

## Multiple Faces Tracking Using Feature Fusion and Neural Network in Video

Boxia Hu<sup>1,2,\*</sup>, Huihuang Zhao<sup>1</sup>, Yufei Yang<sup>1,3</sup>, Bo Zhou<sup>4</sup> and Alex Noel Joseph Raj<sup>5</sup>

<sup>1</sup>College of Mathematics and Econometrics, Hunan University, Changsha, 410082, China

<sup>2</sup>College of Mathematics and Statistics, Hengyang Normal University, Hengyang, 421002, China

<sup>3</sup>School of Computer Engineering and Applied Mathematics, Changsha University, Changsha, 410003, China

<sup>4</sup>College of Computer Science and Electronic Engineering, Hunan University, Changsha, 410082, China

<sup>5</sup>Key Laboratory of Digital Signal and Image Processing of Guangdong, Shantou, 515063, China

\*Corresponding Author: Boxia Hu. Email: huboxia@hynu.edu.cn

Received: 26 May 2020; Accepted: 07 August 2020

**Abstract:** Face tracking is one of the most challenging research topics in computer vision. This paper proposes a framework to track multiple faces in video sequences automatically and presents an improved method based on feature fusion and neural network for multiple faces tracking in a video. The proposed method mainly includes three steps. At first, it is face detection, where an existing method is used to detect the faces in the first frame. Second, faces tracking with feature fusion. Given a video that has multiple faces, at first, all faces in the first frame are detected correctly by using an existing method. Then the wavelet packet transform coefficients and color features from the detected faces are extracted. Furthermore, we design a backpropagation (BP) neural network for tracking the occasional faces. At last, a particle filter is used to track the faces. The main contributions are. Firstly, to improve face tracking accuracy, the Wavelet Packet Transform coefficients combined with traditional color features are utilized in the proposed method. It efficiently describes faces due to their discrimination and simplicity. Secondly, to solve the problem in occasional face tracking, and improved tracking method for robust occlusion tracking based on the BP neural network (PFT\_WPT\_BP) is proposed. Experimental results have been shown that our PFT\_WPT\_BP method can handle the occlusion effectively and achieve better performance over other methods.

**Keywords:** Face tracking; feature fusion; neural network; occlusion

### 1 Introduction

The problem of face tracking can be considered as finding an effective and robust way. It is based on using the geometric dependence of facial features to combine the independent face detectors of various facial features, and then get an accurate estimation of the position of each image's facial features in the video sequence [1]. Particle filter realizes recursive Bayesian filtering by the Nonparametric Monte Carlo simulation method [2]. It can be applied to any nonlinear system which can be described by the state-space model, and its accuracy can approach the optimal estimation. The Particle filter is simple and easy



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

to implement and provides an effective method for the analysis of nonlinear dynamic systems [3,4]. It has attracted extensive attention in the fields of target tracking, signal processing, and automatic control Bui Quang et al. [5]. It approximates the posterior distribution through a set of weighted assumptions which are called particles. During the tracker based on particle filter, there is a likelihood function that generates a weight for each particle and the particles are distributed according to a tracking model. Then, the particles are properly placed, weighted, and propagated. After calculating the posterior distribution of the particles, the most likely position of a face can be estimate sequentially [6,7]. In many cases including some complex backgrounds, particle filter achieves good performance and is used in more and more applications [8]. Though particle filter has a good performance in target tracking, there still exist some problems in face tracking. The common particle filter methods cannot handle an occlusion especially a full occlusion [9,10]. Some tracking results with and without occlusions for particle filter are shown in Fig. 1. The tracking performance becomes poor when a face occlusion occurs. The main reason is faces are similar and the re-sampling would propagate the wrong random samples since the likelihood of those occluded faces and lead to meaningless tracking. Therefore, dealing with occlusions is a crucial part of multiple faces tracking.



**Figure 1:** Examples of multiple faces tracking with and without occlusion

This paper presents an occlusion robust tracking (PFT\_WPT\_BP) method for multiple faces tracking. The three main contributions of this paper are summarized as follows:

- After detecting faces, wavelet packet decomposition is used to generate some frequency coefficients of images. We separately use its higher and lower frequency coefficients of the reconstructed signal to improve the faces tracking performance.
- We define a neural network for tracking the faces with occlusion. When faces tracking fails due to face-occlusion, the neural network is used to predict the next step of the occlusion face/faces.
- A method based on particle filter and multiple feature fusion for faces tracking in the video is proposed. The proposed method has good performance and is robust in multiple faces tracking.

## 2 Related Work

The particle filter algorithm is derived from the idea of Monte Carlo [11], which refers to the probability of an event by its frequency. Therefore, in the process of filtering, where probability such as  $P(x)$  is needed, variable  $x$  is sampled, and  $P(x)$  is approximately represented by a large number of samples and their corresponding weights. Thus, with this idea, particle filter can deal with any form of probability in the process of filtering, unlike Kalman filter [12], which can only deal with the probability of linear Gaussian distribution. This is one of the advantages of a particle filter.

