Texture and Shape Information Fusion for Facial Action Unit Recognition

Irene Kotsia, Stefanos Zafeiriou, Nikolaos Nikolaidis and Ioannis Pitas
Aristotle University of Thessaloniki, Department of Informatics
Box 451, 54124 Thessaloniki, Greece
Tel. ++ 30 231 099 63 04, Fax ++ 30 231 099 63 04
email: {ekotsia, dralbert, nikolaid, pitas}@aiia.csd.auth.gr

***Abstract*—A novel method that fuses texture and shape information to achieve Facial Action Unit (FAU) recognition from video sequences is proposed. In order to extract the texture information, a subspace method based on Discriminant Non-negative Matrix Factorization (DNMF) is applied on the difference images of the video sequence, calculated taking under consideration the neutral and the most expressive frame, to extract the desired classification label. The shape information consists of the deformed Candide facial grid (more specifically the grid node displacements between the neutral and the most expressive facial expression frame) that corresponds to the facial expression depicted in the video sequence. The shape information is afterwards classified using a two-class Support Vector Machine (SVM) system. The fusion of texture and shape information is performed using Median Radial Basis Functions (MRBFs) Neural Networks (NNs) in order to detect the set of present FAUs. The accuracy achieved in the Cohn-Kanade database is equal to 92.1% when recognizing the 17 FAUs that are responsible for facial expression development.**

***Index Terms*—Facial Action Unit Recognition, Discriminant Non-negative Matrix Factorization, Support Vector Machines, Radial Basis Functions Neural Networks, Fusion.**

I. Introduction

Several research efforts have been done during the past two decades regarding facial expression recognition, as applications such as smart environments require efficient facial expression recognition. A set of six basic facial expressions (anger, disgust, fear, happiness, sadness and surprise) that are thought to be expressed in a similar way all over the world was defined [1], thus making the facial expression recognition more standard. A set of muscle movements known as Facial Action Units (FAUs) was also defined. These FAUs are combined in order to create the rules responsible for the formation of facial expressions as proposed in [2]. The above mentioned basic facial expressions along with the neutral state are the target of facial expression recognition systems developed nowadays. A survey on automatic facial expression recognition can be found in [2].

As can be seen from the rules proposed in [2] (Table I), the FAUs 1, 2, 4, 5, 6, 7, 9, 10, 12, 15, 16, 17, 20, 23, 24, 25 and 26 are necessary for fully describing all facial expressions. Therefore, we concentrate on the detection of these 17 FAUs. The operators $+$, *or* in Table I refer to the logical AND, OR operations, respectively. The neutral state is not taken under consideration, as no FAUs are present in it.

TABLE I
THE FAUS TO FACIAL EXPRESSIONS RULES AS PROPOSED IN [2].

Expression	FAU coded description [2]
Anger	$4 + 7 + (((23 \text{ or } 24) \text{ with or not } 17) \text{or}$ $(16 + (25 \text{ or } 26)) \text{ or } (10 + 16 + (25 \text{ or } 26)))$ with or not 2
Disgust	$((10 \text{ with or not } 17) \text{ or } (9 \text{ with or not } 17)) + (25 \text{ or } 26)$
Fear	$(1 + 4) + (5 + 7) + 20 + (25 \text{ or } 26)$
Happiness	$6 + 12 + 16 + (25 \text{ or } 26)$
Sadness	$1 + 4 + (6 \text{ or } 7) + 15 + 17 + (25 \text{ or } 26)$
Surprise	$(1 + 2) + (5 \text{ without } 7) + 26$

In this paper, a novel method for video based Facial Action Units (FAUs) recognition that exploits both the texture and shape information is proposed. The features of the facial texture are obtained by applying a subspace representation method based on a discriminant extension of the Non-negative Matrix Factorization (NMF) algorithm (the so-called DNMF algorithm [3]) on the images derived from the video sequence. The DNMF algorithm is applied on the difference images calculated by subtracting the neutral frame of the video sequence from the fully expressed one. Thus, the set of present FAUs in the difference image under examination is detected. The shape information is extracted by calculating the Candide node displacements between the neutral and the expressive frame [4]. The FAUs classification is obtained using a bank of two-class SVM systems. Thus, the set of FAUs that are adequate for facial expression representation is detected [2]. The texture and shape information are then fused using a Median Radial Basis Function (MRBF) Neural Network, to provide the final classification regarding the set of present FAUs in the video sequence under examination.

