Journal of Ubiquitous Computing and Communication Technologies (ISSN: 2582-337X)
www.irojournals.com/jucct/

Improved Adaboost-Camshift Face Tracking System in Complex Background

Xin-ni Li[1], Ya-jun Wang[2]

Electronics and Communication Engineering, School of Electronic and Information Engineering, Liaoning University of Technology, Liaoning, Jinzhou, China

E-mail: [1]lixinni1564@163.com, [2]wyj123126@126.com

Abstract

With the rapid growth of science and technology, people pay more and more attention to pattern recognition and computer interaction. Therefore, in the last few years, face detection and tracking technology in video sequence has become a hot topic for people to study. Face tracking detection has a wide application prospect in human-computer interaction, intelligent monitoring, video conference and other aspects. In this paper, the problem of face tracking in video sequences is divided into two aspects: face detection and moving object tracking algorithm. In the face detection problem, the face detection based on Adaboost algorithm is described in detail, and the three-frame difference method is added to make the algorithm better and enhance the speed of face detection. In terms of moving object tracking algorithm, Camshift face tracking algorithm based on color histogram is adopted, which is not affected by the shape and size of the target and has good real-time performance. However, under the influence of color interference and occlusion, the algorithm will make tracking errors. Therefore, Kalman filter is introduced. The algorithm can directly delineate the candidate areas of face to be detected, so as to ensure the feasibility of face tracking. The simulation video image face tracking system is verified by Matlab software. The experimental results show that the system can accurately detect and track the faces in the video image sequence, not only in the simple background, but also in the complex background and multiple faces can also be well detected and tracked, and the tracking ensures real-time performance.

Keywords: Face detection; Face tracking; Adaboost algorithm; Camshift algorithm; Kalman filter

DOI: https://doi.org/10.36548/jucct.2023.1.004
Received: 01.02.2023, received in revised form: 01.03.2023, accepted: 15.03.2023, published: 25.03.2023

1. Introduction

With the rapid economic growth, the application of various intelligent systems is more and more popular. Intelligent systems often require identification to provide personalized services. Identification is a very difficult problem. Traditional methods mainly identify things with personal identity characteristics, such as identifiers such as certificates and keys, or identity knowledge such as user names and passwords. In more demanding systems, these two approaches are often used together. For example, when using an ATM, as long as the ATM card and password are correct, the specific user account can be accessed and a series of operations can be performed. However, the disadvantages of traditional authentication methods are very obvious. For example, items that can be identified are easily lost or forged, and identifying marks are easily forgotten. In this case, using each person's own biometrics (such as facial features, fingerprints, etc.) can work well [1].

At present, face recognition technology has become a biological authentication means, as the world's most potential means of authentication. The premise of its application is the need for automatic identity identification system with a certain image recognition ability. In order to achieve this purpose, many researchers have carried out in-depth research on it, and it as an independent subject. Face detection is widely used, it has a wide range of uses in many aspects, such as image monitoring, digital image processing, image detection and other aspects, which have very important application value.

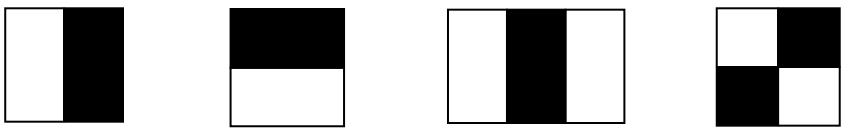
Following the rapid expansion of science and technology, face detection technology and tracking methods are numerous, and constantly new technical methods have been proposed, but, to be able to accurately identify and track the face in any background is very difficult, not only because the facial features of the face is complex, but the face detection and tracking involve many aspects of knowledge. This technology involves not only general image processing, but also the understanding and description of image processing, artificial intelligence, neural networks, monitoring systems, etc. Therefore, face detection and face tracking in specific application background are urgent problems to be solved in this field [2].

