

Review Article

Implementation of Camshift Target Tracking Algorithm Based on Hybrid Filtering and Multifeature Fusion

Sijie Du,¹ Hongxin Xu,¹ and Tianping Li ²

¹Shandong Key Laboratory for Normal Distributed Computer Software Technology, School of Physics and Electronic Science, Shandong Normal University, Jinan 250014, China

²School of Physics and Electronic Science, Shandong Normal University, Jinan 250014, China

Correspondence should be addressed to Tianping Li; sdsdltp@sdsnu.edu.cn

Received 4 September 2020; Revised 24 September 2020; Accepted 2 November 2020; Published 25 November 2020

Academic Editor: Manuel Aleixandre

Copyright © 2020 Sijie Du et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In recent years, the Mean shift algorithm has extensive applications in the field of video tracking. It has some advantages of low cost, small memory, and good tracking effect. However, there are some shortcomings in the existing algorithm; for example, it cannot produce adaptive changes as the target size changes. And when there are similar objects, it is prone to target positioning errors and tracking failures caused by occlusion. In this paper, an improved method of continuous adaptive change Mean shift (Camshift) for high-precision positioning and tracking is proposed. The traditional Camshift method only uses hue components in HSV to extract features. This paper uses the combination of H and S components in HSV space to build a two-dimensional color feature histogram and with the image's LBP feature histogram to increase tracking accuracy. Meanwhile, for the sake of target occlusion and nonlinear changes in the tracking process, this paper introduces a Gaussian-Hermit particle filter that is updated by the Kalman filter. Experimental result demonstrates that the real-time performance of the proposal in this paper is better than Mean shift, Camshift, simple particle filter, and Kalman filter.

1. Introduction

With the wider application of video tracking, scholars continue to propose various methods [1, 2]. In video tracking algorithms [3–6], the Mean shift [7, 8] has many merits such as small calculation amount, good robustness, and high tracking accuracy. But it has certain limitations [9]. For example, fixed tracking window and reverse mapping simply relied on the brightness structure in HSV will cause the tracking effect to deteriorate. Besides, when the target rotates and deforms greatly, it is easy to lose the target and cause tracking failure. For this reason, [10, 11] proposed a method of adaptively changing the tracking window. This method adds adaptive capabilities to the original algorithm to handle the impact of target size changes. In addition, in order to cope with the tracking environment where similar object interference appears, [12] uses the H and V vectors on HSV to construct the histogram to enhance the features of the target so that the tracking accuracy is increased. How-

ever, the issue of the target moving faster was not discussed. [13, 14] raises a method that combined with the Kalman algorithm. The Kalman algorithm [15] is treated as predictor to forecast the target globally and to obtain more accurate target coordinates. This improvement can reduce the error of the tracking when the object moves too fast. [16] proposed a tracking algorithm combining multiple feature templates. This algorithm is combined with the SIFT algorithm to make its tracking effect better. However, the number of array operation is too large to satisfy the requirements of real-time tracking. [17] combines the back projection of Camshift and the motion state obtained by the optical flow method to achieve target prediction, but this algorithm also has the shortcomings mentioned in the previous literature and poor real-time performance. [18] uses chromaticity-differential two-dimensional joint feature to establish a target model. The maximum difference value of each pixel in the 8-neighborhood is used to describe the relative information of the pixel and the detailed information. The

experimental result of this method is good, but the calculation amount is too large. In [19], back projection is used to obtain the chromaticity-differential two-dimensional joint probability distribution map of the tracking image.

In [20, 21], Kalman filter is utilized to enhance robustness when the object is partially occluded. Kalman filter is used as a processor to process partial occlusion when the target's motion mode is linear. However, this method fails to consider the processing of nonlinear problems. [22] proposed a particle filter-based method that improved nonlinear motion tracking. [23] proposed to find the local maximum of the particle iteratively. In this way, some individual particles can be aggregated into a larger particle set, thereby reducing the divergence and obtaining a better tracking result. In [24], Mean shift particle filter requires only a few particles when dealing with fast-moving targets and when encountering occlusions. But since each particle needs to be iterated by Mean shift, even if the quantity of particles is reduced, the tracking speed is still very slow. In addition, as [25] performs arithmetic operations on each particle set, this leads to a large amount of calculation and also makes the particle set too concentrated in this method. At this time, when the tracked object is partially or completely occluded and then reappears, which will cause object tracking failure. Aiming at the shortcomings of the abovementioned documents, [26, 27] proposed the ideas from the local characteristics of the target to the overall characteristics. Among them, the overall feature model of the target mostly meets the characteristics of linear or Gaussian motion, while only a small part of the object does not meet the above characteristics. For objects that meet linear or Gaussian models, Kalman filter is used for motion estimation; for objects that do not meet linear motion, particle filters are used for local estimation. [28] linearizes each particle to obtain more aggregated particles and perform nonlinear processing on each particle. The quantity of particles can be able to greatly cut without any iterative addition in this method. Therefore, their calculation time remains the same. The algorithm proposed in [29] has a degradation phenomenon, and the key to suppressing the degradation phenomenon is to choose a good important density function. In view of the above shortcomings, we improve as follows: (1) we use the original H component to improve the H and S components to build the target histogram; (2) use the target's LBP feature histogram and H and S histograms to establish an inverse mapping graph; (3) use a hybrid filter to improve the tracking effect during occlusion.

In this paper, we propose to take advantages of the weighted feature histograms of H and S in *HSV* to describe the characteristics of the target and then use the inverse mapping distribution of the color histogram to improve the tracking accuracy. The Gaussian-Hermitian particle filter proposed in [30] is intended to generate the importance density function of the particle filter, and the Kalman filter is used for global motion estimation to obtain better state estimation accuracy. The structure of this paper is as follows: Section 2 introduces the proposed algorithm. Section 3 details the improved Camshift method. Section 4 is the experiment, and Section 5 gives the conclusion. In Section 6, the references are listed.

2. The Proposed Method

Camshift algorithm is evolved from Mean shift algorithm. Among them, Mean shift has the superiority of simple operation and small memory space. It is a method that uses the non-parametric function of density gradient estimation to locate and track objects by iterative optimization to seek out the extremum value of the probability distribution. So this article briefly introduces the process of Mean shift algorithm.

The algorithm flow in this paper is as follows:

- (1) Select the object and record the input position and object information W
- (2) Find the zero-order distance of the object:

$$M_{00} = \sum_x \sum_y xI(x, y). \quad (1)$$

Find the first-order distance:

$$M_{10} = \sum_x \sum_y xI(x, y), \quad (2)$$

$$M_{01} = \sum_x \sum_y yI(x, y). \quad (3)$$

Compute the centroid of the search window W :

$$x_c = \frac{M_{10}}{M_{00}}, \quad (4)$$

$$y_c = \frac{M_{01}}{M_{00}}. \quad (5)$$

- (3) The size of W as $s = \sqrt{M_{00/256}}$; the length is 1.2 s
- (4) Adjust the size of the window and find the centroid and specify the measure of the fixed threshold. If the distance moved each time is greater than the value we set, it will repeat (2), (3), and (4) until the distance they move each time is less than the set value, or the quantity of calculations reaches the maximum amount of times set will stop the calculation

2.1. Target Model. When the tracking target is selected, [31] proposes the histogram information of the target color to construct the histogram, and then, the kernel density estimation function Epanechnikov [32] weights the pixels at different positions, highlighting the color characteristics of pixels at different positions contribution. From K_E in (7), we can see that the pixels in the center of the target area are counted for color information. It can be seen that the contribution of the pixels in the center of the target is the largest, and the contribution at the edge of the target is the smallest:

$$\hat{q}_u = C \sum_{i=1}^n K_E(\|x_i\|) \delta(C(x_i) - u) \quad u = 1, 2, 3, 4 \dots n, \quad (6)$$

where u is the histogram interval obtained by segmentation, $C(x_i)$ represents the set of pixel values in the corresponding interval, and x_i represents each pixel value. $\delta()$ is the Kronecker del function [33]; C is the normalization constant $C = [\sum_i^n K_E(\|x_i\|^2)]^{-1}$ and $\sum_i^n \hat{q}_u = 1$. The function δ is used to determine which color histogram $C(x_i)$ belongs to in order to construct the color space. $K_E(\cdot)$ is the kernel density estimation function, which is applied to weight and highlight the features of the target. Its expression is as (7):

$$K_E = \begin{cases} c * (1 - \|x_i\|), & \|x_i\| < 1 \\ 0 & \text{otherwise,} \end{cases} \quad (7)$$

where x_i represents the pixel position of the target area.

