ENHANCING FACIAL EXPRESSION CLASSIFICATION BY INFORMATION FUSION

I. Buciu[1], *Z. Hammal* [2], *A. Caplier*[2], *N. Nikolaidis* [1], *and I. Pitas* [1]

[1] AUTH/Department of Informatics/ Aristotle University of Thessaloniki
GR-54124, Thessaloniki, Box 451, Greece
phone: + 30(2310)99.6361, fax: + 30(2310)99.8453, email: {nelu,nikolaid,pitas}@aiia.csd.auth.gr
web: http://www.aiia.csd.auth.gr
[2] Laboratoire des Images et des Signaux / Institut National Polytechnique de Grenoble
38031 Grenoble, France
phone: + 33(0476)574363, fax: + 33(0476)57 47 90, email: alice.caplier@inpg.fr
web: http://www.lis.inpg.fr

ABSTRACT

The paper presents a system that makes use of the fusion information paradigm to integrate two different sorts of information in order to improve the facial expression classification accuracy over a single feature based classification. The Discriminant Non-negative Matrix Factorization (DNMF) approach is used to extract a first set of features and an automatic extraction algorithm based on geometrical feature is used for retrieving the second set of features. These features are then concatenated into a single feature vector at feature level. Experiments showed that, when these mixed features are used for classification, the classification accuracy is improved compared with the case when only one type of features is used.

1. INTRODUCTION

Studied for decades by psychologists for its important role inside the community, nowadays, facial expression classification issue attracts increasing interest from the computer scientists community. From the psychology perspective, an emotion expressed by facial features deformation contributes to the communication between humans and can help or emphasize the verbal communication. As far as the computer scientists are concerned, their efforts are focusing toward creating more friendly human-computer interfaces which would be able to recognize human facial expression and act accordingly. In principle, facial expression classification methods can be divided into three categories: statistical methods [1] that use characteristic points or characteristic blocks in the face; template based methods [2] using models of facial features or models of facial motion and rule based methods [3]. A survey on automatic facial expression analysis can be found in [4].

Information fusion is a hot research topic in biometrics, where a multifeature system usually achieves higher recognition rate than a single feature one [5]. Other application areas have also benefited by information fusion strategies. Regardless of the application, in a system that combines different types of information the fusion can be done, basically, at three levels: feature level, matching score level, and decision level. Although it is believed that integration at the earlier stage (feature level) of such a system leads to better performance than integration at the final level, just a few works dealt with this type of information fusion mainly due to features incompatibility. Although promising, no much work has been dedicated for employing information fusion in facial expression recognition tasks. One such work is presented in [6], where the authors developed a system which uses two types of features. The first type is the geometric positions of a set of fiducial points on a face. The second type is a set of multi-scale and multi-orientation Gabor wavelet coefficients extracted from the face image at the fiducial points. These are further used independently and jointly as the input of a multi-layer perceptron. However, when combined, the features do not lead to a high improvement in the classification accuracy (especially when the number of layers is high) with respect to the single feature case.

A system which uses two types of information is described in this paper: appearance - based features and geometry-based features. These features are concatenated into a single feature vector at feature level. The appearance-based features are extracted with the help of the Discriminant Non-negative Matrix Factorization (DNMF) algorithm [7]. The geometric features are extracted with the help of an automatic system that segments the salient facial features represented by eyebrows, eyes and mouth, which are relevant for the facial expression. The used geometric features are the difference of five geometrical distances between the salient facial features on the neutral and other facial expressions (such as disgust, happiness, surprise).

2. GEOMETRY BASED APPROACH

Let us consider a sequence of images that contains a human face acting an expression, starting with the neutral face and ending with the full facial expression. The contours of salient facial features (eyes, mouth and brows) are automatically extracted in every frame. This is accomplished by using parametric models. First of all, a coarse localization of these features based on luminance information (valley images for example) is extracted [8]. Then models related to the searched contours are introduced [9]. In this paper, the following parametric models are considered: a circle for the iris, a parabola for the lower eye boundary, Bezier curves for upper eye