These problems are being deeply discussed and studied in all nations. There are a lot of foreign universities that conduct studies on face detection, such as MIT, CMU and other famous universities, and in the home country, Tsinghua University, Beijing Institute of Technology, etc., are conducting relevant research. On the other hand, the MPEG-7 standards body has set up a draft face recognition panel to collect face detection algorithms. With the in-depth study of face detection, a large number of related papers published has increased significantly. Every year, there are a lot of papers on face detection, accounting for one third of research papers on people's face, it can be seen that the world's attention to face detection technology has been widely advanced.

2. Face detection based on Adaboost

2.1 Haar-like features

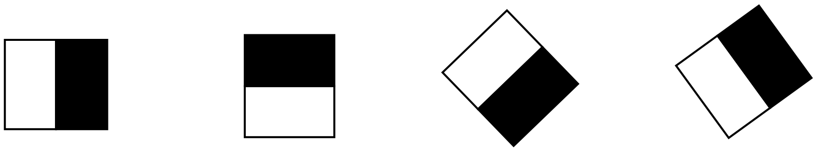
At first, Haar-like features [3] came in the form of two rectangular features, three rectangular features and diagonal features, with only black and white rectangles in the module. The characteristic value of Haar-like is white rectangle pixel sum minus black rectangle pixel sum. For example, in Figure 1(a), (b) and (d), the three types of characteristic values are: Sumwhite-Sumblack, and the characteristic values of class (c) in Fig. 1 are: Sumwhite-2Sumblack. Double the number of pixels in the black area is used to ensure that the number of pixels in the black and white rectangles is the same. The eigenvalue of Haar-like features reflects the grayscale variation of the picture, and so Viola and Jones use four formal features of three types as shown in Figure 1, (a), (b), (c), and (d).



(a) (b) (c) (d)

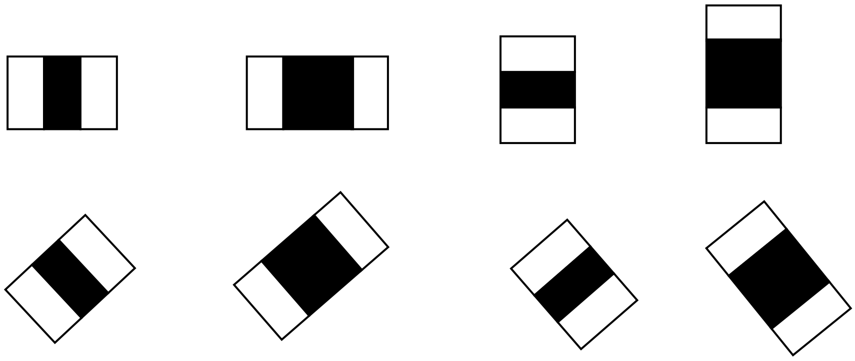
Figure 1. Four Haar-like characteristic forms

After Viola and Jones used Haar-like features to implement a face detection system, many researchers studied Haar-like features deeply and extended them. Among them, the most famous is the Haar-like feature with rotation of 45° proposed by Lienhart et al. Figure 2 is an extension of the Haar-like feature.



(a) (b) (c) (d)

(1) Edge features



(2) Linear characteristics

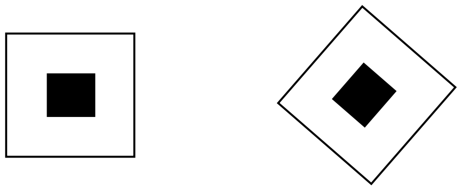


Figure 2. Extended Haar-like features

More rectangular features can be derived from each rectangular feature listed in Figure 2. The vertical and horizontal rectangular features can be arbitrarily scaled in the direction of their location, and the rectangular features with a 45° tilt can be arbitrarily scaled along the direction of their tilt angle. These rectangular features can also be translated within the target template. The target window is W in length and H in width, then the size of the window is L×H, then the size of the rectangular feature is l×h, A=L/ l, D=H/m, and A and D are called maximum scaling factors.

