

Article

Research on a Face Real-time Tracking Algorithm Based on Particle Filter Multi-Feature Fusion

Tao Wang ¹, Wen Wang ², Hui Liu ³ and Tianping Li ^{2,*}

¹ Shandong Key Laboratory of Medical Physics and Image Processing, School of Physics and Electronics, Shandong Normal University, Jinan 250014, China; wwwlljw@163.com

² School of Physics and Electronics, Shandong Normal University, Jinan 250014, China; w851839145@163.com

³ School of Computer Science & Technology, Shandong University of Finance and Economics, Jinan 250014, China; 19862127463@163.com

* Correspondence: sdsdltp@163.com; Tel.: +86-133-0640-8558

Received: 28 December 2018; Accepted: 7 March 2019; Published: 12 March 2019



Abstract: With the revolutionary development of cloud computing and internet of things, the integration and utilization of “big data” resources is a hot topic of the artificial intelligence research. Face recognition technology information has the advantages of being non-replicable, non-stealing, simple and intuitive. Video face tracking in the context of big data has become an important research hotspot in the field of information security. In this paper, a multi-feature fusion adaptive adjustment target tracking window and an adaptive update template particle filter tracking framework algorithm are proposed. Firstly, the skin color and edge features of the face are extracted in the video sequence. The weighted color histogram are extracted which describes the face features. Then we use the integral histogram method to simplify the histogram calculation of the particles. Finally, according to the change of the average distance, the tracking window is adjusted to accurately track the tracking object. At the same time, the algorithm can adaptively update the tracking template which improves the accuracy and accuracy of the tracking. The experimental results show that the proposed method improves the tracking effect and has strong robustness in complex backgrounds such as skin color, illumination changes and face occlusion.

Keywords: video face tracking; particle filter (PF); features fusion; updating model; template drift

1. Introduction

Target tracking technology is an important computer vision research field, which is widely used in the Internet of Things and artificial intelligence [1,2]. Face recognition technology can effectively realize real-time multi-objective online retrieval and comparison in crowded areas such as banks, and the actual application effect is good. Moreover, face information [3] is easy to collect, difficult to copy and steal, and natural and intuitive. Therefore, face recognition technology has become the preferred choice for commercial banks' security prevention and control measures. Many face tracking methods have been proposed in the past few years. At present, two excellent algorithms, namely mean shift [4] and particle filter [5], have been widely used for target tracking. The mean shift algorithm is a kind of deterministic tracking algorithm, which looks for the nearest pattern of points in the sample distribution. The particle filter algorithm is a Monte Carlo simulation method based on non-parametric to achieve recursive Bayesian estimation algorithm, which can effectively solve nonlinear and non-Gaussian state estimation problems. The most popular target tracking cues include color feature [6], edge feature [7], texture feature [8] and motion feature [9]. Each feature has its own advantages and disadvantages in the application. For example, the tracking method based on color features is insensitive to the rotation and posture of the face in which the face can be tracked

in real time. However, when the illumination changes and the skin-like object appears [10] in the scene, it is difficult to accurately track the face, so the face tracking based on a single clue is not robust. Therefore, more and more researchers combine multiple features to improve tracking performance. Particle filter-based tracking algorithms are becoming more and more popular because of their high tracking accuracy and strong anti-interference [11].

The aim of this study is the tracking of a single face in a video in which the images are often affected by complex backgrounds such as illumination change, face rotation and scale change [12], and face occlusion. The adaptive fusion of color and edge features reduces the complexity of the environment. The particle filter target tracking window is adaptively adjusted [13] and the template is adaptively updated [14] in the face tracking system. In addition, the particle filter algorithm together with the histogram algorithm improve the speed of the algorithm [15]. The improved tracking algorithm is capable of adaptively adjusting the [16] tracking window scale to obtain stable tracking of objects with significant dimensional changes. In the tracking process, the template is adaptively updated with the change of the object [17]. The experimental results show that the algorithm has anti-background interference ability, good stability and robustness [18], and finally realizes accurate real-time tracking of the face.

2. Particle Filter Algorithm

The particle filter tracking algorithm is a Bayesian recursive estimation algorithm whose purpose is to construct the posterior probability distribution of the target state. The essence of Bayesian recursive filtering is to use the prior knowledge to construct the posterior probability density function of the random state of the target system, and use some estimation criterion to estimate the state value of the target. When the mean square error of this algorithm is the smallest, it can be considered to be optimal. It is generally assumed that the dynamic model has a first-order Markov state transition property, and the observations are conditionally independent with respect to the state [19]. The initial state probability density function of the target system is assumed to be known, where the initial state vector of the target system the initial information of the system measurement value is expressed. The System State Space Model (SSD: State Space Model) is established as follows:

$$x_k = f_k(x_{k-1}, v_k) \quad (1)$$

$$y_k = h_k(x_k, n_k) \quad (2)$$

Among them, $h(\cdot)$ $f(\cdot)$ are the target state transfer function and the observation function observation equation v_k n_k system noise and observed noise, respectively.

In order to obtain the exact solution of the posterior probability distribution $p(x_k | y_{1:k})$ of the target in each frame, it is necessary to predict and update the target. The target state can be estimated after repeated iterations:

$$\begin{aligned} p(x_k | y_{1:k-1}) &= \int p(x_k | x_{k-1}) p(x_{k-1} | y_{1:k-1}) dx_{k-1} \\ p(x_k | y_{1:k}) &= \frac{p(y_k | x_k) p(x_k | y_{1:k-1})}{p(y_k | y_{1:k-1})} \end{aligned} \quad (3)$$

Through the Bayesian recursive filtering process, the target posterior probability density function $p(x_k | y_{1:k})$ to be tracked can be obtained. Then a state quantity can be optimally estimated as the current tracking result of the target object. The target state estimation result is:

$$\hat{x}_k = \int x_k p(x_k | y_{1:k}) dx_k \quad (4)$$

When Bayesian recursive filtering is used to solve the posterior probability density $p(x_k | y_{1:k})$ of the target state, there is integral operation in the formula, which cannot be completed analytically. For

a general nonlinear non-Gaussian system, it is very difficult to solve the high-dimensional variables. Therefore, a suboptimal solution under the Bayesian framework is used:

$$\lim_{N \rightarrow \infty} \hat{E}[f(X_{0:k})] = E[f(X_{0:K})] \quad (5)$$

2.1. Sequential Importance Sampling

Sequential importance sampling avoids the disadvantages of taking too long to run to a stationary state and not being clear when reaching a stationary state. Sequential importance sampling is based on the sequential analysis method in statistics to estimate the posterior probability density. In fact, it is difficult to sample from the posterior probability density function $p(x_{0:k} | y_{1:k})$ of the real target. Assuming that the importance density function can be decomposed into:

$$q(x_{0:k} | y_{1:k}) = q(x_k | x_{0:k-1}, y_{1:k}) q(x_{0:k-1} | y_{1:k-1}) \quad (6)$$

The analytical formula of the posterior probability distribution at the current moment is:

$$\begin{aligned} p(x_{0:k} | y_{1:k}) &= \frac{p(x_{0:k}, y_k | y_{1:k-1})}{p(y_k | y_{1:k-1})} \\ &= \frac{p(y_k | x_{0:k}, y_{1:k-1}) p(x_{0:k} | y_{1:k-1})}{p(y_k | y_{1:k-1})} \\ &= \frac{p(y_k | x_{0:k}, y_{1:k-1}) p(x_k | x_{0:k-1}, y_{1:k-1}) p(x_{0:k-1} | y_{1:k-1})}{p(y_k | y_{1:k-1})} \end{aligned} \quad (7)$$

According to the Markov nature, the above formula can be written as:

$$p(x_{0:k} | y_{1:k}) = \frac{p(y_k | x_k) p(x_k | x_{k-1}) p(x_{0:k-1} | y_{1:k-1})}{p(y_k | y_{1:k-1})} \quad (8)$$

Particle weights ω_k^i can be recursively expressed as follows:

$$\begin{aligned} \omega_k^i &\propto \frac{p(x_{0:k}^i | y_{1:k})}{q(x_{0:k}^i | y_{1:k})} \\ &= \omega_{k-1}^i \propto \frac{p(y_k | x_k^i) p(x_k^i | x_{k-1}^i)}{q(x_k^i | x_{0:k-1}^i, y_{1:k})} \end{aligned} \quad (9)$$

If $q(x_k | x_{0:k-1}^i, y_{1:k})$ satisfies the first-order Markowitz conditions: $q(x_k | x_{0:k-1}^i, y_{1:k}) = q(x_k | x_{k-1}^i, y_k)$ then:

$$\omega_k^i \propto \omega_{k-1}^i \frac{p(y_k | x_k^i) p(x_k^i | x_{k-1}^i)}{q(x_k^i | x_{k-1}^i, y_k)} \quad (10)$$

The posterior probability density function $p(x_k | y_{1:k})$ at the current moment of the target can be approximated as:

$$p(x_k | y_{1:k}) \approx \sum_{i=1}^N \tilde{\omega}_k^i \delta(x_k - x_k^i) \quad (11)$$

where N is the number of particles, at $N \rightarrow \infty$ that time, the result of this formula is getting closer to the true target state posterior probability density $p(x_k | y_{1:k})$.