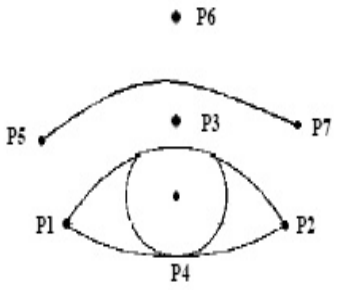

Figure 1: Model of the right eye and eyebrow along with their keypoints.

boundary and brows and cubic curves for the mouth. Figure 1 shows the model for eye and eyebrow. Iris contour being the boundary between the dark area of iris and the eye white, is supposed to be a circle made of points of maximum luminance gradient. The circle of the iris maximizes [10]:

$$E = \sum_{p \in C} \overrightarrow{\nabla} I(p) \overrightarrow{n}(p) \tag{1}$$

where I is the luminance at point p, $\overrightarrow{n}(p)$ is the normal of the boundary at point p and C is a circle. Several circles scanning the search area of each iris are tested and the circle which maximizes E is selected. The radius of the circle is supposed to be known in order to reduce the computational cost. However, it is possible to test several radius values. The lower boundary is represented by a parabola which is defined by points P_1, P_2, P_4 and for the upper boundary, a Bezier curve is defined by the three control points P_1, P_2, P_3. For the eyebrow, the usual model is generally very simple since it is a broken line defined by three points (both corners and a middle point). In this paper, we consider a Bezier curve with three control points P5, P6, P7 as the right model for eyebrows. If $A(a_1, b_1)$, $B(a_2, b_2)$, $C(a_3, b_3)$ are chosen to be three control points related to the eye and eyebrow then, the coordinates (x,y) of each point of the associated Bezier curve are defined by:

$$\begin{aligned} x &= (1-t)^2 a_1 + 2t(1-t)(a_3 - a_1) + t^2 a_2 \\ y &= (1-t)^2 b_1 + 2t(1-t)(b_3 - a_1) + t^2 b_2 \end{aligned} \tag{2}$$

The upper and lower eye model defined above is fitted on the image to be processed by the automatic extraction of keypoints (P_1, P_2, P_3 and P_4) and by the deformation of the model according to information about the maximum of luminance gradient, since, the eye boundary is the limit between the eye white and the skin which is a darker area. For eyebrows, P_5 and P_7 are taken into account. These two points are the corners of each eyebrow. The abscissa x_5 and x_7 of both points correspond to the left and right zero crossing of the derivative of the quantity $H(x) = \sum_{y=1}^{N_y}[255 - I(x,y)]$ and the ordinate $y_5 = y_7$ corresponds to the maximum of the quantity $V(y) = \sum_{x=1}^{N_x}[255 - I(x,y)]$, where I(x,y) is the luminance at pixel (x,y) and (N_x, N_y) represents

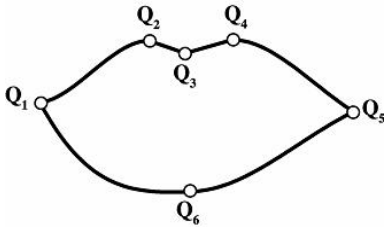

Figure 2: Model of the mouth and its keypoints.

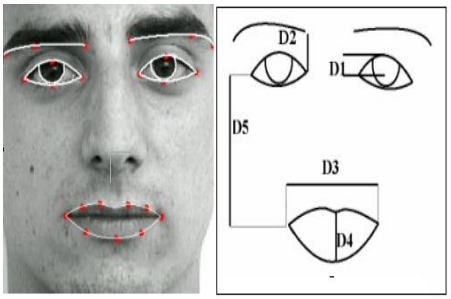

Figure 3: Segmented face (left) and its corresponding skeleton (right).

the dimensions of the region of interest (ROI) for each eyebrow. The third control point P_6 is computed using P_5 and P_7 as

$$\{x_6 = (x_5 + x_7)/2; y_6 = y_7\}. \tag{3}$$

The mouth model is more complex because the mouth deforms more than the eye-related features described above. We follow the model described in [10]. This approach relies on an accurate and robust quasi automatic lip segmentation algorithm. First, the upper mouth boundary and several characteristic points are detected in the first frame by using a new variant of active contours the so-called "jumping snake". Unlike classic snakes, the jumping snake can be initialized far from the final edge and the adjustment of its parameters is easy and intuitive. Then, to achieve the segmentation a parametric model composed of several cubic curves is used. Its high flexibility enables accurate lip contour extraction even in the challenging case of very asymmetric mouth. Compared to existing models, this model brings significant accuracy and realism improvements. The segmentation is achieved by using interframe tracking of the keypoints and the model parameters. The lip segmentation is depicted in Figure 2. Further details about the method can be found in [10]. Figure 3 (left) gives an example of facial feature segmentation.