2.2. Candidate Target Model. In the selection of candidate targets, it is assumed that $\{y_1, y_2, y_3, y_4, \dots, y_n\}$ are n_k pixels in the area to be selected. h is the bandwidth that determines the value of pixels in the positioning process. y is the normalized pixel position of the target candidate centered on y in the current frame; then, the expression of its probability function is as (8):

$$\hat{p}_u = C_h \sum_{i=1}^N K_E\left(\frac{\|y - y_i\|}{h}\right) \delta(C(y_i) - u) \quad u = 1, 2, 3, 4 \dots n. \quad (8)$$

C_h is the normalization constant so that $\sum_i^n \hat{p}_u = 1$, $C_h = [\sum_i^n K_E(\|(y - y_i)/h\|^2)]^{-1}$.

2.3. Similarity Measure. After using the calculation process in (6) and (8) to obtain the weighted probability function of the target and the candidate target, respectively, we then use the similarity function to quantitatively express their similarity. We calculate the similarity between these two discrete representations \hat{p}_u and \hat{q}_u using the Bhattacharyya distance in [34]. Among them, the value range of their similarity is as the (BH) coefficient in (10) is the interval [0, 1]. The size of BH responses the extent of similarity between the target and the candidate target. If the value is larger, the candidate region model and the target model are more similar, and the candidate target we get is the greater the probability of our tracking target.

$$d(y) = \sqrt{1 - \hat{p}(y)}, \quad (9)$$

$$\hat{p}(y) = \rho[\hat{q}, \hat{p}(y)] = \sqrt{\hat{p}(y)\hat{q}_u}. \quad (10)$$

3. Improved Camshift Tracking Algorithm

Camshift is an improved algorithm extended from Mean shift [35] algorithm to consecutive video sequence, which

performs Mean shift [36, 37] operation on all video frames of the video and takes advantage of the size and center of the search window of the previous frame as the initial value of the search window of the Mean shift algorithm in the next frame to find the optimal iterative result. Through the above calculations, we can tail the target after. The Mean shift algorithm process is:

- (1) Initialize search window W
- (2) Establish a back projection in the light of the HSV feature of the video frame
- (3) Utilize the Mean shift algorithm to obtain the position and the size of target to search for the new window
- (4) Reinitialize the size and position of the search window with the value in 3 in the next frame of video image and then jump to 2 to continue

Camshift is able to effectively improve the issue of target size change and partial occlusion, and it does not require high system resources, and the real-time tracking is good, and good tracking results can be achieved when the background conditions are not complicated. However, when the background is more complicated, or there are interferences parallel to the target color pixels, it is going to cause the tracking effect to deteriorate and even the tracking failure because it simply considers the H component in the color histogram to construct color features. For the sake of improving the performance of the system, this paper divides the H and S components into different intervals. For the number of intervals divided by H, we take the value m in (6); then, the quantization step size of each interval is $360/m$; in the same way, S is also divided into different intervals, and the number of intervals is n ; then quantization, the step size is $255/n$. We obtain the color feature histogram of the target based on the obtained spatial range of H and S and then through the action of the weighting function. And the H and S components are transformed by (11) to obtain the standardized distribution table of the inverse mapping distribution, and then, the histograms of hue and saturation obtained are back projected through Mean shift operation to obtain the position of the region of interest. The function of (11) is the amount to a color probability search table that converts the H and S components in the inverse mapping into the same distribution as in formula (11):

$$\rho(h_0, s_0) = \frac{S(h) * S(s)}{\max[S(h)] * \max[S(s)]}. \quad (11)$$

Among them, $S(h)$ and $S(s)$ represent the pixel values in the H and S components, respectively, and $\max[S(h)]$ and $\max[S(s)]$ represent the maximum value of $S(h)$ and $S(s)$. The $p(h_0, s_0)$ is on behalf of the probability distribution of the colors of the H and S components.

The basic principle of LBP is to regard a pixel as a center point, and its value is calculated by comparing the relative gray value of the point and its neighboring 3×3 pixels as a

response, by comparing the pixel value of the surrounding neighborhood. The most primitive LBP calculation formula is described as (12) and (13):

$$\text{LBP}_{P,R} = \sum_{I=0}^{P-1} S(g_i - g_c) x 2^I, \quad (12)$$

$$s(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{else} \end{cases}. \quad (13)$$

Add the result of the inverse mapping transformation of the feature map obtained by LBP and the inverse mapping image obtained by HSV according to the $\alpha, \beta (\alpha + \beta = 1)$ ratio. The above improvement process is as follows:

The first step is to initialize:

Initialize the search window W and initial position of the selected target in the search window.

Step 2. Establish HSV and LBP feature map:

Get the HSV and LBP features of the selected target and select LBP, H, and S to build a two-dimensional weighted feature histogram.

Step 3. Calculate the back projection:

Using the H and S components and LBP feature of picture, a back projection is established.

Step 4. Track the target:

We use the initialized window position to back project, call the Mean shift algorithm to acquire the position and size of the target, and use this as the initial value of the next frame of image. Camshift adjusts and shrinks the window in the light of the calculation information of Mean shift to determine the location information of the target.

Step 5. Loop:

In the next search, we utilize the position information acquired in the previous step as the initial value to continuously adjust the search window and execute step 3.

In the improvement process of the above-mentioned Camshift, the change of the target size and the tracking phenomenon of the target and the background resemblance to the situation can be improved. However, the target model that appears in the target tracking process sometimes fails to meet linear or Gaussian conditions, which will lead to tracking failure. For this possible situation, this paper introduces the method of mixing Gaussian-Hermit particle filter and Kalman filter. Specifically, the algorithm in this paper uses the Gaussian-Hermit particle filter on the basis of the improved Camshift method to improve the nonlinear and non-Gaussian problems and then uses the Kalman filter to renew the prediction, which be able to enhance tracking accuracy. Compared with the Kalman filter algorithm, the method in this paper is also suitable for nonlinear system models. It only needs to know the prior distribution of the target variable and does not strongly depends on the initial conditions. In addition, the Kalman filter algorithm may lead

to divergence, failure, and tracking failure under nonlinear systems and inappropriate initial condition models. The improved algorithm in this paper passes the region of interest tracked by Camshift as input to the Gaussian-Hermit filter. The particle filter considers the state of Camshift to define the number of particles. Then, the processed state is applied to the [38] Kalman filter for prediction update, and the updated tracking result has better tracking accuracy than before. At the same time, unlike Kalman filter tracking in the global scope, the method in this paper restricts the particles to tracking moving targets in the local field of view. Through the combination of these two kinds of filtering, the problems of occlusion and nonlinear motion are improved. In addition, this paper is based on the feature extraction of color density features to meet the application features of Gaussian-Hermit particle filter [39], so GH is used to construct the features of the importance density function. Its GH one-dimensional integral formula (16) is expressed as:

$$I = \int f(p) e^{-p^2} dp \approx \sum_i^n w_i f(p_i). \quad (14)$$

Gauss point is p_i ; the weight coefficient is w_i , and n is the value of Gauss points.