II. System description

The system is composed of three subsystems: texture information extraction, shape information extraction and their fusion for final classification. The texture subsystem operates on the difference images, created from the available image sequences by subtracting the neutral image from the corresponding fully expressive image (see Figure 1). The difference images are used instead of the original facial expression images, due to the fact that they emphasize the facial regions in motion and reduce the variance related to the identity-specific aspects of the facial image [5]. The same image sequences

Fig. 1. Difference images between neutral pose and fully expressive one.

are also used as input to the shape extraction information subsystem. The texture extraction subsystem specifies which FAUs are activated in the difference images are examination.

The information obtained from the grid tracking system is used to calculate the Candide node difference between the neutral and fully expressive frame. The node differences are used as an input to a bank of 17 two-class SVM systems, each one corresponding to a FAU to be detected. Each SVM system is able to recognize if the FAU under examination is present or absent in the video sequence being examined. The output information from both the texture and shape classifiers consists of a set of activated FAUs in the examined video sequence. This set is fed to the fusion subsystem to provide the final classification result, i.e. the set of activated FAUs in the examined video sequence. The diagram of the system used for FAU recognition is shown in Figure 2.

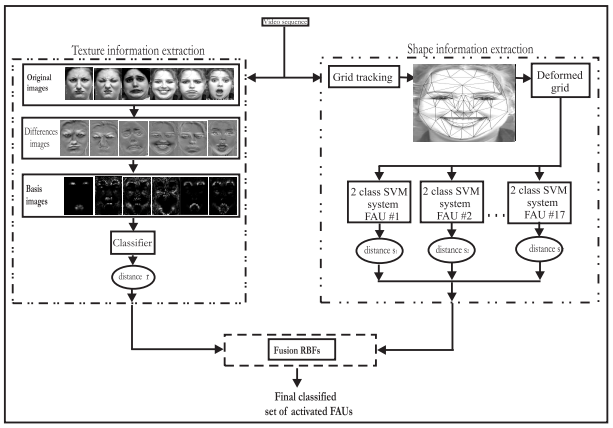


Fig. 2. System architecture for FAU recognition in facial videos.

III. Texture information extraction

Let $\mathcal{U}$ be a database of facial videos. The facial expression depicted in each video sequence is dynamic, evolving through time as the video progresses. We take under consideration the frame that depicts the facial expression in its greatest intensity, i.e. the last frame, to create a facial image database $\mathcal{Y}$ that consists of the difference images of each video sequence (calculated by subtracting the neutral frame from the expressive one). Thus, $\mathcal{Y}$ consists of the difference images. In each image $\mathbf{y} \in \mathcal{Y}$ one or more FAUs are activated. The database that contains the difference images is clustered into 17 different classes $\mathcal{Y}_k,\ k = 1, \ldots, 17$, each one representing one of the 17 basic FAUs. The images are labelled properly with $\{-1, 1\}$ if a FAU is absent or present. In that way, the activation of more than one FAUs simultaneously is possible, thus allowing the application of the rules presented in [2]. Thus, each image can belong to more than one classes. Each difference image is initially normalized. The smallest intensity value for every row is found and its absolute value is added to each pixel in the row, thus resulting in a positive image. In both cases, the input image is afterwards scanned row-wise to form a vector $\mathbf{x} \in \Re_+^F$ of dimension F.

The algorithm used for texture extraction was the DNMF algorithm, which is an extension of the NMF algorithm. The NMF algorithm algorithm is an image decomposition algorithm that allows only additive combinations of non negative components. DNMF was the result of an attempt to introduce discriminant information to the NMF decomposition. Both NMF and DNMF algorithms will be presented analytically below.

