Research on Multi-feature Adaptive Fusion Face Tracking Algorithm

Qing Lei*, Zhijun Li, Motao Wang, Jun Feng and Rui Zhang
School of Automation, Wuhan University of Technology, Wuhan 430070, China

*525968513@qq.com

Abstract. Under the premise of fixed computer performance, it is necessary to take into account the accuracy and real-time of face tracking. For this detection requirement, the image processing method of skin color segmentation is incorporated into AdaBoost's face detection algorithm to accurately and quickly locate the face position. However, in the previous stage of face detection, there is a case where the real-time detection does not respond fast enough and fails the detection. In this case, a particle filter tracking algorithm based on multi-feature adaptive fusion is proposed, which uses the face area detected in the first frame as the tracking target, and achieves face detection by self-adjusting CS-LBP and the weight of how the skin color influences tracking effect, in which way the computational efficiency of face detection between frames is improved when the detection accuracy is maintained. And it is also robust to various complex external factors. It has been proved by experiments.

1. Introduction
In the video sequence, face tracking belongs to target tracking, that is, to track and locate the target position between different frames of adjacent images with the face as the target. There are many different types of tracking algorithms, such as detection-based, depth-based neural networks and correlation-based filtering.

The detection-based algorithm includes the TLD (Tracking Learning Detection) algorithm proposed by Kalal[1]. The algorithm proposes a tracking-learning-detection framework structure, which uses the optical flow method to achieve short-term tracking of the target, and learns the detected shape features online through the cascade detector, and adaptively calibrates the error of the tracking system. The algorithm based on deep neural network includes the convolutional neural network target tracking algorithm proposed by Wang et al[2]. The tracking model is obtained through the training of a large amount of data, and the target can be tracked more accurately, but the real-time performance is poor. The algorithm based on correlation filtering is proposed by Bolme. Through the adaptive training process, target tracking with strong robustness and high stability can be achieved in only one frame of image[3].

Real-time and accuracy are important indicators for evaluating tracking algorithms. In the field of machine vision technology research, face tracking as the front-end technology plays an interface role in the connection process, so reliability and predictability are still the focus of attention. Therefore, this paper will adopt an adaptive feature fusion particle filter target tracking improvement algorithm to obtain a fast, stable and accurate face tracking method that meets the system requirements.
2. Face detection based on skin color segmentation and AdaBoost

The color reflects the optical properties of the object, and the face skin color has a higher degree of discrimination. It is helpful to limit the face detection area by detecting the face color in advance to improve the detection efficiency.

In the general skin color model, mainly statistical histogram model, Gaussian model, simple threshold model and mixed Gaussian model. After considering and weighing the accuracy and complexity of the model calculation, we chose to use the simple threshold model for skin color segmentation. The threshold of the skin color division of each pixel point \( P(i,j) \) in the image is as follows:

\[
P(i,j) = \begin{cases} 1, \text{while } Cb \in [77,127] \text{ and } Cr \in [133,173] \\ 0, \text{ others} \end{cases}
\]

(1)

After the skin color is divided, the image is converted into a binary image. The effect is that the skin color is determined to be white, and the other regions are black, thereby obtaining the face detection region, and then the face detection is performed by the AdaBoost algorithm [4].

The AdaBoost algorithm flow is as follows:

First, the construction of the \( j \)-th feature in the weak classifier \( h_j \) for each sample can be expressed as:

\[
h_j(x_j, f_j, p_j, \theta_j) = \begin{cases} 1, p_j f_j < \theta_j \\ -1, \text{ others} \end{cases}
\]

(2)

Where \( x_j \) is the window to be tested, \( f_j(x) \) is the eigenvalue calculation function, \( \theta_j \) is the weak classifier eigenvalue threshold obtained by the training, and \( p_j \) determines the inequality orientation.

In the iterative process of AdaBoost, the weight of each training sample represents the probability that the weak classifier will extract the sample into the training set, thus for a given training set \( \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\} \), among them \( x_i \in X, y_i \in \{0,1\} \), \( y_i = 0 \) mark the non-face samples, and \( y_i = 1 \) mark the face samples.

Initialize the weight of the sample, \( \omega_{1,i} = \frac{1}{2k}, y_i = 1 \), \( \omega_{1,i} = \frac{-1}{2s}, y_i = -1 \), where \( k \) is the number of positive samples and \( s \) is the number of negative samples.