Some researchers use a histogram method of tracking face [13]. Nevertheless, this method has a significant limitation, and many factors can affect the similarity of the two images, such as light, posture,

face vertical or left-right angle deviation, and so on. As a result, the face tracking result sometimes is poor, and it is hard to use in practical application. Reference [14] proposed a new face-tracking method which is based on the Meanshift algorithm. In these methods, the face position of the current frame is updated according to the histogram of the target in the previous frame and the image obtained in the current frame. These methods are suitable for single target tracking, and the effect is charming. However, when the non-target and target objects are occluded, it often leads to the temporary disappearance of the target [15]. When the target reappears, it is often unable to track the target accurately. Therefore, the robustness of the algorithm is reduced. Because color histograms are robust to partial occlusion, invariant to rotation and scaling, and efficient in computation, they have many advantages in tracking nonrigid objects. Reference [16] proposed a color-based particle filter for face tracking. In this method, Bhattacharyya distance is used to compare the histogram of the target with the histogram of the sample position during the particle filter tracking. When the noise of the dynamic system is minimal, or the variance of observation noise is microscopic, the performance of the particle filter is terrible. In these cases, the particle set quickly collapses to a point in state space. Reference [17] proposed a Kernel-based Particle Filter for face tracking. The standard particle filter usually cannot produce the set of particles that capture the “irregular” motion, which leads to the gradual drift of the estimated value and the loss of the target. There are two difficulties in tracking different numbers of nonrigid objects: First, the observation model and target distribution can be highly nonlinear and non-Gaussian. Secondly, the existence of a large number of different objects will produce overlapping and fuzzy complex interactions [18,19]. A practical method is to combine the hybrid particle filter with AdaBoost. The critical problem of the hybrid particle filter is the selection of scheme distribution and the processing of objects leaving and entering the scene. Reference [20] proposed a three-dimensional pose tracking method that mixes particle filter and AdaBoost. The hybrid particle filter is very suitable for multi-target tracking because it assigns a hybrid component to each player. The proposed distribution can be built by using a hybrid model that contains information from each participant’s dynamic model and test assumptions generated by AdaBoost.

### 3 The Framework of Our Multiple Faces Tracking Approach

Given a face model, a state equation is defined as

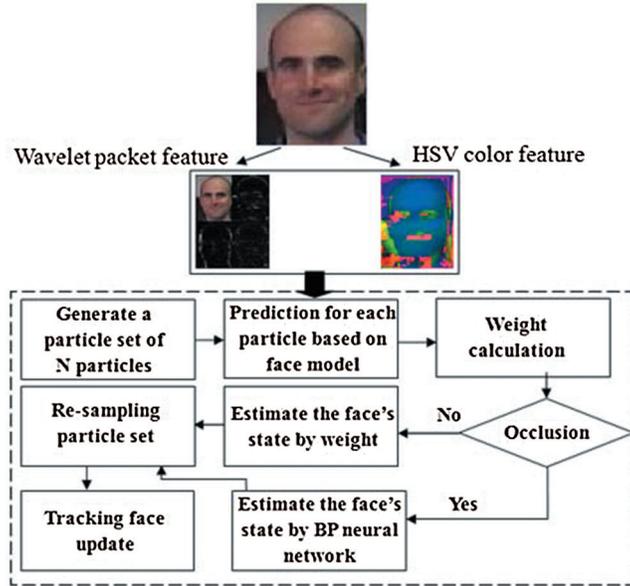
$$\begin{cases} x(t) = f(x(t-1), u(t), w(t)) \\ y(t) = h(x(t), e(t)) \end{cases} \quad (1)$$

where  $x(t)$  is the state at time  $t$ ,  $u(t)$  is the control quantity,  $w(t)$  and  $e(t)$  are the state noise and observation noise respectively. The former equation describes the state transition, and the latter is the observation equation. The framework of our multiple faces tracking approach is shown in Fig. 2.

For such faces tracking problems, particle filter filters out the real state  $x(t)$  from observation  $y(t)$ , and  $x(t-1)$ ,  $u(t)$  by following steps.

**Prediction stage:** Particle filter firstly generates a large number of samples according to the probability distribution of  $x(t-1)$ , which are called particles. Then the distribution of these samples in the state space is the probability distribution of  $x(t-1)$ . Well, then according to the state transfer equation and the control quantity, we can get a prediction particle for each particle.

**Correction stage:** After the observation value  $y$  arrives, all particles are evaluated by using the observation equation, i.e., conditional probability  $p(y|x_i)$ . To be frank, this conditional probability represents the probability of obtaining the observation  $y$  when assuming the real state  $x(t)$  takes the  $i$ th particle  $x_i$ . Let this conditional probability be the weight of the  $i$ -th particle. In this way, if all particles are evaluated, the more likely they are to get the particles observing  $y$ , the higher the weight of course.



**Figure 2:** The framework of multiple faces tracking algorithm using feature fusion

Resampling algorithm: Remove the particles with low-weight and copies the particles with high weight. What we get is, of course, the real state  $x(t)$  we need, and these resampled particles represent the probability distribution of the real state. In the next round of filtering, the resampled particle set is input into the state transition equation, and the predicted particles can be obtained directly.

Since we know nothing about  $x(0)$  at first, we can think that  $x(0)$  is evenly distributed in the whole state space. So the initial samples are distributed in the whole state space. Then all the samples are input into the state transition equation to get the predicted particles. Then we evaluate the weights of all the predicted particles. Of course, only some of the particles in the whole state space can get high weights. Finally, resampling is carried out to remove the low weight, and the next round of filtering is reduced to the high weight particles.