2.2. Importance Density Function Selection

In the proposed algorithm, the appropriate importance density function $q(x_k | x_{k-1}^i, y_k)$ is used as it is related to the effective sample size of the particle filter. The optimal choice is expressed as follows:

$$\begin{aligned} q(x_k | x_{k-1}^i, y_k)_{opt} &= p(x_k | x_{k-1}^i, y_k) \\ &= \frac{p(y_k | x_k^i, x_{k-1}^i) p(x_k^i | x_{k-1}^i)}{p(y_k | x_k^i) p(x_k^i | x_{k-1}^i)} \\ &= \frac{p(y_k | x_k^i)}{p(y_k | x_{k-1}^i)} \end{aligned} \quad (12)$$

Substituting (13) into (14) gives the updated particle weights:

$$\omega_k^i = \omega_{k-1}^i p(y_k | x_{k-1}^i) = \omega_{k-1}^i \int p(y_k | x_k^i) p(x_k^i | x_{k-1}^i) dx_k^i \quad (13)$$

From the above derivation, it can be seen that the probability density function of optimal importance needs to be sampled from $q(x_k | x_{k-1}^i, y_k)$ and every new state has to be integrated, In order to obtain the recommended distribution simply and effectively, the prior probability density function is often taken as the importance density function, namely:

$$q(x_k^i | x_{k-1}^i, y_k) = p(x_k^i | x_{k-1}^i) \quad (14)$$

The particle weight is updated by:

$$\omega_k^i = \omega_{k-1}^i p(y_k | x_k^i) \quad (15)$$

2.3. Resampling Technique

The importance sampling algorithm is prone to particle degradation in practical applications [20]. The main reason is that with the process of iteration, the weight of most particles becomes very small or even zero, and a large amount of time will be wasted in the calculation of small weight particles, and the variance of the importance weight will gradually increase, resulting in the error of the posterior probability density function of the target will also increase. Researchers have tried to increase the number of samples N , but the effect is unsatisfactory. At present, the commonly used solutions are: (1) choosing a good recommendation distribution; (2) using re-sampling technology [21,22].

In order to accurately measure the degree of particle weight degradation, the concept of effective sample size N_{eff} is proposed:

$$N_{eff} = \frac{N}{1 + \text{var}(\omega_k^i)} \approx \frac{1}{\sum_{i=1}^N (\omega_k^i)^2} \quad (16)$$

From the above formula, it can be seen that the larger the particle weight, the smaller the number of effective particles, and the more serious the degradation of the particle weight. Scholars have studied many re-sampling strategies, such as uniform sampling, Markov chain Monte Carlo (MCMC) moving algorithm, hierarchical sampling [23], and evolutionary algorithm [24]. Among them, the uniform sampling method is the most widely used, that is, the particle weight is $1/N$. Because of the continuous replication of large-weight particles, resulting in the lack of diversity of particles, not all States of particles need to be re-sampled. A threshold N_{th} (usually $2/3N$) is set in advance. When $N_{eff} < N_{th}$, the algorithm is re-sampled, otherwise it will not be re-sampled.

3. Results Particle Filter with Multi Features Fusion

In particle filter, observation is based on features. This section will mainly describe the human face features, which are utilized in tracking human face of interest by combining color feature and edge feature.

3.1. Color Feature Description

Color histograms [25] have been widely applied because of their insensitivity of rotation and scales. Meanwhile, its calculation is simple. The difference of face color is mainly reflected in the brightness, whereas the hue is uniform, so this paper employs the hue-saturation-value (HSV) color model. Furthermore, this paper only calculates H-S histogram using 16×8 bins. The color histogram is shown in Figure 1b. To a certain extent, this model reduces the effect of illumination variation.

The weighted color histogram [26] is constructed as:

$$p_c(n) = C_c \sum_{i=1}^N K_E \left(\frac{x_i - x_0}{a} \right) \delta[h(x_i) - n] \quad (17)$$

where the pixel $n \in [1, M]$ and M is the total number of color weighted histogram bin, the function $h(x_i)$ maps the pixel x_i location to the corresponding histogram bin, x_0 is the center position of observation area, and $K_E(\cdot)$ represents the Epanechnikov kernel profile with the nuclear bandwidth a .

The Epanechnikov kernel is described as:

$$K_E(x) = \begin{cases} c(1 - \|x\|^2) & \|x\| < 1 \\ 0 & \|x\| \geq 1 \end{cases} \quad (18)$$

where δ denotes the Kroenke delta function and C_c is a normalization constant:

$$C_c = \frac{1}{\sum_{i=1}^N K_E \left(\frac{x_i - x_0}{a} \right)} \quad (19)$$

The distance between reference target template $q_c(n)$ and candidate target template $p_c(n)$ can be measured by the Bhattacharyya distance d :

$$d_c = \sqrt{1 - \rho_c[p_c(n), q_c(n)]} \quad (20)$$

where $\rho_c[p_c(n), q_c(n)]$ is the Bhattacharyya coefficient [23]:

$$\rho_c[p_c(n), q_c(n)] = \sum_{n=1}^M \sqrt{p_c(n)q_c(n)} \quad (21)$$

Face color likelihood function is defined as:

$$p(y_c|x) = \frac{1}{\sqrt{2\pi\sigma_c}} \exp\left(-\frac{1 - \rho_c}{2\sigma_c^2}\right) \quad (22)$$

where σ_c is the Gaussian variance which is selected as 0.2.

3.2. Edge Feature Description

Edge feature is another key feature of human face. It is insensitive to illumination variation and the background of similar face color. This paper employs edge orientation histogram [24] to depict face edge feature. Meanwhile, we adopt Sobel operator to detect edge contour. The edge orientation histogram is shown in Figure 1c. Firstly, the RGB image was converted to the gray image.

Then, we calculate the gradient G and orientation τ of each pixel in the target region of each particle as follows:

$$p(y_e|x) = \frac{1}{\sqrt{2\pi\sigma_e}} \exp\left(-\frac{1-\rho_e}{2\sigma_e^2}\right) \quad (23)$$

where G_x and G_y represent the level and vertical kernel of images respectively, as well as $0 \leq \tau \leq \pi$. Then the edge orientation histogram was constructed as:

$$p_e(n) = C_e \sum_{i=1}^N K_E\left(\frac{x_i - x_0}{a}\right) G(x_i) \delta[h(x_i) - n] \quad (24)$$

Face edge likelihood function is defined as:

$$\begin{cases} G = \sqrt{G_x^2 + G_y^2} \\ \tau = \arctan\left(\frac{G_y}{G_x}\right) \end{cases} \quad (25)$$

where σ_e is the Gaussian variance and is selected as 0.3, and $\rho_e = \frac{1}{\sum_{n=1}^M} \sqrt{p_e(n)q_e(n)}$.

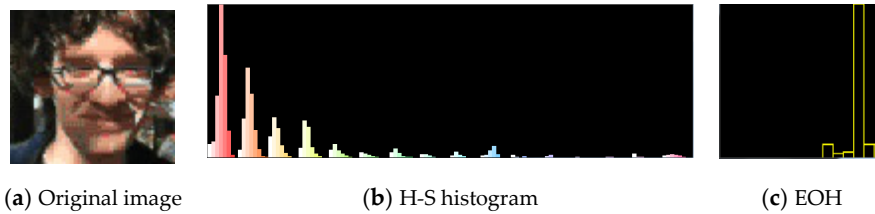


Figure 1. Color histogram and edge orientation histogram.

3.3. Features Fusion Strategy

The observation model $p(y_t|x_t)$ in this paper concerns two image cues: color feature and edge feature. According to linear fusion method, the entire observation likelihood function can be calculated as:

$$p(y_t|x_t) = \theta_c p(y_c|x) + \theta_e p(y_e|x) \quad (26)$$

where $p(y_c|x)$ and $p(y_e|x)$ are the likelihood function of the color feature and edge feature respectively. The terms θ_c and θ_e ($0 \leq \theta_c, \theta_e \leq 1$) are the weights of the color feature and edge feature, where $\theta_c + \theta_e \leq 1$. The weights in most algorithms are assumed to be unchanged during the tracking, and $\theta_c = \theta_e = 0.5$. This is an equal method, whereas the contribution of each feature is different in the actual video tracking. In this paper, we proposed a strategy of self-adaptive multi-features fusion strategy, so that it can make up for the deficiency of each characteristic.

For $s = \{c, e\}$ its likelihood observation result is p_s^i . The characteristic likelihood value formation observation peak is expressed as:

$$\delta_x = \frac{1}{N} \sum_{i=1}^N |p_x^i - \bar{p}_x| \quad (27)$$

when $\bar{p}_s = \frac{1}{N} \sum_{i=1}^N p_s^i$, the value of Equation (17) is as big as possible.