In the \( T \) round training process, \( t = 1,2,\cdots,T \) for each round of training, get a weak classifier, and minimize the error rate \( \varepsilon_t \):

\[
\varepsilon_t = \sum_{n=1}^{N} \omega_n^{(t)} I(h_t(x_n) - y_n)
\]

(3)

Calculate \( \alpha_t \), the formula is as follows:

\[
\alpha_t = -\ln \beta_t
\]

(4)

Among them, the update factor \( \beta_t = \frac{\varepsilon_t}{1-\varepsilon_t} \).

Adjust the weight of the training sample:

\[
\omega_{t+1,i} = \begin{cases} \omega_{t,i} \beta_t, x_i \text{ is correctly classified} \\ \omega_{t,i}, \text{ others} \end{cases}
\]

(5)

Through the \( T \) iteration process, the resulting strong classifier consists of the \( T \) optimized weak classifiers with the lowest error rate:

\[
H(x) = \begin{cases} 1, \sum_{t=1}^{T} \alpha_t r_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0, \text{ others} \end{cases}
\]

(6)
After the face detection intercepts and acquires the image of the face region, the face image needs to be normalized.

The trained face detector is used for face detection of real-time video, and the experimental results are shown in figure 1.

![Image](image.png)

**Figure 1.** Face detection based on skin segmentation and AdaBoost algorithm.

The algorithm based on skin color segmentation and AdaBoost can basically obtain a more accurate face region with high accuracy and low false detection rate. However, the FPS is only about 15 in the video, which obviously causes the video to be unsmooth and stuck, and does not meet the real-time requirements. And when the image in the video has a large posture change, local occlusion and illumination changes drastically, the face detection failure in some frames will occur. To this end, it is necessary to introduce the idea of tracking to ensure that the face area is always acquired, at the same time, improve efficiency and enhance real-time performance.

3. **Particle filter target tracking improved algorithm based on adaptive feature fusion**

The emergence of particle filter theory[5] is based on the filtering of Bayesian sampling estimates. Hammersley and Morton et al. first proposed SIS (Sequence Important Sampling) in the 1950s, namely the sequence importance sampling method[6]. However, due to the influence of the number of particles in the iterative process and the large computational complexity, it has not received enough attention. At the end of the last century, in 1993, Gordon and Salmond proposed a new Bootstrap nonlinear filtering method based on SIS[7], which attracted many scholars to study SIS and determined the research basis of particle filtering algorithm.

The particle filter algorithm flow is as follows:

(1) **Initialization**

The stage of initializing the particles is to arrange the particles in an arbitrary or predetermined manner in the image. Generate N particles in the first video frame, set the initial state \( S_0^i = \{x_0^i, w_0^i\} (i = 1, \cdots, N) \), among them \( x_0^i \) is the state variable, \( w_0^i \) is the weight.

(2) **State transition**

The process of mapping the state of a particle in the image of the previous frame to the state of the particle in the current frame is called state transition.

\[
x_k^i = f(x_{k-1}^i) + w_k^i
\]  

Equation (7) is the state transition equation, \( w_k^i \) represents the state noise.

\[
y_k^i = h(x_k^i) + v_k^i
\]  

Equation (8) is the observation equation, \( v_k^i \) represents the state noise.

(3) **Calculating weights**

The formula for calculating the weight is as follows:

\[
w_k^i = w_{k-1}^i p(y_k^i | x_k^i)
\]  

Equation (9) is the observation equation, \( p(y_k^i | x_k^i) \) is the observation likelihood function.
Among them, \( p(y_k^i | x_k^i) \) is the likelihood function under the observation model.

(4) Resampling
The posterior probability density is resampled, leaving particles with larger weights and deleting particles with smaller weights. The following formula is the definition of the normalized weight:

\[
w_k^i = \frac{w_k^i}{\sum_{i=1}^{N} w_k^i}
\]

(10)

(5) State estimation
The tracking results are derived from state variables and weight estimates:

\[
E[X_k] = \sum_{i=1}^{N} x_k^i w_k^i
\]

(11)

In order to improve the accuracy and real-time performance of particle filter tracking face, it is necessary to select appropriate target features. The CS-LBP feature calculation process optimizes the feature dimension, maintains excellent computational complexity, and has a strong ability to describe the target. Therefore, the CS-LBP feature is used as the texture feature in the tracking process.

In the real scene, due to the interference of target movement and background noise, the tracking process for a single target feature is easy to misjudge or lose the target. The method based on multiple target feature information fusion effectively utilizes the advantages of various features, can improve the robustness of the target tracking process, and improve the tracking effect.