In our method, the state of a face at time  $t$  is denoted as  $s_t$  and its history is  $S = \{s_1; s_2; \dots; s_t\}$ . The basic idea of the particle filter algorithm is to compute the posterior state-density at time  $t$  using process density and observation density. Aiming to improve the sampling of the particle filter, we propose an improved algorithm based on combining with Wavelet Packet Transform and HSV color feature.

## 4 Face Tracking Model with Wavelet Packet Transform and Color Feature Fusion

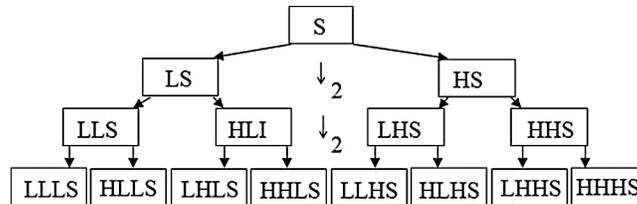
### 4.1 Face Feature Extraction Base on the Wavelet Packet Transform

The theory of Wavelet Packet Transform and feature extraction are introduced in this part. Wavelet packet analysis is an extension of wavelet analysis and will decompose not only the approximate but also the detail of the signal [21]. Wavelet packet decomposition provides the finer analysis as follows:

$$\begin{cases} u_{2n}(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} g_k(t) u_n(2t - k) \\ u_{2n+1}(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} h_k(t) u_n(2t - k) \end{cases} \quad (2)$$

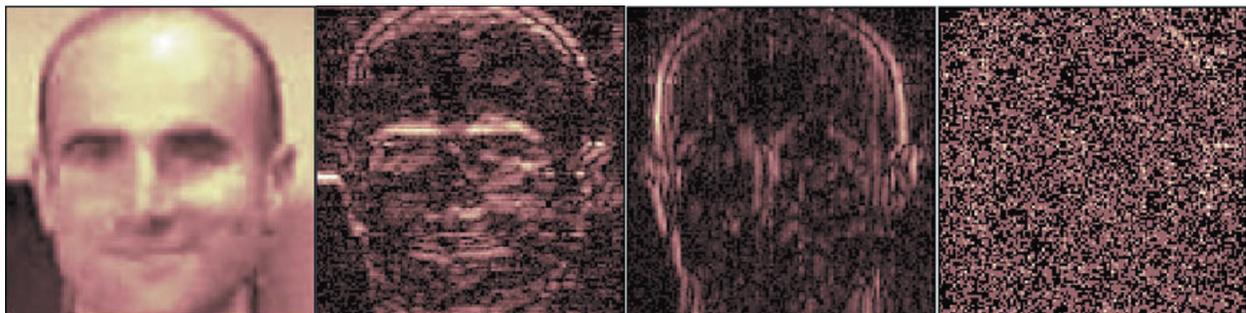
where  $g_k(t)$  and  $h_k(t)$  are a pair of complementary conjugate filters [22],  $t$  is a parameter in the time-domain,  $k = 1, 2, 3, \dots, N$ . The result of the Wavelet Packet Transform is shown as a full decomposition tree, as depicted

in Fig. 3. A low (L) and high (H) pass filter is repeatedly applied to the signal S, followed by decimation by Eq. (2), to produce a complete subband tree decomposition to some desired depth.



**Figure 3:** The wavelet packet decomposition tree

More face features are generated by wavelet packet decomposition and are used in face tracking. An example of face image decomposition is shown in Fig. 4.



**Figure 4:** Four wavelet packet coefficient features

### 4.2 Face Tracking Model

In our work, the face model is defined as parameters set  $s = (F_{color}, F_{wp}, R)$ , where R is a rectangle represented by  $R = (C_x, C_y, W, H)$ , and  $(C_x, C_y)$  is the center position, and W, H are the width and height of the rectangle. We consider the face as a discrete-time 2-dimensional motion with constant velocity. The state at time step k is denoted by  $s_k$ , our face dynamics model are modeled as

$$x_k = Px_{k-1} + Qn_k \tag{3}$$

where P is the transition matrix, Q is the system noise matrix and  $n_k$  is a Gaussian noise vector. This dynamic model is used in the sampling step of particle filters. With an assumption that the object moves with constant velocity, we can describe the motion in the following equations.

$$\begin{cases} x_{k+1} = x_k + (x_k - x_{k-1}) \\ y_{k+1} = y_k + (y_k - y_{k-1}) \end{cases} \tag{4}$$

where  $x_k, y_k$  represent the center of the target region at time step k. From this, the state vector  $x_k^{(i)}$  of the i-th particle at time step k, the system matrices P and Q, and noise vectors  $n_k$  are defined as Eqs. (5)–(7), respectively.

$$x_k^{(i)} = [x_k^{(i)}, y_k^{(i)}, x_{k-1}^{(i)}, y_{k-1}^{(i)}]^T \tag{5}$$

$$P = \begin{bmatrix} 2 & 0 & -1 & 0 \\ 0 & 2 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (6)$$

$$n_k^{(i)} = [n_{x_k}^{(i)}, n_{y_k}^{(i)}]^T \quad (7)$$

The observation process is performed to measure and weigh all the newly generated samples. The visual observation is a process of visual information fusion including two sub-processes: The computation sample weights  $w_k^{color}$ ,  $w_k^{wp}$  based on color histogram features and wavelet packet features, respectively. To consider timing issue, one- to four-level wavelet packet decomposition is used.