Feature weight is:

$$\tilde{\theta}_s = \frac{\delta_s}{|x_s^{peak} - \bar{x}^{peak}|} \quad (28)$$

The weight of each feature can be normalized as:

$$\theta_s = \frac{\tilde{\theta}_s}{\sum_{s=1}^M \tilde{\theta}_s} \quad (29)$$

Particle weight is updated as:

$$w_t^i \propto w_{t-1}^i p(y_t | x_t^i) = w_{t-1}^i [\theta_c p(y_c | x) + \theta_e p(y_e | x)] \quad (30)$$

4. Face-Tracking System

In this section, the proposed face-tracking system is implemented in detail.

4.1. Dynamic Model

The auto-regressive process (ARP) model [19] has been widely used for the purpose of formulating such a dynamical model. The ARP is divided into first-order and second-order. The first-order ARP only considers the displacement and noise of the target. However, moving targets usually have physical properties of velocity and acceleration. Hence, the second-order ARP is adopted in this work. The dynamical model can be represented as:

$$X_t = AX_{t-1} + BX_{t-2} + CN_{t-1} \quad (31)$$

where A and B are the drift coefficient matrix, C is the random diffusion coefficient matrix, N_{t-1} is the noise matrix at time $(t-1)$. These parameters can be obtained from experience or video sequence training.

4.2. Face Tracking with Self-Updating Tracking Window

When the target area of the moving target and the number of particles in the sample are determined, the average distance of the weighted particles in the moving target area to the moving target center is related to the size of the moving target. When the size of the moving target is diminished, the distribution of particles is relatively concentrated, so the average distance of the weighted particles in the moving target area to the moving target center is diminished.

The tracking window size will be smaller or larger than the face size which needs adjustment as the face size changes [9]. The self-updating tracking window model is represented as:

$$\begin{cases} s_t(x) = s_{t-1}(x) \times d/d_1 \\ s_t(y) = s_{t-1}(y) \times d/d_1 \end{cases} \quad (32)$$

where s_t represents the current frame tracking window size, s_{t-1} represents the previous frame tracking window, and d_1 is the average distance from the previous frame to the target center.

We define the state variable (x,y) as the center of the rectangle. The center of the particle is (x_i,y_i) , and $\omega^i (i = 1, 2, \dots, M)$ are the weights of the samples. M is the number of particles whose weight is greater than a certain threshold, T. The average distance between the particle and the target center is defined as:

$$d = \frac{1}{M} \sum_{i=1}^M \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (33)$$

4.3. Updating Model

In the actual process of human face tracking, the environment or the target may be under various conditions (such as illumination variation, posture variation, and occlusion). A single fixed target model

would not be stable for a long time and is prone to drift. In this paper, we present a template updating technology [10]. Threshold value is set at π . If the average status $P < \pi$ the template updates as:

$$H_{new} = \tau H_{old} + (1 - \tau) H_{current} \quad (34)$$

where H_{new} is the new reference histogram, H_{old} is the initial reference histogram, $H_{current}$ is the current reference histogram. If $\pi = 0.3$ and τ is a constant $0 \leq \tau \leq 1$, then $\tau = \rho^{k-1} / \rho^{k-1} + \rho^k$.

4.4. The Integral Histogram of the Image

During the process of particle filtering, the purpose is mainly to calculate the similarity coefficient of the histogram of the target and the histogram of all particles, and then determine the observed value of the target. This process is an exhaustive search process. If there are more particles in a video, it will take a long time in calculation [26]. In order to solve this problem, the approach of integrating histogram was used to simplify the calculation of particles histogram, which greatly saved the computation time. The methodology of this method is to simply add and subtract the upper left corner, the upper right corner, the lower left corner and the lower right corner of the particle. Finally, we can get the histogram of the whole region.

4.5. Tracking Algorithm Procedure

The specific steps of this algorithm are summarized as follows:

Step 1: Initialize $t = 0$, the known initial state is $\{x_0, \omega_0\}_{i=1}^N$, and calculate the template color histogram, edge orientation histogram p_c^0 and p_e^0 .

Step 2: Predict the face current state according to Equation (32).

Step 3: Weight updating:

- 3.1 Sample particles x_t^i from $p(x_t^i | x_{t-1}^i)$, $i = 1, \dots, N$;
- 3.2 Calculate the color likelihood $p(y_c | x)$ according to Equation (22);
- 3.3 Calculate the gradient likelihood $p(y_e | x)$ according to Equation (25);
- 3.4 Calculate the weight of each feature according to Equation (27), and normalize θ_c, θ_e by Equation (18);
- 3.5 Calculate the entire observation likelihood $p(y_t | x)$ according to Equation (26);
- 3.6 Calculate the weight w_t^i according to Equation (30);
- 3.7 Normalize weight $\tilde{\omega}_t^i = \frac{\omega_t^i}{\sum_{i=1}^N \omega_t^i}$

Step 4: Output target state estimation

Step 5: The particle resamples: $E(x_t) = \sum_{i=1}^N \tilde{\omega}_t^i x_t^i$

Step 6: Update the model according to Equation (31).

Step 7: Continue to track and return to Step 2.

5. Experimental Results and Analysis

In this section, we will test the effectiveness and accuracy of our proposed algorithm. The data set was the Visual Tracker Benchmark. This benchmark includes the results from 100 test sequences and 29 trackers. This website contains data and code of the benchmark evaluation of online visual tracking algorithms. Visual Tracker Benchmark Database is applied in experiment which provides a standard video database for evaluating face tracking and recognition algorithms.

5.1. Tracking Effect and Error Analysis

In order to assess the tracking performance of this proposed method, we manually selected a rectangle window of face as the matching template, and set $N = 100$. This experiment platform is based on Visual Studio 2010 OpenCV 2.4.8. Euclidean distance is used to measure the result of target tracking. The sequences are obtained from reference [27], and the information of them is shown in Table 1.

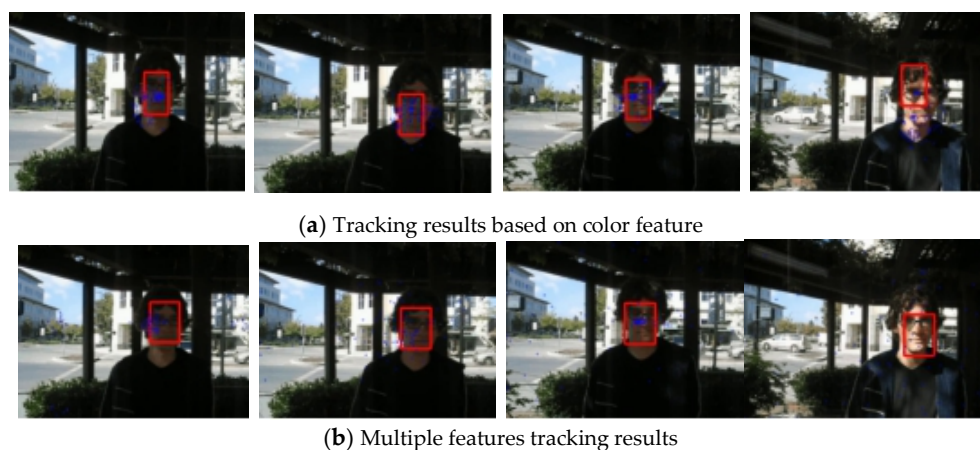
Table 1. Video Sequences Used in Our Experiments.

Sequence	Frame Size	Sequence Characteristics	Total Frames
Test 1	480 × 360	Illumination variation	182
Test 2	480 × 360	Similar background	119
Test 3	480 × 360	Object scaling	198
Test 4	480 × 360	Object rotation	45
Test 5	480 × 360	Occlusion	310
Test 6	480 × 360	High speed operation and lens stretching	180
Test 7	480 × 360	Light change	60

We carried out experiments on different video sequences. In the framework of the improved particle filter algorithm, we extracted face color features and edge features. The adaptive fusion strategy proposed in this paper was used to track video faces. At the same time, the test results of this algorithm were compared with the tracking results of the traditional particle filter based on the color feature, so as to verify the advantages of our improved tracker in the presence of occlusion, similar background and illumination changes.

Furthermore, Root Mean Squared Error (RMSE) is used as one objective metrics to evaluate the quantitative quality of different experimental results. The smaller the RMSE, the closer the tracking result is to the ground truth.

The first sequence in Figure 2 is about drastic variations in illumination, where the cast shadow drastically change the appearance of the target face when a person walks underneath a trellis covered by vines. The frame indexes are the 24th, 36th, 52th and 138th, respectively, shown as Figure 2, where the blue points denote the particles and red boxes are the tracking results. Figure 2a is the tracking result based only on color features. In the tracking process, the performance is ideal for the condition that the outdoor illumination variation is not obvious. When the light of target tracking based on color feature changes dramatically, the rectangular box cannot select the correct face location. This happens because the color features are very sensitive to illumination variation. Figure 2b shows the tracking results of this proposed algorithm. At the 36th frame and 138th frame, compared to the undesirable results based on color feature, our algorithm overcomes the problems and tracks the face during the entire challenging sequence successfully. Figure 3 is the quantitative analysis of the tracking results measuring the position error, which indicates that the error rate of our algorithm is obviously lower than single color feature tracking. This is mainly attributed to the full use of the latest observation information in the improved particle filtering algorithm and the fusion strategy of two features, so that the target can be tracked more accurately.

