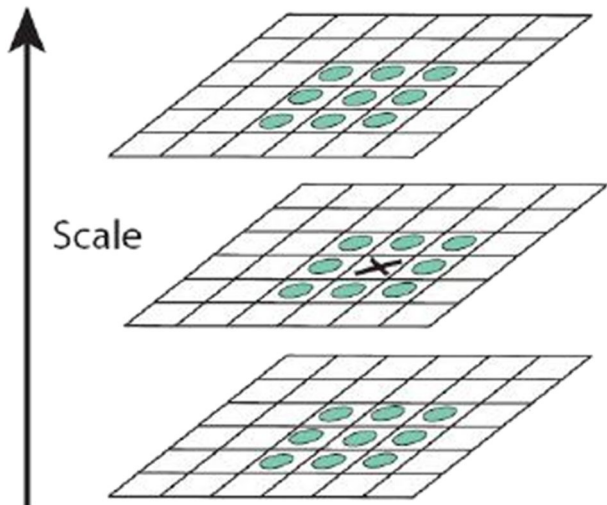
(2) Precise positioning of feature points

After finding the position of the feature point, the precise positioning of the subpixel is carried out. The method adopted is the method proposed by Lowe in BMVC2002 [17].

3) Determine the main direction of the feature points

If the scale parameter of the feature point is σ_i , the search radius is set to $6\sigma_i$. Make a 60° fan-shaped area in this circular area, and count the sum of Haar wavelet features in the fan-shaped area [33]. Then rotate the fan-shaped area, and then count the sum of wavelet features. The direction with the largest sum of wavelet features is the main direction. The main direction of the feature is determined as shown in Fig. 3.

Fig. 2 Determination of the extremum point



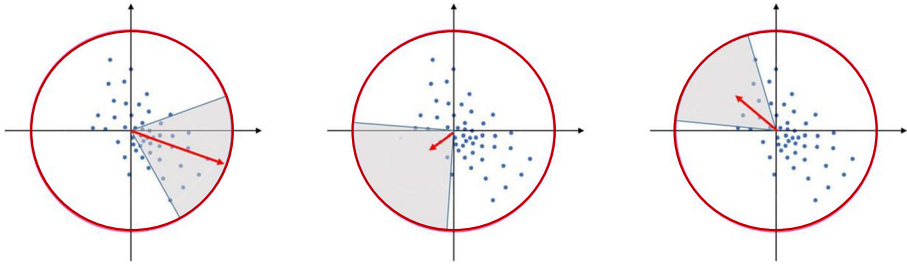


Fig. 3 Features the main direction of the point

4) Descriptor generation

For the feature point with the scale parameter σ_i , a window of $24\sigma_i \times 24\sigma_i$ is taken on the gradient image with the feature point as the center, and the window is divided into 4×4 subregions. Each sub-region has a size of $9\sigma_i \times 9\sigma_i$, and adjacent subregions have overlapping bands with a width of $2\sigma_i$. Each subregion is weighted with a Gaussian kernel, and a subregion description vector of length 4 is calculated:

The vector of each subregion is weighted through another Gaussian window with a size of 4×4 and finally normalized.

The disadvantage of the KAZE algorithm is its high time consumption, so instead the AKAZE algorithm [32] is proposed, which reduces the time consumption of the algorithm while ensuring the robustness of matching.

The improvements of the AKAZE algorithm over the KAZE algorithm mainly pertains to the following two aspects:

- 1) When constructing the nonlinear scale space, the KAZE algorithm adopts the AOS algorithm to calculate the diffusion equation. Although this method is stable, the solution is time-consuming. The AKAZE algorithm adopts the Fast Display Diffusion (FED) algorithm for fast solving. The scale space of the KAZE algorithm is constructed by nonlinear interpolation, while the AKAZE algorithm uses the image pyramid method, which can speed up the extraction of feature points.
- 2) The KAZE algorithm uses MSURF descriptors. The process of obtaining the MSURF descriptors is to obtain the local gradients of feature points, which is time-consuming. The AKAZE algorithm uses the LDB descriptor (binary descriptor), which is faster. At the same time, the LDB descriptor is further improved to obtain the M-LDB descriptor, which has better support for rotation robustness.