Assuming that the posterior probability of the system state is approximated by a Gaussian distribution, w_k and v_k are mutually independent Gaussian noises with zero mean, and the variances of [40] are Q and R , respectively. Gaussian-Hermitian filtering, like Bayes filtering, also has an iterative process of prediction and update. It also uses transfer equations and observation equations to obtain posterior probability estimates. Specific steps are as follows:

- (1) Forecast: suppose the x_{k-1} is system state at the previous moment, and the variance is $p_{(k-1)}$. Use the Gaussian point transformation formula $x_i = (\sqrt{p_{k-1}})^T q_i + x_{k-1}$ to carry out the transition equation between Gaussian transformation and system state and calculate the system state and its variance; then, the system can be obtained The prediction state and variance are expressed as:

$$x_{k|k-1} = \sum_{i=1}^m w_i f(x_i), \quad (15)$$

$$p_{k|k-1} = Q + \sum_{i=1}^m [f(x_i) - x_{k-1}]^T w_i. \quad (16)$$

- (2) Update: according to the prediction results, continue the Gaussian transformation to obtain a new Gaussian point and observation model. The system state and its variance are updated as follows:

$$x_k = x_{k-1} + K(z_k - y_k), \quad (17)$$

$$P_k = P_{k-1} - K_k P_{xz}^T, \quad (18)$$

among them

$$y_k = \sum_{i=1}^m w_i h(x_i), \quad (19)$$

$$P_{xz} = \sum_{i=1}^m w_i (x_i - x_{k-1})(h(x_i) - y_k)^T, \quad (20)$$

$$P_{zz} = \sum_{i=1}^m w_i (h(x_i) - y_k)(h(x_i) - y_k)^T, \quad (21)$$

$$K_k = P_{xz} P_{zz}^{-1}. \quad (22)$$

Based on the above theory, the GHPF algorithm steps are as follows:

$$q(x_k | x_{k-1}^i, y_k) = N(x_k^i, P_k^i). \quad (23)$$

Step 1. Importance sampling. Gauss-Hermitian filter is performed on the sample set $\{x_{k-1}^i\}$ ($i = 1, 2, 3 \dots N$) at $k-1$ time to obtain the state x_k and variance P_k of each particle and take them as importance density functions, namely:

Generate prediction sample set:

$$\{\hat{x}_k^i, i = 1, 2, 3 \dots N\}, \hat{x}_k \sim q(x_k | x_{k-1}^i, y_k). \quad (24)$$

The importance weights corresponding to the particles are:

$$\left\{ \hat{w}_k^i \propto \hat{w}_{k-1}^i \frac{p(y_k | \hat{x}_k^i) p(\hat{x}_k^i | x_{k-1}^i)}{q(\hat{x}_k^i | x_{k-1}^i, y_k)} \right\}. \quad (25)$$

Normalized weight is:

$$\tilde{w}_k^i = \frac{\hat{w}_k^i}{\sum_{i=1}^N \hat{w}_k^i}. \quad (26)$$

$$p(x_k | y_{1:k}) = \frac{1}{N} \sum_{i=1}^N \delta(x_k - x_k^i). \quad (27)$$

$$\hat{x}_k = \frac{1}{N} \sum_{i=1}^N x_k^i. \quad (28)$$

Step 2. Resampling of particles. Get the particle sample set $\{x_k^i, 1/N\}$ ($i = 1, 2, \dots, N$), then the posterior probability density is:

Step 3. Estimate the posterior mean of the state.

The predicted x_k in (19) is updated through the Kalman filter to obtain a satisfactory tracking effect. The introduction is as follows: when the observation noise is large, the particles cannot accurately describe the posterior probability, which will affect the estimation performance. Considering the influ-

ence on the order estimation of the state x_n , we pass the predicted x_k through the Kalman filter for prediction update. Since equation (18) and equation (19) are approximately linear equations, Kalman filtering can be considered to obtain its optimal estimate $\hat{x}_{k|k}$, where we define a state space equation for the Kalman filter system:

Equation of state:

$$Z_k = H_k x_k + v_k. \quad (29)$$

Measurement equation:

$$Z_k = H_k x_k + v_k, \quad (30)$$

where x_k is the state vector, Z_k is the measurement vector, A is the state transition matrix, u_k is the control vector, B_k is the control matrix, and w_k and v_k are the system noise and satisfy the Gaussian noise. The process of Kalman filter is a recursive process. The constant "prediction update process", its prediction status value is as (31)

$$\hat{x}_{k|k-1} = A_k \hat{x}_{k-1} + B_k u_k. \quad (31)$$

Minimum prediction error:

$$P_{k|k-1} = A_k P_{k-1|k-1} A_k^T + Q_k. \quad (32)$$

Updated status:

Measurement error:

$$\tilde{y}_k = z_k - H_k \hat{x}_{k|k-1}. \quad (33)$$

Measurement covariance:

$$S_k = H_k P_{k|k-1} H_k^T + R_k. \quad (34)$$

Optimal Kalman gain:

$$K_n = P_{k|k-1} H_k^T + R_k. \quad (35)$$

Correction status value:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k. \quad (36)$$

Corrected minimum mean square error:

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}. \quad (37)$$

The location information of the tracking target is obtained after predictive update. The steps of the aforementioned hybrid particle filter algorithm 1 are as follows:

4. Experiment

In this paper, the experiment environment is under the 64-bit system of Windows10, and the processor is Intel(R) Core (TM)i5-9400 CPU@2.900GHz 2.90GHz and RAM is 16.0GB. The experiment is carried out in python3.7 under

1. Initialization: sample the prior probability distribution $p(x_0)$ to get particles $\{X^i\}_{i=1}^N$
2. For $n = 1, 2, \dots, n$
3. Update the particles according to formula (19) to obtain x_k
4. Update the importance weight according to equation (29) and get \tilde{w}_i^k
5. Normalize the weights according to equation (32) to obtain w_i^k
6. Resample the particles to obtain the particle density (34) $p(x_k | y_{1:k})$
7. According to equation (35), the Kalman gain K_k is obtained
8. According to formula (37), the posterior estimation error covariance matrix is $P_{k|k}$ is obtained
9. According to equation (36), the optimal estimate of state x_k is obtained \hat{x}_k
10. End n

ALGORITHM 1: Hybrid particle filter algorithm steps.

the editor of *PyCharm*. The relevant information of these three groups of experimental videos is shown in Table 1. The video sets of these three groups of experiments contain different experimental information. In this paper, three sets of video sets in Table 1 are used to verify our tracking effect. These three sets of experiments contain environments of different complexity. In the first group of experiments, the tracking of a car on the expressway is a part of the experiment. The tracking environment of this group of experiments is relatively complex, including all occlusions and partial occlusion, as well as abrupt changes of motion state. The second group of experiments was conducted in an intersection environment. Compared with the first group of experiments, it had more kinds of interference, among which the interference not only included vehicles, pedestrians, zebra crossings, and roadside public facilities. The third experiment was conducted under intense light conditions. In this experiment, the interference was mainly caused by light and roadside trees and moving vehicles. These three groups of experiments have different emphases. For the sake of analyzing the tracking consequence of our experiment in detail, we conducted quantitative analysis on the data of the three groups of experiments. In this paper, we analyze the change of the centroid coordinates of the tracking object to represent the change of the position of the object and use the difference of the centroid of the tracking object on the X- and Y-axes to represent the tracking error. We verify the tracking effects of different algorithms by analyzing the errors of different experiments. At the same time, the advantages and disadvantages of different algorithms are analyzed to verify the advantages of this paper. The video set selected in the experiment is shown in Table 1, which records the name of the video, the frame number of the video set, and the size of the video in detail.