**Figure 2.** Tracking results under the illumination variation.

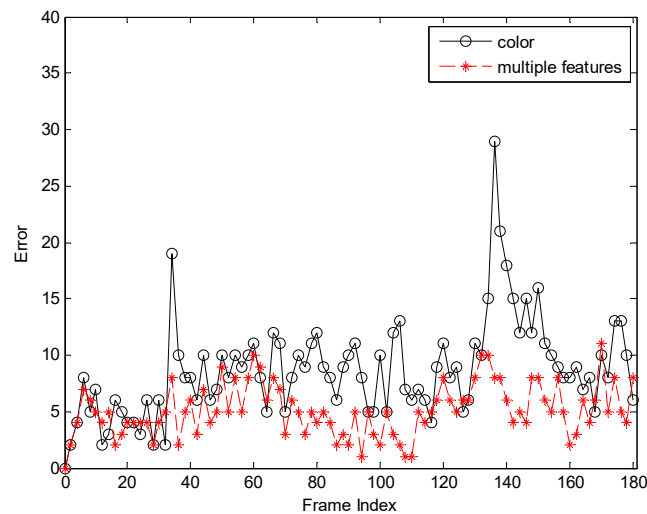


Figure 3. The RMSE comparison corresponding to illumination variation.

Figure 4 allows us to test the robustness of the experiment. It can be seen from the test results that the tracking effect of the algorithm is very stable.

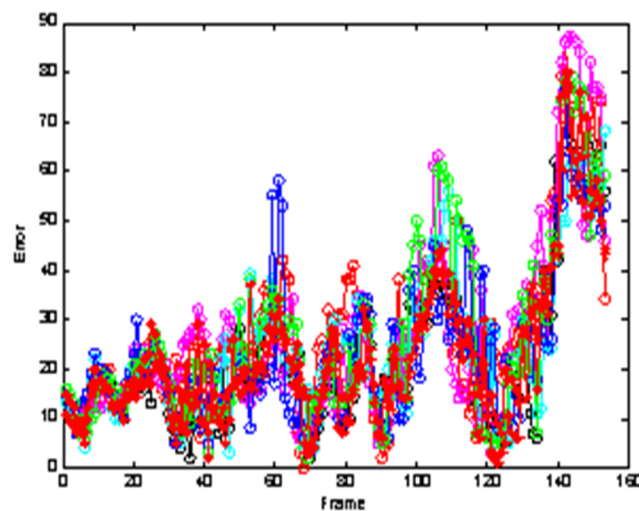


Figure 4. The analysis robustness.

The second sequence in Figure 5 is about the variations in pose and background, corresponding to the 6th, 38th, 82th, 117th frame snapshots. Figure 5 shows the comparison results between our algorithm and the use of color features. If we only use color features for a particle filter, the tracking algorithm is not sensitive to the changes of facial expression and posture. However, when there are skin-like objects in the video, the algorithm considers the object which is similar to the human face as the face area and causes a tracking failure, as shown in Figure 5a. For example, in the 82th frame, the human face is interfered by hands, which leads to the tracking deviating from the real face position and tracking to the hand. In comparison, our method can track the face very well, as shown in Figure 5b. Furthermore, we can see from Figure 6 that the error of our algorithm is very small, conforming to the observation of Figure 5. Figure 7 allows us to test the robustness of the experiment. It can be seen from the test results that the tracking effect of the algorithm is very stable.

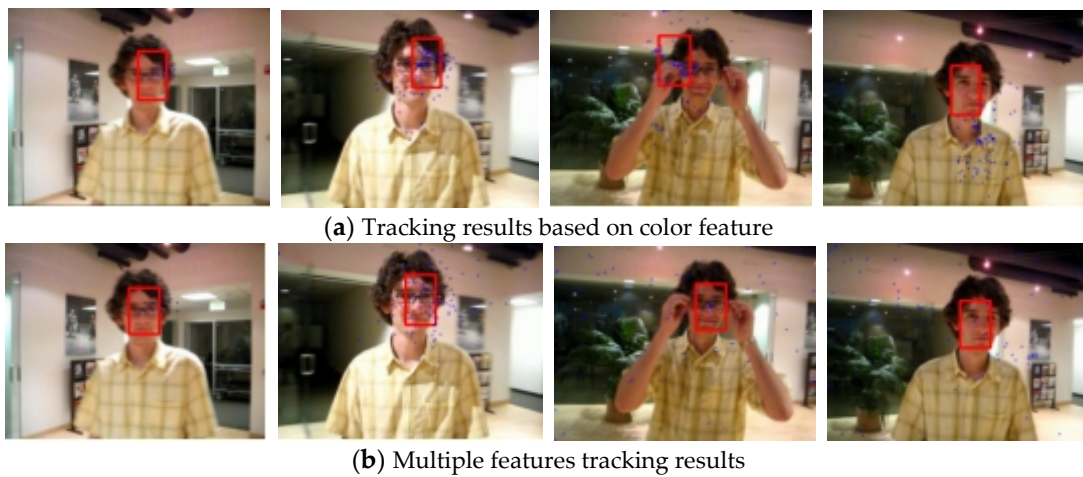


Figure 5. Tracking results under the background changes.

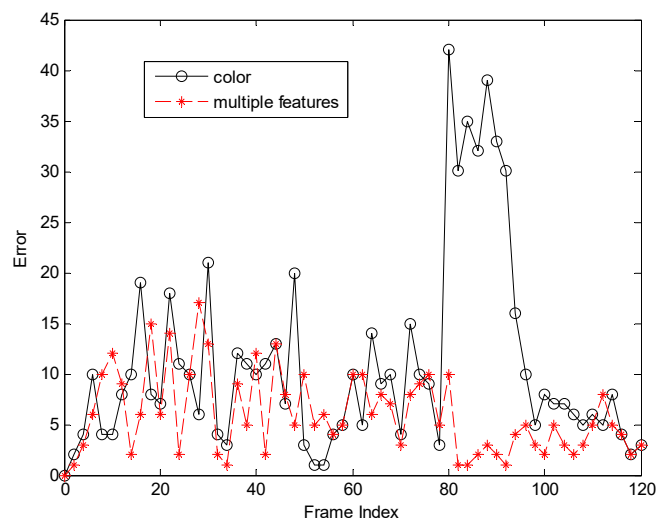


Figure 6. The RMSE comparison corresponding to background changes.

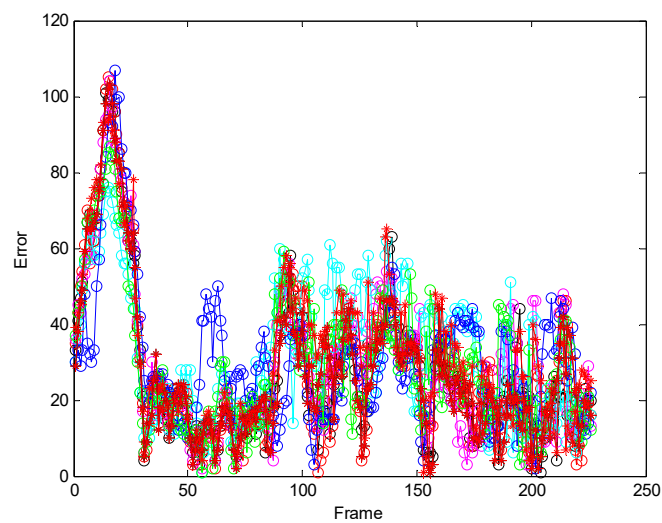


Figure 7. The analysis robustness.

The third sequence in Figure 8 is about enlarged objects and diminished objects. The size of the video sequence is 480×360 pixels. The frame indices are 8, 16, 19, and 40. A girl in the computer room (a frame with a simpler background) is used to evaluate the performance of the proposed algorithm in handling pose variation. The results are shown in Figure 8. This is a stretched video shot where the scale of the human face is variable. Scale variation occurs when the girl goes toward the camera, as seen from Figure 6. Figure 8a shows results of the skin color tracker. The red rectangle is always the same size throughout the entire process, but the face partially goes outside the red rectangle (such as the 16th frame and the 19th frame). Then, the girl slowly turns away from the camera, and the red rectangle contains other information except for the human face. The algorithm excellently scaled the size of the tracking window to the human face as it changed. As shown in Figure 8b, it mainly benefits from adaptively adjusting the tracking window in the tracking algorithm. Figure 9 gives the error curves comparison about Test 3. The tracking error of the algorithm our proposed is obviously lower than that based on color feature.