The number of eigenvalues M derived from vertical or horizontal eigen archetypes is：

$$M = AD\left(L+1-l\frac{A-1}{2}\right)\left(H+1-\mathrm{h}\frac{D+1}{2}\right) \quad (2.1)$$

The number of eigenvalues N derived from the characteristic prototype of 45° inclination angle is:

$$N = AD\left(L+1-(l+h)\frac{A+1}{2}\right)\left(H+1-(l+h)\frac{D+1}{2}\right) \quad (2.2)$$

Since the workload of a large number of Haar-like eigenvalues is very heavy, the concept of integral graph is introduced to quickly calculate the Haar-like features, which reduces the amount of computation.

2.2 Integral graph

The calculation of Haar-like features requires repeated calculation of pixel values in the target region, and the use of integral map greatly reduces the amount of calculation, which provides convenience for the calculation of Haar-like features. That is to say, first of all, an integral graph needs to be constructed, and then any Haar-like rectangle feature can be obtained through table lookup or finite simple operations, which greatly reduces the number of operations and the amount of computation.

The integral of an image is essentially a sum, as the image is composed of discrete pixels. Each point in the integral image graph represents the sum of all pixel values in the upper left corner of that point in the original image. First, an array A is created as an integer image so that its width and height are equal to the original image. Then this array is allocated, and the pixels of the rectangle formed by this point and the origin of the image are stored by each point:

$$SAT(\mathrm{x}, y) = \sum_{x_i \le x, y_i \le y} I(x_i, y_i) \tag{2.3}$$

In the formula, I (x, y) indicates the pixel value of the image at (x, y).

For example, the calculation of a group of Haar-like edge features is considered. The form of Haar-like edge features is shown in Figure 3.

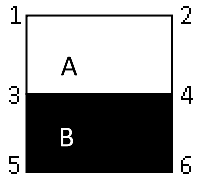


Figure 3. Haar-like feature form graph

Then, the Haar-like edge features formed by regions A and B are:

$$\begin{aligned} Haar_{A-B} &= Sum(A) - Sum(B) \\ &= [SAT_4 + SAT_1 - SAT_2 - SAT_3] - [SAT_6 + SAT_3 - SAT_4 - SAT_5] \end{aligned} \tag{2.4}$$

Obviously, for a grayscale map, its integral map should be built in advance. To calculate the pixel value of all pixel points in a certain area, the integral chart is used to quickly get the result by checking the table of operations.

2.3 Classifier training based on Adaboost algorithm

In order to increase the speed of face detection and reduce the difficulty of face detection, the complex classifier is divided into several simple classifiers and simple classifiers for screening; the new classifier after screening forms a complex classifier, and so on; these complex classifiers cascade into a face classifier to achieve face detection. And those classifiers that have been simplified are weak classifiers, which are defined as:

$$h_j(x) = \begin{cases} 1, & p_j k_j(x) < p_j Q_j \\ 0, & \text{other} \end{cases} \tag{2.5}$$

Pj in the formula is a parameter set to control the direction of unlimited, and kj(x) is a function used to calculate the Haar-like eigenvalue, Qj. To train the classifier, the threshold of Haar-like eigenvalue is obtained, that is, the threshold of weak classifier. 1 and 0 represent consecutive outputs, 1 represents the face and 0 represents non-human faces.

Strong classifier is the complex classifier mentioned above, which is formed by the combination of several weak classifiers, which is defined as:

$$H_t(x) = \begin{cases} 1, & \left(\sum_{j=1}^{N} h_j(x) \right) > \mathrm{b}_t \\ 0, & \text{other} \end{cases} \tag{2.6}$$

The output of Ht(x) can be 0 or 1. When the result of the output is 0, it means reject, and when the output result is 1, it means accept.