For the  $k$ th sample, we obtain the weight  $w_k^{color}$  through a Bhattacharyya similarity function [16] as shown in Eq. (8), calculating the similarity between sample histogram features  $F_k^{color}$  and the reference histogram features template  $F_{ref}^{color}$ .

$$w_k^{color} = \exp\{-D^2(F_k^{color}, F_{ref}^{color})\} \quad (8)$$

where  $D(F_k^{color}, F_{ref}^{color}) = (1 - \sum_{i=0}^m \sqrt{f_{i,k}^{color} \times f_{i,ref}^{color}})$  and  $m$  is the number of segments of the segmented color space. Here, we use color space with  $6 \times 6 \times 6$  bins.

To compute the sample weight  $w_k^{wp}$  based on the wavelet packet feature, the Euclidean distance between the sample feature vector  $v_k^{wp}$  and the reference feature vector  $v_{ref}^{wp}$  is employed in our system. The expression of  $w_k^{wp}$  is as follows:

$$w_k^{wp} = \exp\{-Eu(V_k^{wp}, V_{ref}^{wp})\} \quad (9)$$

where  $Eu(V_k^{wp}, V_{ref}^{wp}) = (\sqrt{\sum_{i=0}^{d-1} (v_{i,k}^{wp} - v_{i,ref}^{wp})^2})$ ,  $d$  is the number of feature dimensions.

With two different visual cues, we obtain the final weight for the  $k$ th sample as:

$$w(z_t^k | s_t^k) = \lambda_{color} w_k^{color} + \lambda_{wp} w_k^{wp} \quad (10)$$

where  $\lambda_{color} + \lambda_{wp} = 1$ ,  $\lambda_{color}$  and  $\lambda_{wp}$  are the coefficient values the weights of color histogram features and wavelet packet features, respectively. We can determine their values from experiences.

### 4.3 Multi-Face Tracking Algorithm Based on Particle Filter

Our multi-face tracking system consists of two parts: Automatic face detection and particle filter tracking. In the tracking system, the boosted face detector which is introduced above achieves automatic initializations when the system starts or when a tracking failure occurs. The face detection results are used to update the reference face model. The updating criterion is confidence values that are less than a threshold valued for  $M$  successive frames.

The proposed tracking algorithm includes four steps, as shown below.

#### 1. Initialization:

- i) Automatic face detection and  $n$  faces are detected.
- ii) Initialize the  $K$  particles.

2. Particle filter tracking: Probability density propagation from  $\{(s_{t-1}^{(k)}, \rho_{t-1}^{(k)}, c_{t-1}^{(k)})\}$  to  $\{(s_t^{(k)}, \rho_t^{(k)}, c_t^{(k)})\}$ , where  $\rho_t^k$  is the weights for the  $k$ th sample at time  $t$ , and  $c_t^k$  indicates the cumulative weights for the  $k^{th}$  sample at time  $t$ .

For  $k = 1:K$

For  $j = 1:n$

(1) Sample selection, generate a sample set as follows

- i) Generate a random number  $\lambda^j \in [0,1]$ , uniformly distributed;
- ii) Find, by binary subdivision, the smallest  $p$  for which  $c_{t-1}^{(jk)} \geq \lambda^j$ ;
- iii) Set  $s_t^{(jk)} = s_{t-1}^{jp}$ .

(2) Prediction, obtain  $s_t^{(jk)}$  with  $s_t^{(jk)} = Ps_t^{(jk)} + Qn_t^k$ .

(3) Weight measurement

- i) Calculate the color histogram feature according to Eq. (8).
- ii) Calculate the wavelet packet decomposition feature according to Eq. (9).
- iii) Calculate the sample weight according to Eq. (10).

End

End

3. For each face, normalize the sample set so that weight  $\sum_k w_t^{jk} = 1$ , and update them  $c_t^{(jk)} \geq \lambda^j$  as follows:

$$c_t^{(jk)} = c_t^{j(k-1)} + w_t^{jk}, c_t^{j(0)} = 0$$

4. Estimate state parameters of the pointing gesture at time-step  $t$ :

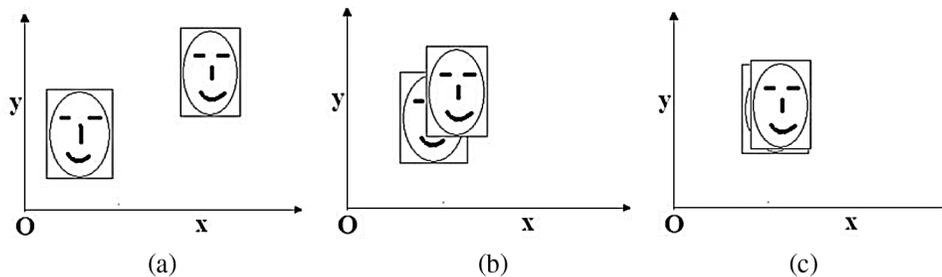
$$\hat{s}^j = \sum_{i=1}^N w_t^{j(i)} s_t^{j(i)} / \sum_{i=1}^N w_t^{j(i)}$$

5. If updating criteria are satisfied, go to Step 1; otherwise, go to Step 2.

### 5 Multiple Faces Tracking in Occlusions

An occlusion usually exists in multiple faces tracking and, it could cause a failure in the tracking of multiple faces because two objects are of high similarity. In our study, an occlusion tracking method combined with a neural network algorithm is proposed.