Finally, on the resulting facial skeleton, five characteristic distances are estimated as described in Figure 3 (right): eye opening ($D1$), distance between the inner corner of the eye and the corresponding corner of the eyebrow ($D2$), mouth opening width ($D3$), mouth opening height ($D4$), distance between a mouth corner and the outer corner of the corresponding eye ($D5$)[11].

3. APPEARANCE BASED APPROACH - DNMF

Let us suppose now that we have n face images that are lexicographically scanned and stored in the columns of a $m \times n$ non-negative matrix $\mathbf{X}$. Then, each image is described by the vector $\mathbf{x}_j = [x_1, x_2, \ldots, x_m]^T$, where $j = 1, \ldots, n$ and m is the number of pixels in the image. DNMF approximates $\mathbf{X}$ by a product of two non-negative matrices (factors) $\mathbf{Z}$ and $\mathbf{H}$ of size $m \times p$ and $p \times n$, respectively, i.e. $\mathbf{X} \approx \mathbf{ZH}$. Several constraints (described below) are involved in this approximation. The columns of $\mathbf{Z}$ form the basis images and $\mathbf{H}$ contains in its rows the decomposition coefficients. Let us now further suppose that we have $\mathcal{Q}$ distinct image classes and $n_{(c)}$ is the number of image samples in a certain class c, $c = 1, \ldots, \mathcal{Q}$. Each image from the image database corresponds to one column of matrix $\mathbf{X}$ and belongs to one of these classes. Therefore, each column of the $p \times n$ matrix $\mathbf{H}$ can be considered as an image representation coefficient vector $\mathbf{h}_{(c)l}$, where $c = 1, \ldots, \mathcal{Q}$ and $l = 1, \ldots, n_{(c)}$. The total number of coefficient vectors is $n = \sum_{c=1}^{\mathcal{Q}} n_{(c)}$. We denote the mean coefficient vector of class c by $\boldsymbol{\mu}_{(c)} = \frac{1}{n_c} \sum_{l=1}^{n_{(c)}} \mathbf{h}_{(c)l}$ and the global mean coefficient vector by $\boldsymbol{\mu} = \frac{1}{n} \sum_{c=1}^{\mathcal{Q}} \sum_{l=1}^{n_{(c)}} \mathbf{h}_{(c)l}$. If we express the within-class scatter matrix by $\mathbf{S}_w = \sum_{c=1}^{\mathcal{Q}} \sum_{l=1}^{n_{(c)}} (\mathbf{h}_{(c)l} - \boldsymbol{\mu}_{(c)})(\mathbf{h}_{(c)l} - \boldsymbol{\mu}_{(c)})^T$ and the between-class scatter matrix by $\mathbf{S}_b = \sum_{c=1}^{\mathcal{Q}} (\boldsymbol{\mu}_{(c)} - \boldsymbol{\mu})(\boldsymbol{\mu}_{(c)} - \boldsymbol{\mu})^T$, the cost function $\mathcal{L}_{DNMF}$ associated with the DNMF algorithm is written as [7]:

$$\mathcal{L}_{DNMF} = KL(\mathbf{X}||\mathbf{ZH}) + \alpha \sum_{i,j} u_{ij} - \beta \sum_i v_{ii} + \\ + \gamma \mathbf{S}_w - \delta \mathbf{S}_b, \tag{4}$$

