JOURNAL OF VISUAL COMMUNICATION AND IMAGE REPRESENTATION
Vol. 8, No. 3, September, pp. 278–290, 1997
ARTICLE NO. VC970359

Facial Expression Recognition Using Model-Based Feature Extraction and Action Parameters Classification

Chung-Lin Huang* and Yu-Ming Huang

Institute of Electrical Engineering, National Tsing-Hua University, Hsin-Chu, Taiwan, Republic of China

Received August 5, 1996; accepted July 16, 1997

This paper introduces an automatic facial expression recognition system which consists of two parts: facial feature extraction and facial expression recognition. The system applies the point distribution model and the gray-level model to find the facial features. Then the position variations of certain designated points on the facial feature are described by 10 action parameters (APs). There are two phases in the recognition process: the training phase and the recognition phase. In the training phase, given 90 different expressions, the system classifies the principal components of the APs of all training expressions into six different clusters. In the recognition phase, given a facial image sequence, it identifies the facial expressions by extracting the 10 APs, analyzes the principal components, and finally calculates the AP profile correlation for a higher recognition rate. In the experiments, our system has demonstrated that it can recognize the facial expression effectively. © 1997 Academic Press

1. INTRODUCTION

The human visual system can easily understand different facial expressions. However, using the computer to recognize human facial expression is a notrivial task. Recently, several researchers [2–11] have attempted the automatic recognition of facial expression. It can be widely applied as a part of an effort to develop basic techniques for virtual space teleconferencing in which the machine can recognize human facial expressions and then reproduce the human facial images with realistic expressions in a remote location.

Ekman and Friesen [1] have proposed the facial action coding system (FACS) which represents the facial expression by a set of facial action units. An expression can be viewed as a point in the space spanned by the action units. Mase [2] presented the facial expression recognition method in two ways. First, the optical flow field of skin movement is evaluated in muscle windows, each of which defines one primary direction. Second, the expression recognition system uses the 15 feature vectors for facial expression categorization. He showed an accuracy rate of nearly 80% for recognizing four types of expressions: happiness, anger, disgust, and surprise. Morishima [8] developed a facial emotional model which employs a representation facial feature action based on the description of the epic of facial expression. Facial expression is described as the point of the space and the face animation can be described by the locus in 3D emotion space.

Yacoob and Davis [3] proposed an approach for analyzing and representing the dynamics of facial expression. Their system consists of locating of tracking prominent facial features, optical flow analysis, and the classification. It achieves a recognition rate above 80% for all six expressions. Rosenblum *et al.* [4] extended the work of [3] by using a connectionist architecture. Individual emotion networks were trained by viewing a set of sequences of one emotion for many objects. The trained neural network was then tested for emotion recognition. However, their works are limited by the motion in six predefined and hand-initialized rectangular regions on a face that is not fully automatic. Essa and Penland [5] provided a facial expression representation by characterizing facial muscle activation. The facial motion estimation is operated by fitting the 3D deformable facial model to the face in an image for the muscle-based representation. Then, the facial motion is used to recognize the facial expression in two ways: using the physics-based model directly and generating the spatio-temporal motion-energy templates for recognition.

Black and Yacoob [6] used local parameterized models of image motion for recovering the nonrigid motions of the human face which provides a concise description of facial motions in terms of a small number of parameters. However, they did not address the problem of locating the various facial features (they are manually selected as mentioned in [7]). The effectiveness of the extracted features will affect the accuracy of the parameters estimation. Vanger *et al.* [8] developed a facial expression recognition method by using a synergetic pattern recognition approach.

* To whom all the correspondence should be addressed. Fax: 886-3-5715971. E-mail: clhuang@ee.nthu.edu.tw.

They created a prototype index for expressions corresponding to each motion by averaging the image of all eye and mouth parts for each motion. Then, they checked each part of the facial image against the prototype and match to the most similar prototype to identify the emotion represented by that particular facial image. The recognition rate of their method is claimed to be almost 70%.

Moses *et al.* [9] proposed a facial expression recognition method to identify the shape of the mouth feature only. The mouth was described by a valley contour shown to exist independently of illumination and viewpoints. They have shown their method to recognize five simple expressions with the correct recognition ratio above 80%. Kitamura *et al.* [10] used simple measurements (0 or 1) of the forehead wrinkle, eye opening, nostril furrow deepening, mouth opening, and eyebrow motion to recognize human facial expression. Matsuno *et al.* [11] used a potential net on a facial area in an image. The net is deformed by image forces because each node is moved to the position of facial features. They measured the movement of each node for facial expression recognition.

Originated from the active contour models (or snakes) [12], Cootes *et al.* [13–14] proposed a point distribution model (PDM) which differs in that the global shape constraints are applied. An instance of a model can only deform in ways found in its training set. The average example is calculated and the deviation of each example from the mean is established. A principal component analysis of the covariance matrix of deviations reveals the main mode of variation. Usually only a small number of model parameters is required to reconstruct the training examples. Lanitis *et al.* [16] applied the PDM-based method to track human face. Baumberg *et al.* [19] used the PDM and Kalman filter to track the silhouette of a walking pedestrian in real time. Heap *et al.* [20] extended the PDM by proposing a Cartesian–polar hybrid PDM which allows the angular movement to be modeled directly. Recently, Cootes and Taylor [18] proposed statistical feature detector to locate faces.

In this paper, we assume that the head motion is very trivial, so we do not have to consider the face locating problem. To extract the facial features (except the mouth feature) effectively, we have proposed a modified gradient-descent-based shape parameter estimation method which is simpler than the previous methods. For mouth feature (or lips) extraction, a method proposed by Bregler *et al.* [21] applies the active contour model (or snake) to extract the lip contour for visual speech recognition. Instead, we use a *deformable mouth model* to extract the mouth which is more efficient than the snake.

This paper proposes a different approach which consists of three main processes: (1) a modified PDM-based facial feature extraction (i.e., eyes, eyebrow, mouth, and jaw); (2) an action parameter analysis which characterizes the motion of each facial feature (i.e., eye, eyebrow, and mouth motion); (3) a two-stage facial expressions recognition (i.e., happiness, sadness, anger, surprise, smile, and fear). The feature points in the first frame (expression starts) and the last frame (expression ends) are analyzed to characterize the motion of these features in the image sequence capturing a human facial expression. We try 15 training image sequence sets (every set contains six different emotions) and use a 2D emotion space to model all characteristics of the six expressions.

In the recognition process, we find action parameters (APs) which are used to identify some facial expressions. APs can be obtained directly by analyzing the variations of the feature contours. Then, we apply the principal component analysis to reduce the statistical dimension of APs. Since the first two components represent over 90% variation of APs, we use a 2D emotional space model to classify the six expressions into six clusters. A simple *Gaussian model* is taken to model the six emotional clusters as well as to classify the input unknown emotions. This paper proposes a two-stage recognition process which includes a distance-based classification in 2D emotion space and a facial expression identification using the AP profile correlation. The facial expression recognition system is illustrated in Fig. 1.

2. FACIAL FEATURE EXTRACTION

The PDM may be used to generate new examples of the shape, which will be similar to those in the training set, by varying the parameters within certain limits. The mean shape model is placed in the image and is allowed to interact dynamically until it fits to the location of a newly suggested position for each model point based on the matching of the local intensity model. Different from [13, 14] which deforms each model point individually, we propose another approach: (1) moving and deforming the entire PDM shape model simultaneously by changing the shape parameters and (2) measuring the model-image fitness by using the overall gray-level fitness measure. Here, we apply the gradient-descent-based shape parameters estimation which minimizes the overall gray-level model fitness measure. By varying the shape parameters which are consistent with the training set, we can find the best shape model fitted with the real face in the image. However, in [13, 14], each model point moves independently and the movements are not consistent with the PDM shape model; therefore, they need to adjust the model points by estimating the PDM shape parameters and then readjusting the movements which are computation intensive operations.