In the first experiment, we chose a video frame number of 2480 with a resolution of 217×290 to track a video on a highway with a relatively complex environment. The experimental environment includes partial occlusion of the object, total occlusion, sudden change of target speed, lane change of the target vehicle, and tracking under the interference of similar targets. The tracking effect of this experiment is shown in Figure 1. We select 8 frames of the video to represent the experiment of this group. Under the same conditions, it can be known that the tracking effect of

TABLE 1: The experimental video sets and their size and frame number.

Video name	Number of videos	Picture size
Car chase	2480	217×290
Car	252	360×480
Crossroads	2160	480×640

our means is relatively good, which can not only track the target in the case of occlusion after updated prediction but also combine the improved particle filter and Kalman filter in the face of velocity change. It can not only improve the tracking performance but also realize various tracking effects when the target has lane change. At the same time, for the sake of demonstrating the robustness of our method, this paper compares the predicted value of each method's tracking target with the actual value and then utilize the difference value to the error graph to make quantitative analysis, which is the experimental effect. The tracking effect of experiment 1 is shown in Figure 2, and the experimental error results are shown in Figure 3.

The second group selected a 360×480 crossroad video clip with 252 frames of video for the experiment. The jamming objects in this set of videos are more complex. In this experiment, a video of a crossroad was selected. The disturbing objects in the video included not only the interference of pedestrians, vehicles in different directions, and the white zebra section on the sidewalk but also the interference of two cars parked on the roadside with similar outlines. The experimental environment is complex, and the experimental results of various methods are also different. The results of this group of experiments are revealed in Figure 4. It can be concluded from the figure 4 that other methods can still track the target in the first few frames when interference exists. After some tracking updates, other methods all stay on some different interferences, which cannot overcome the conditions of interference, resulting in the failure of tracking. The algorithm presented in this paper not only has good experimental results in interference and turning but also has good tracking effect when the size of the target changes. In order to

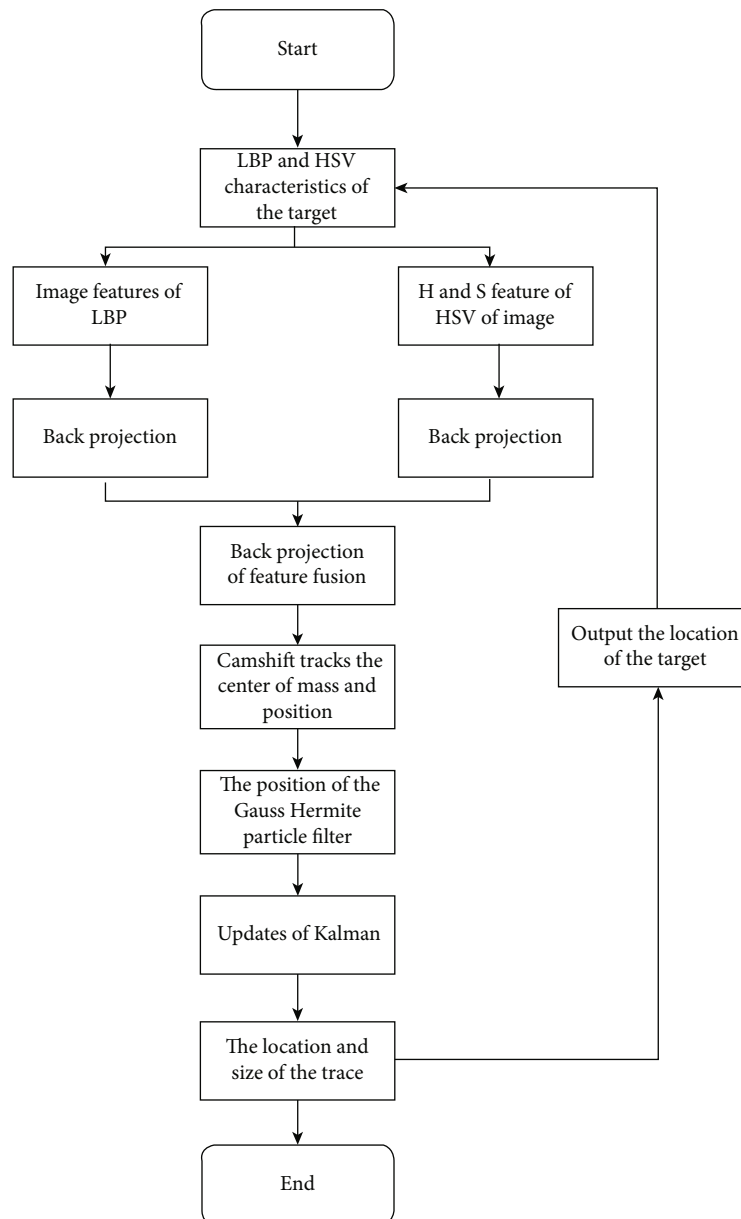


FIGURE 1: The flow chart of our way.

better analyze the tracking effect, we analyzed the centroid value of the target. It can be seen from Figure 5 that our method has better effect on the distribution and real position of the centroid, which be able to meet the precision requirements of real-time tracking.

In the third experiment, the frame number of the video was 2160, and the resolution of the video was 480×640 . The video environment of this group of experiments is under the condition of intense illumination. The main interfering factor in this experiment is the intensity of light. In the experiment, we can see that our method can always track the target from Figure 6 even if there is interference from similar items in the tracking process. However, the tracking effect of other methods cannot adapt to the change of the object quickly and

requires more frames to be initialized. It can be known from the experiment that the tracking impact of other methods is relatively poor under the same conditions. However, with the passage of time, the tracking infection of other ways gradually gets better, and the tracking result is also better. Compared with our method, the tracking accuracy of other methods is low, and the tracking effect is not very good. It can be concluded from Figure 7 that our method has a good tracking effect with low relative error and good tracking effect.

Meanwhile, for the sake of quantitatively analyzing our experimental results, this paper uses the actual centroid coordinates of the tracking target position and the centroid coordinates of the observation position to verify