Literature [8] uses color features and texture features to describe the tracking target, and fuses the features in the particle filter algorithm, successfully solving the problem of poor stability of color features under illumination and environmental interference. However, the texture features used in it are the most basic LBP features, with some room for improvement.

In the target tracking based on vision system, the limitation of single feature is caused by the change of background environment and the impact of noise, resulting in serious tracking performance degradation. In order to dynamically assign weights of texture features and color features, it can be implemented by adaptive weight calculation, and the calculation formula is as follows:

\[
w_m^i = \alpha \times \text{exp}\{-(d_{\text{skin}}^i)/\sigma^2\} + \beta \times \text{exp}\{-(d_{\text{cs}}^i)/\sigma^2\}
\]

(12)

Where the sum of the weights in the feature fusion is \( \alpha + \beta = 1 \), \( \sigma^2 \) is the variance of Gaussian noise, \( d_{\text{skin}}^i \) and \( d_{\text{cs}}^i \) are the similarity distance between the current state and the target state of the skin color and the texture feature. The formula for calculating the skin color weight \( \alpha \) and the texture weight \( \beta \) is as follows:

\[
\alpha = \frac{f_{\text{skin}}^i}{f_{\text{skin}}^i + f_{\text{cs}}^i}
\]

(13)

\[
\beta = \frac{f_{\text{cs}}^i}{f_{\text{skin}}^i + f_{\text{cs}}^i}
\]

(14)

Among them, \( f_{\text{skin}}^i \) represents the sum of the squares of the similarities between the candidate template of the particle skin color feature and the target template:

\[
f_{\text{skin}}^i = \sum_{i=1}^{N} (d_{\text{skin}}^i)^2
\]

(15)

Similarly, for CS-LBP features:

\[
f_{\text{cs}}^i = \sum_{i=1}^{N} (d_{\text{cs}}^i)^2
\]

(16)

When matching targets, algorithms such as Euclidean distance and Manhattan distance are used to calculate the similarity between templates. However, in the histogram feature, the Bhattacharyya coefficient is commonly used as a measure of similarity. The Bhattacharyya coefficient has a value...
between 0 and 1, which can clearly show the degree of matching of features. Two consecutive distributions can be expressed by the following formula:

$$\rho(p, q) = \int \sqrt{p(u)q(u)} du$$

(17)

After normalizing the color histogram and texture histogram of the region to be tracked, the similarity between the current target region and the starting target region can be expressed as:

$$d(p^n(x,y), q^n(x_0,y_0)) = \sqrt{1 - \rho(p^n(x,y), q^n(x_0,y_0))}$$

(18)

In this formula:

$$\rho(p^n(x,y), q^n(x_0,y_0)) = \sum_{i=1}^{m} \sqrt{p^n_i(x,y) \cdot q^n_i(x_0,y_0)}$$

(19)

Among them, \(p^n(x,y)\) represents the histogram of the normalized candidate region obtained in \(X^k\) at time \(k\), and \(q^n_i(x_0,y_0)\) represents the histogram of normalized starting target region, \(m\) is the number of bins in the histogram. The smaller \(d\) is, the more it can reflect the high similarity between the candidate region histogram and the target region histogram. At this time, the weight of the particles of the candidate region needs to be increased, if \(d\) is larger, the weight of the particle should be larger.

Finally, the weight \(w^i_{ck}\) is normalized and then substituted into the particle filter framework of Equation (11) to derive the state of the current system.

The specific implementation steps of face tracking based on particle filter improved algorithm combined with face detection and multi-feature adaptive fusion are as follows:

1. In the first frame image of the video, the face region of the target is obtained by the face classifier based on the skin color segmentation and the AdaBoost algorithm, and the region is marked.

2. Determining the center coordinate \((x_0,y_0)\) according to the obtained face target region. Calculate the color histogram and texture histogram of the target area respectively, and the initial color feature and initial texture feature and probability distribution \((q_{\text{skin}}(u) \text{ and } q_{\text{csibp}}(u))\) of the target are obtained. Then, select the number of particles \(N\), and the dimension of the state space is positively correlated with the number of particles, so the total amount of particles will affect the face tracking effect. Usually the total number of particles \(N\) is selected according to the needs of the detection system, and then the weight of each particle is initialized to \(w^i_0 = 1/N, i = 1, 2, \cdots, N\), that is, the sum of the total weights of the particles is 1.