2.1. *Point Distribution Model*

To deal with various facial expressions on different persons, we need to build a model which describes both shape

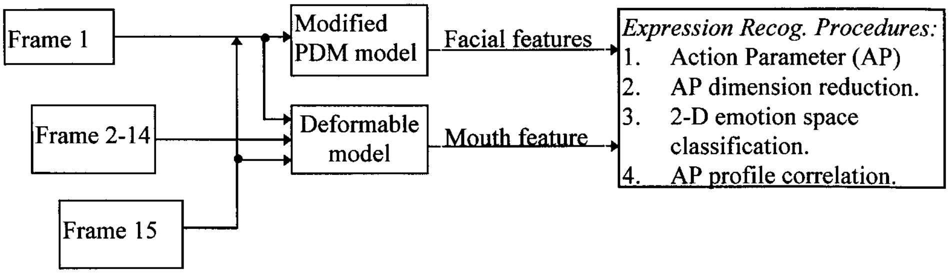
Frame 1
Frame 2-14
Frame 15
Modified PDM model
Deformable model
Facial features
Mouth feature
Expression Recog. Procedures:
1. Action Parameter (AP)
2. AP dimension reduction.
3. 2-D emotion space classification.
4. AP profile correlation.

FIG. 1. The flow diagram of the expression identification system.

and variability. We manually locate the feature points on the training set images by following some rules to ensure that each point plays an essential role on the boundary of the images. This will ensure the coherence of points on the different features. We call these points "landmark points" [1]. If the choice of landmark points is improper, the method may fail to capture shape variability reliably. We select the landmark points (see Fig. 2) based on (1) the points mark some parts of the object with particular application-dependent significance, such as the center of an eye on the face model or sharp corners of a boundary, and (2) the points can be interpolated from the preselected points, for instance, the landmark on the boundary at equal distances to the other two neighboring landmarks.

(1) *Aligning the training set.* The PDM-based method analyzes the statistics of the coordinates of the labeled points over the training set. To have a concise shape model, we must label (using landmark points) different features on the images in the training set. These landmark points on different images have minimal difference, so that we can align them with different scale, rotation, and translation before training. By minimizing a weighted sum of squares of distances between corresponding points on different shapes, we align every shape to the first shape; calculate the mean shape of the N shapes; and then align every shape to the mean shape. The aligned shapes of the training set are illustrated in Fig. 3. The details of the alignment processing of the training set can be found in [13, 14].

(2) *Statistical analysis of the aligned shapes.* Having generated the N aligned shapes and the mean shape $\bar{\mathbf{x}}$, we may calculate the deviation of the aligned shapes from the mean shape, $d\mathbf{x}_i$ as

$$d\mathbf{x}_i = \mathbf{x}_i - \bar{\mathbf{x}}. \tag{1}$$

Then, we can obtain the $2n \times 2n$ covariance matrix, $\mathbf{S}$, as

$$\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N} d\mathbf{x}_i\, d\mathbf{x}_i^{\mathrm{T}}. \tag{2}$$

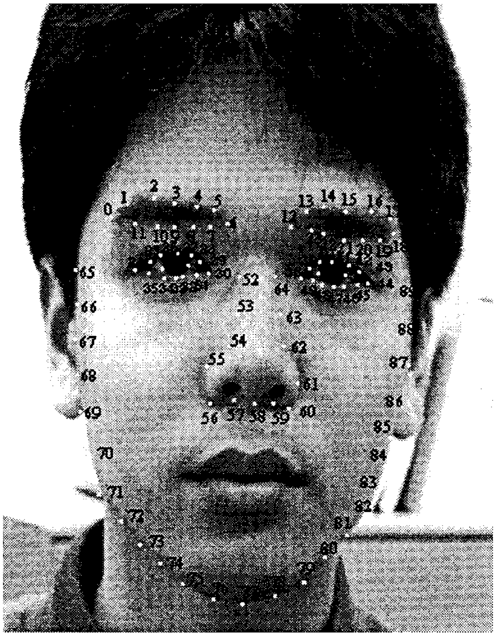

FIG. 2. We select 90 feature points on each image for the training model.

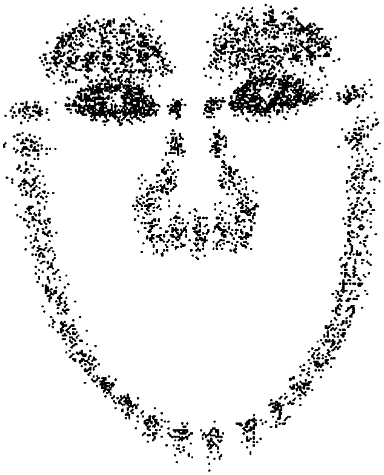

FIG. 3. The clouds of the aligned training set.

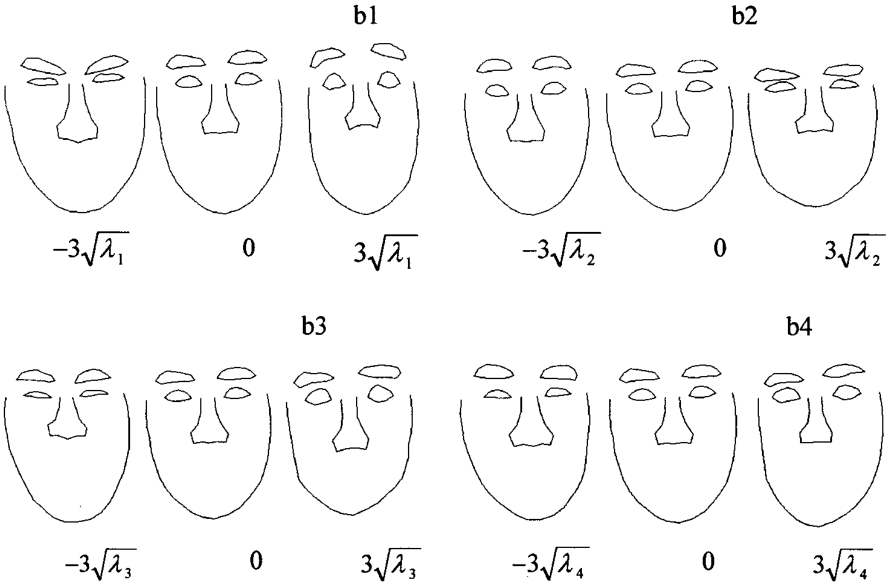

FIG. 4. The shape variation by changing shape parameters: b1, b2, b3, and b4.

Applying the principal component analysis, we can project the original $2n$ dimension shape points vector to another axis to reduce the dimension. We first calculate the eigenvectors of the covariance matrix **S** (i.e., $p_1, \ldots, p_{2n}$) such that

$$\mathbf{S}\mathbf{p}_k = \lambda_k \mathbf{p}_k \quad \text{with } \mathbf{p}_k^{\mathrm{T}}\mathbf{p}_k = 1, \tag{3}$$

where λ_k is the kth eigenvalue of **S**, with $\lambda_k \geq \lambda_{k+1}$. According to the principal component analysis, it is sufficient to use the first t eigenvectors to describe the shape variation. Another advantage of this method is that the models represent the global variation rather than the local variation of the shape.

Now we determine how many terms is enough for us to describe the shape variation. If we define λ_{T} as

$$\lambda_{\mathrm{T}} = \sum_{k=1}^{2n} \lambda_k \text{ and } \lambda_t = \sum_{k=1}^{t} \lambda_k, \tag{4}$$

then, based on the experimental results, $\lambda_t/\lambda_{\mathrm{T}} = 0.8$ is sufficient. We use 90 landmark points (n = 90) and 10 eigenvectors (t = 10) which suffice the constraint. Given an arbitrary shape, we can use $\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P} \cdot \mathbf{b}$ to approximate it, where $\mathbf{P} = (\mathbf{p}_1, \ldots \mathbf{p}_t)$ is the matrix of the first t eigenvectors, and $\mathbf{b} = (b_1, \ldots, b_t)^{\mathrm{T}}$ is a vector of weights which are determined by the eigenvalues $(\lambda_1, \ldots, \lambda_t)$. The shape variations can be described by the first four principal components illustrated in Fig. 4.

2.2. *The Gray-Level Model*

Since the facial contours do not indicate the existence of strong edges, whereas, some face feature points are so close to one another that the edge information on one point may interfere with the edge of the other point. To resolve these drawbacks, Cootes *et al.* [18] introduced the gray-level model. Since every point on the face is on a particular position, its gray-level appearance for every face in the training set will be similar. There are several ways to describe the gray-level appearance. We may use a rectangular window with the centroid located on the feature point and find the one-dimensional profile which is normal to the curve passing through the feature point to record the gray-level appearance. To reduce the error caused by the background luminance variation, we sample the difference of the gray-level along the profile and then normalize it.