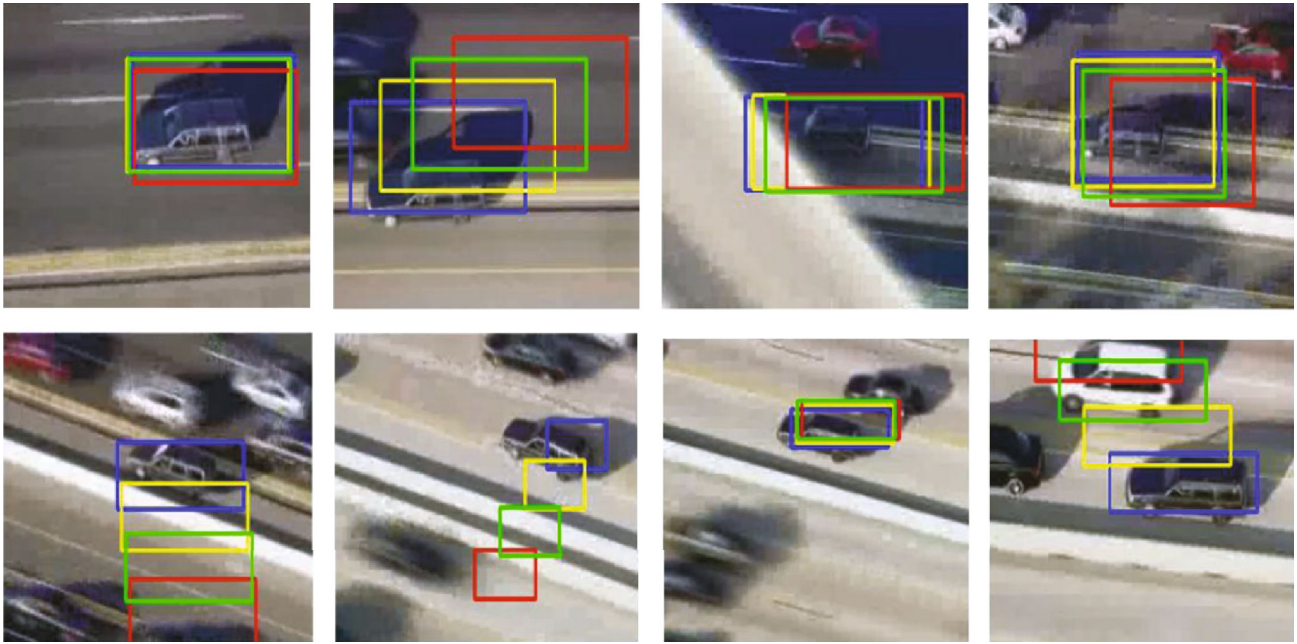


FIGURE 2: Tracking effect of car chase.

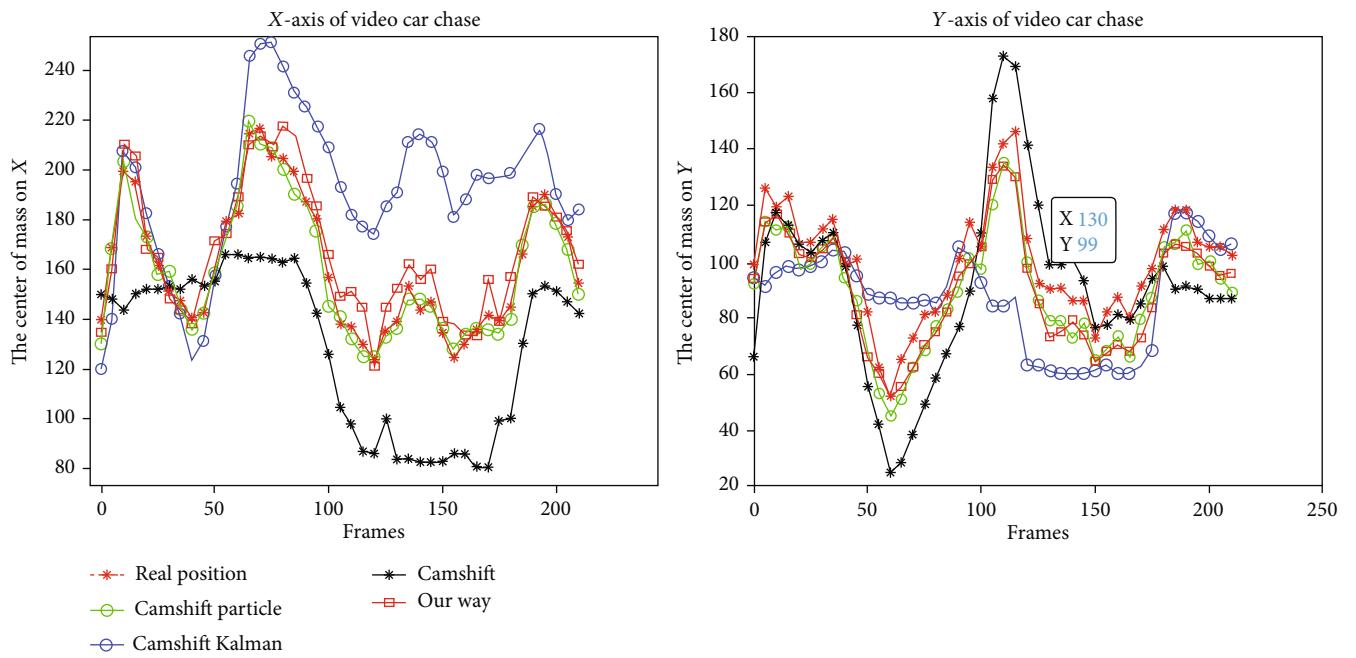


FIGURE 3: Car_chase's the centroid coordinates on the X- and Y-axis.

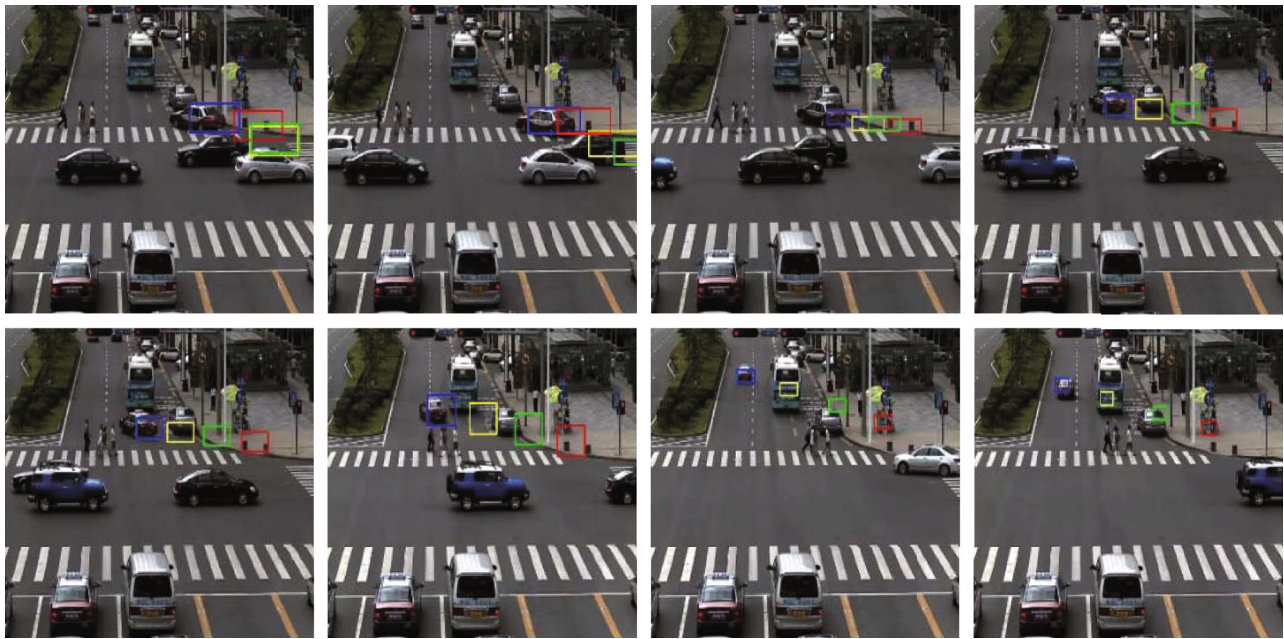


FIGURE 4: Tracking effect of crossroad.