Take two faces as an example, if no faces are occluded, there are very few relationships between the particles in different faces. The two faces can be tracked by using a traditional particle filter. When a facial occlusion occurs, the particles in different faces will be occluded, as shown in Fig. 5. The overlapped area will affect the tracking result, even causes tracking failure.



**Figure 5:** Illustration of face occlusion ((a) no occlusion, (b) a part occlusion, (c) complete occlusion)

## (1) Occlusion detection

During initializing, the face  $i$  is marked as a rectangular region  $R_i^k$ , which is shown as in Fig. 1, where  $k$  is the frame,  $a_i$  is the area of  $R_i^k$ .

The overlapped area between two faces in  $k$ th frame is defined as

$$A_{ij}^k = a_i^k \cap a_j^k, \quad i \neq j \quad (11)$$

If  $\frac{A_{ij}^k}{\min(a_i^k, a_j^k)} > \lambda$ , face-occlusion occurs, otherwise, there is no face-occlusion, where  $\lambda$  is the threshold value.

## (2) The spatial position of overlapped area judgment

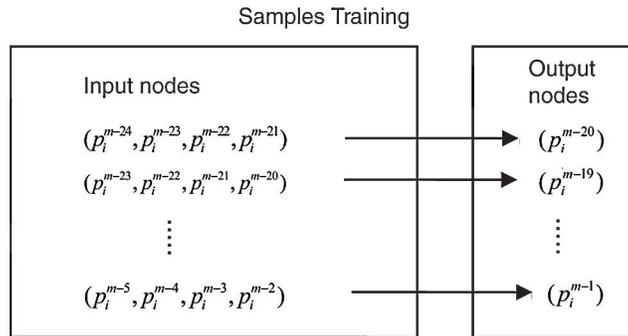
After the face-occlusion detecting, the spatial position of these faces must be judged. We can define the likelihood between  $A_{ij}^k$  and the detected face  $i$  area based on their color histogram.

$$L_i^k = \rho[A_{ij}^k, Q_i] \quad (12)$$

If  $L_i^k < L_j^k$   $i \neq j$ , face  $i$  is occluded in  $k$ th frame, otherwise face  $j$  is occluded in  $k$ th frame.

## (3) Face tracking update

Define  $p_i^k(\bar{x}_i^k, \bar{y}_i^k)$  as the center position of the rectangular region  $R_i^k$ , where  $k = 1, 2, 3 \dots N$ . If face  $i$  is occluded in  $m$ th frame, where  $m > 24$ .  $p_i^m(\bar{x}_i^m, \bar{y}_i^m)$  is the center position of the rectangular region  $R_i^m$ . The BP neural network training processing can be shown as in Fig. 6.



**Figure 6:** BP neural network training

Define  $X = \{\bar{x}_i^1, \bar{x}_i^2, \bar{x}_i^3 \dots \bar{x}_i^{m-1}\}$ , and  $Y = \{\bar{y}_i^1, \bar{y}_i^2, \bar{y}_i^3 \dots \bar{y}_i^{m-1}\}$ , during the three-layer BP neural network,  $X$ ,  $Y$  are chosen as samples to train BP neural network. For example, input cells are  $\{\bar{x}_i^k, \bar{x}_i^{k+1}, \bar{x}_i^{k+2}, \bar{x}_i^{k+3}, \bar{y}_i^k, \bar{y}_i^{k+1}, \bar{y}_i^{k+2}, \bar{y}_i^{k+3}\}$ , output cells are  $\{\bar{x}_i^{k+4}, \bar{y}_i^{k+4}\}$ , where  $0 < k < m-4$ . The tracker position  $(\bar{x}_i^m, \bar{y}_i^m)$  in  $m$ th frame can be estimated by BP neural network.

Choosing input nodes  $\{\bar{x}_i^{m-4}, \bar{x}_i^{m-3}, \bar{x}_i^{m-2}, \bar{x}_i^{m-1}, \bar{y}_i^{m-4}, \bar{y}_i^{m-3}, \bar{y}_i^{m-2}, \bar{y}_i^{m-1}\}$ , BP neural network can get output  $(\bar{x}_i^m, \bar{y}_i^m)$ , and the position of the target face is updated with  $(\bar{x}_i^m, \bar{y}_i^m)$ .