Adaboost algorithm [4] is a typical algorithm with repeated feedback process. Its core idea is to train plenty of samples in the same training set. Different weak classifiers are trained and then cascaded to form a strong classifier. The implementation process of Adaboost is: firstly, each sample in the same training set is divided into different categories by using the classifier. At the same time, the classification of each sample is correct or wrong after recording classification, and the accuracy of the overall classification is recorded, which is used as the basis to determine the weight of each sample. The image with modified weight value is added to the original training set instead of the original image to form a new training set. The above classification process is repeated to obtain a series of weak classifiers in turn. Finally, the weak classifiers obtained in each training are fused to generate a strong classifier.

In the concrete realization, the main steps are:

(1) Input sample set N={(x1,O1),(x2,O2),(x3,O3),(x4O4)... (xn,On)}, where Oi =yi is the sample category, yi belongs to {-1, 1}, that is, positive samples and negative samples respectively, n is the overall number of samples.

(2) Initialize the sample weight, and assign D1 to the average weight of positive and negative samples;

(3) For iteration number t= 1,2,3,4... T,

1. Based on the initialized sample weights, a vertical line is used as the classifier to obtain a sub-classifier H1;

2. According to the classification results of the weak classifier model H1, the weight of misclassified samples is re-increased, and a new sample weight D2 is updated. By using the new weight, a horizontal line is used as the classifier, and a sub-classifier H2 is obtained.

3. Then, according to the classification results of weak classifier H2, the sample weight is re-adjusted to obtain a new sample weight D3, and a sub-classifier H3 is obtained by using the new weight to make a vertical line classification.

4. Repeat steps (1-3) until the classification accuracy or number of iterations is reached, and then combine all weak classifiers linearly.

AdaBoost is a classifier with very high accuracy. Various methods can be used to construct sub-classifiers. Weak classifiers are very simple to construct and do not need feature screening, and overfitting can be ignored. The purpose of weight modification is to weaken some unimportant training feature data. Of course, some key training data can also be reused by increasing the weight. The weak classifier of Adaboost algorithm is obtained by extracting the Haar-like features of the face, so that several haar-like features can be used to describe the facial features of the face, and these features are extracted by using the iterative algorithm.

2.4 Motion detection based on three frame difference method

Face detection mainly includes two parts: motion detection and face detection. Motion detection is mainly used for preliminary screening of face areas, in order to speed up the face detection speed. Face detection mainly uses Adaboost algorithm for face detection, and adopts three-frame difference method for motion detection [5].

The three-frame difference method is an enhancement of the two-frame difference algorithm for analyzing video sequences. It involves selecting three consecutive frames and computing the differences between adjacent frames. These differences of images are then thresholded to produce a binary image, which highlights areas of motion. By comparing these binary images, the contour information of moving objects can be extracted. The operation process of the three-frame difference method is shown in Figure 4.

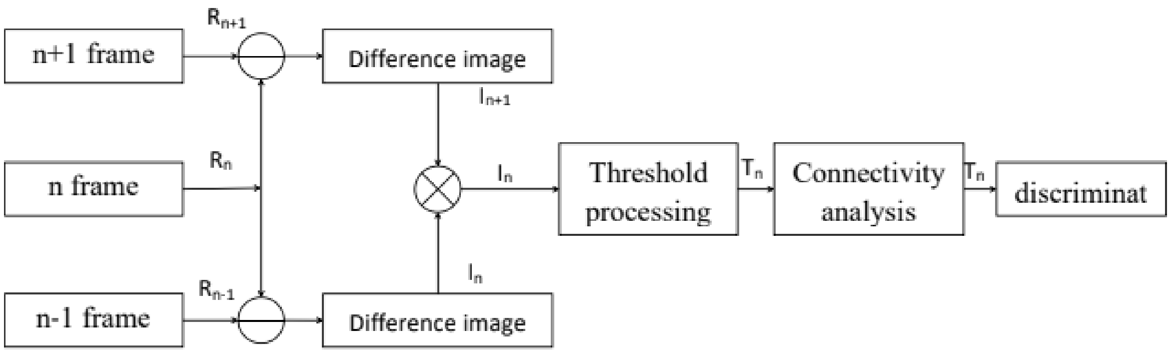


Figure 4. Diagram of three-frame difference method

According to the idea of two-frame difference method, any three connected frames are denoted as Rn+1, Rn and Rn−1 respectively, and the gray values of pixels corresponding to each frame are denoted as rn+1(x,y), rn(x,y) and rn−1(x,y) respectively. Differential images In+1 and In are obtained according to the equation (2.7).