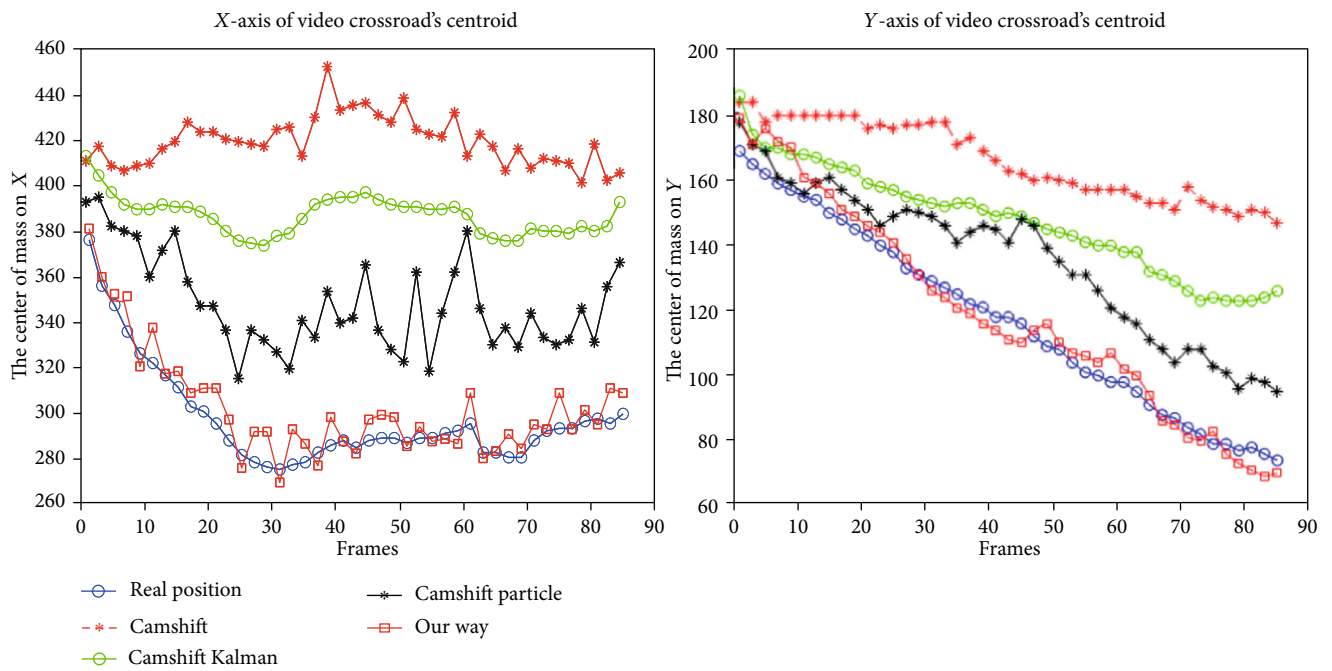


FIGURE 5: Cross_roads track the centroid coordinates on the X- and Y-axis.



FIGURE 6: Tracking effect diagram of a car.

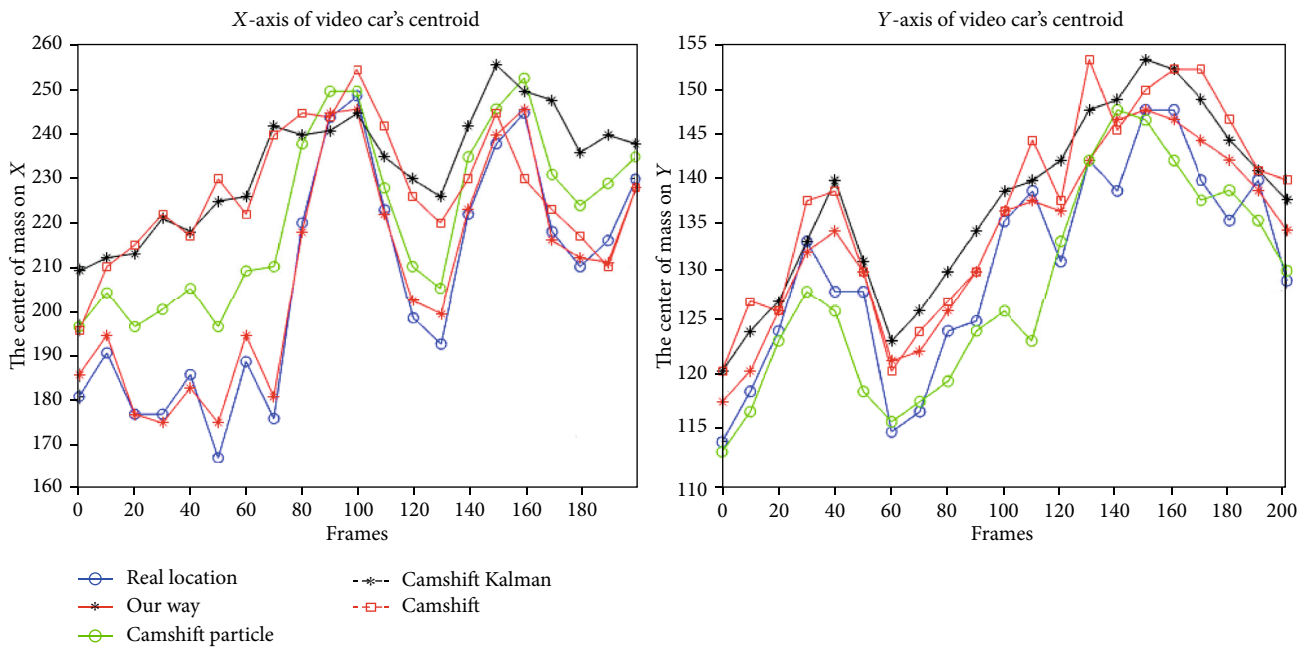


FIGURE 7: Car's tracking target's centroid coordinates on X- and Y-axis.

our method, where x and y is on behalf of the coordinates of the centroid of the tracking object on the X- and Y-axes. As indicates in the figure, Figures 3–7 represent the changes in the centroid points of the tracking objects in videos 1, 2, and 3, respectively. From Figures 3–7, we can see that under the same conditions, our method has a relatively good tracking impression, and the real-time tracking and tracking accuracy meet the tracking requirements.

It can be observed that the coordinate position of the centroid of the tracking target on the x and y coordinate axes from Figures 3–7. This article analyzes the video tracking error based on the coordinate value of the tracking target. From the changes in the line graphs of Figures 8–10, we can come to the following conclusions. In the first set of videos, large errors will occur due to occlusion and sudden changes in the motion state, but the tracking effect is adaptive to Mean shift and particle. The method is not much different

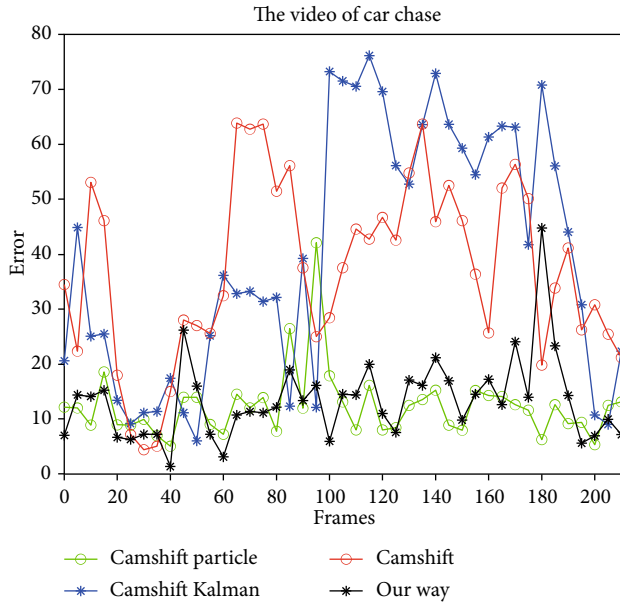


FIGURE 8: Car_chase's error graph.