A. The Non-negative Matrix Factorization Algorithm

In order to apply NMF in the database $\mathcal{Y}$, the matrix $\mathbf{X} \in \Re_+^{F \times G} = [x_{i,j}]$ should be constructed, where $x_{i,j}$ is the i-th element of the j-th image, F is the number of pixels and G is the number of images in the database. In other words, the j-th column of $\mathbf{X}$ is the $\mathbf{x}_j$ facial image in vector form (i.e. $\mathbf{x}_j \in \Re_+^F$). NMF aims at finding two matrices $\mathbf{Z} \in \Re_+^{F \times M} = [z_{i,k}]$ and $\mathbf{H} \in \Re_+^{M \times L} = [h_{k,j}]$ such that :

$$\mathbf{X} \approx \mathbf{ZH}, \tag{1}$$

where M is the number of dimensions taken under consideration (usually $M \ll F$), $\mathbf{Z}$ is a matrix that consists of basis images and $\mathbf{H}$ is the matrix that contains the corresponding weight vectors.

A facial image $\mathbf{x}_j$ after the NMF decomposition can be written as $\mathbf{x}_j \approx \mathbf{Zh}_j$, where $\mathbf{h}_j$ is the j-th column of $\mathbf{H}$. Thus, the columns of the matrix $\mathbf{Z}$ can be considered as basis images and the vector $\mathbf{h}_j$ as the corresponding weight vector. Vectors $\mathbf{h}_j$ can also be considered as the projection vectors of the original image vectors $\mathbf{x}_j$ on a lower dimension feature space.

Let $\mathbf{x} = [x_1, \ldots, x_F], \mathbf{q} = [q_1, \ldots, q_F]$ be positive vectors $x_i > 0,\ q_i > 0$, then the Kullback-Leibler (KL) Divergence (or relative entropy) between $\mathbf{x}$ and $\mathbf{q}$ is defined [6] as:

$$KL(\mathbf{x}||\mathbf{q}) \triangleq \sum_i (x_i \ln \frac{x_i}{q_i} + q_i - x_i). \tag{2}$$

The defined cost for the decomposition (1) is the sum of all KL divergences for all images in the database. This way the

following metric can be formed :

$$\begin{aligned} D_N(\mathbf{X}||\mathbf{ZH}) &= \sum_j KL(\mathbf{x}_j||\mathbf{Zh}_j) \\ &= \sum_{i,j}(x_{i,j}\ln(\frac{x_{i,j}}{\sum_k z_{i,k}h_{k,j}})+ \\ &+\sum_k z_{i,k}h_{k,j} - x_{i,j}) \end{aligned} \quad (3)$$

as the measure of the cost for factoring $\mathbf{X}$ into $\mathbf{ZH}$ [7].

The NMF factorization is the outcome of the following optimization problem :

$$\min_{\mathbf{Z},\mathbf{H}} D_N(\mathbf{X}||\mathbf{ZH}) \text{ subject to} \quad (4)$$
$$z_{i,k} \geq 0,\ h_{k,j} \geq 0,\ \sum_i z_{i,j} = 1,\ \forall j.$$

The update rules for the weight matrix $\mathbf{H}$ and the bases matrix $\mathbf{Z}$ can be found in [7].

B. The Discriminant Non-negative Matrix Factorization Algorithm

In order to incorporate discriminants constraints inside the NMF cost function (4), we should use the information regarding the separation of the vectors $\mathbf{h}_j$ into different classes. Let us assume that the vector $\mathbf{h}_j$ that corresponds to the jth column of the matrix $\mathbf{H}$, is the coefficient vector for the ρth facial image of the rth class and will be denoted as $\boldsymbol{\eta}_\rho^{(r)} = [\eta_{\rho,1}^{(r)} \dots \eta_{\rho,M}^{(r)}]^T$. The mean vector of the vectors $\boldsymbol{\eta}_\rho^{(r)}$ for the class r is denoted as $\boldsymbol{\mu}^{(r)} = [\mu_1^{(r)} \dots \mu_M^{(r)}]^T$ and the mean of all classes as $\boldsymbol{\mu} = [\mu_1 \dots \mu_M]^T$. The cardinality of a facial class $\mathcal{Y}_r$ is denoted by N_r. Then, the within scatter matrix for the coefficient vectors $\mathbf{h}_j$ is defined as:

$$\mathbf{S}_w = \sum_{r=1}^{6}\sum_{\rho=1}^{N_r}(\boldsymbol{\eta}_\rho^{(r)} - \boldsymbol{\mu}^{(r)})(\boldsymbol{\eta}_\rho^{(r)} - \boldsymbol{\mu}^{(r)})^T \quad (5)$$

whereas the between scatter matrix is defined as:

$$\mathbf{S}_b = \sum_{r=1}^{6} N_r(\boldsymbol{\mu}^{(r)} - \boldsymbol{\mu})(\boldsymbol{\mu}^{(r)} - \boldsymbol{\mu})^T. \quad (6)$$