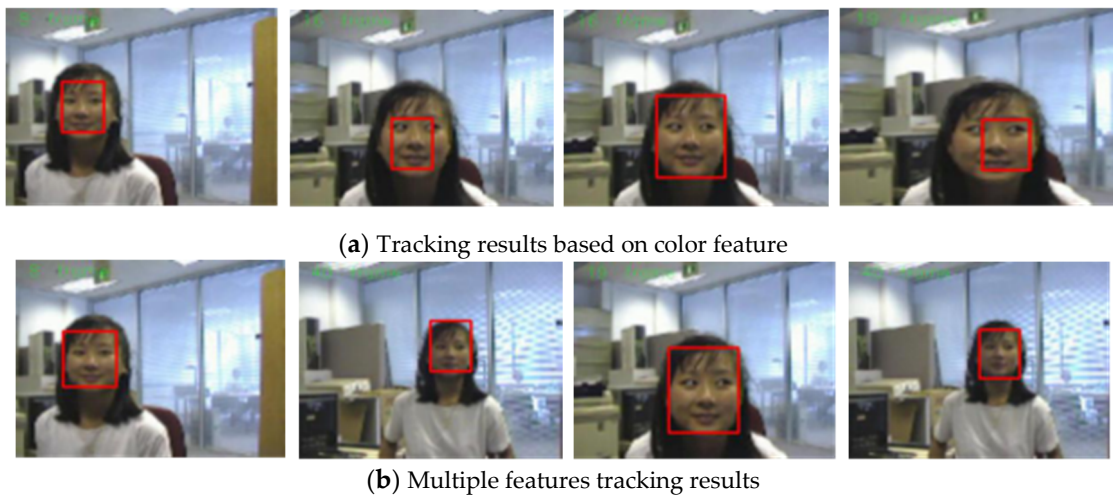


Figure 8. Tracking results under the background changes.

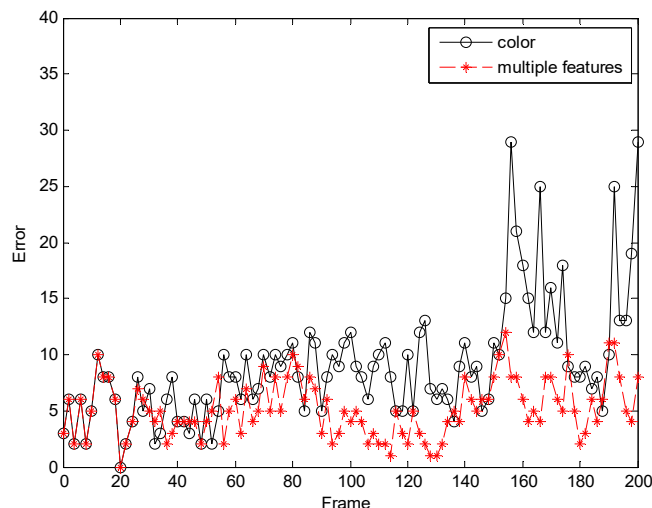


Figure 9. The RMSE comparison corresponding to background change.

The fourth sequence in Figure 10 is about the rotation of a man's head. The frame indices are 6, 27, 33, and 41. Some samples of the final tracking results are shown in Figure 10. When the man turned his head, his head looks changed. The result of Figure 10a benefits from the color feature. In the 27th frame, the man's face drastically changes, and the target is lost in the succeeding frames.

Especially, as shown in the 33th frame, the overlarge tracking window has made the tracker lose the target. The results of the tracking algorithm proposed in this paper are shown in Figure 10b. This method is adaptive template updating, which accurately tracks the target and is robust to pose variations. Based on Table 1, the tracking performance of our method is the best. Figure 11 gives the error curves comparison about Test4. The tracking error of the algorithm our proposed is obviously lower than that based on color feature.

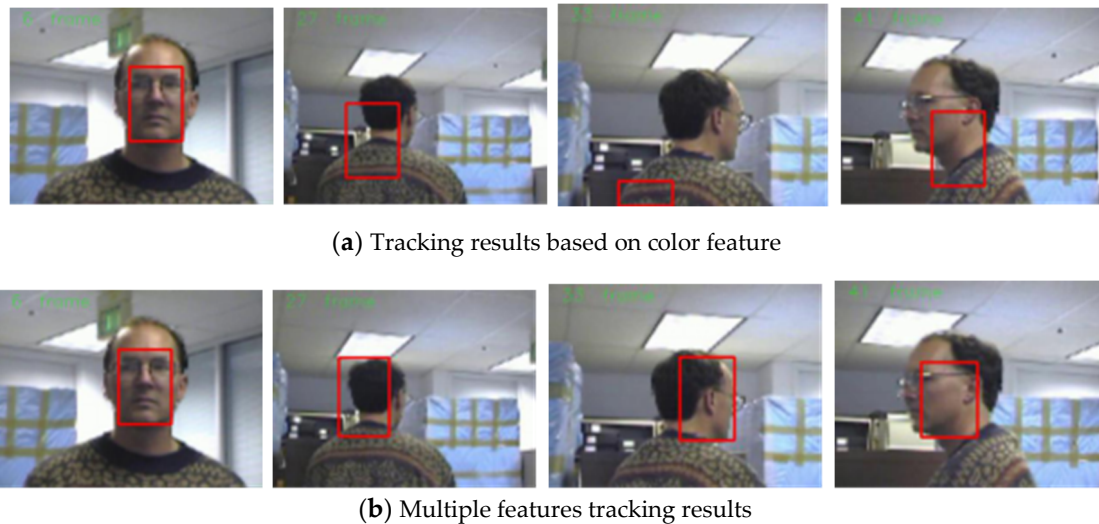


Figure 10. Tracking results under the background changes.

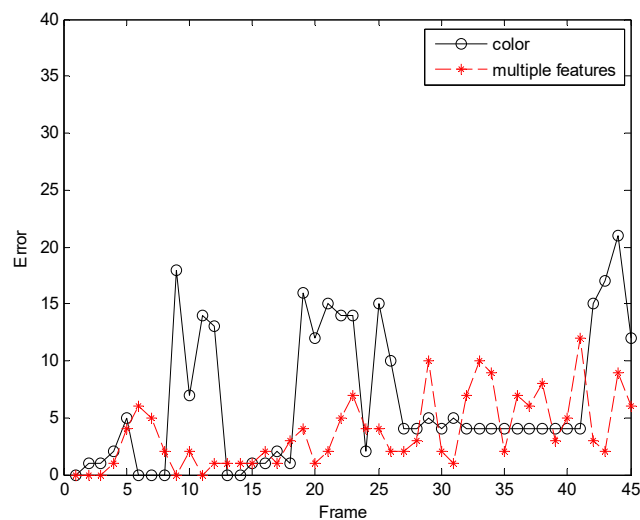


Figure 11. The RMSE comparison corresponding to background changes.

The fifth sequence in Figure 12 is about the face blocked by a book. Some samples, corresponding to the 13th, 167th, 268th and 278th frame snapshots, of the final tracking results are shown in Figure 12. Performance on this sequence exemplifies the accuracy and robustness of our fusion method to partial occlusion. As we can observe, when the face is occluded, the visible face target range becomes smaller and smaller, and the face information is not comprehensive. At the 167th frame that the man is covered partly by a book, the tracking method based on color feature loses the human face obviously, as shown in Figure 12a. Moreover, when the book is taken away, it still fails to track the man's face, which confuses the tracker from following the right face within the interruption. In like manner the tracking method based on edge feature loses the human face obviously, as shown in Figure 12b. The tracking results of our method are shown in Figure 12c. Although both methods can

track faces, when a face is occluded or the illumination changes, the single feature method cannot track the face in real time. Nevertheless, our fusion tracker follows the man's face accurately and robustly. Figure 13 gives the error curves comparison about Test5 the tracking error of the algorithm our proposed is obviously lower than that based on color and egde feature. It is mainly benefited from the proposed fusion scheme, which can enhance the one that is reliable for tracking.