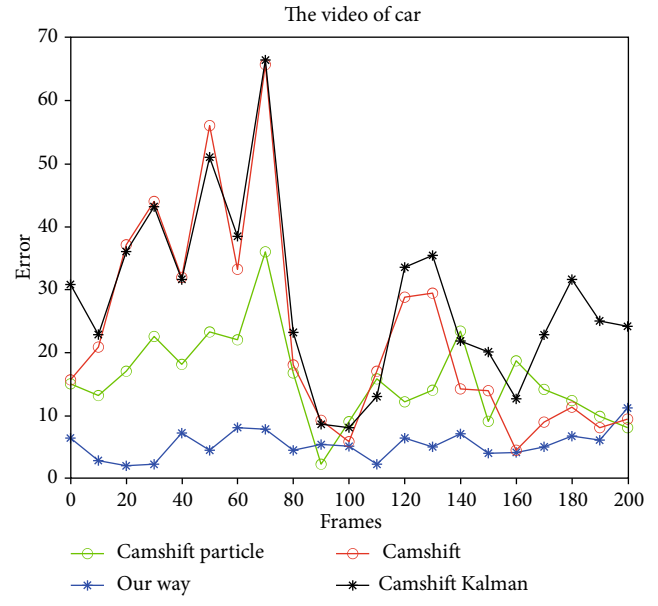


FIGURE 10: Car's error graph.

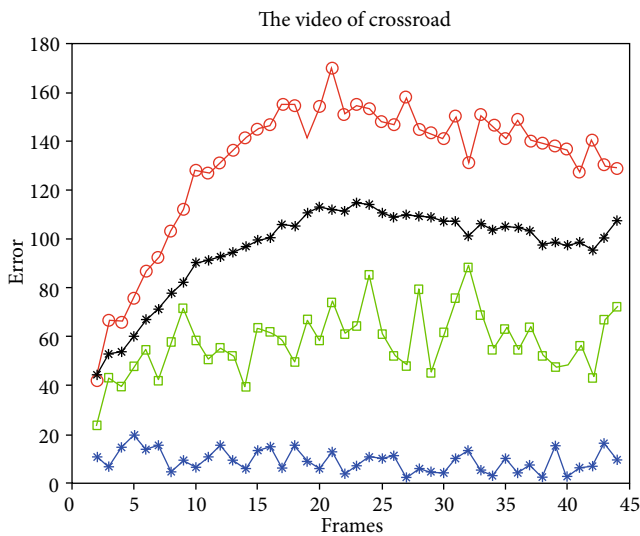


FIGURE 9: Cross_road's error graph.

from our method, and the tracking effect is relatively good. But in the second complex environment, the advantage of the tracking effect of our method is shown. In the tracking, the error of our method is small, and the accuracy is high, but other methods have failed to track. The effect of the third set of experiments is not much different from the tracking errors of other methods. In the later stages of tracking, the tracking accuracy of these methods is improving and can basically meet the tracking requirements, but the error of our method is relatively small, and the tracking effect is better, and the error effect diagram is shown in Figure 10.

In order to further prove the accuracy of our method, we selected the data of experiment two and compared our method with Kcf and TLD methods. It can be seen from Table 2 that our method has small errors and meets the accuracy requirements of real-time tracking.

TABLE 2: Comparison table of tracking error of different methods.

Tracking way	Our way	KCf	TLD
Tracking average error	8.5	15	22

5. Conclusion

The improved Camshift method proposed in this paper uses a weighted kernel function to weight the pixels around the target to increase the weight of the pixels close to the target in order to highlight the target features. Then, the original Camshift algorithm is improved from a one-dimensional histogram of HSV (hue, saturation, value) that only contains hue components to a two-dimensional histogram of color features using hue and saturation components in the HSV space and LBP feature of image to highlight the target. Features can enhance the accuracy of tracking similar targets. Meanwhile, for the sake of tracking the occlusion and nonlinear non-Gaussian changes in the tracking model, we introduced a Gaussian-Hermit particle filter to improve the nonlinearity and occlusion problems in motion, and then, we use the Kalman filter to predict the update that makes the tracking accuracy more accurate. Through experiments, we can conclude that the improved method put forward in this paper not only satisfies the effect of real-time tracking in time. In terms of tracking accuracy, we can see that the coordinates of the actual position coordinates, and the predicted coordinates have a small difference to meet the tracking requirements. Through the analysis of experiments, our method is capable of being better than Mean shift and Continuously Adaptive Mean shift (Camshift) and simple particle filter and Kalman filter in accuracy. Consequently, this algorithm has great application prospect in the field of target tracking and location.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Ethical Approval

This article does not contain any studies with human participants performed by any of the authors.

Conflicts of Interest

Sijie DU declares that he has no conflict of interest. Hongxin Xu declares that he has no conflict of interest. Tianping Li declares that he has no conflict of interest.

Authors' Contributions

Funding acquisition was done by Tianping Li. Methodology was done by Sijie Du and Hongxin Xu. Project administration was done by Tianping Li. Software was done by Sijie Du. Validation was done by Hongxin Xu. Writing the original draft was done by Sijie Du. Writing the review and editing were done by Tianping Li. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

This work was supported in part by the NSFC (61572286 and 61472220), the NSFC Joint with Zhejiang Integration of Informatization and Industrialization under Key Project (U1609218), and the Fostering Project of Dominant Discipline a Talent Team of Shandong Province Higher Education.