subject to $\mathbf{Z}, \mathbf{H} \geq 0$. Here $[u_{ij}] = \mathbf{U} = \mathbf{Z}^T\mathbf{Z}$, $[v_{ij}] = \mathbf{V} = \mathbf{H}\mathbf{H}^T$, α, β, γ and δ are constants. The other terms appearing in the cost function have the following meaning. The first term $KL(\mathbf{X}||\mathbf{ZH}) = \sum_{i,j} \left(x_{ij} \ln \frac{x_{ij}}{\sum_k z_{ik} h_{kj}} + \sum_k z_{ik} h_{kj} - x_{ij} \right)$, $k = 1, \ldots, p$, is the Kullback-Leibler divergence and ensures that the product $\mathbf{ZH}$ approximates as much as possible the original data $\mathbf{X}$. The second term can be further split in two as $\sum_{i,j} u_{ij} = \sum_{i \neq j} u_{ij} + \sum_i u_{ii}$, where the minimization of the first sum forces the columns of $\mathbf{Z}$ to be orthogonal in order to reduce the redundancy between basis images, while the minimization of the second sum guarantees the generation of sparse features in the basis images. The third term $\sum_i v_{ii}$ aims at maximizing the total "energy" on each retained component. To achieve a better discrimination between classes the term $\mathbf{S}_w$ should be minimized while the term $\mathbf{S}_b$ should be maximized.

Starting at iteration $t = 0$ with random positive matrices $\mathbf{Z}$ and $\mathbf{H}$, the algorithm updates their values according to an Expectation-Maximization (EM) approach, leading to the following updating rules (i - v)for each iteration $t > 0$ [7]:

$$(i) \quad h_{(c)kl}^{(t)} = \frac{2\mu_c - 1}{4\xi} + \\ + \frac{\sqrt{(1 - 2\mu_c)^2 + 8\xi h_{(c)kl}^{(t-1)} \sum_i z_{ki}^{(t)} \frac{x_{ij}}{\sum_k z_{ik}^{(t)} h_{(c)kl}^{(t-1)}}}}{4\xi} \tag{5}$$

Here $h_{(c)kl}$ refers to the element k of the vector $\mathbf{h}_{(c)l}$ corresponding to the class c. For each class c, the vectors $\mathbf{h}_{(c)l}$ pertaining to the same class (i.e. $l_1, l_2, \ldots, \in c$) are then concatenated into a matrix

$$(ii) \qquad \mathbf{H}_{(c)}^{(t)} = \left[\mathbf{h}_{(c)l_1}^{(t)} \,|\, \mathbf{h}_{(c)l_2}^{(t)} \,|\, \ldots \right] \tag{6}$$

where "|" denotes concatenation. These matrices are further concatenated for all $\mathcal{Q}$ classes as:

$$(iii) \qquad \mathbf{H}^{(t)} = \left[\mathbf{H}_{(1)}^{(t)} \,|\, \mathbf{H}_{(2)}^{(t)} \,|\, \ldots \,|\, \mathbf{H}_{(\mathcal{Q})}^{(t)} \right] \tag{7}$$

The basis images are updated as:

$$(iv) \qquad z_{ik}^{(t)} = \frac{z_{ik}^{(t-1)} \sum_j \frac{x_{ij}}{\sum_k z_{ik}^{(t-1)} h_{kj}^{(t)}} h_{jk}^{(t)}}{\sum_j h_{kj}^{(t)}} \tag{8}$$

$$(v) \qquad z_{ik}^{(t)} = \frac{z_{ik}^{(t)}}{\sum_i z_{ik}^{(t)}}, \quad \text{for all } k \tag{9}$$

The final $\mathbf{Z}$ and $\mathbf{H}$ found at the last iteration are such that approximation $\mathbf{X} \approx \mathbf{ZH}$ is as good as possible, $\mathbf{S}_w$ is as small as possible and $\mathbf{S}_b$ is as large as possible.

4. FEATURES AND INFORMATION FUSION

For each subject and expression, we subtract $D1$, $D2$, $D3$, $D4$, $D5$ corresponding to the expressions *disgust*, *joy*, and *surprise* from $D1$, $D2$, $D3$, $D4$, $D5$ corresponding to *neutral*. If we denote, for example, with $D1_d$ the first distance associated to *disgust* and with $D1_n$ the first distance associated to *neutral*, we have $diff_1 = D1_d - D1_n$. We collect all five differences $\{diff_1, diff_2, diff_3, diff_4, diff_5\}$ in a 5-dimensional geometrical feature vector $\mathbf{g}$. Obviously, the process is repeated for the other two facial expressions, too.