$$I_n(x,y) = \left|R_n(x,y) - R_{n-1}(x,y)\right| \quad (2.7)$$

The difference image In+1 and In are logically operated according to the Formula (2.8) to obtain the image In'. Then threshold processing and connectivity analysis are performed to extract exercise goals.

$$I_n^{'}(x,y) = \left|r_{n+1}(x,y) - r_n(x,y)\right| \cap \left|r_n(x,y) - r_{n-1}(x,y)\right| \quad (2.8)$$

3. Face tracking based on improved Camshift algorithm

3.1 CamShift algorithm

MeanShift, an effective feature space analysis method, is widely used in image filtering, image segmentation and object tracking. However, the size of MeanShift's detection window is fixed, and the target is a process of decreasing gradually from near to far, hence a fixed window is not appropriate. So, the size and angle of the window should be adjusted to match the size and angle of the target. The CamShift algorithm is an enhanced version of the MeanShift algorithm designed to address this issue. It can dynamically adapt the search window size to match changes in the tracked object's dimensions, resulting in improved tracking performance.

CamShift algorithm utilizes a color histogram model of the target to create a probability distribution map, and then initializes a search window with a specific size and position. It dynamically adjusts the search window's position and size based on the previous frame's results to locate the target's center in the current image. The CamShift algorithm tracks a target in a video by repeatedly applying the MeanShift operation to each frame. It uses the size and center of the search window from the previous frame as the initial values for the search window in the current frame. By iterating this process, the algorithm can track the target over time. The process is as follows:

(1) Initialize the search window;

(2) Calculate the probability distribution of colors within the search window, use the MeanShift algorithm to iterate over the rectangular window, calculate the centroid, zero order moment, and first order moment of the search window, and save them;

(3) Update the location and size of the search window using the centroid and zero-order moments obtained from (2), where the updated search window has a height of 1.2 s and a width of s;

$$s = 2 \times \sqrt{\frac{M_{00}}{256}} \tag{3.1}$$

4) Re-apply the updated search window obtained to the next frame of video sequence, and then re-perform the operation of step (2) to continue the next cycle.

Face tracking system based on CamShift algorithm uses H component of face skin color to build color probability distribution map, and calls CamShift algorithm in back projection map for face tracking. CamShift algorithm has a fast iteration speed and can continuously modify the position and size of the tracking window in an adaptive manner, thus ensuring the real-time tracking of video sequences [7].

However, CamShift algorithm also has its own shortcomings. If there are objects with similar colors around the tracking target, the algorithm will be interfered by the surrounding objects, resulting in tracking failure.

3.2 CamShift algorithm based on Kalman filter

Kalman filter is an algorithm which can be applied to linear equation and discrete equation [8]. Based on Kalman filter theory, a discrete system that can be described using linear stochastic differential equations and system measurements is introduced. Let the moment of the system be e, then the system is expressed as:

$$X(e) = SX(\mathrm{e}-1) + GU(\mathrm{e}) + W(e) \tag{3.2}$$

$$Z(e) = WX(e) + V(e) \tag{3.3}$$

When all the conditions mentioned above are satisfied, Kalman filter can be used as the optimal information processor. Next, the optimal output of the system is calculated using the covariance Q and R of the system. To predict the next state of the system, the process model of the system must be used first. It is assumed that the system is now in the state of moment e. According to the formula of the system model, the previous state can be used to predict the state of the system at this moment. Then, the equation of state of the system at this moment (that is, moment e) can be obtained:

$$X(e) = SX(e-1) + GU(e) \tag{3.4}$$

The covariance P(e) of the system prediction is updated as:

$$P(e) = SP(e-1)S^T + Q \tag{3.5}$$

Equations (3.4) and (3.5) complete the prediction of the system. The measured value of the current state must be combined to get the optimal estimate of the current state.