References

- [1] S. R. Jondhale and R. S. Deshpande, "Modified Kalman filtering framework based real time target tracking against environmental dynamicity in wireless sensor networks," *Ad Hoc & Sensor Wireless Networks*, vol. 40, no. 1-2, pp. 119–143, 2018.
- [2] S. R. Jondhale and R. S. Deshpande, "Kalman filtering framework-based real time target tracking in wireless sensor networks using generalized regression neural networks," *IEEE Sensors Journal*, vol. 19, no. 1, pp. 224–233, 2019.
- [3] S. R. Jondhale and R. S. Deshpande, "Efficient localization of target in large scale farmland using generalized regression neural network," *International Journal of Communication Systems*, vol. 32, no. 16, 2019.
- [4] S. R. Jondhale and R. S. Deshpande, "GRNN and KF framework based real time target tracking using PSOC BLE and smartphone," *Ad Hoc Networks*, vol. 84, pp. 19–28, 2019.
- [5] R. J. Satish and S. D. Rajkumar, "Self recurrent neural network based target tracking in wireless sensor network using state observer," *International Journal of Sensors, Wireless Communications and Control*, vol. 9, no. 2, pp. 165–178, 2019.
- [6] A. Carrio, C. Sampedro, A. Rodriguez-Ramos, and P. Campoy, "A review of deep learning methods and applications for unmanned aerial vehicles," *Journal of Sensors*, vol. 2017, Article ID 3296874, 13 pages, 2017.
- [7] Z. Xiang, T. Tao, L. Song et al., "Object tracking algorithm for unmanned surface vehicle based on improved mean-shift method," *International Journal of Advanced Robotic Systems*, vol. 17, no. 3, 2020.
- [8] C. Jung, S. Shin, J. Lee, and W. Kim, "Objective evaluation of image decomposition algorithms for depth map upsampling," *Journal of Electrical Engineering and Technology*, vol. 14, no. 6, pp. 2619–2624, 2019.
- [9] X. Sun, W. Wang, D. Li, B. Zou, and H. Yao, "Object contour tracking via adaptive data-driven kernel," *EURASIP Journal on Advances in Signal Processing*, vol. 2020, no. 1, 2020.
- [10] X. Zeng, L. Xu, Y. Cen, R. Zhao, S. Hu, and G. Xiao, "Visual tracking based on multi-feature and fast scale adaptive kernelized correlation filter," *Ieee Access*, vol. 7, pp. 83209–83228, 2019.
- [11] W. Kim, S. W. Moon, J. Lee, D. W. Nam, and C. Jung, "Multiple player tracking in soccer videos: an adaptive multiscale sampling approach," *Multimedia Systems*, vol. 24, no. 6, pp. 611–623, 2018.
- [12] W. Ge, Z. Guo, Y. Dong, and Y. Chen, "Dynamic background estimation and complementary learning for pixel-wise foreground/background segmentation," *Pattern Recognition*, vol. 59, pp. 112–125, 2016.
- [13] V. Karavasili, K. Blekas, and C. Nikou, "A novel framework for motion segmentation and tracking by clustering incomplete trajectories," *Computer Vision and Image Understanding*, vol. 116, no. 11, pp. 1135–1148, 2012.
- [14] J. H. Lee, H. J. Jung, and J. Yoo, "A real-time face tracking algorithm using improved CamShift with depth information," *Journal of Electrical Engineering & Technology*, vol. 12, no. 5, pp. 2067–2078, 2017.
- [15] S. Moontaha, A. Galka, M. Siniatchkin et al., "SVD square-root iterated extended Kalman filter for modeling of epileptic seizure count time series with external inputs," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Berlin, Germany, 2019.
- [16] G. Du, Y. Lei, H. Shao, Z. Xie, and P. Zhang, "A human-robot interface using particle filter, Kalman filter, and over-damping method," *Intelligent Service Robotics*, vol. 9, no. 4, pp. 323–332, 2016.
- [17] C. Gao, H. Chen, and T. Li, "Target tracking based on Camshift algorithm and multi-feature fusion," *Journal of Physics Conference*, vol. 1550, article 032015, 2020.
- [18] W. Howard, N. S. Kiong, and W. Jiwei, "Robust video tracking algorithm: a multi-feature fusion approach," *IET Computer Vision*, vol. 12, no. 5, pp. 640–650, 2018.
- [19] W. Mao, H. Xie, Z. Tan, Z. Liu, and M. Liu, "High precision indoor positioning method based on visible light communication using improved Camshift tracking algorithm," *Optics Communications*, vol. 468, 2020.
- [20] X. Cui, Z. Jing, M. Luo, Y. Guo, and H. Qiao, "A new method for state of charge estimation of lithium-ion batteries using square root cubature Kalman filter," *Energies*, vol. 11, no. 1, p. 209, 2018.
- [21] N. Dionelis and M. Brookes, "Phase-aware single-channel speech enhancement with modulation-domain Kalman filtering," *IEEE/ACM Transactions on Audio Speech & Language Processing*, vol. 26, no. 5, pp. 937–950, 2018.
- [22] H. Wang, M. W. Ren, and J. Y. Yang, "New infrared object tracking algorithm based on smog model," *Journal of Infrared & Millimeter Waves*, vol. 27, no. 4, pp. 252–256, 2008.
- [23] G. Du and P. Zhang, "A novel human-manipulators interface using hybrid sensors with Kalman filter and particle filter," *Robotics and Computer-Integrated Manufacturing*, vol. 38, pp. 93–101, 2016.
- [24] I. S. Topkaya and H. Erdogan, "Using spatial overlap ratio of independent classifiers for likelihood map fusion in mean-

- shift tracking,” *Signal Image & Video Processing*, vol. 13, pp. 61–67, 2018.
- [25] C. Wang, H. Chen, X. Zhang, and C. Meng, “Evaluation of a laying-hen tracking algorithm based on a hybrid support vector machine,” *Journal of Animal Science & Biotechnology*, vol. 7, 2017.
- [26] J. Radosavljević, D. Klimenta, M. Jevtić, and N. Arsić, “Optimal power flow using a hybrid optimization algorithm of particle swarm optimization and gravitational search algorithm,” *Electric Power Components & Systems*, vol. 43, no. 17, pp. 1958–1970, 2015.
- [27] I. A. Iswanto and B. Li, “Visual object tracking based on mean-shift and particle-Kalman filter,” *Procedia Computer Science*, vol. 116, pp. 587–595, 2017.
- [28] S. W. Wang, L. I. Yan-Jun, and K. Zhang, “Multimodal IR 3-d target tracking based on non-uniform sampling,” *Journal of Infrared and Millimeter Waves*, vol. 28, no. 6, pp. 445–449, 2009.
- [29] G. Wu, B. Li, Q. Zhu, M. Huang, and Y. Guo, “Using color and 3D geometry features to segment fruit point cloud and improve fruit recognition accuracy,” *Computers and Electronics in Agriculture*, vol. 174, p. 105475, 2020.
- [30] A. A. Masoud, “Motion planning with gamma-harmonic potential fields,” *IEEE Transactions on Aerospace & Electronic Systems*, vol. 48, no. 4, pp. 2786–2801, 2016.
- [31] J. Xia, W. Rao, W. Huang, and Z. Lu, “Automatic multi-vehicle tracking using video cameras: an improved CAMShift approach,” *KSCE Journal of Civil Engineering*, vol. 17, no. 6, pp. 1462–1470, 2013.
- [32] S. G. Lee and J. H. Jo, “An object tracking system for PTZ cameras using CAM shift,” in *Proceedings of the 6th WSEAS international conference on Computer Engineering and Applications, and Proceedings of the 2012 American conference on Applied Mathematics*, Korea, 2012.
- [33] A. Yilmaz, “Kernel-based object tracking using asymmetric kernels with adaptive scale and orientation selection,” *Machine Vision and Applications*, vol. 22, no. 2, pp. 255–268, 2011.
- [34] B. Yousefi, C. I. Castanedo, X. P. V. Maldague, and G. Beaudoin, “Assessing the reliability of an automated system for mineral identification using LWIR Hyperspectral Infrared imagery,” *Minerals Engineering*, vol. 155, p. 106409, 2020.
- [35] L. Yu and J. Y. Hou, “Large-screen interactive imaging system with switching federated filter method based on 3D sensor,” *Complexity*, vol. 2018, Article ID 8730281, 11 pages, 2018.
- [36] P. Shan, “Image segmentation method based on K-mean algorithm,” *EURASIP Journal on Image and Video Processing*, vol. 2018, no. 1, 2018.
- [37] C. Y. Zhang, L. Chen, and R. B. Gou, “High-speed vision extraction based on the CamShift algorithm,” *Cluster Computing*, vol. 22, pp. 555–564, 2017.
- [38] X. Yang, W.-A. Zhang, A. Liu, and L. Yu, “Linear fusion estimation for range-only target tracking with nonlinear transformation,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6403–6412, 2020.
- [39] N.-S. Zheng, L.-S. Cai, H.-F. Bian, and C. Lin, “Hybrid particle filtering algorithm for GPS multipath mitigation,” *Transactions of Nonferrous Metals Society of China*, vol. 24, no. 5, pp. 1554–1561, 2014.
- [40] W. Gao, Y. Tang, and M. Zhu, “Study on the cascade classifier in target detection under complex background,” *Acta Physica Sinica*, vol. 63, no. 9, p. 9, 2014.