A Method of Improving Accuracy in Expression Recognition

Zhi-Jie Li

Abstract — In order to improve the accuracy of a special kind of facial expression recognition problem, a method for precise face detection and segmentation combined with the particle swarm optimization is proposed. The method uses three key technologies: skin color segmentation, particle swarm search and curve approximation. Firstly, the face contour is roughly obtained through skin color segmentation. Secondly, the accurate face position is detected by particle swarm optimization. Thirdly, the face contour is reduced and regulated further via the curve approximation. The experimental results show that this method can eliminate the interference factor, and then improve the accuracy of expression recognition.

Keywords — Expression recognition; face detection; particle swarm optimization; skin model.

I. INTRODUCTION

In recent years, along with the increase in computer hardware performance and the expectations of human-computer interaction, expression recognition has become a hot subject. As a typical advanced biometric identification technology, expression recognition [1]-[4] mainly contains three core steps: face detection, expression feature extraction, and emotion classification. That is to say, after the face detection, the facial expression information is extracted using the feature extraction method, and then the expression is classified according to extracted expression information. Among them, face feature extraction is the key step in facial expression recognition. It decides the final recognition result and then affects the recognition rate. The feature extraction method easiest to understand is a geometric method, which deals with expression recognition through shape variables of key face parts. The representative algorithms are the face movement coding system (FACS) [5], [6]and face animation parameter (FAP) [7], [8]. But this kind of method is not only complex but also requires manual intervention. The feature extraction method easiest to realize is the statistical method, which deals with expression recognition through a certain transformation projection. Now the statistical methods have been successfully applied to expression recognition are principal component analysis (PCA) [9] and independent component analysis (ICA) [10]. This kind of method is not only simple, and tight but achieves good recognition results. Our work on facial expression recognition is based on feature extraction using statistical method.

It has been found that the existing studies on expression recognition mainly contain two categories: One is expression recognition focus on different person. The other is expression recognition focus on the same person. Expression recognition focuses on the same person and is usually used to investigate individual emotional change. Relevant studies are rare. The typical research is in 2011, Tanveer [11] developed a face expression classification system based on PCA. This system can distinguish different expressions of the same individual. PCA method is used for reducing dimensions on the complex expression data sets. The revealed implicit information is the most significant expression characteristic. But the recognition accuracy of the system needs to be improved further.

This paper focuses on the second category of expression recognition problem — recognition for the same individual. We investigate how to improve the accuracy of facial expression recognition.

II. PREPROCESS

In order to ensure the consistency of face size, position and quality in the image, we must preprocess the face image. Preprocessing steps includes grey transformation, binary, cropping and normalization. After preprocessing, image quality can be improved and get a standard, segmented face image. In this paper, we use a standard face image to perform the experiment.

First, the image is transformed from RGB color space to the YCbCr color space [12], [13]. Images in YCbCr color space are stored in the computer as a three-dimensional matrix. The first dimension of the matrix \(Y\) represents the luminance component. The second dimension \(Cb\) represents the blue component. The third dimension \(Cr\) represents red component. The element’s value of each dimension in the matrix represents the size of the \(Y\), \(Cb\), \(Cr\), respectively. As mentioned before, in YCbCr color space, the pixel value distribution is different between the skin region and other regions in the image. So, we extract the \(Cr\) component in the three-dimensional matrix of YCbCr. Then the grey image of \(Cr\) component is saved as a two-dimensional matrix. In order to quantitatively detect the face area, the grey image of \(Cr\) component needs binary processing, which makes the face area more prominent. Thus, we can get the formula for judging face color as follows:

\[
f(i, j) = \begin{cases} 1, & C_r \in [a, b] \\ 0, & \text{else} \end{cases}
\]

in which \(f(i, j)\) is the image after threshold segmentation; \(a, b\) on behalf of the selected threshold range; \(i, j\) represent the two-dimensional coordinates of each pixel.
The binary image contains a lot of noise after the color segmentation. In addition, different illumination leads to some discontinuous black areas on the face. Therefore, we need to use morphology to preprocess the binary image to get rid of these holes that existed in the binary image. The basic morphology computing includes expansion, corrosion, open operation, and close operation. In this paper, we use a close operation to get face region images without holes. Finally, although the skin region has been extracted, the facial part still cannot be distinguished strictly from other body parts such as arms. This is the key to the low accuracy of expression recognition.