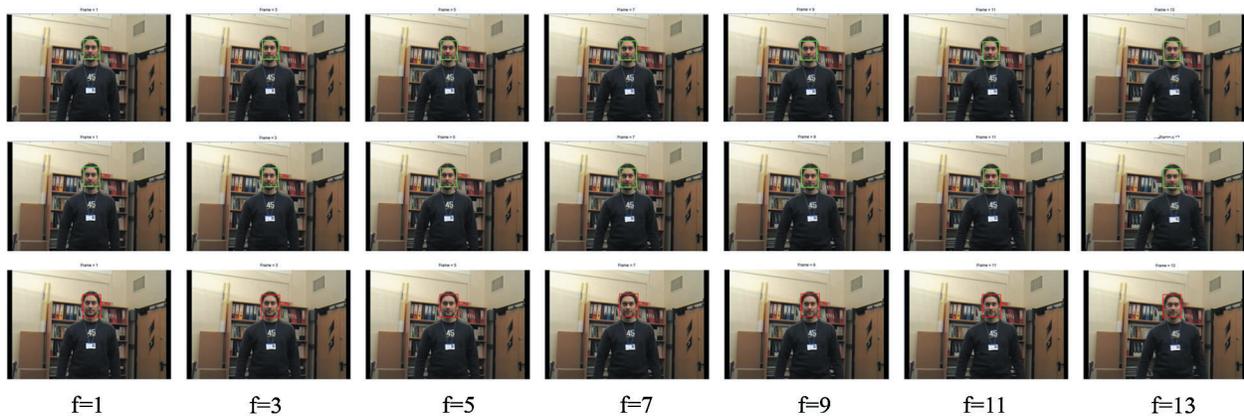
## 6 Experiment and Comparison

We experimented using video data sets downloaded from [20]. The experiments were implemented on a Intel(R) Xeon(R) E31220 3.1 GHz CPU and 8192 MB RAM. The resolution of each frame was  $720 \times 480$  pixels per image. During our research, we do not focus on face detection, since many existing

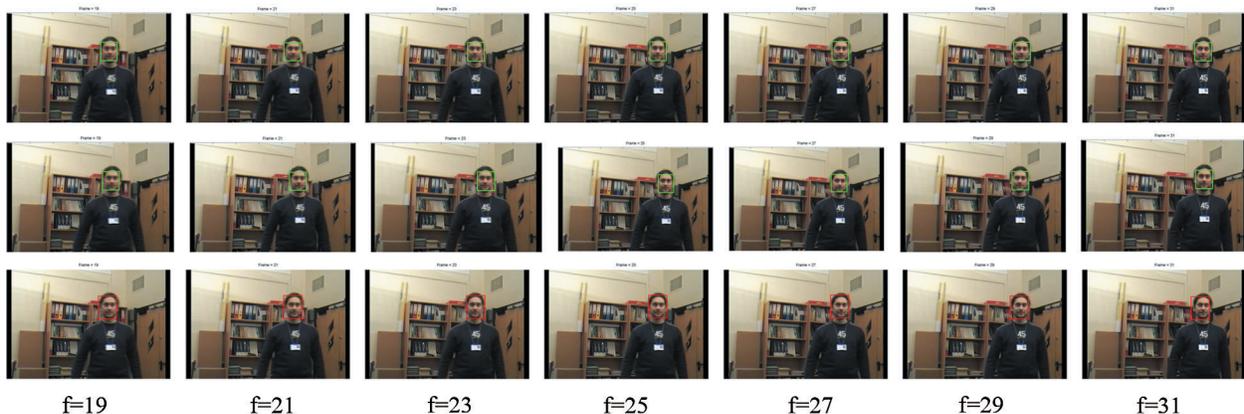
methods can be used to detect faces [21,22]. Comparing with other methods, the method [22] performed well and provided excellent results achieving a higher detection rate, and it was used in our face detection.

### 6.1 One Face Tracking Results

We have carried out some experiments to track one face with our proposed method PFT\_WPT\_BP, and the particle filter number was 200. The color square showed the region of the tracked face. Fig. 7 showed the experimental results of one face tracking (even numbers in frame 1 to 13) based on different methods with Kalman Filter [12], Particle Filtering [10] and PFT\_WPT\_BP, where  $f$  is a label of the frame in the video. And Fig. 8 showed the tracking results of one face tracking in frame 19 to 31. We can find that satisfactory experimental results were achieved in three methods.



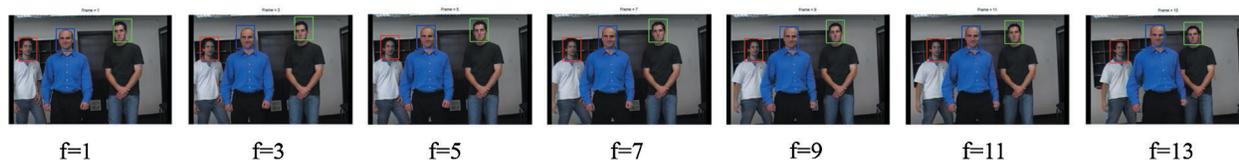
**Figure 7:** One face tracking base on different methods (line 1 is based on Kalman Filter [12], line 2 is based on particle filtering [10], line 3 is based on PFT\_WPT\_BP)



**Figure 8:** One face tracking base on different methods (line 1 is based on Kalman Filter, line 2 is based on particle filtering, line 3 is based on PFT\_WPT\_BP)