The discriminant constraints are incorporated by requiring tr$[\mathbf{S}_w]$ to be as small as possible while tr$[\mathbf{S}_b]$ is required to be as large as possible. Thus, the new cost function is given by:

$$D_d(\mathbf{X}||\mathbf{Z}_D\mathbf{H}) = D_N(\mathbf{X}||\mathbf{Z}_D\mathbf{H}) + \gamma\text{tr}[\mathbf{S}_w] - \delta\text{tr}[\mathbf{S}_b]. \quad (7)$$

where γ and δ are constants and D is the measure of the cost for factoring $\mathbf{X}$ into $\mathbf{ZH}$ [3].

Following the same Expectation Maximization (EM) approach used by NMF techniques [3], the following update rules for the weight coefficients $h_{k,j}$ that belong to the r-th facial class become:

$$h_{k,j}^{(t)} = \frac{T_1 + \sqrt{T_1^2 + 4(2\gamma - (2\gamma+2\delta)\frac{1}{N_r})h_{k,j}^{(t-1)}}}{2(2\gamma - (2\gamma+2\delta)\frac{1}{N_r})} \frac{\sum_i z_{i,k}^{(t-1)} \frac{x_{i,j}}{\sum_l z_{i,l}^{(t-1)} h_{l,j}^{(t-1)}}}{2(2\gamma - (2\gamma+2\delta)\frac{1}{N_r})}. \quad (8)$$

where T_1 is given by:

$$T_1 = (2\gamma + 2\delta)(\frac{1}{N_r}\sum_{\lambda,\lambda\neq l} h_{k,\lambda}) - 2\delta\mu_k - 1. \quad (9)$$

The update rules for the bases $\mathbf{Z}_D$, are given by:

$$z_{i,k}^{(t)} = z_{i,k}^{(t-1)} \frac{\sum_j h_{k,j}^{(t)} \frac{x_{i,j}}{\sum_l z_{i,l}^{(t-1)} h_{l,j}^{(t)}}}{\sum_j h_{k,j}^{(t)}} \quad (10)$$

and

$$z_{i,k}^{(t)} = \frac{\acute{z}_{i,k}^{(t)}}{\sum_l \acute{z}_{l,k}^{(t)}}. \quad (11)$$

The above decomposition is a supervised non-negative matrix factorization method that decomposes the facial images into parts while, enhancing the class separability. The matrix $\mathbf{Z}_D^\dagger = (\mathbf{Z}_D^T\mathbf{Z}_D)^{-1}\mathbf{Z}_D^T$, which is the pseudo-inverse of $\mathbf{Z}_D$, is then used for extracting the discriminant features as $\acute{\mathbf{x}} = \mathbf{Z}_D^\dagger\mathbf{x}$. The most interesting property of DNMF algorithm is that it decomposes the image to facial areas, i.e. mouth, eyebrows, eyes, and focuses on extracting the information hiding in them. Thus, the new representation of the image is a better one compared to the one acquired when the whole image was taken under consideration.

For testing, the facial image $\mathbf{x}_j$ is projected on the low dimensional feature space produced by the application of the DNMF algorithm:

$$\acute{\mathbf{x}}_j = \mathbf{Z}_D^\dagger\mathbf{x}_j. \quad (12)$$

For the classification of the facial image $\acute{\mathbf{x}}_j$, its distance from each class center is calculated. The smallest distance from a class (presence or absence of the specific FAU) specifies the class r_j to which the sample under examination belongs:

$$r_j = \underset{k=1,2}{\text{argmin}}(\min\|\acute{\mathbf{x}}_j - \boldsymbol{\mu}^{(k)}\|). \quad (13)$$

IV. Shape information extraction

The geometrical information extraction is achieved using a grid tracking system, based on deformable models [4]. The tracking is performed using a pyramidal implementation of the well-known Kanade-Lucas-Tomasi (KLT) algorithm. The user has to place manually a number of Candide grid nodes on the corresponding positions of the face depicted at the first frame of the image sequence. The algorithm automatically adjusts the grid to the face and then tracks it through the image sequence, as it evolves through time. At the end, the grid tracking algorithm produces the deformed Candide grid that corresponds to the last frame i.e. the one that depicts the greatest intensity of the facial expression. An example of the Candide grid for every facial expression can be seen in Figure 3.