$$X(e) = X(e) + K_g(e)\left[Z(e) - WX(e)\right] \tag{3.6}$$

After obtaining the optimal value of the current state, in order to ensure that Kalman filter can maintain the running state before the end of the system running process, it is also necessary to update the covariance P(e) under the current state.

$$P(e) = [I - K_g(e)W]P(e) \tag{3.7}$$

Due to the characteristic of Kalman filter that can predict the location of the aim in the current image, it can be embedded in the Camshift algorithm during the tracking process of the system to improve the Camshift algorithm, thus tracking the target more accurately. Therefore, the state vector can be set as: L(k)=[x(k),y(k),sx(k),sy(k)], where x(k) and y(k) respectively represent the x and y components of the centroid coordinates in the current state, and sx(k) and sy(k) represent the x and y components of the moving target velocity respectively [8]. Let Z(k)=[xz(k),yz(k)]T represent the observation vector in the current state, then the signal model and observation model in practical application can be expressed as:

$$\begin{bmatrix} x(k) \\ y(k) \\ s_x(k) \\ s_y(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & t & 0 \\ 0 & 1 & 0 & t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x(k\text{-}1) \\ y(k\text{-}1) \\ s_x(k\text{-}1) \\ s_y(k\text{-}1) \end{bmatrix} + \begin{bmatrix} \frac{t^2}{2} \\ \frac{t^2}{2} \\ t \\ t \end{bmatrix} w(k) \tag{3.8}$$

$$\begin{bmatrix} x_Z(k) \\ y_Z(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} x(k) \\ y(k) \\ s_x(k) \\ s_y(k) \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} s(k) \tag{3.9}$$

The Kalman filter is capable of estimating the current motion state of a target [9]. The specific tracking process can be divided into:

(1) Initialization of the filter;

(2) The signal model of Kalman filter (Eq. 3.8) and the state information of the aim in the previous frame are used to predict the state of the goal in the current frame and set the search area;

(3) Camshift algorithm is used to search for the best matching template based on the probability distribution features of colors of the aim in the search area set in the previous step;

(4) Take the target information obtained in the previous step as the observation of Kalman filter to update its status, to predict the target of the next frame more accurately.

When the moving target is seriously blocked, the target position information searched by Camshift is not accurate, and updating the Kalman filter status with Camshift may lead to the error estimation of the moving target status by the filter. To solve this problem, the predicted value of Kalman filter is used to replace the optimal value calculated by Camshift as the observation to update the state of the filter. The residue r(k) between the predicted value obtained by Kalman filter and the observed value calculated by Camshift algorithm can be used to determine whether the object has serious occlusion.

$$r(k) = \sqrt{\left(x_Z(k) - x'(k)\right)^2 + \left(y_Z(k) - y'(k)\right)^2} \qquad (3.10)$$

When the value of r(k) is small, it means that the target has no serious occlusion. When r(k) exceeds a certain threshold, the target may have serious occlusion. If there is a serious occlusion of the aim, the area of the target tracked by Camshift algorithm becomes significantly smaller, and a decision condition can be further obtained to determine whether there is occlusion.

$$R < \lambda R_0 \qquad \lambda \in (0,1) \qquad (3.11)$$

When r(k) meets certain threshold conditions and formula (3.11) is established, it is determined that the moving target has serious occlusion. At this time, the target state calculated by Camshift algorithm cannot be used to update Kalman filter, and the predicted value of Kalman filter should be used to update the state. Since the state information used at this time is mainly generated by the motion state of the target before it is blocked, it has high credibility. In the tracking process of subsequent frames, if the goal is still in severe occlusion, the forecast value of Kalman filter will continue to be used to update the filter state [10].