2.3 Kalman filtering algorithm

The Kalman filter algorithm is an optimal recursive filter algorithm that uses the minimum mean square error estimation for the state sequence of the linear system [10, 18], which consists of a prediction part and a parameter update part. First, the prediction mechanism is used to predict the position information of the moving object in the current image in the next frame. Then the predicted new object position is provided to the system for updated parameters. Finally, the system feeds back the updated parameters to the prediction mechanism and continuously predicts parameter updates.

The prediction stage is used to predict the state value and the minimum mean square error. Since the Kalman filtering algorithm has a memory effect, the state value and error estimated at the previous time must be used to predict the state value and error at the next time. In the update phase, the recursive idea is mainly used to update and correct the movement state of the object at the next moment.

The Kalman filtering algorithm has the advantages of high accuracy, low computational complexity, and strong real-time performance. However, when the object has irregular movement and cross-occlusion, the Kalman filter will diverge, which will lead to the failure of moving object tracking.

3 Improved Camshift algorithm based on Kalman and AKAZE

When the contrast between the object color and the background color is minimal or the object is severely occluded, the probability distribution map of the Camshift algorithm through the back projection of the object histogram is prone to a large range of diffusion, which will lead to the rapid expansion of the search box and ultimately lead to tracking failure. The AKAZE algorithm does not depend on the color characteristics of the object. Instead, through the feature matching between the image motion in the video sequence and the object template, so that the two establish a one-to-one correspondence, the position information of the moving object is obtained. Combining the AKAZE algorithm and the Camshift algorithm can solve the interference of the background color on the moving object. Simultaneously, when the object is severely occluded, the Kalman filter algorithm is used to prejudge the position of the tracked object in the image at the next moment. Then the Camshift algorithm is used to search for the object and adjust the size and position of the search box, so that the tracking is not lost after the tracking object is occluded.

Due to the complex environment of moving objects, the problem of occlusion often occurs. In the Camshift algorithm, the color histogram of the tracked object model is compared with the object color histogram in the current video sequence. The similarity between the two histograms is used as the basis for judging whether the object is occluded [8].

Assume that H_{obj} is the color histogram of the object model, H_{sce} represents the color histogram of the object in the current video sequence, and the value of $D(H_{obj}, H_{sce})$ is used to measure the similarity. The value range of $D(H_{obj}, H_{sce})$ is [0, 1]. The smaller the value, the more similar the object histogram in the video sequence is to the object model and the accurate tracking of the object can be achieved. Use the preset thresholds T_{th} and $D(H_{obj}, H_{sce})$ to judge whether the object is blocked.

The similarity is measured by Bhattacharyya distance and its calculation formula is as follows:

$$D(H_{obj}, H_{sce}) = \sqrt{1 - \frac{1}{\sqrt{\bar{H}_{obj}\bar{H}_{sce}N^2}} \sum_I \sqrt{H_{obj}(I) * H_{sce}(I)}}} \quad (1)$$

where N represents the number of matching points using AKAZE optimal feature, I represents the frame of the current sequence.

Set an appropriate threshold N_{th} . When the object is not occluded, the matching number N_{th} of feature points is obtained through the AKAZE algorithm. When the number of matching points $N < N_{th}$, it means that the object is occluded. When the object is occluded,

the Kalman algorithm is used to predict the position information of the moving object in the next frame of the image, so as to achieve the tracking of the foreground object. If the number of matching points $N > N_{th}$, it means that the object is not occluded, but at this time, because the number of feature matching exceeds the obtained threshold, re-sampling is required to determine the best N_{th} .