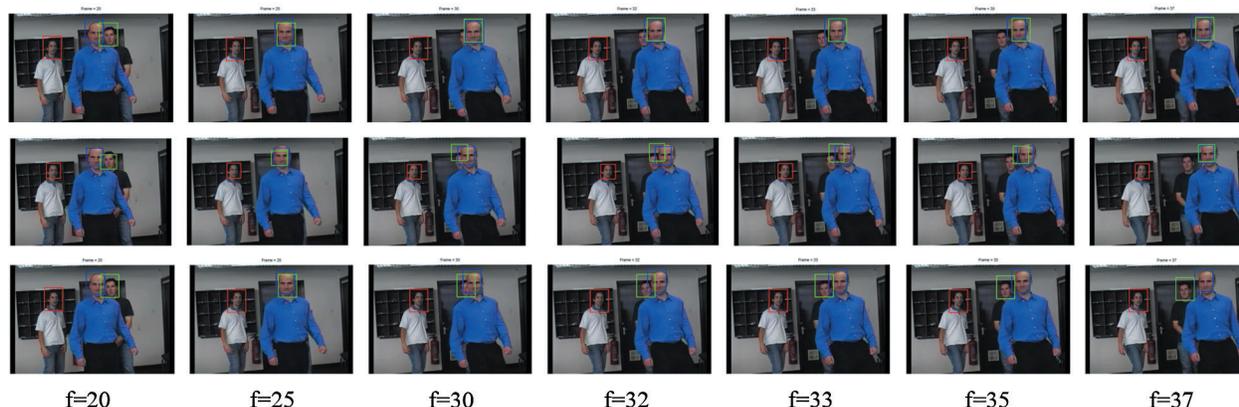
### 6.2 Multiple Faces Tracking Results

We have carried out some experiments to track multiple faces too, and the particle filter number was 200. The colored square shows the region of the tracked face. Fig. 9 shows the experimental results of three faces tracked (frame 1 to 13) which provided satisfactory results.



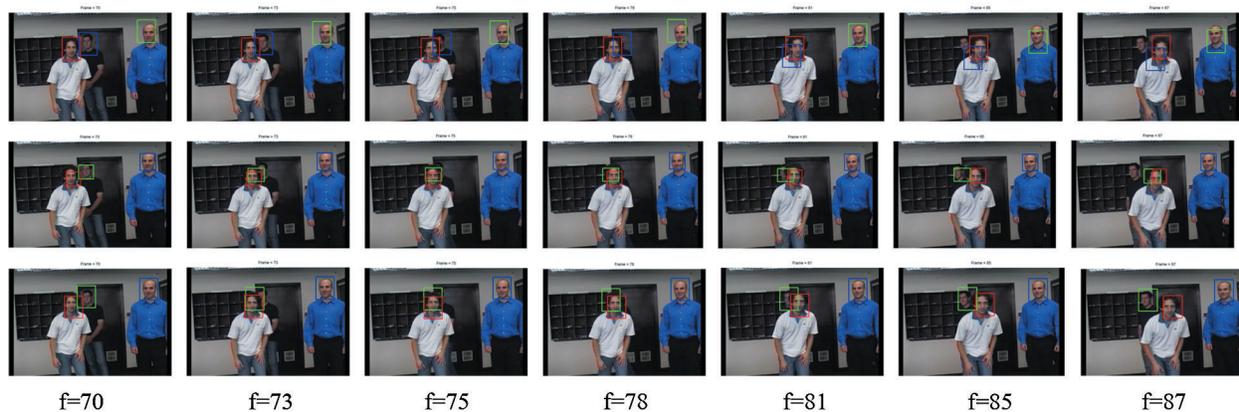
**Figure 9:** Multiple faces tracking in no-occlusion based on PFT\_WPT\_BP

With face-occlusion (frame 20 to 37), we experimented using different tracking methods. The face of the first person (blue cloth) occluded the face of the second person (black cloth) in frame 20 to 33. And Fig. 10 shows faces tracking with occlusion based on different methods. We found that the faces tracking failed for occluded faces based on Kalman Filter and Particle Filtering methods as indicated in Fig. 10 (line 1 and 2). But our method could achieve acceptable results in Fig. 10 (line 3).



**Figure 10:** Multiple faces tracking in occlusion based on different methods (line 1 is based on Kalman Filter, line 2 is based on particle filtering, line 3 is based on PFT\_WPT\_BP)

After several frames, the third person's face (white cloth) occluded the second person's face (black cloth) at the beginning of the frame. We detected the faces again, and found that the faces tracking failed in face-occlusion after frame 70. The results based on Kalman Filter and particle filtering is shown in Fig. 11 (lines 1 and 2). The occluded face was successfully tracked based on our method as shown in Fig. 11 (line 3). The system successfully recovered the faces from occlusion. After the occlusion, each face was normally re-sampled and the face appearances were updated again.



**Figure 11:** Multiple faces tracking in occlusion based on different methods (line 1 is based on Kalman Filter, line 2 is based on particle filtering, line 3 is based on PFT\_WPT\_BP)

## 7 Conclusion

This paper presents an occlusion robust tracking method for multiple faces. Experimental results have been shown that our PFT\_WPT\_BP method can handle the occlusion effectively and achieve better performance than several previous methods. BP neural network is used to predict the occasional faces. We assume that the occasional face would not miss a long time. If a face is missing for a long time, it is difficult to track it, and we can find the face by face detection. The faces tracking in a more complex environment will be researched in our future work.

**Funding Statement:** The Project Supported by Scientific Research Fund of Hunan Provincial Education Department (16C0223), the Project of “Double First-Class” Applied Characteristic Discipline in Hunan Province (Xiangjiaotong [2018] 469), the Project of Hunan Provincial Key Laboratory (2016TP1020).