4. Implementation of video image face tracking system

4.1 Video image face tracking system flow

The improved Adaboost-Camshift algorithm adopted herein is mainly used to track the face of the people in the video array. Figure 5 shows the overall flow chart of the algorithm.

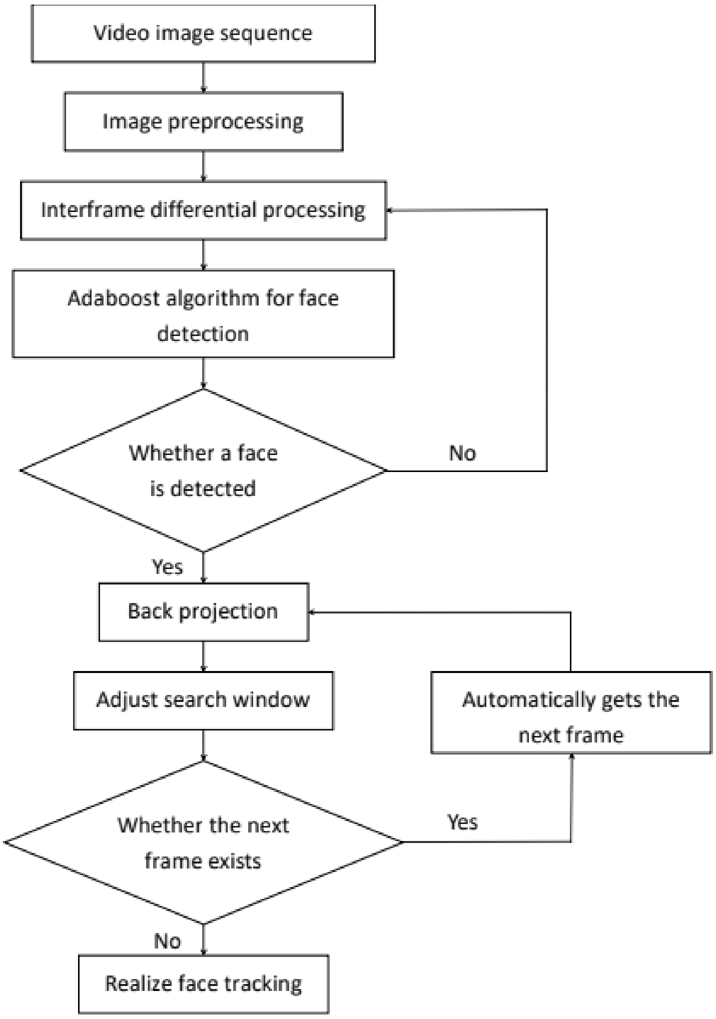


Figure 5. Flow chart of improved Adaboost-Camshift algorithm

The improved Adaboost-Camshift algorithm is mainly divided into the following aspects:

(1) Face detection part: Image pre-processing, three-frame difference method is used to screen face candidate regions, and face color classifier trained by Adaboost algorithm is used to realize face detection.

(2) Face tracking part: RGB space is converted into HSV space, and H component histogram is used for reverse projection, the search window is selected and the center of the search window is adjusted to the centroid, and the face position in the next frame is predicted using a Kalman filter. The improved Camshift algorithm is used to realize face tracking.

4.2 Experimental results of video image face tracking system

As seen in Figure 6, the experimental results of Adaboost algorithm for face detection are as follows. According to the above results, the algorithm adopted in this article can accurately detect the faces in the image, even if there is a side face or poor lighting in the image.

(a) (b) (c) (d)

The improved Adaboost Camshift method proposed in this article can accurately track faces in video images, successfully solving the problem of false tracking of faces caused by occlusion. Fig. 7 is a group of image examples that use this algorithm to track faces in video images.