When the AKAZE algorithm is used for feature matching, there will be mismatches. That is, the feature points in the object template and the points in the image sequence that are not related to the object are also matched. The points that are not related to the object must be included in the sequence image. Then perform culling. This paper uses the random sampling consensus algorithm [9] to match the image features of the object template with the object sub-images in the video sequence according to a single mapping. The four corner points of the object template image are mapped in the current image, thereby obtaining the position of the moving object in the current image sequence and then obtaining a rectangular frame containing the characteristic points. At this time, the lost object will appear in the area again. The improved algorithm steps are as follows:

- 1) Read the image sequence, convert the image sequence from the original RGB color space to the HSV color space, and extract the H channel of the image.
- 2) Select the tracking object and use the selected object frame as the initial state of the object. Calculate and display the color histogram H_{obj} of the object, find the AKAZE feature points of the object template, and prepare for the subsequent feature matching.
- 3) Initialize each parameter of the Kalman filter.
- 4) Predict the possible position of the foreground object in the scene at the next moment and the width and height of the object frame.
- 5) Calculate the object color histogram H_{sce} in the scene, and combine the H_{obj} calculated in step (2) to obtain the similarity $D(H_{obj}, H_{sce})$.
- 6) Determine the size of $D(H_{obj}, H_{sce})$ and T_{th} . When $D(H_{obj}, H_{sce}) > T_{th}$, it means that the object is blocked. At this time, AKAZE feature point matching is performed between the object sub-image in the video sequence and the object template. Calculate the optimal matching number N between the two. Compare N with the threshold N_{th} , if $N < N_{th}$, the Kalman algorithm is used to predict the position information of the moving object that may appear in the next frame of the image. Update the parameters in the Kalman filter to achieve object tracking. If $N \geq N_{th}$, it means that the object is not occluded, but the number of feature matches exceeds the obtained threshold. It is necessary to re-adopt a random sampling consensus algorithm to determine the position information of the tracked object in the scene, the width and height of the rectangular frame, and the best N th value at this time. Update the parameters in the Kalman filtering algorithm.

When $D(H_{obj}, H_{sce}) \leq T_{th}$, it means that the object is not blocked. The Camshift algorithm is used to track the foreground object in real time, and the obtained foreground object position information and the size of the rectangular frame are used as the parameters of the Kalman filter.

- 7) Mark the moving object in the image sequence.
- 8) Judge whether it is the last frame of video, if not, skip to step (4), if it is, end the tracking process.

The specific process of improved moving object tracking is shown in Fig. 4 below.