The shape information used from the j-th video sequence is the displacements $\mathbf{d}_j^i$ of the nodes of the Candide grid, defined as the difference between coordinates of this node in

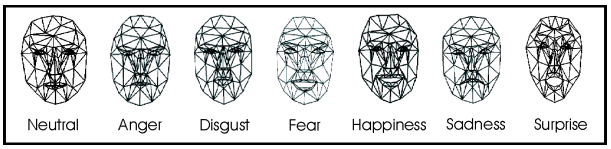


Fig. 3. An example of the Candide grid for every facial expression.

the first and last frame [4]:

$$\mathbf{d}_j^i = [\Delta x_j^i \Delta y_j^i]^T \quad i \in \{1, \ldots, K\} \quad \text{and} \quad j \in \{1, \ldots, N\} \tag{14}$$

where i is an index that refers to the node under consideration. In our case $K = 104$ nodes were used.

For every facial video in the training set, a feature vector $\mathbf{g}_j$ of $F = 2 \cdot 104 = 208$ dimensions, containing the geometrical displacements of all grid nodes is created:

$$\mathbf{g}_j = [\mathbf{d}_j^1 \quad \mathbf{d}_j^2 \quad \ldots \quad \mathbf{d}_j^K]^T. \tag{15}$$

Let $\mathcal{U}$ be the video database that contains the facial videos, that are clustered into 17 different classes $\mathcal{U}_k,\ k = 1, \ldots, 17$, each one representing one of the 17 basic FAUs. The feature vectors $\mathbf{g}_j \in \Re^F$ labelled properly with $\{-1, 1\}$ if a FAU is absent or present are used as a training input to a multi-class SVM as will be described in the following section.

An example of several posers for each facial expression with the corresponding deformed grid is shown in Figure 4.

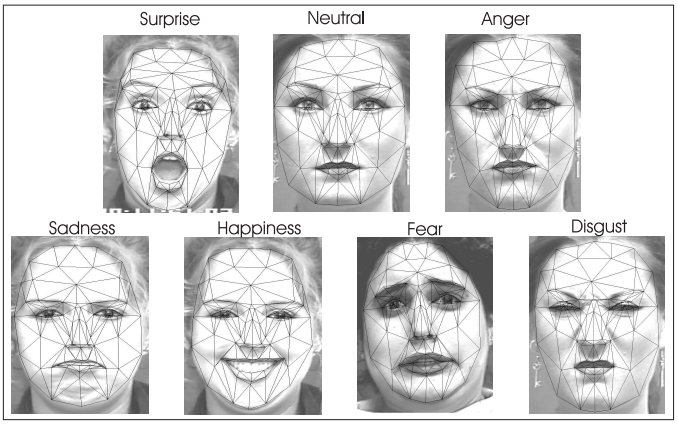


Fig. 4. An example of several posers for each facial expression from the Cohn-Kanade database.

A. Support Vector Machines

A two-class SVM classifier finds a hyperplane or surface that separates the two-classes $\mathcal{F}^1$ and $\mathcal{F}^2$ with the maximum margin [8]. In order to train a two-class SVM network using soft margin formulation, the following minimization problem has to be solved [8]:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \qquad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{j=1}^{N}\xi_j \tag{16}$$

subject to the separability constraints:

$$y_i(\mathbf{w}^T\phi(\mathbf{x}_j) + b) \geq 1 - \xi_j, \xi_j \geq 0, \quad j = 1, \ldots, N \tag{17}$$

where $\mathbf{w}$ is the vector of hyperplane coefficients, b is the bias, $\boldsymbol{\xi} = [\xi_i, \ldots, \xi_w]$ is the slack variable vector, C is the term that penalizes the training errors and y_i is the class label of the vector $\mathbf{x}_i$ that takes values in $\{-1, 1\}$.