III. FEATURE EXTRACTION

After image preprocessing, we begin to extract the expression feature. Obviously, feature recognition using all the pixels in the image is not feasible on efficiency. People cannot tolerate a very long time. So, these pixels need filtering, conversion, and abandonment. Only the representative pixels are selected to be involved in recognition. There are many methods of simplifying the data set or reducing dimensions. Now, the widely used method is the principal component analysis (PCA). Usually, PCA method has three advantages, such as dimension reduction, correlation removal, and probability estimation. In statistics, PCA is the linear transformation technology mainly used to simplify the data set. This transformation can transform data into a new coordinate system so that the first large variance of any data projection is in the first coordinate (referred to as the first principal component), and the second large variance is in the second coordinate (referred to as the second components), the rest can be done in the same manner. PCA is often used to reduce the dimension of the data set while maintaining the characteristic of the data set having the largest contribution to the variance at the same time. This is achieved by preserving the principal components of low order and ignoring those of higher order. Because the principal components of low order always keep the most important aspect of the data set. The theoretical description about PCA is as follows:

A face image of \( m \times n \) size can be represented as a one-dimensional vector in space \( R^{mn} \). Then the training set \( X_{\text{training}} \) composed of \( D \) face images is represented as \( mn \times D \) matrix, namely, \( X_{\text{training}}=[x_1, x_2, \ldots, x_D] \). In which, each face is a \( mn \times 1 \) vector, namely, \( x_i=[x_{i1}, x_{i2}, \ldots, x_{imn}] \). Training set \( X_{\text{training}} \) is a matrix composed of all training images, also called a training matrix. Then, the training matrix should be normalized. That is to say, let \( X_{\text{training}} \) minus average column vector \( \bar{x} \) to get standard training matrix \( \mathbf{X} \).

\[
\mathbf{X} = [x_1 - \bar{x}, x_2 - \bar{x}, \ldots, x_D - \bar{x}]
\]

(2)

where, \( \bar{x} = \frac{1}{D} \sum_{i=1}^{D} x_i \). Next, we solve the eigenvalue \( \lambda \) and eigenvector \( \mathbf{P} \) of the covariance matrix \( \mathbf{S} = \frac{1}{D-1} \mathbf{XX}^T \). Eigenvalue \( \lambda \) is taken as the contribution rate of the principal component. If arranged in descending order, Eigen value is \( \lambda = [\lambda_1, \lambda_2, \ldots, \lambda_N] \) \( (\lambda_1 > \lambda_2 > \ldots > \lambda_N) \). Eigen vector \( \mathbf{P} \) is called transformation matrix or projection matrix, which is correspondence with descending Eigen value \( \lambda \).

Let \( \mathbf{X}' \) multiply projection matrix \( \mathbf{P} \) to get new matrix \( \mathbf{Y} \). \( \mathbf{Y} \) is a \( D \times mn \) matrix, which is the same size with \( \mathbf{X}' \). \( \mathbf{Y} \) is also called the principal component matrix. And \( \mathbf{P} \) is a \( mn \times mn \) square matrix. Then, the relationships between \( \mathbf{X}' \) and \( \mathbf{Y} \) is:

\[
\mathbf{Y} = \mathbf{X}' \mathbf{P}
\]

(3)

In the transformed principal component matrix, \( \mathbf{x}_1 \) and \( \mathbf{y}_i \) is one-to-one correspondence. That is to say, \( \mathbf{y}_i \) is the updated representation of \( \mathbf{x}_1 \). \( \mathbf{Y}_1 \) has the largest variance and is called the first principal component. \( \mathbf{Y}_2 \) has the second largest variance and is called the second principal component. The rest can be done in the same manner. \( mn \) original variables will be transformed into \( mn \) principal components. Since the total variance is constant, the variances of frontal variables are larger (such as \( \mathbf{Y}_1, \mathbf{Y}_2 \)), and the variances of latter variables are smaller (such as \( \mathbf{Y}_{mn-1}, \mathbf{Y}_{mn} \)). Strictly, only the frontal variables are the principal component, while the latter variables are the secondary components. So, we keep the frontal variables and ignore the latter variables. As for how many frontal variables we should keep, it is thought that the determinant is the percentage of the cumulative variance of the kept variables to the total variance. The percentage can be also regarded as the cumulative contribution rate, which means the information quantity is represented by frontal principal components. Usually, we set 95% to be the threshold to keep how many principal components. The contribution rate can be calculated using the Eigen value \( \lambda \).