For every feature point in the training set, we can extract a profile, $\mathbf{g}_j$ ($j = 1, \ldots, n$), of length $n_p + 1$ pixels, centered at the point j. If the profile's samples starts at $\mathbf{x}_{\text{start}}$ and ends at $\mathbf{x}_{\text{end}}$ with length $n_p + 1$ pixels (see Fig. 5), the intensity of the kth element of the profile is

$$g_{jk} = I_j(\mathbf{y}_k), \tag{5}$$

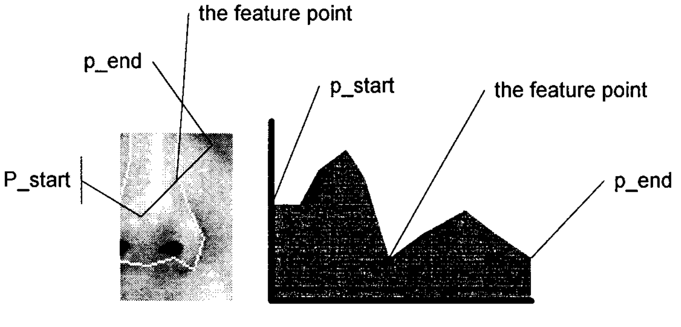

FIG. 5. The gray-level profile extraction on the landmark point.

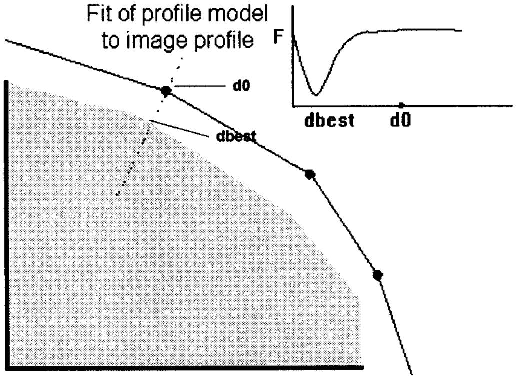

FIG. 6. Suggested movement of points is along the normal direction to boundary toward the best-fit position.

where $\mathbf{y}_k$ is the location of the point along the profile

$$\mathbf{y}_k = \mathbf{x}_{\text{start}} + \frac{k-1}{n_p}(\mathbf{x}_{\text{end}} - \mathbf{x}_{\text{start}}) \tag{6}$$

and $I_j(\mathbf{y}_k)$ is the gray-level at the position $\mathbf{y}_k$. Then, we calculate the normalized difference of g_j by using

$$\mathbf{g}'_j = \frac{\mathbf{g}''_j}{\sum_{k=1}^{n_p} |g''_{jk}|}, \tag{7}$$

where $\mathbf{g}''_j = [g''_{j1}, g''_{j2}, \ldots g''_{j(n_p+1)}]$, $g''_{jk} = g_{jk} - g_{j(k-1)}$, $k = 1 \ldots n_p + 1$, and g_{jk} is the kth pixel for the jth feature point's gray-level profile on the current frame. For convenience, we will simply substitute $\mathbf{g}_j$ for $\mathbf{g}'_j$. Here, we use principal component analysis to describe the statistics property of the gray-level. For each feature point, we calculate a mean profile $\bar{\mathbf{g}}$, then get a $n_p \times n_p$ covariance matrix $\mathbf{S}_g$, an eigen-matrix $\mathbf{P}_g$ and a set of eigenvalue λ_k ($k = 1, \ldots, n_p$). For an arbitrary sampled profile $\mathbf{g}$, we apply the following function to evaluate how well it can be fitted to a particular landmark point j (with position $\mathbf{x}_j$) as

$$\mathrm{F}(\mathbf{x}_j) = \sum_{j=1}^{n_p} \frac{b_{gj}^2}{\lambda_j}, \tag{8}$$

where $\mathbf{b}_g = \mathbf{P}_g^{\mathrm{T}}(\mathbf{g} - \bar{\mathbf{g}})$ and $\mathbf{b}_g = (\mathrm{b}_{g1}, \mathrm{b}_{g2}, \ldots, \mathrm{b}_{gn_p})$. In the fitting process (see Fig. 6), we measure the F value to determine the displacement of a particular point from the initial position to the best fit position. Along the normal direction of each model point, we find the smallest F value that indicates the best match between the gray-level profile of the current position of the test model point and the mean profile of the corresponding feature point. Suppose the displacement is d_{best}, then the adjusted displacement $|d\mathbf{X}| = 0.5d_{\text{best}}$ if $d_{\text{best}} < d_{\max}$ otherwise $|d\mathbf{X}| = 0.5d_{\max}$. We set the $d_{\max}$ value adaptively to reduce the calculation time, it decreases as the number of iterations increases.

Here, we assume (1) the background does not change much during the gray-level model generation phase and the facial expression recognition phase and (2) the illumination variation is linear. If facial expressions are to be recognized before a totally different background, then we may neglect the influence of the background on the gray-level generation. Here, we apply the differentiation and normalization on gray-level profile (Eq. (7)) to reduce the error caused by the illumination changes. However, the gray-level model may not fit well for some features, such as the mouth and eyes. We need to apply the parabolic deformable model (see Section 3) to fit the mouth and eyes. Since we are interested in finding the variation vectors of these features, the exact location fitting of these features is not necessary.

2.3. Shape Model and Feature Points Interaction

This section describes how to use the PDM and the grey-level model to extract the facial features. Suppose the current shape position is $\mathbf{X}$ (with centroid $\mathbf{X}_c$) and we need to adjust the global shape variation (including the translation $d\mathbf{X}_c = (dX_c, dY_c)$, rotation $d\theta$, the scale ds) and the local shape variation $d\mathbf{b}$ to find the next fitting position $\mathbf{X} + d\mathbf{X}$,

$$\mathbf{X} + d\mathbf{X} = (\mathbf{X}_c + d\mathbf{X}_c) + \mathbf{M}((s + ds), (\theta + d\theta)) \cdot [\bar{\mathbf{x}} + \mathbf{P} \cdot (\mathbf{b} + d\mathbf{b})], \tag{9}$$

where $\mathbf{M}(s, \theta)$ is a 2×2 rotation matrix. By finding gray-level profiles of every point j on $\mathbf{X} + d\mathbf{X}$ ($\mathbf{x}_j \in \mathbf{X} + d\mathbf{X}$) as $\mathbf{g}_j$, we calculate the gray-level profile fitness value $F(\mathbf{x}_j)$ and find the overall F values (i.e., $\sum_j F(\mathbf{x}_j)$ for $\mathbf{x}_j \in \mathbf{X} + d\mathbf{X}$) of all landmark points. If the $\sum_j F(\mathbf{x}_j)$ is minimized then the position $\mathbf{X} + d\mathbf{X}$ indicates the best-fit shape. In the following, we illustrate a modified PDM-based fitting process.

(1) *Initial facial model position estimation.* In the facial feature extraction process, we may encounter the problem that if the positions of some fitting points are too far away from the actual positions, then the adjustment may require a lot of iterations to pull the landmark points to the proper place. To avoid this kind of problem, we introduce a Canny edge detector to find the horizontal valley contour that lies between two lips and the two symmetric vertical edges which are supposed to be the two vertical outer boundary of the face. From these extracted edges, we can roughly estimate the position of the face to place the initial PDM shape model.

(2) *Shape Adjustment Process.* Here, we apply the two-

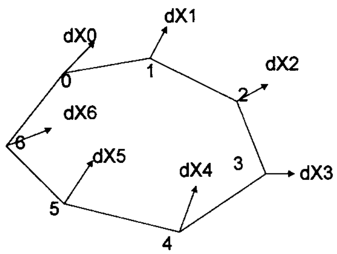

FIG. 7. The best $d\mathbf{X}$ of every landmark point found by the gray-level model.

step estimations for the global shape variation parameters (i.e., the translation $d\mathbf{X}_{\mathrm{c}}$, the rotation $d\theta$, the scale ds) and the local shape variation parameter (i.e., $d\mathbf{b}$). First, we assume that the current global shape is $\mathbf{X}$, then we can do the global shape variation for the new global shape as $\mathbf{X} + d\mathbf{X} = \mathbf{M}(s + ds, \theta + d\theta)\cdot[\mathbf{x}] + (\mathbf{X}_{\mathrm{c}} + d\mathbf{X}_{\mathrm{c}})$, where $\mathbf{M}$ is a 2×2 rotation matrix, $\mathbf{x}$ represents the aligned shape, and $\mathbf{X}_{\mathrm{c}}$ represents the central point of current shape. Second, we may also deform the current local shape $\mathbf{x}$, by changing local shape parameter $d\mathbf{b}$ to generate the new local shape as $\mathbf{x} + d\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}(\mathbf{b} + d\mathbf{b})$.