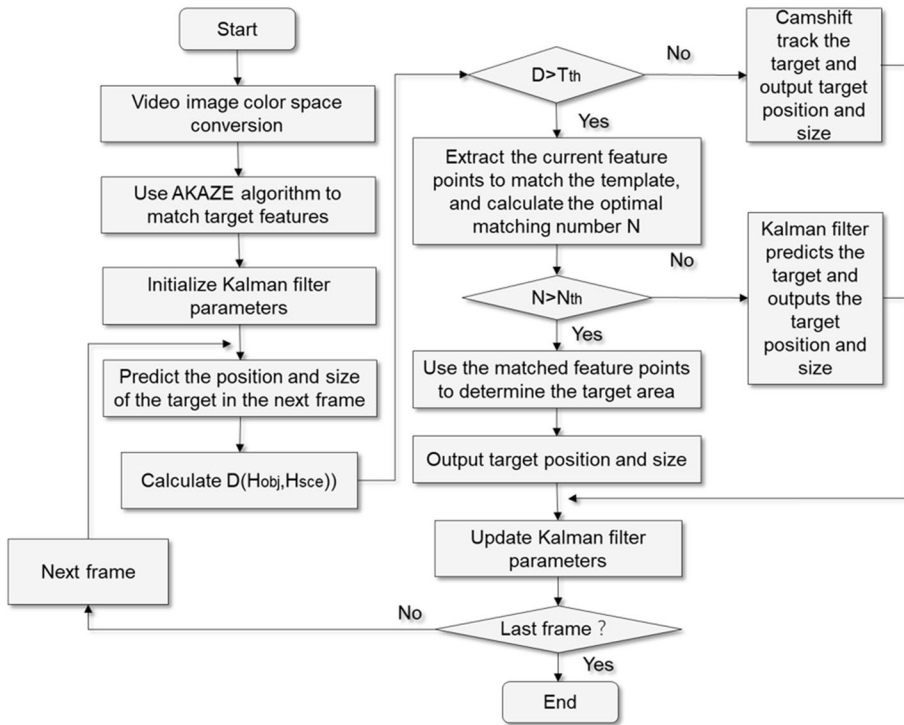


Fig. 4 The flow diagram of the proposed improved Camshift algorithm

4 Results and analysis

In order to verify the correctness and effectiveness of the improved algorithm in object tracking, comparative experiments were carried out from two different complex scenarios. The video information of the scene is shown in Table 1. The main experimental environment is Windows7 system, AMD Athlon(th) X4740 Quad Core Processor CPU, Visual Studio 2015 integrated open method environment, and Opencv3.1 computer vision algorithm library.

In experiment I, use the test video vtest.avi that comes with Opencv3.1 to test the Camshift algorithm, the algorithm combining the Kalman filter algorithm and the Camshift algorithm, and the improved object tracking algorithm. The AKAZE feature matching result in experiment I is shown in Fig. 5 and the object tracking results are shown in Table 2.

Table 1 Two basic video information

ID	Video source	Video scene	resolution	Test frame number
I	Opencv3.1	outdoor	768*576	75
II	camera	indoor	640*302	120

Fig. 5 The AKAZE feature matching result for frame 11 in experiment I



Table 2 The object tracking results for different frames in experiment I

frame	Camshift	Kalman-Camshift	the proposed algorithm
11			
41			
49			

It can be seen from Table 2 that when the object is not occluded, the three algorithms can effectively track the object. When the moving object is partially occluded and there is interference in the background, the Camshift algorithm is unable to track the object, whereas the combined Kalman and Camshift tracking algorithm can effectively track but the tracking accuracy is not good. In this case, only the improved algorithm can accurately and continuously track the moving object.

In order to more clearly and diversely verify the object tracking effect of the proposed algorithm in complex scenes, the second experiment uses a selfie video to test the three algorithms. Figure 6 is the AKAZE feature matching result for frame 17 in experiment II. Table 3 shows the tracking results in more different situations.

From Table 3, the moving object has gone through the following sequence: unoccluded → partially-occluded → completely-occluded → partially-occluded → unoccluded.



Fig. 6 The AKAZE feature matching result for frame 17 in experiment II

When the object is not occluded, the three algorithms can accurately track the object. When the object is partially occluded, the Camshift object-tracking algorithm and the Kalman filtering algorithm combined with the Camshift algorithm can effectively track the unoccluded part of the object. The improved algorithm can predict and track the entire object well. When the object is completely occluded, the object template and the feature matching in the current image sequence will be useless. However, because the Kalman filter algorithm is used in the improved algorithm to predict the position of the moving object, the improved algorithm can more accurately detect the position and size of the moving object. The Camshift tracking algorithm and the combined Kalman, Camshift algorithm cannot track the object. When a moving object changes from a partially occluded state to an unoccluded state, the Camshift tracking algorithm and the combined Kalman, Camshift tracking algorithm can no longer track the object. The improved algorithm continues to effectively track the object.

In order to quantitatively evaluate the three algorithms, the tracking success rate, average processing time, and other evaluation indicators are used for comparison. The calculation results of indicators are shown in Table 4.

The object tracking success rate of the improved algorithm is much higher than the other two algorithms. However, the algorithm speed is slower than the other two algorithms, which is shown in Table 4.