IV. EXPRESSION CLASSIFICATION AND RECOGNITION

To simplify the studied problem, we labelled the face images using four expressions in the training set. Therefore, the expression recognition problem can be regarded as a distance matching problem among images, after the feature extraction step. Through PCA dimension reduction, the face image of the training set is projected into a new subspace and obtained a set of coordinates. In other words, the face image containing tens of thousands of variables is simplified into a few dozens of representative variables. Similarly, test images need to be projected into this new subspace. Thus, test images are also simplified into a few dozens of variables. At this time, matching the training images and the test images would be much easier. Matching is done through calculating the distance between the test images and the training images, according to certain distance criterion. Finally, based on the obtained similarity with various expressions, the facial expression is classified. Because there are not many feature points after dimension reduction, this paper takes the general Euclidean distance to define the true distance between any two points in multi-dimensional space. The shorter the distance, the more similarity between the two expressions. If there are \( r \) principal components kept after dimension reduction, then the \( t \)-dimensional Euclidean space is a point set. The distance between two points \( x_1 \) and \( x_2 \) is defined as the following formula.

\[
d(x_1, x_2) = \sqrt{\sum_{i=1}^{r} (x_{i1} - x_{i2})^2}
\]

(4)

where \( x_{il} \) represents the \( l \)-dimensional coordinate of the
first point; \( x_{i2} \) represents the \( i \)-dimensional coordinate of the first point. The smaller the value of \( d(x_1, x_2) \), the more similar the two facial expressions of \( x_1 \) and \( x_2 \).

V. IMPROVING RECOGNITION ACCURACY

A. Face Detection Based on PSO

Particle swarm algorithm [14]-[17] is a global optimization algorithm; it can solve a large number of nonlinear, non-differentiable, and multi-peak complex optimization problems. Individual in PSO represents a possible solution. It is known that most information is stored on the image edge. Also, face location is based on edge information. After image preprocessing, a binary image of the face image can be obtained. The next work is all performed on the edge of the binary image. The face region can be located roughly using particle swarm searching on the edge image. Due to page limitations, the detailed process of detecting faces using PSO will not be discussed here. For more information, see our previous work [18].

B. Face Contour Smooth by Curve Fitting

After the above treatment, we have got an initial face contour. Obviously, this is not our desired result and needs to be further refined. Compared with irregular face contour, standard face contour is easier to be recognized. Hence, in order to get a regular face contour, we need to smooth the initial contour. In order to avoid the endpoint oscillation phenomenon, the cubic smoothing spline is selected to curve fitting. We use the piecewise cubic polynomial to interpolate, i.e., for a given sequence \( \{(x_k, y_k)\}_{k=0}^{N} \) with \( N+1 \) nodes, solving \( N \) three-degree polynomial. The coefficients of the three-degree polynomial \( s_{k,0}, s_{k,1}, s_{k,2}, s_{k,3} \) should satisfy the following equation (5).

\[
S(x) = S_k(x) = s_{k,0} + s_{k,1}(x-x_k) + s_{k,2}(x-x_k)^2 + s_{k,3}(x-x_k)^3 \quad \text{ for } x \in [x_k, x_{k+1}], \quad k = 0, 1, \ldots, N-1
\]

That is to say, function \( S(x) \) is a three-degree polynomial at each interval \([x_k, x_{k+1}]\). At the same time, the three-degree polynomial must satisfy three conditions: pass by a given sequence of nodes, be smooth continuous functions, the second derivative is continuous as well. Now, the curve fitting problem using cubic smoothing spline is reduced to solving a set of coefficients \( s_{k,0}, s_{k,1}, s_{k,2}, s_{k,3} \). which satisfy the above conditions. According to the property of the spline curve, the following relationship of the second derivative is obtained.

\[
h_k m_{k-1} + 2(h_{k-1} + h_k) m_k + h_k m_{k+1} = u_k, \quad k = 1, 2, \ldots, N-1
\]

where, \( u_k = 6(d_k - d_{k-1}) \), \( d_k = \frac{y_{k+1} - y_k}{h_k} \), \( h_k = x_{k+1} - x_k \).

Using the natural cubic spline constraints \( m_0 = 0, m_N = 0 \), we can rewrite equations (6) from equation 1 to equation \( N-1 \). Then, we get triangular linear equations \( HM = V \) containing \( m_1, m_2, \ldots, m_{N-1} \), which is expressed as equation (7).

\[
\begin{bmatrix}
b_1 & c_1 \\
a_1 & b_1 & c_2 \\
\vdots & \vdots & \ddots & \vdots \\
a_N \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_{N-1} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{N-1} \end{bmatrix}
\]

Equation (7) has strict diagonal dominance and has unique solution. When the coefficient \( \{m_k\} \) is obtained, we can use the following equations to calculate spline coefficients \( \{s_{k,i}\} \) of \( S_k(x) \).

\[
s_{k,0} = y_k, \quad s_{k,1} = d_k - \frac{h_k (2m_k + m_{k+1})}{6}, \quad s_{k,2} = \frac{m_{k+1} - m_k}{6h_k}, \quad s_{k,3} = \frac{m_{k+1} - m_k}{6h_k}
\]

According to the above coefficient \( \{s_{k,i}\} \), we can solve the corresponding polynomial \( S_k(x) \) for each curve section.