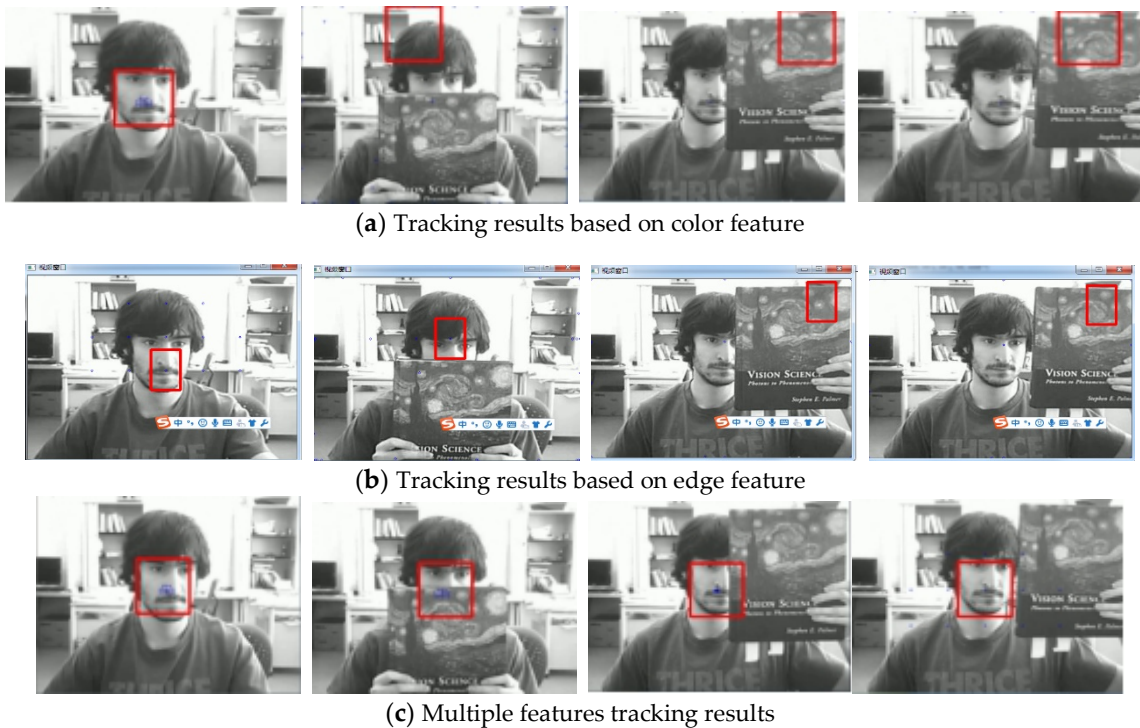


Figure 12. Tracking results under the background changes.

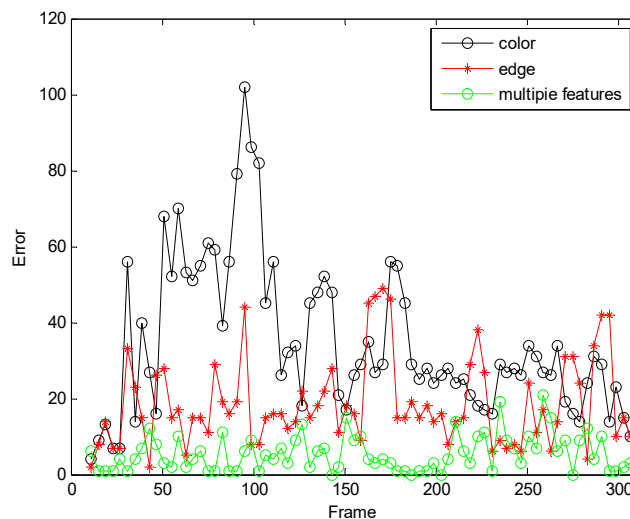


Figure 13. The RMSE comparison corresponding to background changes.

The fourth sequence in Figure 14 is about high-speed face movement and lens stretching. The frame indices are 16, 54, 101 and 118. In this sequence, Figure 14a is based on the algorithm in the literature. When the rapid rotation occurs, the template cannot be updated in time to cause the tracking effect to be extremely poor. When the lens is stretched, the tracking window size cannot be adjusted to cause redundant or missing information. Figure 14b is a comparison of the algorithm in

this paper. Our tracker can accurately locate the face even if the face part moves at high speed. The red rectangular frame always surrounds the human face, which can track the face very well. It mainly benefits from the improvement of the particle filter algorithm and the adaptive fusion strategy using two features to make up for the disadvantages of each feature. In addition, the real-time update of the face template and the window adaptive adjustment reduce the impact of the target high-speed motion and stretching. Figure 15 gives the error curves comparison about Test6 the tracking error of the algorithm our proposed is obviously lower than that based on color and egde feature.

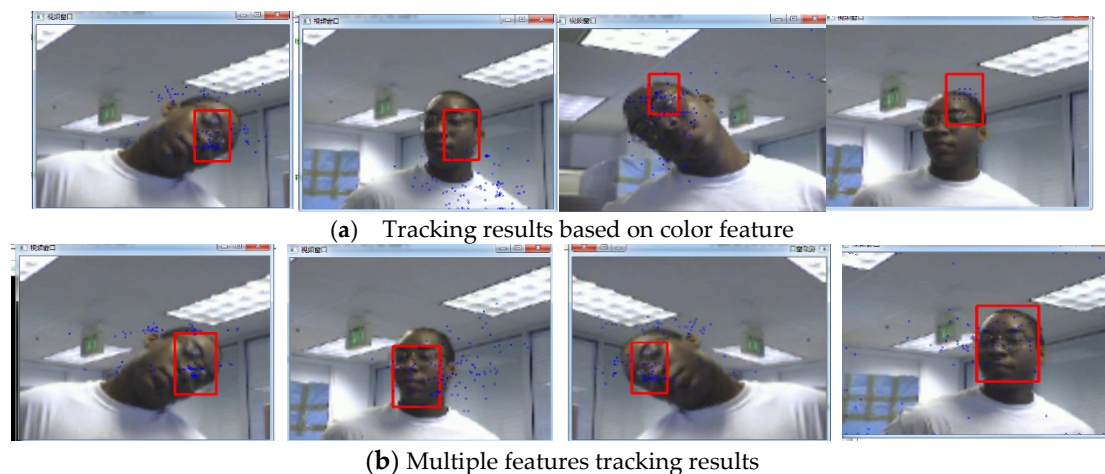


Figure 14. Tracking results under the background changes.

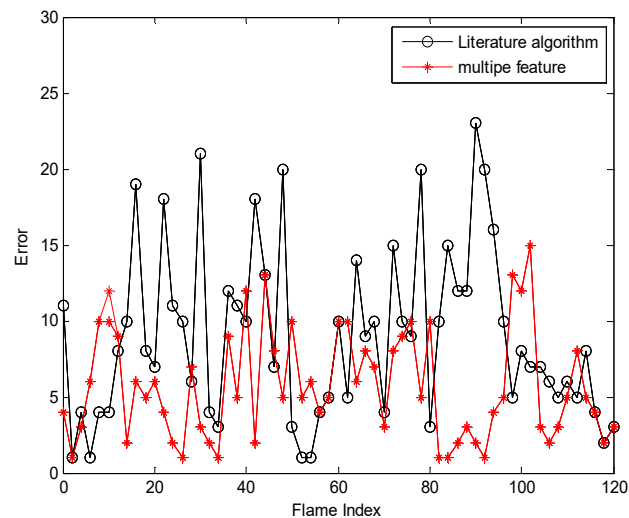


Figure 15. The RMSE comparison corresponding to background changes.

The third sequence in Figure 16 is about light change. The size of the video sequence is 480×360 pixels. The frame indices are 12, 22, 34, and 47. When the illumination changes, Figure 16a is a face tracking based on the single feature of color. From the figure we can see that only the single feature of color is easy to cause information loss. When we combine edge features and color features, Through Figure 16b, we can see that the face part can be accurately tracked regardless of the change in illumination. When the illumination changes, Figure 16a is a face tracking based on the single feature of color. From the figure we can see that only the single feature of color is easy to cause information loss. When we combine edge features and color features, Through Figure 16b, we can see that the face part can be accurately tracked regardless of the change in illumination. Figure 17 gives the error curves comparison for Test 7. The tracking error of the algorithm our proposed is obviously lower than that based on color feature.

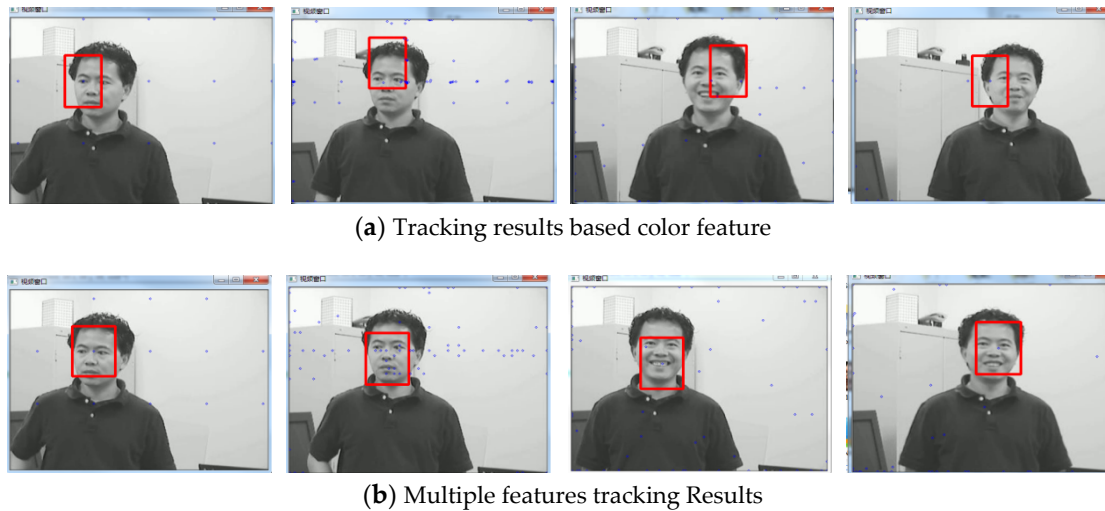


Figure 16. Tracking results under the background changes.