In the case of DNMF approach, each image $\mathbf{x}$ is projected into the pseudoinverse of the basis images learned by DNMF through the training procedure. In other words, the appearance-based feature vector $\mathbf{f}$ is formed as $\mathbf{f} = \mathbf{Z}^\sharp \mathbf{x}$, where "$\sharp$" denotes matrix pseudoinversion. Notice that the size of $\mathbf{f}$ is $p \times 1$.

Information fusion here, simply consists in concatenating both feature vectors, resulting a new $(p + 5)$ - dimensional vector $\mathbf{r} = [\mathbf{f} \,|\, \mathbf{g}]$.

Figure 4: Nine basis images generated by the DNMF algorithm.

5. EXPERIMENTAL RESULTS AND DISCUSSIONS

The experiments were performed by using two facial expression databases. Hammal - Caplier database [11] was used as training data for DNMF and geometric distance approach. The facial images used for testing came from the Cohn-Kanade AU-coded facial expression database [12]. This database was originally created for Action Units (AU) representation appearing in the FACS coding system and not for explicit facial expression recognition. Prior to testing, the facial action (action units) have been converted into emotion class labels according to [3]. Hammal - Caplier database was recorded using regular (non-actor) people. Due to the difficulty for a non actor to simulate all the six universal emotions, images for only four expressions (*disgust, joy, surprise* and *neutral*) exist in the database. Therefore, in our case, the geometrical-based feature vectors form only three classes. The total number of samples taken from the Hammal - Caplier and Cohn-Kanade database is 192 and 104, respectively, including the neutral pose. By eliminating the neutral pose, a number of 144 and 73 samples are attained for the aforementioned databases. Nine basis images retrieved by DNMF through the training step applied to Hammal - Caplier database are depicted in Figure 4.

The experiments were undertaken in the following three scenarios: (a) classification using only appearance-based features extracted by DNMF, (b) classification using solely geometry-based features, and (c) classification by using geometry-appearance features. Experiments with different numbers p, $p \in \{9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169\}$, of basis images retrieved by DNMF were carried out. We used kNN and kmeans as classifiers. The number k of neighbors for kNN were selected from the set $\{1, 2 \ldots, 15\}$. Table 1 presents the results corresponding to the lowest p, and k values that achieved the maximum accuracy in the case of the three scenarios.

As it can be seen from the first two rows of the Table, the accuracy corresponding to the geometry features is much lower than the one obtained by appearance features. Moreover, eleven neighbors are necessary for the kNN to reach the maximum accuracy. Concerning the first scenario, it must be noted that DNMF has also been applied for facial expression classification in the case of Cohn-Kanade database in [7]. However, there, the same database was used for both training and testing (which is a standard procedure), while, here, different databases were used for training and testing. The good classification results obtained here, in the first scenario, emphasize the capability of the algorithm for generalization. The last row shows the performance yielded by both classifiers for the fused (geometry-appearance) feature vector. The results reveal that, compared with the case (a), the accuracy increased from 86.30% to 90.41% for kNN and from 87.67% to 91.78% for *kmeans* classifier. Another issue that can be noticed is related to the minimum number p of basis images necessary to achieve a maximum accuracy. As can be seen only 9 DNMF basis images were sufficient for achieving the highest recognition accuracy, which makes this algorithm attractive from the storage view point (especially when the dimension of the images is high).

Table 1: Classification accuracy (%) for kNN and kmeans classifiers in the three experimental scenarios. p refers to the number of DNMF basis images and k represents the minimum number of neighbors corresponding to the maximum accuracy achieved by kNN classifier.

	kNN			*kmeans*	
	accuracy	k	p	*accuracy*	p
a)	86.30	9	9	87.67	16
b)	69.86	11	-	73.97	-
c)	90.41	7	9	91.78	9

Table 1 only presents the average (total) accuracy. To show the feature extraction systems and classifiers performance in the various expressions, the confusion matrix for each feature extraction system and classifier was computed. The confusion matrix is the matrix whose diagonal entries are the number of facial expressions that are correctly classified, while the off- diagonal entries correspond to misclassification. The rows of the matrix describe the actual (correct) class labels and its columns the predicted ones. Figure 5 depicts the confusion matrix for all experiments. The first two tables associated with DNMF features show that both classifiers confuse six times "disgust" as "joy". When geometry-based features are used, kNN makes the same confusion while the *kmeans* was able to correctly identify "disgust" but had confused "joy" with "disgust" two times. This last confusion applies to kNN as well. Moreover, kNN, which performs the worst, wrongly classified "surprise" as "disgust" nine times. For both classifiers, employing mixed geometry-appearance features reduces the confusion of "disgust" with "joy", which were the most confused facial expressions.