Next, we determine which node should be selected for curve fitting. The investigation process experienced three stages. In the first stage, we plan to select four nodes. Selecting method is as follows: first, locate the center point \( O \) of the rectangular; then, make four rays from \( O \) to the center points of four edges of the rectangular. These four rays form many intersection points with the contour of rectangular and the initial face contour. Only the internal intersection points are selected. Finally, four fitting nodes are found, i.e., top, below, left, and right (see Fig. 1(a)).

![Fig. 1. Results of curve fitting based on different nodes. (a) selecting 4 nodes; (b) curve fitting based on 4 nodes; (c) face segmentation based on 4 nodes; (d) selecting 8 nodes; (e) curve fitting based on 8 nodes; (f) face segmentation based on 8 nodes; (g) selecting 6 nodes; (h) curve fitting based on 6 nodes; (i) face segmentation based on 6 nodes.](http://dx.doi.org/10.24018/ejece.2021.6.3.440)
sequence to join the first node and end node. That means the first node is overlapped with the end node. Thus, the final node sequence is top, below, left, right, and top.

According to the sequence of the five nodes, we can make a curve fitting for the face contour marked with a blue line, shown as Fig. 1(b). Based on the contour, the face region can be selected, shown as in Fig. 1(c). This smooth curve has the advantages of good chin fitting but has the disadvantages of too narrow forehead fitting.

So, we continue to adjust in the second stage. We plan to add more nodes by selecting eight nodes. Selecting method adds four nodes besides the former four nodes. Four rays are made from O to the four corners of the rectangular. These four rays form many intersection points with the contour of rectangular and the initial face contour. Only the internal intersection points are selected. Now, we get a node sequence composed of eight nodes, i.e., top, top-right, right, lower-right, below, lower-left, left, and top-left (see Fig. 1(d)). According to the sequence, we can make a curve fitting for face contour, as shown in Fig. 1(e). This smooth curve has the advantages of good forehead fitting but has the disadvantages of too wide chin fitting. This result is just opposite to the result with four node sequence. The two methods have complementary advantages. Therefore, we determine a compromise scenario by selecting six nodes.

In the third stage, two nodes (lower-right and lower-left) of the eight nodes are removed. A sequence with six nodes is formed, i.e., top, top-right, right, below, left, top-left (see Fig. 1(g)). According to the sequence, we can make curve fitting for face contour, shown in Fig. 1(h). The final smooth curve is quite similar to the face contour, retaining important information.

After the above processing, we completely remove the unrelated parts with the expressions, and then get a precise face contour which plays an important role in improving the accuracy of face segmentation, and hence will improve the accuracy of the expression recognition.

VI. CONCLUSION

In order to improve the accuracy of expression recognition, we proposed a method utilizing the particle swarm optimization (PSO) algorithm and curve fitting for precise face location and segmentation. We focused on the facial expression classification for the same individual. Although different expressions have different appearances, but the accurate expression recognition is still difficult because of the similarity in the expressions. So, the application of statistical methods in expressions recognition has not high accuracy. As we know that, the more similar part, the more susceptible. We start from this point, and refine the results of face preprocessing, because image preprocessing is the main factor of accuracy in expression recognition. The PSO algorithm is introduced into face detection and segmentation. Based on the skin segmentation, the PSO algorithm takes particle advantage of global optimization searching to face region, which removes the hair, ears, neck, and other interference regions. The cubic smoothing spline is selected to curve fitting to get a regular face contour. In order to avoid the endpoint oscillation phenomenon. We use the piecewise cubic polynomial to interpolate. These treatments play key roles in expression recognition. Especially for similar expressions, only the accurate face can get the correct matching results. Our experimental results show that facial expression recognition using PSO algorithm and curve-fitting produces precise facial contour. The conclusion demonstrates that the elimination of excessive interference factors can increase the matching accuracy, and hence will improve the recognition accuracy.

REFERENCES


DOI: http://dx.doi.org/10.24018/ejece.2021.6.3.440