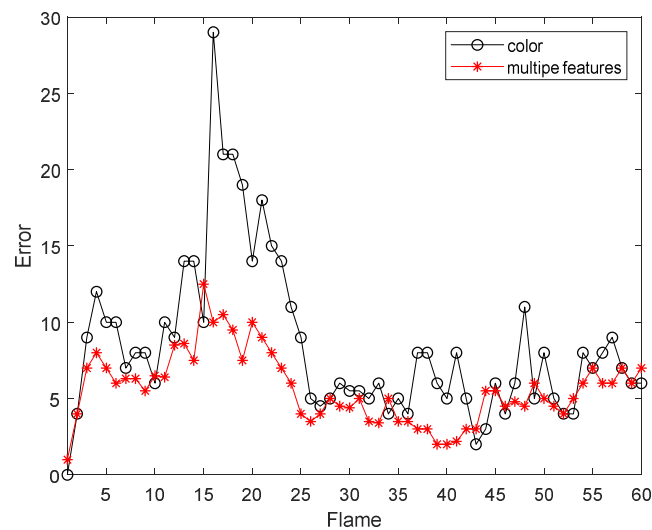


Figure 17. The RMSE comparison corresponding to background change.

5.2. Comparison with Other Algorithms

In order to further verify the accuracy of our algorithm, we further compare the algorithm of this paper with the others algorithm. Figures 18 and 19 are comparisons with the KCF algorithm. They are the quantitative analysis of the tracking results measuring the position error, which indicates that the error rate of our algorithm is obviously lower than with single color feature tracking.

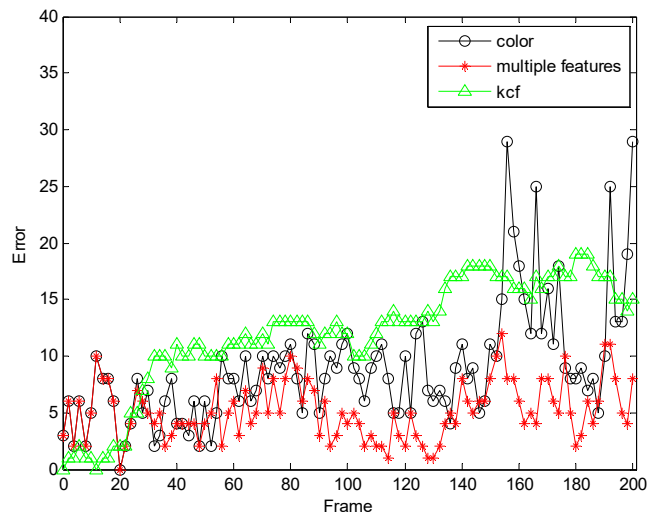


Figure 18. The RMSE comparison corresponding with KCF.

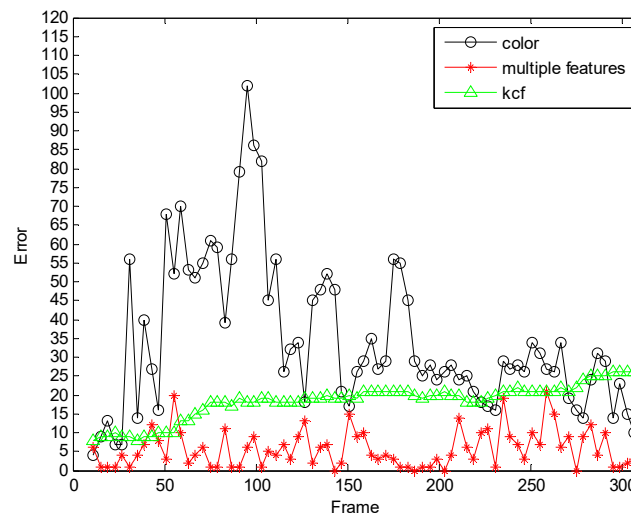


Figure 19. The RMSE comparison corresponding with KCF.

Figures 20 and 21 are comparisons with the Fiserface algorithm. By comparison, we find that the tracking accuracy of our algorithm is higher. It can be seen from the test results that the tracking effect of the algorithm is very stable. Our comparison with the Fiserface algorithm is mainly to compare the accuracy of the tracking effect, and the accuracy is used to show that our algorithm is feasible. Moreover, compared to Fiserface as the number of frames increases, the experimental results obtained by our algorithm are better, and our algorithm requires less computational time and cost under the same situation.

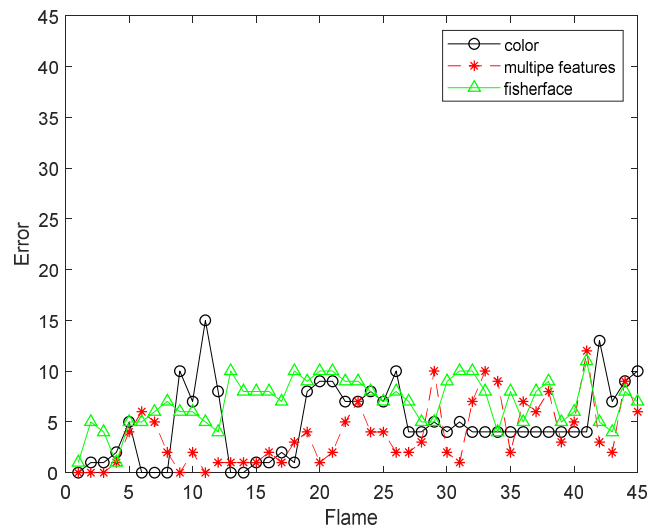


Figure 20. The RMSE comparison corresponding with Fisherface.

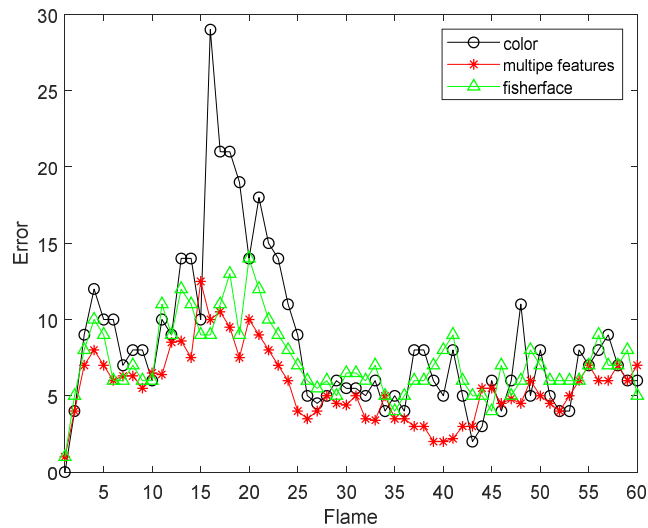


Figure 21. The RMSE comparison corresponding with Fisherface.

Figures 22 and 23 are comparisons with the R-CNN algorithm. By comparison, we find that the tracking accuracy of our algorithm is higher. Our comparison with the R-CNN algorithm is mainly to compare the accuracy of the tracking effect and the accuracy is used to show that our algorithm is feasible. Moreover, compared to R-CNN as the number of frames increases, the experimental results obtained by our algorithm are better, and our algorithm requires less computational time and cost under the same situation.

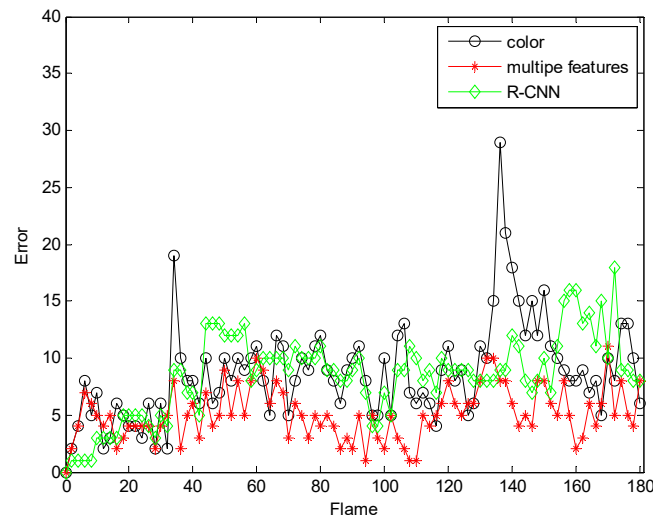


Figure 22. The RMSE comparison corresponding with R-CNN.