(3) *Gradient-descent-based shape parameters estimation.* To find the best fitted shape, we propose a gradient-descent-based shape parameters estimation method. The global and local shape parameters estimation for the ith iteration is illustrated in the following steps:

(1) Find the next shape $\mathbf{X} + d\mathbf{X}$ by using the new global shape parameters $((\mathbf{X}_{\mathrm{c}} + d\mathbf{X}_{\mathrm{c}}), s + ds, \theta + d\theta)$.

(2) Find the gray-level profile $(\mathbf{g}_j)$ of each landmark point j on $\mathbf{X} + d\mathbf{X}$ $(\mathbf{x}_j \in \mathbf{X} + d\mathbf{X})$ and calculate the corresponding fitness value $F(\mathbf{x}_j)$.

(3) Add the F values for all landmark points on $\mathbf{X} + d\mathbf{X}$ to see if $\sum_j F(\mathbf{x}_j)$ exceeds the preselected threshold F_m. If $\sum_j F(\mathbf{x}_j) > F_m$ then it indicates that the shape model does not fit to the real face on the image at all. Choose another initial value of $\mathbf{X}_{\mathrm{c}}$ by adding a larger variation $d\mathbf{X}_{\mathrm{c}}$. Determine the $d\mathbf{X}_{\mathrm{c}}$ by selecting the median one of all the best $d\mathbf{X}$ of the landmark points (see Fig. 7). If $\sum_j F(\mathbf{x}_j) > F_m$ go to step 1; otherwise continue (it indicates a rough shape fitness).

(4) Determine the decrement or increment of the global shape parameters (i.e., $\pm ds$ and $\pm d\theta$) by examining $\sum_j F(\mathbf{x}_j)$ (i.e., $\{[\sum_j F(\mathbf{x}_j)]_i - [\sum_j F(\mathbf{x}_j)]_{i+1}\} > 0$ or <0).

(5) If $\sum_j F(\mathbf{x}_j)$ does not decrease (i.e., $\{[\sum_j F(\mathbf{x}_j)]_i - [\sum_j F(\mathbf{x}_j)]_{i+1}\} > 0$) for all small variations ds and $d\theta$ then continue else go to step 4.

(6) Examine the final $\sum_j F(\mathbf{x}_j)$. If $\sum_j F(\mathbf{x}_j) > F_n$ (another preselect threshold); then go back to step 3 (to avoid being trapped in the local minimum); otherwise continue.

(7) Change the local shape parameters $d\mathbf{b}$ for the new local shape $\mathbf{x} + d\mathbf{x}$ and then find the minimum $\sum_j F(\mathbf{x}_j)$, which indicates the best fitness of the PDM shape model. The decrement or increment the local shape parameters $d\mathbf{b}$ is determined by the value of overall gray-level profile fitness (i.e., $\{[\sum_j F(\mathbf{x}_j)]_i - [\sum_j F(\mathbf{x}_j)]_{i+1}\} > 0$ or <0).

(8) Stop if $\{[\sum_j F(\mathbf{x}_j)]_i - [\sum_j F(\mathbf{x}_j)]_{i+1}\} > 0$ for all variations of $d\mathbf{b}$, otherwise go to step 7.

3. OTHER FACIAL FEATURE FITTING

The gray-level model will fail in the points of which the gray-level appearance has a large variety that cannot be trained using a gray-level model. In our experiment, we find out that this model fails on the cheek and mouth portions. For the cheek part (landmark point numbers 70, 71, 72, 82, 83, and 84), the background behind the head will affect the gray-level appearance. So in our system, we simply use edge detection on these points. Another problem occurs when this model is applied for locating the eyes. The system needs to extract the accurate positions of the feature points of which their motion parameters will be used for facial expression recognition. However, the gray-level model may not fit the eyes accurately. In our experiments, the points on the leftmost and rightmost sides of the iris (landmark point numbers 36, 37, 50, and 51) are well identified, so we use these points to aid the eye fitting.

(1) *Eye fitting.* The iris is always the darkest portion of eyes, so we define a block of which the leftmost position is the position of landmark point number 36 (for the left eye) and the rightmost position is the position of landmark point number 37 (for the right eye). This block will contain all of the iris and some regions above and below the eye. Using the mean gray-level value of this block as the threshold, we set the points (whose gray-levels are above the threshold) to value 1 and the others to value 0. Then the region growing method is used to remove some points which are not in the iris but the gray-level values are below the threshold. After applying the region growing method, we can get the precise locations of the upper and lower points of eyes.

(2) *Mouth fitting.* The problem of mouth extraction is more difficult. Points in the mouth region do not indicate strong edges nor gray-level invariant. The interior portion of lips has at least three kinds of gray-level appearances caused by different mouth actions: one for the mouth closed, another one for grinning, and the other one for the mouth opened without showing the teeth. We need to propose another method to extract the mouth region.

When the mouth is closed, we find that the gray-level value of the point located between the lips is always darker than the surrounding region. This is a useful property for us to track the mouth region. To fit the mouth region, we must roughly know the position of the mouth. Fortunately, we have already fitted the other features of the face, and we can estimate the initial position of the mouth for the following mouth fitting process. We define a block as the search region which is below landmark point number 58, above landmark point number 77, to the right of landmark point number 70, and to the left of landmark point number 84. The intensity of the pixels located between two lips is often the darkest part in this block, so we search for the darkest point in every vertical strip. These darkest points are not necessarily located on the position between the two lips, so that we can apply a simple gray-level thresholding technique to eliminate these misleading points and use a parabolic curve to approximate the remain points.

For n landmark points located at (x_j, y_j) with $j = 1, \dots, n$, we can approximate the curve as

$$\mu(x) = a + bx + cx^2. \tag{10}$$

The errors between the darkest points and the approximated curve are

$$E = \sum_{j=1}^{n} (\mu(x_j) - y_j)^2. \tag{11}$$

To minimize the errors, the following equations must be satisfied: $\partial E/\partial a = 0$, $\partial E/\partial b = 0$, and $\partial E/\partial c = 0$. After a simple deduction, we can get a, b, and c. Let A, B, C, D, E, F, and G be defined as

$$A = \sum_{j=1}^{n} x_j, \quad B = \sum_{j=1}^{n} x_j^2, \quad C = \sum_{j=1}^{n} x_j^3, \quad D = \sum_{j=1}^{n} x_j^4,$$
$$E = \sum_{j=1}^{n} y_j, \quad F = \sum_{j=1}^{n} x_j y_j, G = \sum_{j=1}^{n} x_j^2 y_j. \tag{12}$$

we can have a, b, and c by using the following equations:

$$a = \frac{\begin{vmatrix} E & A & B \\ F & B & C \\ G & C & D \end{vmatrix}}{\begin{vmatrix} n & A & B \\ A & B & C \\ B & C & D \end{vmatrix}} b = \frac{\begin{vmatrix} n & E & B \\ A & F & C \\ B & G & D \end{vmatrix}}{\begin{vmatrix} n & A & B \\ A & B & C \\ B & C & D \end{vmatrix}} c = \frac{\begin{vmatrix} n & A & E \\ A & B & F \\ B & C & G \end{vmatrix}}{\begin{vmatrix} n & A & B \\ A & B & C \\ B & C & D \end{vmatrix}}. \tag{13}$$

To fit the upper lip, we find the edges with the strongest gradient located a few pixels above the parabolic curve and then approximate it with another parabolic curve. The same method is applied to find the lower lip. The appearance of teeth may confuse the mouth contour extraction process, so, in our system, the initial picture frame starts with the mouth-closed image. The fitting of mouth in the next picture starts around the regions near the first fitted mouth. The results of the mouth fitting process are illustrated in Fig. 8.

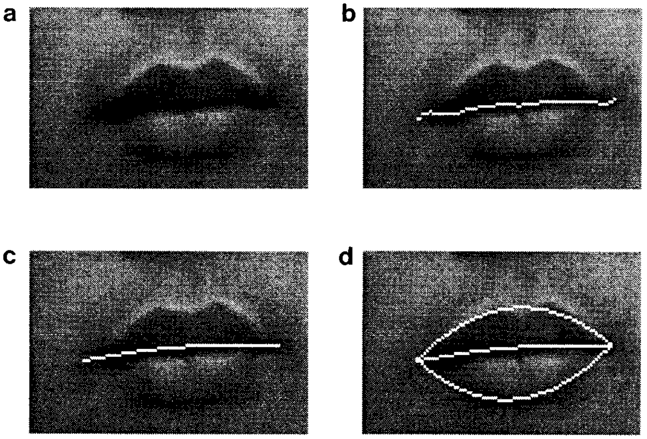

FIG. 8. The mouth fitting process.