6. CONCLUSION

In this paper, we have attempted to improve the accuracy of two single feature facial expression recognition systems by applying information fusion. As experimental results show, the idea of combining geometry and appearance-based features has leaded to a more accurate facial expression recognition system, increasing the accuracy by 4 percentage for both kNN and *kmeans* classifiers with respect to the second best case, i.e. DNMF appearance-based features one.

	disgust	joy	surprise
disgust	14	1	2
joy	6	25	0
surprise	1	0	24

(1)

	disgust	joy	surprise
disgust	16	1	0
joy	6	24	1
surprise	1	0	24

(2)

	disgust	joy	surprise
disgust	13	2	2
joy	6	25	0
surprise	9	3	13

(3)

	disgust	joy	surprise
disgust	4	2	11
joy	0	27	4
surprise	0	2	23

(4)

	disgust	joy	surprise
disgust	16	0	1
joy	4	25	2
surprise	0	0	25

(5)

	disgust	joy	surprise
disgust	15	0	2
joy	3	27	1
surprise	0	0	25

(6)

Figure 5: Confusion matrices corresponding to (1) DNMF appearance-features and kNN, (2) DNMF appearance features and $kmeans$ (3) geometry-based features and kNN (4) geometry-based features and $kmeans$ (5) geometry-appearance features and kNN (6) geometry-appearance features and $kmeans$.

Acknowledgment

This work has been conducted in conjunction with the "SIMILAR" European Network of Excellence on Multimodal Interfaces of the IST Programme of the European Union (www.similar.cc).

REFERENCES

[1] Y. Shinza, Y. Saito, Y. Kenmochi, and K. Kotani, "Facial expression analysis by integrating information of feature-point positions and gray levels of facial images," *Proc. IEEE Proc. Int. Conf. on Image Processing*, 2000.

[2] Y. Tian, T. Kanade, and J.Cohn, "Recognition actions units for facial expression analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 97–115, 2001.

[3] M. Pantic and L. J. M. Rothkrantz, "Expert system for automatic analysis of facial expressions," *Image and Vision Computing*, vol. 18, no. 11, pp. 881–905, 2000.

[4] B. Fasel and J. Luettin, "Automatic facial expression analysis: a survey," *Pattern Recognition*, vol. 1, no. 30, pp. 259–275, 2003.

[5] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, no. 1, pp. 4–20, 2004.

[6] Z. Zhang, M. Lyons, M. Schuster, and S. Akamatsu, "Comparison between geometry-based and Gabor-wavelets-based facial expression recognition using multi-layer perceptron," *Proc. of Third IEEE Int. Conf. Automatic Face and Gesture Recognition*, pp. 454–459, 1998.

[7] I. Buciu and I. Pitas, "A new sparse image representation algorithm applied to facial expression recognition," *Proc. IEEE Workshop on Machine Learning for Signal Processing*, pp. 539–548, 2004.

[8] K. Sobottka and I. Pitas, "Looking for faces and facial features in color images," *Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications, Russian Academy of Sciences*, vol. 7, no. 1, pp. 124–137, 1997.

[9] Y. Tian, T. Kanade, and J.Cohn, "Dual state parametric eye tracking," *Proc. of the 4th Int. Conf. on Automatic Face and Gesture Recognition*, pp. 110–115, 2000.

[10] Z. Hammal, N. Eveno, A. Caplier, and P-Y Coulon, "Parametric models for facial features segmentation," *Signal Processing*, , no. 86, pp. 399–413, 2006.

[11] Z. Hammal, A. Caplier, and M. Rombaut, "A fusion process based on belief theory for classification of facial basic emotions," *Proc. Fusion'2005 the 8th International Conference onInformation fusion (ISIF 2005)*, 2005.

[12] T. Kanade, J. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," *Proc. Fourth IEEE Int. Conf. Face and Gesture Recognition*, pp. 46–53, March 2000.