In order to further quantify the tracking effect and accuracy, we chose the Average Overlap Rate (AOR) evaluation method to measure the tracking effects of different algorithms. The calculation method is as formula (2):



















$$AOR = \frac{A_p \cap A_{gt}}{A_p \cup A_{gt}} \quad (2)$$

where the A_p is the area of prediction, the A_{gt} is the area of ground truth.

First, select 50 consecutive frames from scene one and scene two, and manually label them to obtain ground truth. Then, in order to further explain the detection situation in the target loss state, the pictures are divided into three categories: unoccluded, partially-occluded, and completely-occluded. Finally, AOR is calculated based on the result of target tracking. The result is shown in Fig. 7.

As can be seen from the Fig. 7, in the case of unoccluded, the AOR of the three algorithms is basically the same, all above 0.9, and the accuracy of the Camshift algorithm

Table 3 The object tracking results for different frames in experiment II

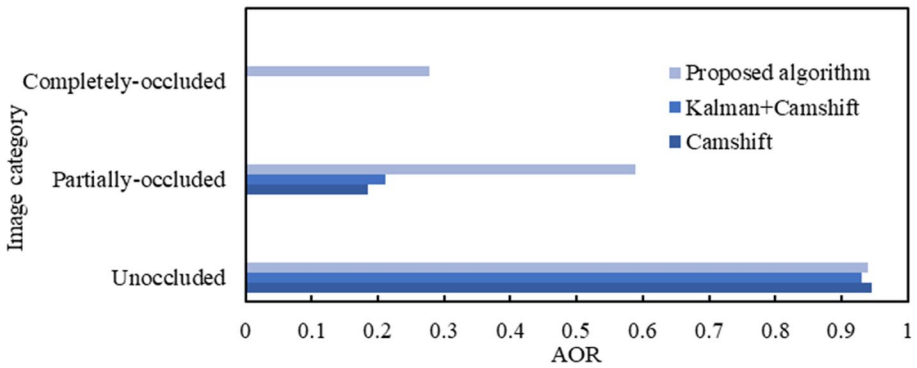
frame	Camshift	Kalman-Camshift	the proposed algorithm
17			
30			
35			
41			
72			
84			

is slightly higher. With the change of the occlusion situation, the AOR of Camshift and Kalman-Camshift dropped sharply to 0.2, indicating that these two algorithms can basically detect the target at this time, but the calibration position is not accurate. At this time, the improved algorithm AOR can still maintain a detection accuracy of about 0.6. In the case of being completely occluded, both Camshift and Kalman-Camshift approach 0, indicating that the target cannot be tracked. The improved algorithm AOR is close to 0.3, indicating that the target object can still be tracked even under occlusion, but the position can not be calibrated accurately. What needs to be explained here is that when the target is completely occluded, manual labeling can only infer the target position based on experience, and there will be certain errors.

Deep learning algorithms also detect objects without being occluded through transfer learning. However, because there are not a large number of occluded samples, deep

Table 4 Compares the results of the three algorithms

Algorithm	ID	Number of test frames	Tracking successful frames	Average processing time /ms	Tracking success rate
Camshift	I	75	33	23.52	0.440
	II	122	38	17.53	0.311
	I&II	197	71	41.05	0.360
Kalman+Camshift	I	75	44	27.32	0.587
	II	122	75	24.58	0.615
	I&II	197	119	51.9	0.604
Proposed algorithm	I	75	68	30.21	0.907
	II	122	117	26.12	0.959
	I&II	197	185	56.33	0.939

**Fig. 7** Compares the AOR of the three algorithms

learning algorithms cannot complete continuous tracking of occluded objects in complex scenes.