3. According to the particle state in the image of the previous frame, the particle state of the latter frame is predicted and estimated. After the state transition process advances, the state vector is updated, but the weight does not change. Therefore, it is necessary to update the state estimates of each particle by using the observation model. The observation is the calculation of the difference between the candidate region histogram and the target region histogram of particles. The skin color feature histogram \((p_{\text{skin}}^n, n = 1, 2, \cdots, m)\) and \((p_{\text{csibp}}^n, n = 1, 2, \cdots, m)\) is calculated sequentially for the feature extraction process mentioned above.

4. Extracting the skin color feature vector and texture vector feature of each particle region in the particle set, calculating the similarity between the histogram feature in each particle candidate template and the histogram feature of the target template, using the Bhattacharyya coefficient mentioned above as the standard for feature similarity metrics.

5. Analyzing the similarity between the candidate template and the target template obtained in the previous step. Combining the degree of difference between the target face and the background region, the weight \(\alpha\) of the skin color and the weight \(\beta\) of the texture are respectively solved by the formula, then substitute the formula to get the adaptive weight of each particle and then normalize it.

6. After obtaining the state vector and weight of the particle according to the state transition model and the observation model, the target position of the current frame is predicted. Maximum criteria and weighting criteria are often used to determine target state and location. The maximum criterion selects the particle state with larger particle concentration weight as the target state. The weighting criterion
involves the weight state of all particles, the particle similarity is proportional to its proportion, and the target state is obtained through the comprehensive calculation of particle importance. Therefore, the weighting criterion can better predict the current position. The calculation formula of the face target state and position is as follows:

$$E[X_k] = \sum_{i=1}^{N} x_k^i w_k^i$$

(20)

(7) During the tracking process, the weight of the particles similar to the target state will continue to increase, and the remaining particle weights will continue to decrease. After a period of time, most of the particles will have a smaller weight and make a few particles become a key factor in the tracking process. This phenomenon is called particle degradation. In order to avoid particle degradation, it is necessary to add a resampling operation, that is, to disassemble the particles with larger weights. This ensures that the total number of particles is constant and the weight of the particles is updated, so the tracking process becomes reliable and continuous.

(8) Output rectangle to lock face target position.

4. Improved face tracking algorithm effect display

Figure 2. Face tracking effect picture in real-time video.

As shown in figure 2, it is a face tracking effect picture in real-time video. The rectangle is the tracked face region, the point is the particle distribution state. The tracking effect shown in the figure is good, and the FPS is stable at around 40, and the real-time performance is greatly improved compared with the direct face detection. The above results show that the particle filter algorithm combined with CS-LBP and skin color adaptive has better accuracy and robustness, and has good real-time performance, which is suitable for tracking human face.

5. Particle filter target tracking improved algorithm based on adaptive feature fusion

In this paper, the face detection algorithm based on skin color segmentation and AdaBoost is used to accurately and quickly capture the face position in a video frame. Aiming at the low real-time detection of face detection and possible detection failure, an improved particle filter tracking algorithm based on adaptive fusion of CS-LBP and skin color features is proposed to track the first frame face of the detected face target. In the particle filter framework, the CS-LBP features that describe the texture feature are combined with the human skin color, and the adaptive weight adjustment method is used to change the influence of the two features on the tracking results. The experimental results show that this method can achieve accurate face tracking while ensuring high real-time performance.

Acknowledgments

This work was supported by the National Key R&D Program of China under Grant No.2017YFB1300502.
References

[1] Kalal Z, Mikolajczyk K and Matas J 2012 Tracking-learning-detection 34 1409-22
[2] Wang L, Ouyang W, Wang X, et al 2016 Visual tracking with fully convolutional networks 3119-3127
[3] Bolme D S, Beveridge J R, Draper B A, et al 2010 Visual object tracking using adaptive correlation filters 2544-2550
[4] Freund Y, Schapire R E 1996 Experiments with a new boosting algorithm 96 148-156.
[5] Arulampalam M S, Maskell S, Gordon N, et al 2002 A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking 50 174-188
[6] Hammersley J M, Morton K W 1954 Poor man's monte carlo 16 23-38
[7] Gordon N J, Salmond D J and Smith A F M 1993 Novel approach to nonlinear/non-Gaussian Bayesian state estimation 140 107-113
[8] Li Yuanzheng, Lu Chaoyang, Gao Quanxue, et al 2010 Particle Filter and Mean Shift Tracking Method Based on Multi-feature Fusion 32 411-415