After solving the optimization problem (16) subject to the separability constraints (17), the decision function that can be used to classify unlabelled samples is:

$$s(\mathbf{g}) = \text{sign}(\mathbf{w}^T\phi(\mathbf{g}) + b). \tag{18}$$

The output of the decision function is the label of the class the specific FAU under examination belongs to.

In this formulation, a non-linear mapping ϕ is used. On the other hand, if a linear SVM system is to be constructed then $\phi(\mathbf{g}) = \mathbf{g}$. The non-linear mapping is defined by a positive kernel function, $h(\mathbf{g}_i, \mathbf{g}_j)$, specifying an inner product in the feature space and satisfying the Mercer condition [8]:

$$h(\mathbf{g}_i, \mathbf{g}_j) = \phi(\mathbf{g}_i)^T\phi(\mathbf{g}_j). \tag{19}$$

Typical kernels include the polynomial and Radial Basis Functions (RBF) kernels:

$$\begin{aligned} h(\mathbf{x}, \mathbf{y}) &= \phi(\mathbf{x})^T\phi(\mathbf{y}) = (\mathbf{x}^T\mathbf{y} + 1)^d \\ h(\mathbf{x}, \mathbf{y}) &= \phi(\mathbf{x})^T\phi(\mathbf{y}) = e^{-\gamma(\mathbf{x}-\mathbf{y})^T(\mathbf{x}-\mathbf{y})} \end{aligned} \tag{20}$$

where d is the degree of the polynomial kernel and γ is the spread of the Gaussian cluster. These kernels have been used in the experiments conducted in this paper.

B. Fusion of texture and shape information

The application of the DNMF algorithm on the images of the database results in the label that specifies if the FAU being examined is present or not in the difference image under examination. Similarly, the classification procedure performed using the SVM system on the grid following the facial expression through time also results in the FAUs labels.

In more detail, the image $\mathbf{x}_j$ and the corresponding vector of geometrical displacements $\mathbf{g}_j$ are taken into consideration. The DNMF algorithm, applied to the $\mathbf{x}_j$ image, produces the label r_j as a result, while SVMs applied to the vector of geometrical displacements $\mathbf{g}_j$, produce the label s_j as the equivalent result. Thus, a new feature vector $\mathbf{c}_j$, defined as:

$$\mathbf{c}_j = [r_j \quad s_j]^T. \tag{21}$$

containing classification information from both sources was created to be used for fusion purposes.

C. Radial Basis Function (RBF) Neural Networks (NNs)

A series of RBF NNs was used for the fusion of texture and shape results. The number of networks is equal to the number of FAUs that we are trying to detect. Each network provided a binary decision on whether the corresponding FAUs

are activated in the data. The RBF network consists of a linear combination of a set of basis functions [9]:

$$p_k(\mathbf{c}_j) = \sum_{n=1}^{M} w_{k,n}\phi_n(\mathbf{c}_j), \quad k = 1, 2 \tag{22}$$

where M is the number of kernel functions and $w_{k,n}$ are the weights of the hidden unit to output connection. Each hidden unit (kernel) implements a Gaussian function:

$$\phi_n(\mathbf{c}_j) = exp[-(\mathbf{m}_n - \mathbf{c}_j)^T \mathbf{\Sigma}_n^{-1}(\mathbf{m}_n - \mathbf{c}_j)] \tag{23}$$

where $j = 1, \ldots M$, $\mathbf{m}_n$ is the mean vector and $\mathbf{\Sigma}_n$ is the covariance matrix [9].

The decision regarding the class l_j of $\mathbf{c}_j$, namely the existence of the corresponding FAU, is handled by the related RBF NN and is taken as:

$$l_j = \underset{k=1,\ldots,2}{\operatorname{argmax}}\, p_k(\mathbf{c}_j). \tag{24}$$

V. Experimental results

A. Database description

The Cohn-Kanade database has been used in the experiments. This database is annotated with FAUs. All the available subjects and videos were taken under consideration to form the database for the experiments.