5 Conclusion

The main conclusions of this study are as follows:

- 1) An improved Camshift, Kalman algorithm with AKAZE feature matching is proposed to solve the problem of inaccurate object tracking or incomplete detection frame due to poor robustness of object tracking algorithm in complex scenes.
- 2) The improved algorithm is tested in different complex situations and compared with the original algorithm. It is found that the tracking success rate has increased by about 30%, reaching 0.933, which shows that the proposed algorithm can achieve continuous tracking of a completely occluded moving object in a complex background. In terms of algorithm efficiency, the image frame processing speed is faster than 35 ms / frame, which meets the real-time object tracking requirements.

However, compared with other algorithms, the improved algorithm has the lowest average detection speed. Therefore, in future research, the feature point matching algorithm can be optimized to improve the object detection speed. Additionally, achieving high-precision continuous tracking of multiple lost objects is also an important research direction in the future.

Code availability Some or all code that support the findings of this study are available from the corresponding author upon reasonable request.

Author contributions Conceptualization, L.P., H.Z., and B.Y.; Data Curation, L.P., H.Z.; Formal Analysis, L.P., H.Z., and B.Y.; Methodology, L.P., H.Z., and B.Y.; Project Administration, L.P., and H.Z.; Resources, L.P., H.Z., and B.Y.; Funding Acquisition, L.P.; Investigation, L.P., H.Z., and B.Y.; Validation, L.P., H.Z., and B.Y.; Writing – Original Draft Preparation, L.P., H.Z., and B.Y.; Writing – Review & Editing, L.P., H.Z., and B.Y.

Funding This research is funded by National key research and development program (Grant No.: 2018YFB1600202); Chang'an University Doctoral Candidates' Innovative Ability Cultivation Funding Project (Grant No.:300203211241).

Declarations

Conflicts of interest /competing interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Ali NH, Hassan GM (2014) Kalman filter tracking[J]. *Int J Comput Appl* 89(1):15–18
2. Baojun Z (2019) Moving target detection and tracking based on Camshift algorithm and Kalman filter in sport video 15(1)
3. Bradski GR (1998) Real time face and object tracking as a component of a perceptual user interface. *Proceedings Fourth IEEE Workshop on Applications of Computer Vision. WACV'98 (Cat. No.98EX201)*, Princeton, NJ, USA 214–219, <https://doi.org/10.1109/ACV.1998.732882>
4. Cao S, Wang X, Xiang K (2017) Visual object tracking based on motion-adaptive particle filter under complex dynamics[J]. *Eurasip J Image & Video Process* 2017(1):76
5. Chaoqian G, Hu C, Tianping L (2020) Target tracking based on Camshift algorithm and multi 1550(3)
6. Chen Q (2017) Research on real-time single target tracking algorithm based on nuclear correlation filter [D]. *Anhui University*
7. Chengwen D, Ligu D, Zuqun P (2016) Binocular stereo matching algorithm based on KAZE feature[J]. *Industrial Control Computer* 29(07): 93-94+97
8. Hongyun X, Zequn T, Wanchao M et al (2020) High precision indoor positioning method based on visible light communication using improved Camshift tracking algorithm[J]. *Optics Communications* (prepublish).
9. Jacob HW, Randal WB (2020) An Iterative Pose Estimation Algorithm Based on Epipolar Geometry With Application to Multi-Target Tracking[J].*IEEE/CAA. J Automatica Sinica* 7(04):942–953
10. Kanagala SB, Detroja KP (2020) Distributed state estimation through cocting Kalman filters[J]. *Asian J Control*
11. Keehn DG, Lacy JO (2010) VSAM data set design parameters[J]. *IBM Syst J* 13(3):186–212
12. Keli H, En F, Jun Y et al (2018) Element-weighted neutrosophic correlation coefficient and its application in improving CAMShift Tracker in RGBD Video 9(5)
13. Lee E, Gu E, Yoo H J et al (2013) Moving dim-target tracking algorithm using template matching[C]// *International Conference on Technological Advances in Electrical, Electronics Computer Engineering. IEEE* 294-297
14. Li J, Zhou X, Chan S et al (2017) Object tracking using a convolutional network and a structured output SVM[J]. *Computational Visual Media (English)* 003(004):325–335
15. Mingxin J, Hongyu W, Xiaokai L (2012) Multi-target tracking algorithm based on multi-camera[J]. *Acta Automat Sin* 38(04):531–539