Finally, we summarize the facial feature extraction process. Here, we only select the extracted facial features of the first and last pictures of the image sequence for the recognition process. However, for the mouth region, all the pictures in the image sequence have to be analyzed because the mouth contour may change a lot between the first and the last pictures. To extract the feature of the first picture, we must assign an initial position to the mean shape model on the first picture. Then we apply the PDM and the gray-level model to find the best-fit positions of all feature points. After the facial feature extraction, we can estimate the approximate region of the mouth. We then extract the mouth contour by applying the mouth model in this region. The mouth contour of the next 14 pictures is extracted around the region of the mouth in the previous picture. The other feature points in the last picture can be extracted by the same method applied to the first picture.

4. FACIAL EXPRESSION RECOGNITION

To describe human facial expressions, Ekman *et al.* [1] has introduced the FACS which divides the movement of facial muscle action into 44 standard action units (AUs). Our system can accurately track the contour variation of the facial features (i.e., eyes, mouth, and eyebrows), however, it cannot identify all the AUs from the extracted information. Therefore, we try to replace some AUs by other action parameters (APs) to identify the facial expression effectively. Then we apply the principal component analysis on action parameters (APs) to reduce the dimensions of these parameters to two dimensions and simplify

TABLE 1
Action Unit Selection [8]

AU-No	FACS name	AU-No	FACS name
AU-1	Inner brow raiser	AU-14	Dimpler
AU-2	Outer brow raiser	AU-15	Lip corner depressor
AU-4	Blow lower	AU-16	Lower lip depressor
AU-5	Upper lid raiser	AU-17	Chin raiser
AU-6	Cheek raiser	AU-20	Lip stretcher
AU-7	Lid tighter	AU-23	Lip tighter
AU-9	Nose wrinkler	AU-25	Lip part
AU-10	Upper lid raiser	AU-26	Jaw drops
AU-12	Lip corner puller		

the recognition process. The reduced 2D parameters of every expression form clusters. There are six clusters of parameters, and each one implies one expression.

4.1. The AP Acquisition Process

After the face tracking process, we have 193 action parameters. There are 180 position parameters (90 feature points) that have been obtained by active shape extraction and 13 mouth shape parameters (9 for three parabolic curves and 4 for the begin and the end point position). Using all the action parameters for face expression recognition is inefficient and may not generate satisfactory results. Therefore, we need to determine how to select meaningful action parameters.

Morishima [7] listed the AUs that will affect the facial expression in Table 1. However, in Table 2, we remove the disgust, because from the facial expressions of the oriental people, it is not easy to differentiate the disgust expression from the anger expression. In our experiments, we also find that it is very difficult for our volunteers to express the disgust emotion on their faces. Instead, we add the smile expression because the smile is easily differentiated from the happy which is often expressed by the action of laughing.

TABLE 2
AUs for Basic Emotions [8]

Basic emotion	Combination of AU parameters
Happiness	AU-1, AU-6, AU-12, AU-14
Smile	AU-1, AU-6, AU-12, AU-14
Surprise	AU-1, AU-2, AU-5, AU-15, AU-16, AU-20, AU-26
Sadness	AU-1, AU-4, AU-15, AU-23
Anger	AU-2, AU-4, AU-7, AU-9, AU-10, AU-20, AU-26
Fear	AU-1, AU-2, AU-4, AU-5, AU-15, AU-20, AU-26

However, in this paper, we only extract the contour deformation of the facial features, so we cannot use all the AUs mentioned in the above table. The AUs that we are interested in are AU-1, AU-2, AU-4, AU-5, AU-7, AU-12, AU-15, AU-16, AU-25, and AU-26. Due to the limitation of the feature extraction of computer vision technique and the system simplicity, we combine some AUs and substitute some other AUs by the action parameters (APs). Totally, we define four of the 10 action parameters (APs) as:

1. AP-1. It is equivalent to AU-1.
2. AP-2. It is equivalent to AU-2.
3. AP-3. It is equivalent to AU-4.
4. AP-4. It is equivalent to AU-26.

Then, we may calculate the other six APs from the following six feature parameters (FPs) as:

1. FP-1(OPEN). It is the combination of AU5 and AU7 for detecting the motion of the eyelids.
2. FP-2(VER). It is the mouth height.
3. FP-3(HOR). It is the mouth width.
4. FP-4(CUR-1). It is the curvature of the curve between two lips.
5. FP-5(CUR-2). It is the curvature of the upper lip.
6. FP-6(CUR-3). It is the curvature of the lower lip.

All of these action parameters are generated by taking the difference between the feature parameters of the first face picture and the last picture. To reduce the effect of head motion, we align AU-1 to AU-26 with the positions of the pupils. Let y_i denote the y coordinate value of the ith landmark point and *Origin* indicate the y coordinate value of the pupils; then for the first picture of the image sequence of facial expression we have $Origin = (y_{36} + y_{37} + y_{50} + y_{51})/4$. Before calculating the values of APs we subtract *Origin* from the y coordinate of every point to align all points; i.e., $y_i' = y_i - Origin$. For convenience, we substitute y_i for y_i'. Let AP_i denote the variation of the FPs which are measured in the first and the last pictures, we can calculate all APs using the equations

$$\begin{aligned}\text{AP1: } ap_1 = {} & (y_4 + y_5 + y_6 + y_{12} + y_{13} + y_{14})_{\text{last}}/6 \\ & - (y_4 + y_5 + y_6 + y_{12} + y_{13} + y_{14})_{\text{first}}/6;\end{aligned}$$

$$\begin{aligned}\text{AP2: } ap_2 = {} & (y_0 + y_1 + y_2 + y_{15} + y_{16} + y_{17})_{\text{last}}/6 \\ & - (y_0 + y_1 + y_2 + y_{15} + y_{16} + y_{17})_{\text{first}}/6;\end{aligned}$$

$$\begin{aligned}\text{AP3: } ap_3 = {} & (y_7 + y_8 + y_9 + y_{10} + y_{11} + y_{19} + y_{20} + y_{21} \\ & + y_{22} + y_{23})_{\text{last}}/10 - (y_7 + y_8 + y_9 + y_{10} \\ & + y_{11} + y_{19} + y_{20} + y_{21} + y_{22} + y_{23})_{\text{first}}/10;\end{aligned}$$

$$\begin{aligned}\text{AP4: } ap_4 = {} & (y_{75} + y_{76} + y_{77} + y_{78} + y_{79})_{\text{last}}/5 \\ & - (y_{75} + y_{76} + y_{77} + y_{78} + y_{79})_{\text{first}}/5;\end{aligned}$$

$$\begin{aligned}\text{AP5: } ap_5 = {} & ((y_{25} + y_{26} + y_{27} + y_{28} + y_{29} + y_{39} + y_{40} + y_{41} \\ & + y_{42} + y_{43}) - (y_{31} + y_{32} + y_{33} + y_{34} + y_{39} + y_{45} \\ & + y_{46} + y_{47} + y_{48} + y_{49}))_{\text{last}}/10 + ((y_{25} + y_{26} \\ & + y_{27} + y_{28} + y_{29} + y_{39} + y_{40} + y_{41} + y_{42} + y_{43}) \\ & - (y_{31} + y_{32} + y_{33} + y_{34} + y_{39} + y_{45} + y_{46} \\ & + y_{47} + y_{48} + y_{49}))_{\text{first}}/10;\end{aligned}$$

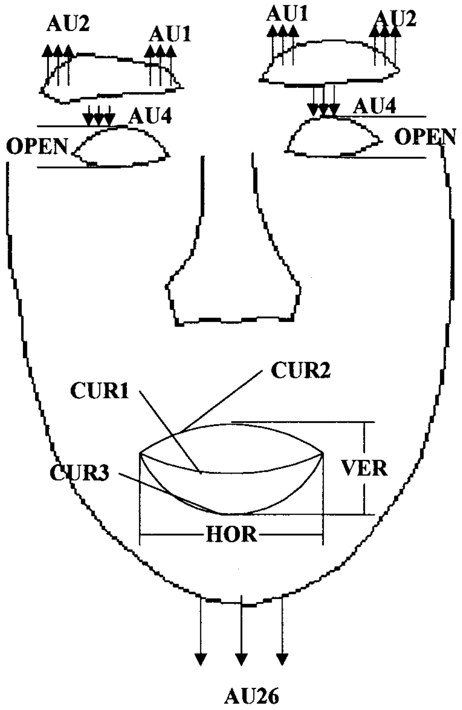

FIG. 9. The Action Parameters.