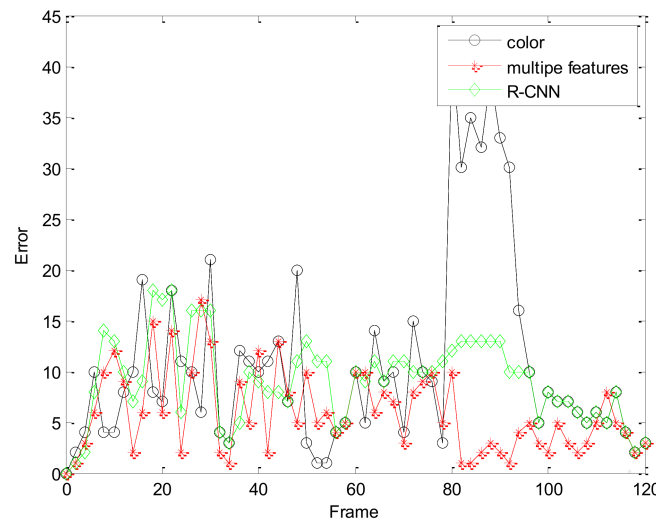


Figure 23. The RMSE comparison corresponding with R-CNN.

5.3. Computational Efficiency

In order to verify the calculation efficiency of integral histogram is higher than normal histogram, two histograms of video face tracking were used to compare their consumption in the process of tracking in the histogram computation time.

Table 2 shows that with the increase of particle number, computational time is also gradually increased, but the time consumption growth speed of the integral histogram is not bigger than for a normal histogram. Also, we found that in the case of few particles, the ordinary histogram calculation time is less than the integral histogram, because the initialization time of the integral histogram is basically fixed. Therefore, more particles exist in a video, more obvious the advantage of the integral histogram.

Table 2. The Effects of Different Particle Number on the Calculation.

Particle Number	Time/s	
	Normal Histogram	Integral Histogram
20	0.028987	0.050296
50	0.085672	0.054322
100	0.148562	0.058970
500	0.765326	0.063952

6. Conclusions

In this paper, a face tracking algorithm based on adaptive fusion of skin color and edge features is proposed, which adaptively updates the template and adaptively adjusts the target tracking window to adapt to complex video background. Our experimental results show the advantages of our algorithm more clearly. It can be seen from the table in Figure 2 that our algorithm takes less time and is more efficient. The experimental results show that compared with the single feature algorithm, the face can be accurately tracked in complex backgrounds with skin color, illumination changes, and especially in face color changes. The algorithm reduces the loss of target information by updating the tracking target template in real time, which further improves the accuracy of the algorithm. In the future, the computational complexity of particle filter algorithm will be thoroughly studied to meet the real-time requirements.

However, there are some limitations of the proposed algorithm. For example, the hardware-assisted approach need to be considered, the initial face template should be well defined, or the target model gets anchored to the first frame. In our future works, we plan to develop a faster and more robust tracking method.

Author Contributions: Formal analysis, T.W., W.W.; Methodology, T.W. and W.W.; Supervision, T.L.; Writing—original draft, T.W. and H.L.; Writing—review & editing, T.W.

Funding: This work was supported in part by NSFC (61572286 and 61472220), 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.

Acknowledgments: We are grateful to the editors and referees for their invaluable suggestions for improving the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, Y.; Nie, L.; Liu, L.; Rosenblum, D.S. From action to activity: Sensor-based activity recognition. *Neurocomputing* **2016**, *181*, 108–115. [[CrossRef](#)]
2. Lu, L.; Zhang, J.; Jing, X.; Khan, M.K.; Alghathbar, K. Dynamic weighted discrimination power analysis in DCT domain for face and palmprint recognition. In Proceedings of the 2010 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea, 17–19 November 2010; Volume 34, pp. 44–48.
3. Liu, Y.; Zheng, Y.; Liang, Y.; Liu, S.; David, S. UrbanWater Quality Prediction based on Multi-task Multi-view Learning. In Proceedings of the 25th International Joint Conference on Artificial Intelligence, New York, NY, USA, 9–15 July 2016; Volume 34, pp. 44–48.
4. Lee, J.; Park, G.Y.; Kwak, H.Y. Two-directional two-dimensional random projection and its variations for face and palmprint recognition. In Proceedings of the ICCSA 2011: International Conference on Computational Science and Its Applications, Santander, Spain, 20–23 June 2011; Volume 32, pp. 707–727.
5. Leng, L.; Li, M.; Kim, C.; Bi, X. Dual-source discrimination power analysis for multi-instance contactless. *Multimed. Tools Appl.* **2015**, *76*, 333–345. [[CrossRef](#)]
6. Leng, L.; Li, M.; Leng, L.; Teoh, A.B.J. Conjugate 2DPalmHash code for secure palm-print-vein verification. In Proceedings of the 2013 6th International Congress on Image and Signal Processing (CISP), Hangzhou, China, 16–18 December 2013; Volume 70, pp. 495–523.
7. Yoon, J.H.; Yang, M.H.; Lim, J.; Yoon, K.J. Bayesian multi-object tracking using motion context from multiple objects. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 5–9 January 2015; pp. 33–40.
8. Welch, G.; Bishop, G. *An Introduction to the Kalman Filter*; University of North Carolina at Chapel Hill: Chapel Hill, NC, USA, 2017; Volume 8, pp. 127–132.
9. Gustafsson, F.; Gunnarsson, F.; Bergman, N.; Forsslund, U.; Jansson, J.; Karlsson, R.; Nordlund, P.J. Particle filters for positioning, navigation, and tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 425–437. [[CrossRef](#)]
10. Sadeghian, A.; Alahi, A.; Savarese, S. Tracking the Untrackable: Learning to Track Multiple Cues with Long-Term Dependencies. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 24–30 July 2017.

11. Lin, Y.; Shen, J.; Cheng, S.; Pantic, M. Mobile Face Tracking: A Survey and Benchmark. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
12. Jiang, X.; Yu, H.; Lu, Y.; Liu, H. A fusion method for robust face tracking. *Multimed. Tools Appl.* **2016**, *75*, 11801–11813. [[CrossRef](#)]
13. Wang, L.; Yan, H.; Lv, K.; Pan, C. Visual Tracking via Kernel Sparse Representation with Multikernel Fusion. *IEEE Trans. Circuits Syst. Video Technol.* **2014**, *24*, 1132–1141.
14. Li, X.; Dick, A.; Shen, C.; Zhang, Z.; Hengel, A.V.D.; Wang, H. Visual tracking with spatio-temporal Dempster-Shafer information fusion. *IEEE Trans. Image Process.* **2013**, *22*, 3028–3040. [[CrossRef](#)] [[PubMed](#)]
15. Yang, D.; Zhang, Y.; Ji, R.; Li, Y.; Huangfu, L.; Yang, Y. An Improved Spatial Histogram and Particle Filter Face Tracking. In *Genetic and Evolutionary Computing, Advances in Intelligent Systems and Computing*; Springer: Beijing, China, 2015; Volume 329, pp. 257–267.
16. Li, Y.; Wang, G.; Nie, L.; Wang, Q. Distance Metric Optimization Driven Convolutional Neural Network for Age Invariant Face Recognition. *Pattern Recognit.* **2018**, *75*, 51–62. [[CrossRef](#)]
17. Arulampalam, M.S.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 174–188. [[CrossRef](#)]
18. Dou, J.; Li, J.; Zhang, Z.; Han, S. Face tracking with an adaptive adaboost-based particle filter. In Proceedings of the 24th IEEE Chinese Control and Decision Conference (CCDC), Taiyuan, China, 23–25 May 2012; pp. 3626–3631.
19. Bos, R.; De Waele, S.; Broersen, P.M.T. Autoregressive spectral estimation by application of the burg algorithm to irregularly sampled data. *IEEE Trans. Instrum. Meas.* **2002**, *51*, 1289–1294. [[CrossRef](#)]
20. Wang, J.; Jiang, Y.X.; Tang, C.H. Face tracking based on particle filter using color histogram and contour distributions. *Opto-Electron. Eng.* **2012**, *39*, 32–39.
21. Swain, M.J.; Ballard, D.H. Color indexing. *Int. J. Comput. Vis.* **1991**, *7*, 11–32. [[CrossRef](#)]
22. Comaniciu, D.; Ramesh, V.; Meer, P. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 564–575. [[CrossRef](#)]
23. Kailath, T. The Divergence and Bhattacharyya Distance Measures in Signal Selection. *IEEE Trans. Commun. Technol.* **1967**, *15*, 52–60. [[CrossRef](#)]
24. Zhu, W.; Levinson, S. Edge Orientation-Based Multi-View Object Recognition. In Proceedings of the 15th International Conference on Pattern Recognition (ICPR), Barcelona, Spain, 3–7 September 2000; pp. 936–939.
25. Sobel, I. *An Isotropic 3×3 Gradient Operator, Machine Vision for Three-Dimensional Scenes*; Freeman, H., Ed.; Academic Press: New York, NY, USA, 1990; pp. 376–379.
26. Wang, J.; Yagi, Y. Integrating color and shape-texture features for adaptive real-time object tracking. *IEEE Trans. Image Process.* **2008**, *17*, 235–240. [[CrossRef](#)] [[PubMed](#)]
27. Visual Tracker Benchmark. Available online: http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html (accessed on 10 March 2019).