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] J. Goenette, L. Unzueta, F. Dornaika and O. Otaegui, “Efficient deformable 3D face model tracking with limited hardware resources,” *Multimedia Tools & Applications*, vol. 79, no. 6, pp. 12373–12400, 2020.
- [2] M. Tian, Y. Bo and Z. Chen, “Multi-target tracking method based on improved firefly algorithm optimized particle filter,” *Neurocomputing*, vol. 359, no. 24, pp. 438–448, 2019.
- [3] G. S. Walia, A. Kumar, A. Sexena, K. Sharma and K. Singh, “Robust object tracking with crow search optimized multi-cue particle filter,” *Pattern Analysis & Applications*, vol. 3, no. 1, pp. 434–457, 2019.
- [4] K. Yang, J. Wang, Z. Shen, Z. Pan and W. Yu, “Application of particle filter algorithm based on gaussian clustering in dynamic target tracking,” *Pattern Recognition and Image Analysis*, vol. 29, no. 3, pp. 559–564, 2019.
- [5] P. B. Quang, C. Musso and F. L. Gland, “Particle filtering and the laplace method for target tracking,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 1, pp. 350–366, 2016.
- [6] T. T. Yang, H. L. Feng, C. M. Yang, G. Guo and T. S. Li, “Online and offline scheduling schemes to maximize the weighted delivered video packets towards maritime CPSs,” *Computer Systems Science and Engineering*, vol. 33, no. 2, pp. 157–164, 2018.
- [7] Y. T. Chen, J. Wang, S. J. Liu, X. Chen, J. Xiong *et al.*, “Multiscale fast correlation filtering tracking algorithm based on a feature fusion model,” *Concurrency and Computation: Practice and Experience*, vol. 31, no. 10, pp. e5533, 2019.
- [8] X. Zhang, W. Lu, F. Li, X. Peng and R. Zhang, “Deep feature fusion model for sentence semantic matching,” *Computers, Materials & Continua*, vol. 61, no. 2, pp. 601–616, 2019.
- [9] Y. T. Chen, W. H. Xu, J. W. Zuo and K. Yang, “The fire recognition algorithm using dynamic feature fusion and IV-SVM classifier,” *Cluster Computing*, vol. 22, no. 3, pp. 7665–7675, 2019.
- [10] T. Wang, W. Wang, H. Liu and T. Li, “Research on a face real-time tracking algorithm based on particle filter multi-feature fusion,” *Sensors*, vol. 19, no. 5, pp. 1245, 2019.
- [11] Q. Wang, M. Kolb, G. O. Roberts and D. Steinsaltz, “Theoretical properties of quasi-stationary Monte Carlo methods,” *Annals of Applied Probability*, vol. 29, no. 1, pp. 434–457, 2019.
- [12] P. R. Gunjal, B. R. Gunjal, H. A. Shinde, S. M. Vanam and S. S. Aher, “Moving object tracking using kalman filter,” in *International Conf. on Advances in Communication and Computing Technology*, Sangamner, India, pp. 544–547, 2018.
- [13] W. Singh and R. Kapoor, “Online object tracking via novel adaptive multicue based particle filter framework for video surveillance,” *International Journal of Artificial Intelligence Tools*, vol. 27, no. 6, 1850023, 2018.
- [14] M. Long, F. Peng and H. Y. Li, “Separable reversible data hiding and encryption for HEVC video,” *Journal of Real-Time Image Processing*, vol. 14, no. 1, pp. 171–182, 2018.

- [15] S. Sonkusare, D. Ahmedt-Aristizabal, M. J. Aburn, V. T. Nguyen, T. Pang *et al.*, “Detecting changes in facial temperature induced by a sudden auditory stimulus based on deep learning-assisted face tracking,” *Scientific Reports*, vol. 9, no. 1, pp. 4729, 2019.
- [16] R. D. Kumar, B. N. Subudhi, V. Thangaraj and S. Chaudhury, “Walsh-Hadamard kernel based features in particle filter framework for underwater object tracking,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 5712–5722, 2019.
- [17] B. Eren, E. Egrioglu and U. Yolcu, “A hybrid algorithm based on artificial bat and backpropagation algorithms for multiplicative neuron model artificial neural networks,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 2, no. 6, pp. 1593–1603, 2020.
- [18] K. Picos, V. H. Diaz-Ramirez, A. S. Montemayor, J. J. Pantrigo and V. Kober, “Three-dimensional pose tracking by image correlation and particle filtering,” *Optical Engineering*, vol. 57, no. 7, 073108, 2018.
- [19] Y. Song, G. B. Yang, H. T. Xie, D. Y. Zhang and X. M. Sun, “Residual domain dictionary learning for compressed sensing video recovery,” *Multimedia Tools and Applications*, vol. 76, no. 7, pp. 10083–10096, 2017.
- [20] J. Wang and Y. Yagi, “Integrating color and shape-texture features for adaptive real-time object tracking,” *IEEE Transactions on Image Processing*, vol. 17, no. 2, pp. 235–240, 2008.
- [21] H. H. Zhao, P. L. Rosin and Y. K. Lai, “Block compressive sensing for solder joint images with wavelet packet thresholding,” *IEEE Transactions on Components Packaging & Manufacturing Technology*, vol. 9, no. 6, pp. 1190–1199, 2019.
- [22] J. Chen, D. Jiang and Y. Zhang, “A common spatial pattern and wavelet packet decomposition combined method for EEG-based emotion recognition,” *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 23, no. 2, pp. 274–281, 2019.