Figure 7. Example of face tracking in video image

The video sequence adopted in the experiment is a 10 second dynamic video sequence. The background in the video sequence is a complex background with rich colors, and there is a fence in the background with a color very similar to the skin color. The six pictures in Figure 7 are periodically captured at the 0, 2, 4, 6, 8 and 10 seconds of the 10 second video. The six pictures clearly demonstrates that the Adaboost algorithm can accurately record the human face in the video. Even the hair occlusion in Figure 7(c) and hand occlusion in Figure 7(d) have no impact on the face detection and face tracking system, and the fences similar to skin color in each picture have no impact on the system for false tracking, and face has been accurately tracked.

A total of 10 10 second dynamic video sequences were tested in this experiment, totalling 100 seconds. The Adaboost-Camshift algorithm [10] is compared with the improved Adaboost-Camshift algorithm, and the results have been shown in Table 1. It can be clearly seen that the improved tracking algorithm has better real-time performance, and the error tracking rate is reduced, which can meet the needs of face tracking and detection.

Table 1. Tracking performance comparison

	Average time per frame/ms	**Mistracking rate**
Adaboost-Camshift	23	6.5%
Improved Adaboost-Camshift	21	2%

5. Conclusion

In this paper, the face detection part using Adaboost algorithm, at the same time added three frame difference method to screen the face area to improve the algorithm, by improving the detection speed of face detection. For the whole system, the improved Camshift algorithm and Adaboost algorithm are combined. The combination of the two algorithms make the system successful in automatically tracking faces. Compared with the traditional face tracking algorithm, it is found that the proposed method shortens the tracking time, improves the detection accuracy, and achieves good results.

References

[1] Adjabi, Insaf, et al. "Past, present, and future of face recognition: A review." Electronics 9.8 (2020).

[2] Dong, Yinpeng, et al. "Efficient decision-based black-box adversarial attacks on face recognition." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[3] Devi, B. T., & Shitharth, S. (2021). Multiple Face Detection Using Haar-AdaBoosting, LBP-AdaBoosting and Neural Networks. In IOP Conference Series: Materials Science and Engineering (Vol. 1042, No. 1, p. 012017). IOP Publishing.

[4] Chengsheng, T., Huacheng, L., & Bing, X. (2017). AdaBoost typical Algorithm and its application research. In MATEC Web of Conferences (Vol. 139, p. 00222). EDP Sciences.

[5] Cheng, Y. H., & Wang, J. (2014). A motion image detection method based on the inter-frame difference method. In Applied Mechanics and Materials (Vol. 490, pp. 1283-1286). Trans Tech Publications Ltd.

[6] NaNa, Z., & Jin, Z. (2018). Optimization of face tracking based on KCF and Camshift. Procedia computer science, 131, 158-166.

[7] Ramakrishna, B. B., & Sharmila Kumari, M. (2019). Human face tracking using normalized CamShift face tracker. Journal of Computational and Theoretical Nanoscience, 16(4), 1485-1488.

[8] Kim, Y., & Bang, H. (2018). Introduction to Kalman filter and its applications. Introduction and Implementations of the Kalman Filter, 1, 1-16.

[9] Xiu, C., Su, X., & Pan, X. (2018, June). Improved target tracking algorithm based on Camshift. In 2018 Chinese control and decision conference (CCDC) (pp. 4449-4454). IEEE.

[10] Kim, J. H., Kim, J. Y., Hong, Y. J., Kwon, J. W., Kang, D. J., & Lho, T. J. (2007). AdaBoost-based Real-Time Face Detection & Tracking System. Journal of Institute of Control, Robotics and Systems, 13(11), 1074-1081.

Author's biography

Xin-ni Li is currently studying for a postgraduate degree at Liaoning University of Technology in Jinzhou, China. Her research topic is multivariate statistical modeling and fault diagnosis and their applications in industrial processes.

Ya-jun Wang was born in Huludao, China in 1978. She received her doctorate in control theory and control engineering from Northeastern University in 2015. She is a professor at Liaoning University of Technology in Jinzhou, China. She has written more than 30 articles. Her research interests include multivariate statistical modeling, process monitoring and fault diagnosis and their applications in industrial processes.