AP6: $ap_6 = \mathrm{VER}_{\mathrm{last}} - \mathrm{VER}_{\mathrm{first}}$;

AP7: $ap_7 = \mathrm{HOR}_{\mathrm{last}} - \mathrm{HOR}_{\mathrm{first}}$;

AP8: $ap_8 = \mathrm{CUR} - 1_{\mathrm{last}} - \mathrm{CUR} - 1_{\mathrm{first}}$;

AP9: $ap_9 = \mathrm{CUR} - 2_{\mathrm{last}} - \mathrm{CUR} - 2_{\mathrm{first}}$;

AP10: $ap_{10} = \mathrm{CUR} - 3_{\mathrm{last}} - \mathrm{CUR} - 3_{\mathrm{first}}$;

where, VER, HOR, CUR-1, CUR-2, and CUR-3 are illustrated in Fig. 9. We generate 10 action parameters (APs) for facial expression recognition.

4.2. *The AP Dimension Reduction*

Facial expression can be represented by the values of these APs which can be further classified to represent different facial expressions. It is not easy to give a proper thresholding for each AP nor appropriate to assign a proper weight for a specific AP during the facial expression recognition process. Therefore, we develop a method which may systematically generate the thresholds. This paper proposes a method which combines principal component analysis of APs and a 2D Gaussian model.

Let n be the number of the samples in the training set and ap_{ij} (with $i = 1, \ldots, N$; $j = 1, \ldots, n$) be the scalar value of the ith AP of the jth training data. Each AP has different variation range. For instance, ap_8 varies with the range in the order of 10^{-3} but ap_4 varies with the range in the magnitude of a few tens. If we train these APs with their original values, the influence of the parameters with small variations will be ignored. So we transform these parameters to another domain before training. Here, we use the weighting matrix to reduce the range of different variations. The diagonal weighting matrix $\mathbf{W}$ is defined as

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & 0 & 0 & 0 & 0 \\ 0 & . & 0 & 0 & 0 \\ 0 & 0 & . & 0 & 0 \\ 0 & 0 & 0 & . & 0 \\ 0 & 0 & 0 & 0 & w_{N,N} \end{bmatrix}, \tag{14}$$

where $w_{ii} = (\sigma_i)^{-1}$ and σ_i is the variance of the ap_i of the ith AP. Then we apply the principal component analysis for n training samples as

(1) The mean value of ap_i is $\overline{ap}_i = (1/n) \sum_{j=1}^{n} ap_{ij}$ for n samples, $i = 1, \ldots, N$, and $N = 10$. (15)

(2) The deviation for jth sample is $d\mathbf{ap}_j = \mathbf{ap}_j - \overline{\mathbf{ap}}$, where $\overline{\mathbf{ap}} = (\overline{ap}_1, \ldots, \overline{ap}_N)^{\mathrm{T}}$. (16)

(3) The $N \times N$ covariance matrix is $\mathbf{S}_{\mathrm{ap}} = (1/n) \sum_{j=1}^{n} d\mathbf{ap}_j \cdot d\mathbf{ap}_j^{\mathrm{T}}$. (17)

The eigenvalues and eigenvectors of S_{ap} are λ_k and $\mathbf{Ap}_k$, respectively, with $k = 1, \ldots, K$. In our experiment, we find that the first two terms ($K = 2$) of eigenvalues can represent over 90% variation of the parameters. Therefore, we choose the first two terms of the principal component as our principal action parameters. Then, we calculate the magnitude of the two principal components (i.e., $\mathbf{b} = (b_1, b_2)$) of any training sample (i.e., $\mathbf{ap}$) by using

$$\mathbf{b} = \mathbf{AP}^{\mathrm{T}}(\mathbf{W} \cdot \mathbf{ap} - \overline{\mathbf{ap}}), \tag{18}$$

where $\mathbf{AP} = (\mathbf{Ap}_1, \mathbf{Ap}_2)$ is a $N \times 2$ matrix. Figure 10 illustrates the two principal components' distribution of six different expressions.

4.3. *The Training Process*

Suppose we have a set of n training expressions, and n corresponding sets of 2D action parameters generated as $\{(b_1, b_2)_k \mid k = 1, \ldots, n\}$ in the 2D emotion space (see Fig. 10). Then we use the minimum distance classifier to cluster the parameters into six clusters representing six expressions. Each expression has $n/6$ sets of action parameters. Let the first parameter cluster (i.e., $\{(b_1, b_2)_{k1} \mid k1 = 1, \ldots, n/6)\}$) represent "happy" expression, the second parame-

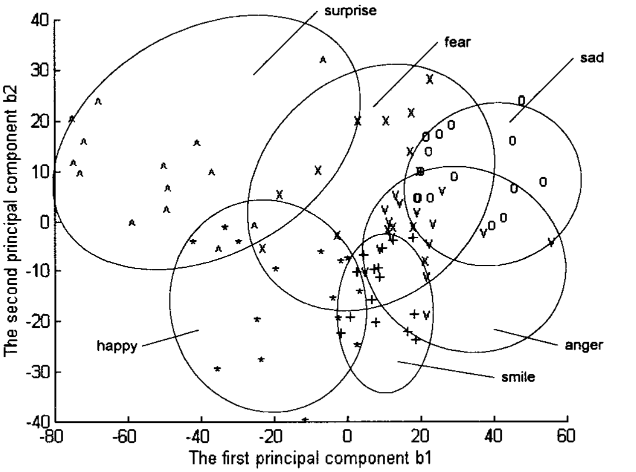

FIG. 10. The distribution of the first two components of the APs for six different facial expression from 15 different training faces: *, happy; +, smile; ^, surprise; o, sad; v, anger; x, fear.

ter cluster (i.e., $\{(b_1, b_2)_{k2} \mid k2 = n/6 + 1, \ldots, n/3)\}$) represent "smile" expression, and so on. Therefore, we can divide all action parameters into six groups of clusters representing six different facial expressions. Let $\mathbf{m}_{\text{happy}}$ be the mean and σ_{happy} be the variance of the action parameter of the "happy" expression which are defined as

$$\mathbf{m}_{\text{happy}} = \frac{1}{n/6}\sum_{i=1}^{n/6} \mathbf{b}_i, \quad \sigma_{\text{happy}} = \frac{1}{n/6}\sum_{i=1}^{n/6} ((\mathbf{b}_i - \mathbf{m}_{\text{happy}})(\mathbf{b}_i - \mathbf{m}_{\text{happy}})^{\mathrm{T}}), \tag{19}$$

where $\mathbf{b} = (b_1, b_2)$. We can also calculate the mean and variance of the other five facial expressions. Because the bases of two principal components of APs are orthogonal, we can manipulate them individually. The density functions of the two principal components (b_1 and b_2) of six different expressions are illustrated in Figs. 11 and 12.

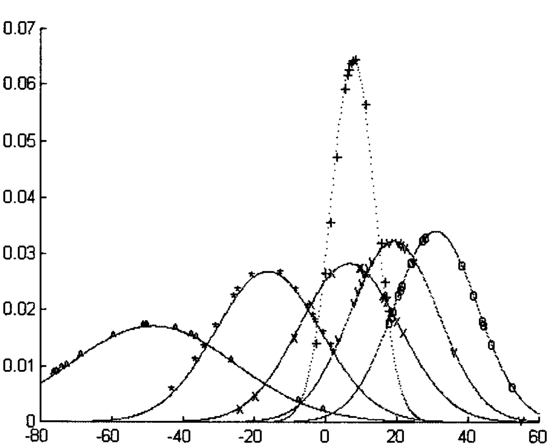

FIG. 11. The density function of the first principal component b1: *, happy; +, smile; ^, surprise; o, sad; v, anger; x, fear.

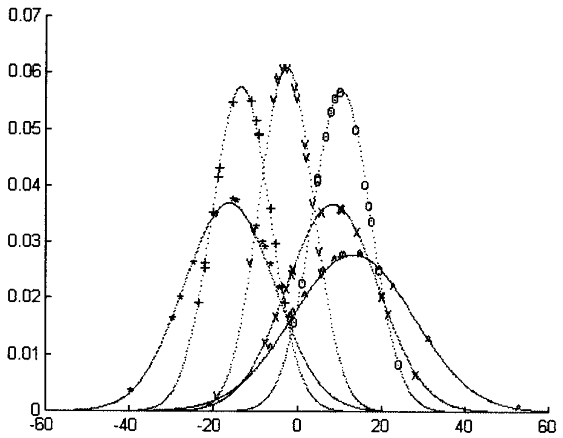

FIG. 12. The density function of the second principal component b2: *, happy; +, smile; ^, surprise; o, sad; v, anger; x, fear.