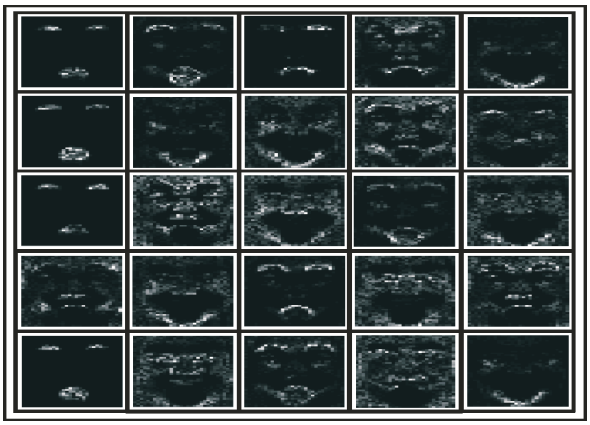
The most frequently used approach for testing the generalization performance of a classifier is the leave-one-out cross-validation approach [10]. A variant of leave-one-out was used (i.e., leave 20% of the samples out) for the formation of the test dataset in our experiments. Five sets containing 20% of the data for each class, chosen randomly, were created. One set containing 20% of the samples for each class is used as the test set, while the remaining sets form the training set. After the classification procedure is performed, the samples forming the test set are incorporated into the current training set, and a new set of samples (20% of the samples for each class) is extracted to form the new test set. The remaining samples create the new training set. This procedure is repeated five times. The average classification accuracy is defined as the mean value of the percentages of the correctly classified facial expressions over all data presentations.

B. FAUs recognition using texture information

An example of the basis images extracted when the DNMF algorithm is applied at the difference images is shown in Figure 5. The accuracy rates obtained for FAUs recognition using only the texture information and for different numbers of basis images are shown in Figure 6. The best classification accuracy achieved was equal to 84.4% and the number of basis images was equal to 180.

C. FAUs recognition using shape information

The best accuracy rate obtained for FAUs recognition using only the shape information was equal to 86.7%. In Figure 7, the accuracy rates achieved for FAUs recognition when using SVMs are shown. The functions used as SVM kernels were the polynomial and RBF functions. The best accuracy was achieved when a polynomial kernel of degree 4 was used.



(c)

Fig. 5. Basis images extracted for the DNMF algorithm.

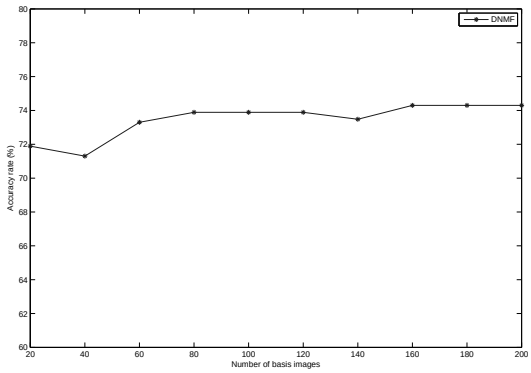
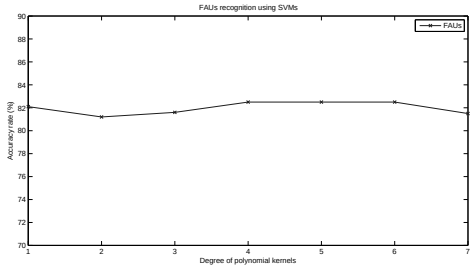


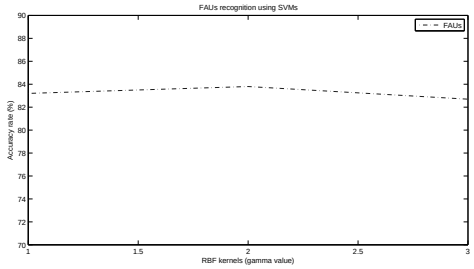
Fig. 6. Recognition accuracies obtained for FAU recognition using the DNMF algorithm.