16. Nadaraya E (1965) On nonparametric estimates of density functions and regression curves[J]. *Theory Prob. Appl. (USSR)*. Vol. 10:186–190
17. Peng L, Wenbo W, Zongwei W (2015) Multi-source remote sensing image matching based on nonlinear scale space[J]. *Science of Surveying and Mapping* 40(07): 41–44+103
18. Singh Phogat K, Eui Chang D (2020) Discrete-time invariant extended Kalman filter on matrix Lie groups[J]. *Int J Robust Nonlinear Control* 30(12)
19. Usman M, Weiqiang W, Abdenour H et al (2019) Bag of words KAZE (BoWK) with two-step classification for high-resolution remote sensing images 13(4):395–403
20. Vasif, VN, Seçkin Y, Asuman G et al (2017) Shredded banknotes reconstruction using AKAZE points[J]. *Forensic Sci Int* 278
21. Wang X, Liu H, Ma WL (2018) Sparse least squares support vector machines based on Meanshift clustering method[J]. *IFAC Papers OnLine* 51(18)
22. Wenbin F, Baohua L (2018) Research on improved SIFT algorithm for image matching[J]. *Comput Eng Appl* 54(03): 200–205+232
23. Xiangnan Z, Wenwen G, Qifeng H et al (2020) Camshift tracking method based on correlation probability graph for model pig[J]. *EURASIP Journal on Wireless Communications and Networking* 2020(1)
24. Xin L, Yingbo Y, Bin L et al (2019) A new payload partition strategy in color image steganography[J]. *IEEE Trans Circuits Syst Video Technol* 1–1
25. Xin L, Jiaojiao Y, Mingliang C et al (2020) Adaptive payload distribution in multiple images steganography based on image texture features[J]. *Ieee Transactions On Dependable And Secure Computing*
26. Xin L, Kaide L, Xinshan Z et al (2020) Robust detection of image operator chain with two-stream convolutional neural network[J]. *Ieee Journal Of Selected Topics In Signal Processing* 14(5):955–968
27. Xu HX, Zhou W, Zheng JH et al (2011) Multi-object visual tracking based on reversible jump Markov chain Monte Carlo[J]. *IET. Computer Vision* 5(5):282–290
28. Xu T, Huang C, Zhu M (2015) Detection and tracking of multiple moving targets in complex conditions[C]. //International Conference on Mechatronics, Materials, Chemistry and Computer Engineering
29. Yan C, Zhang B, Coenen F (2015) Multi-attributes gait identification by convolutional neural networks[C]. *IEEE International Conference on Image and Signal Processing, Shenyang* 642–647
30. Ye L, Hua Z, Chuankai L (2020) Maneuver detection and tracking of a space target based on a joint filter model[J]. *Asian Journal of Control* 05
31. YiGuo C, BingFan W (2014) Research on application of Camshift and Kalman filter algorithm in video object tracking 3593:1685–1689
32. Yilmaz S (2017) Science-forensic science; data on forensic science reported by S. Yilmaz and Colleagues (Shredded banknotes reconstruction using AKAZE points) [J]. *Science Letter*
33. Yun W, Qing T, Jianhua G et al (2017) Multi-vehicle detection algorithm through combining Harr and HOG features[J]. *Elsevier B.V*
34. Yuwei L, Yuan Y, Qi W (2020) A dense connection-based network for real-time object tracking[J]. *Neurocomputing* 410
35. Zhang K, Zhang L, Yang M H, et al (2013) Fast tracking via spatio-temporal context Learning[J]. *Computer Science*
36. Zongwei Z, Wenhan L, Qiang W et al (2020) Distractor-aware discrimination learning for online multiple object tracking[J]. *Pattern Recogn* 107