4.4. The Two-Stage Recognition Process

Here, we propose a two-stage recognition process which includes a distance-based classification in 2D emotion space and a facial expression identification using the AP profile correlation. The first stage process recognizes the input expression by using the AP clusters of the training set in emotion space. However, there is a significant overlap in the emotion space for different clusters (expressions). The second stage process uses the AP profile of the input unknown expression (i.e., $(ap_1, ap_2, \ldots, ap_{10})$) for further identification.

Given an unknown expression, first, we use the evaluation function to measure the similarity between the unknown expression and expression i with the distance E_i defined as

$$E_i = \sum_{j=1}^{2} \frac{(b_j - m_{\text{expression-}i})^2}{s_{\text{expression-}i}}, \tag{20}$$

where expression-i implies one of the six expressions. We may use this distance evaluation function to select two of the most probable facial expressions to which the testing expression may be matched.

Using statistical analysis of the expression to recognize the expression is not reliable because of some confused expressions. For some expressions, of which the 2D emotion parameters located between the transition regions in the 2D emotion space (see Fig. 10), our system may misidentify the expressions. In the first stage, we find the three best-matched expressions which have minimum evaluation distance (Eq. (20)). We select three best matches because Fig. 10 illustrates that the principal component distribution of each expression is overlapped with at least the other two expressions.

In the second stage, we take advantages of the AP profile

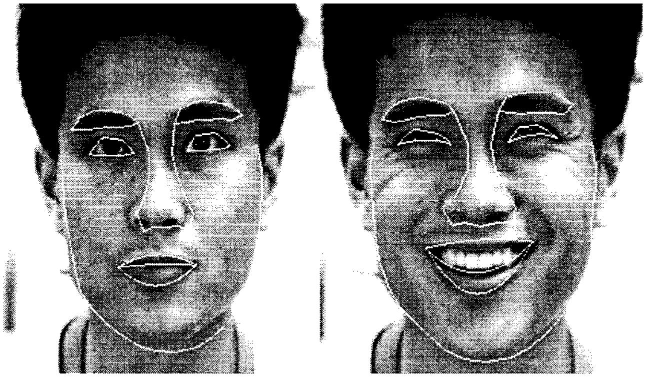

FIG. 13. Face expression 1: happy.

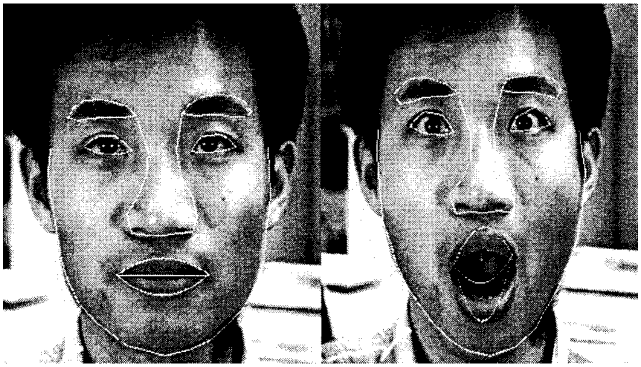

FIG. 15. Face expression 3: surprise.

of the input unknown facial expressions. The AP profiles of different expressions are determined during the training stage. Let the mean AP profile of known expression i is pretrained as $\mathbf{a}\overline{\mathbf{p}}_i = (a\overline{p}_1, \ldots, a\overline{p}_{10})_i$, we may correlate the AP profile of the unknown expression with the prestored mean AP profiles of the three best-matched expressions, respectively. From the highest score of the three correlation we may tell the exact emotion of the input facial expression. From the prestored AP profile, we may also select several dominant APs for each expression presented in Table 5. This table includes the expression behavior and the increment or decrement of the specific AP measurement, it confirms our common sense of emotion identification.

5. EXPERIMENTAL RESULTS AND DISCUSSION

In our experiment, we take the image sequences of different facial expressions from 15 volunteers, each one demonstrates six expressions. Each expression is made twice by each volunteer. We take 12 image sequences for every volunteer, and overall, we take 180 image sequences for 15 volunteers. There are 15 pictures in an image sequence, and the size of the picture is 512 × 512. The camera that we use in our experiment is a SONY XC7500. For each facial expression, an image sequence with 30 frames is taken at 30 frames/s and stored in DRAM on an Oculus-F/64 frame grabber which is transferred to the host computer (PC with Pentium CPU) for further processing.

Here, we do mouth extraction process for every frame in the image sequence (see Fig. 1) because this process is fast and reliable. The location of the mouth can also be roughly estimated by the preprocessing stage of the PDM-based shape fitting (see Section 2.3) especially when the head rigid motion occurs (see Table 5.) Since the PDM-based processing is slow, it is applied to analyze only the first and last frame of the image sequence. The first frame is selected when the people start making expressions which can be easily detected by the frame difference operation, and the last frame is determined when there is no motion detected by the frame difference operation.

In the experiment, we assume that the head motion is negligible. If there is a noticeable head motion, our method can still identify the facial features. However, before finding the APs, we need to register the extracted facial fea-

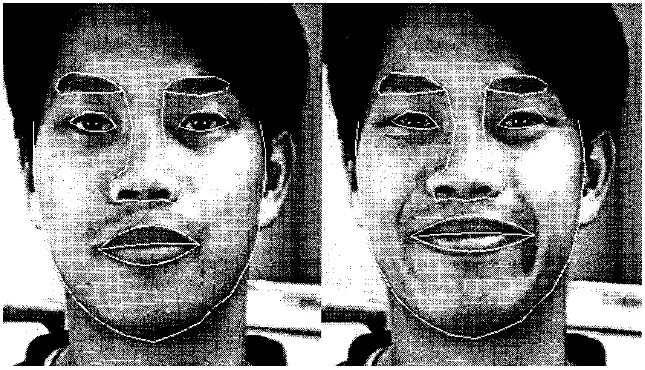

FIG. 14. Face expression 2: smile.

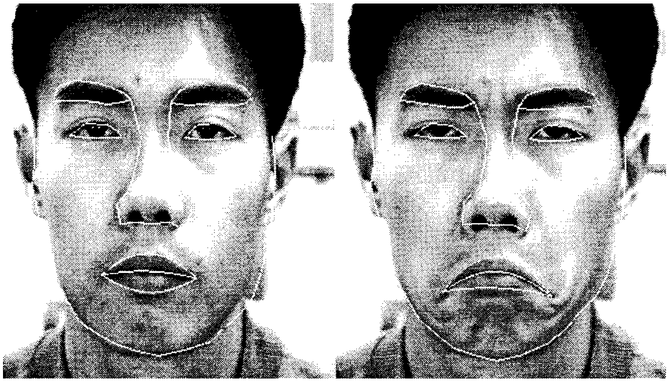

FIG. 16. Face expression 4: sad.

TABLE 5
The Dominant Action Parameter for Verifying the Facial Expression

Facial expression	Behavior	Dominant APs
Happy	Upward curving of mouth and expansion on vertical and horizontal direction	ap6 > 0, ap7 > 0
Smile	Upward curving of mouth	ap8 > 0, ap7 > 0
Surprise	Raising brows and vertical expansion of mouth	ap4 > 0, ap6 > 0, ap5 > 0
Sadness	Downward curving of mouth	ap8 < 0
Anger	Inward lowering of eyebrows	ap1 < 0, ap4 < 0
Fear	Expansion of mouth and raising eyebrows	ap6 > 0, ap4 > 0

tures on the first and last frames to compensate the motion vector due to the head motion. We may use the landmark points near the two ears (landmark point numbers 65–69 and 86–90) to register the extracted facial contours of the first and last frames. To extract the facial motion information, our method is more suitable than the optical flow methods [3, 4, 10] which are sensitive to the head motion and illumination changes. Our 2D flexible model is also simpler than the works [6, 12] which require the 3-D face model fitting to the images.