D. Fusion of texture and shape information for FAUs recognition

The best accuracy achieved when using the proposed fusion approach was equal to 92.1%, which is significantly better than the one obtained when using either texture or shape information. The accuracy rate was increased due to the use of both sources of information. The introduction of texture eliminates some of the confusions observed when using shape information only. This happens as in many FAUs, the shape information is not enough to fully describe its presence. In many cases, the available grid nodes fail to describe all possible texture characteristics, such as furrows and wrinkles that may appear on the face. To be more specific, when FAU 12 is observed (see Figure 8), some vertical furrows appear between the nose and the corners of the mouth (emphasized with a cloud of black dots). These furrows cannot be fully described by the Candide grid deformation due to the absence of properly placed grid nodes. The same happens with FAU 23 (also shown in Figure 8), where horizontal furrows appear between the chin and mouth (emphasized with a cloud of black dots). Texture can capture all the necessary information where



(a)



(b)

Fig. 7. FAUs recognition accuracies using shape and SVMs for various kernels (a) polynomial kernels (b) RBF kernels.

the shape description would fail, thus making the fusion of the two kinds of information more powerful.

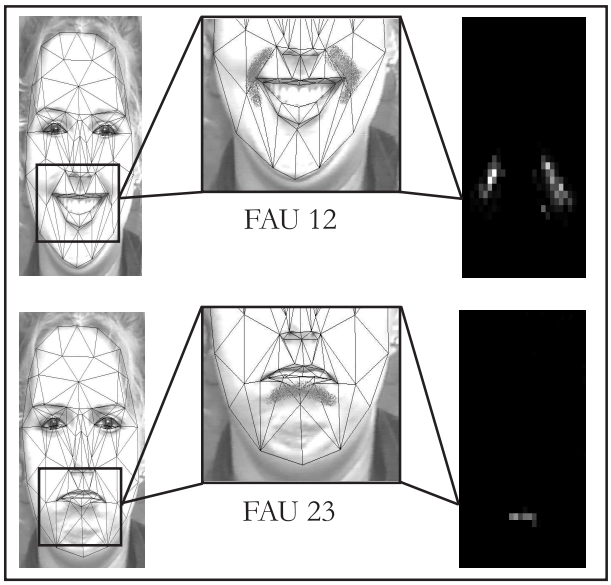


Fig. 8. Furrows that appear when FAUs 12 and 23 are observed and two of the sparse DNMF bases that correspond to the furrows.

VI. Conclusions

A novel method for FAUs recognition is proposed in this paper. The recognition is performed by fusing the texture and the shape information extracted from a video sequence using a subspace representation method and a SVMs system, respectively. The results obtained from the above mentioned methods are then fused using MRBF NNs. The system achieves an accuracy of 92.1% when recognizing the 17 basic FAUs.

References

[1] P. Ekman and W. V. Friesen, *Emotion in the Human Face*, Prentice Hall, New Jersey, 1975.

[2] 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, August 2000.

[3] S. Zafeiriou, A. Tefas, I. Buciu, and I. Pitas, "Exploiting discriminant information in non-negative matrix factorization with application to frontal face verification," *IEEE Transactions on Neural Networks*, vol. 17, no. 3, pp. 683–695, 2006.

[4] I. Kotsia and I. Pitas, "Facial expression recognition in image sequences using geometric deformation features and support vector machines," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 172–187, January 2007.

[5] G. Donato, M.S. Bartlett, J.C. Hager, P. Ekman, and T.J. Sejnowski, "Classifying facial actions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 974–989, Oct. 1999.

[6] M. Collins, R. E. Schapire, and Y. Singer, "Logistic regression, adaboost and bregman distances," *Computational Learing Theory*, pp. 158–169, 2000.

[7] D.D. Lee and H.S. Seung, "Algorithms for non-negative matrix factorization," in *NIPS*, 2000, pp. 556–562.

[8] V. Vapnik, *Statistical learning theory*, Wiley, New York, 1998.

[9] A. G. Bors and I. Pitas, "Median radial basis function neural network," *IEEE Transactions on Neural Networks*, vol. 7, pp. 1351–1364, 1996.

[10] I. Cohen, N. Sebe, S. Garg, L. S. Chen, and T. S. Huanga, "Facial expression recognition from video sequences: temporal and static modelling," *Computer Vision and Image Understanding*, vol. 91, pp. 160–187, 2003.