We select 70 different pictures as the training set for the PDM and the gray-level model. The facial features extraction process can be implemented after PDM and gray-level model training. Some results of facial feature extraction are shown in Figs. 13 to 18. Before recognition process, we choose 90 image sequences as the training set, and 15 image sequences for each facial expression. The APs of all image sequences are tested and then classified into six clusters. All image sequences are then tested in the recognition process, the recognition results are shown in the Tables 3 and 4. From these two tables, we may find the correct recognition ratio for the fear expression recognition is the worst (71%) and the correct recognition ratio for the surprise expression is the best (100%). We can explain the results by analyzing Fig. 10 in that the principal distribution of fear expression is overlapped with the other five expressions, whereas, the principal distribution of surprise expression is only overlapped with the other two expressions.

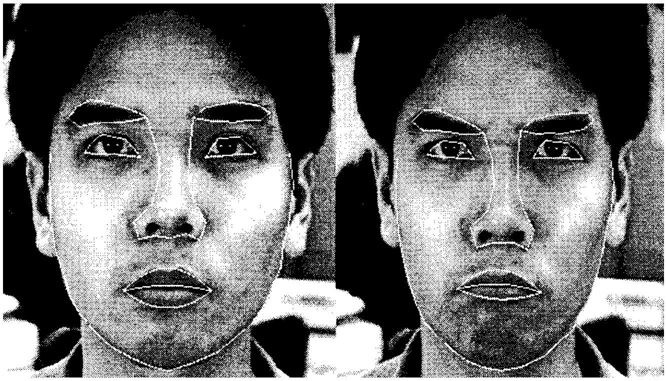

FIG. 17. Face expression 5: anger.

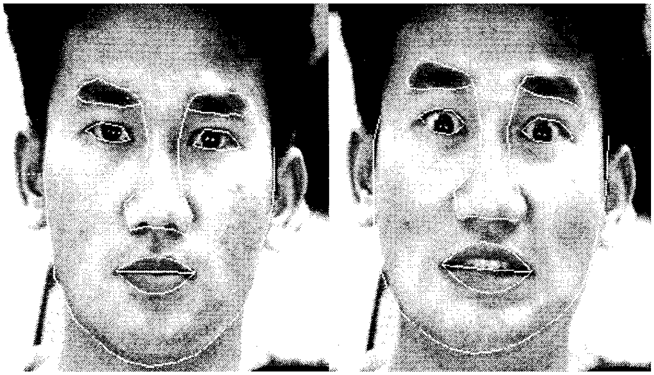

FIG. 18. Face expression 6: fear.

The computation time is less than 7 min in analyzing one image sequence which represents the human expression in the period of about one second. Most of the processing time is spent in gradient-descent-based shape parameters estimation, because there are numerous iterations of matrix calculation. Each image sequence (with 30 consecutive pictures) is off-line captured and stored in the DRAM buffer of the frame grabber. We select the first picture and the last picture of the image sequence when the volunteer starts and ends his facial expression, respectively.

Recognition error is due to two reasons. The first reason is that we took the image sequences from the people who might behave unnaturally in front of the camera, so that the abnormal expressions could not be accurately recognized by our system. The second reason is that in our system, we only extract the contours of the face features, but sometimes the skin wrinkles of the face provides important information for emotion identification. Some expressions (such as anger) contain important features,

TABLE 3
The Correct Recognition Ratio of the First Stage Recognition

	Happy	Smile	Surprise	Sad	Anger	Fear	Correct ratio
Happy	22	4	2			2	73.3%
Smile	7	17			5	1	56.6%
Surprise			30				100%
Sad				18	2	10	60.0%
Anger		4		1	19	6	63.3%
Fear	4	1	1		6	18	60.0%

TABLE 4
The Correct Recognition Ratio after the Second Stage Recognition

	Happy	Smile	Surprise	Sad	Anger	Fear	Correct ratio
Happy	23	3	2			2	76.6%
Smile	2	25			3		83.3%
Surprise			30				100%
Sad				29	1		96.6%
Anger		3		3	24		80.0%
Fear	2	1	2	2		21	70.0%

such as wrinkles, which cannot be recognized in our system.

6. CONCLUSIONS

We have proposed an approach to analyze and classify facial expression based on the facial feature extraction and the action parameters classification. Our system makes mostly correct recognition for the six expressions. However, it is not very satisfactory even though it is also difficult for us human beings to accurately differentiate some facial expressions. Further study is required to improve the accuracy of the facial feature extraction, to increase the effectiveness of the action units classification for expression recognition, and to add more identifiable expressions (such as disgust) to our system.

REFERENCES

1. P. Ekman and W. V. Friesen, Measuring facial movement with facial action coding system, in *Emotion in Human Face* (P. Ekman, Ed.), Cambridge Univ. Press, Cambridge, 1982.
2. K. Mase, Recognition of facial expression from otpical flow, *IEICE Trans. Special Issue on Computer Vision and Its application* **E-74,** No. 10, 1991.
3. Y. Yacoob and L. Davis, Computing spatio-temporal representation of human face, in *CVPR'94, June 21–23, 1994, Seattle,* pp. 70–75. USA.
4. M. Rosenblum, Y. Yacoob, and L. Davis, Human emotion recognition from motion using a radial basis function network architecture, in *Proc. of IEEE Workshop on Motion of Non-Ridgid and Articulated Objects, Nov. 11–12, 1994, Austin,* pp. 43–49, USA.
5. I. A. Essa and A. Pentland, Facial expression recognition using virtually extracted facial action parameters, in *Proc. of Int. Workshop on Auto. Face- and Gesture Recognition,* pp. 35–40, Zurich, 1995.
6. M. J. Black and Y. Yacoob, Recognizing facial expression under rigid and non-rigid facial motions, in *Proc. of Int. Workshop on Auto. Face- and Gesture Recognition,* pp. 12–17, Zurich, 1995.
7. S. Morishima, Emotion model—a criterion for recognition, synthesis and compression of face and emotion, in *Proc. of Int. Workshop on Auto. Face- and Gesture Recognition,* pp. 284–289, Zurich, 1995.
8. P. Vanger, R. Honlinger, and H. Haken, Applications of synergetics in decoding facial expression of emotion, in *Proc. of Int. Workshop on Auto. Face- and Gesture Recognition,* pp. 24–29, Zurich, 1995.
9. Y. Moses, D. Reynard, and A. Blake, Determining facial expression in real-time, in *Proc. of Int. Workshop on Automatic Face- and Gesture Recognition,* pp. 332–337, Zurich, 1995.
10. Y. Kitamura, J. Ohya, N. Ahuja, and F. Kishino, Computational taxonomy and recognition of facial expression, in *Proc. of ACCV'93 Nov. 23–25, 1993, Osaka,* Japan.
11. K. Matsuno, C. W. Lee, and S. Tsuji, Recognition of facial expression with potential net, in *Proc. of ACCV'93 Nov. 23–25, 1993, Osaka,* Japan.
12. M. Kass, A. Witkin, and D. Terzopoulos, Snake: active contour models, in *Proceeding of the First Int. Conf. on Computer Vision,* pp. 259–268, IEEE Comput. Soc. Press, 1987.
13. T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, Active shape models—Their training and application, *Computer Vision Image Understading* **61,** No. 1, 1995, 38–59.
14. T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam, Use of active shape models for locating structures in medical images, *Image Vision Comput.* **12,** No. 6, 1994, 355–365.
15. J. N. Bassili, Emotion recognition: The role of facial movement and the relative importance of upper and lower areas of the face, *J. Personality Social Psychol.* **37,** 1979, 2049–2059.
16. A. Lantis, C. J. Taylor, and T. F. Cootes, Automatic tracking, coding and reconstruction of human faces, using flexible appearance models, *Electron. Lett.* **30,** No. 19, 1994, 1587–1588.
17. A. L. Yuille, D. S. Cohen, and P. W. Hallinan, Feature extraction from faces using deformable template, in *Proc. CVPR-94, June, 1989,* pp. 104–109.
18. T. F. Cootes and C. J. Taylor, Locating faces using statistical feature detector, *Int. Conf. on Automatic Face and Gesture Recognition, Oct. 14–16, 1996, Vermont,* USA.
19. A. Baumberg and D. Hogg, An efficient method for contour tracking using active shape models, *Int. Workshop on Motion of Non-rigid and Articulated Objects, 1994, Austin Texas,* USA.
20. T. Heap and D. Hogg, Extending the point distribution model using polar coordinate, *Image Vision comput.* **14,** 1996, 589–599.
21. C. Bregler and S. M. Omohundro, Nonlinear manifold learning for visual speech recognition, *5th ICCV, June 22–23, 1995, Cambridge, Mass,* USA.