1

Facial expression recognition using shape and texture information

I. Kotsia[1] and I. Pitas[1]

Aristotle University of Thessaloniki
pitas@aiia.csd.auth.gr
Department of Informatics
Box 451 54124
Thessaloniki, Greece

Summary. A novel method based on shape and texture information is proposed in this paper for facial expression recognition from video sequences. The Discriminant Non-negative Matrix Factorization (DNMF) algorithm is applied at the image corresponding to the greatest intensity of the facial expression (last frame of the video sequence), extracting that way the texture information. A Support Vector Machines (SVMs) system is used for the classification of the shape information derived from tracking the Candide grid over the video sequence. The shape information consists of the differences of the node coordinates between the first (neutral) and last (fully expressed facial expression) video frame. Subsequently, fusion of texture and shape information obtained is performed using Radial Basis Function (RBF) Neural Networks (NNs). The accuracy achieved is equal to 98,2% when recognizing the six basic facial expressions.

1.1 Introduction

During the past two decades, many studies regarding facial expression recognition, which plays a vital role in human centered interfaces, have been conducted. Psychologists have defined the following basic facial expressions: anger, disgust, fear, happiness, sadness and surprise [?]. A set of muscle movements, known as Action Units, was created. These movements form the so called $Facial \quad Action \quad Coding \quad System \quad (FACS)$ [?]. A survey on automatic facial expression recognition can be found in [?].

In the current paper, a novel method for video based facial expression recognition by fusing texture and shape information is proposed. The texture information is obtained by applying the DNMF algorithm [?] on the last frame of the video sequence, i.e. the one that corresponds to the greatest intensity of the facial expression depicted. The shape information is calculated as the difference of Candide facial model grid node coordinates between the first and the last frame of a video sequence [?]. The decision made regarding the class the sample belongs to, is obtained using a SVM system. Both the DNMF and SVM algorithms have as an output the distances of the sample under examination from each of the six classes (facial expressions). Fusion of the distances obtained from DNMF and SVMs applications is attempted using a RBF NN system. The experiments performed using the Cohn-Kanade database indicate a recognition accuracy of 98,2% when recognizing the six basic facial expressions. The novelty of this method lies in the combination of both texture and geometrical information for facial expression recognition.

1.2 System description

The diagram of the proposed system is shown in Figure **??**.

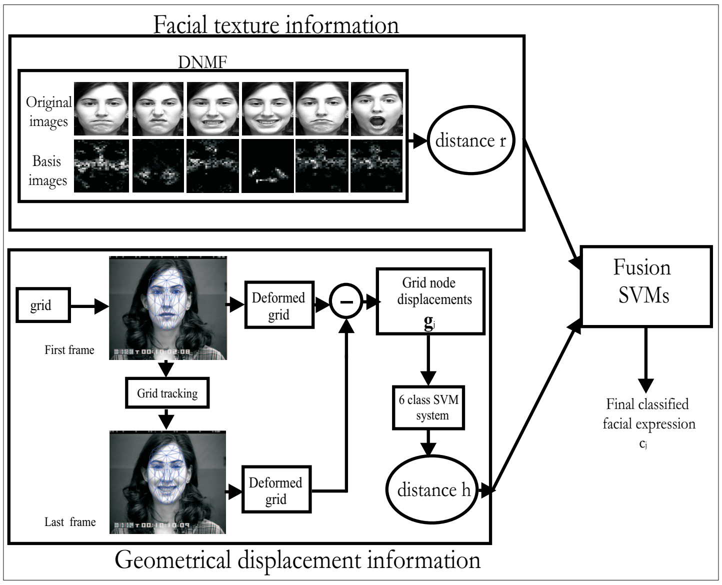

Fig. 1.1. System architecture for facial expression recognition in facial videos

The system is composed of three subsystems: two responsible for texture and shape information extraction and a third one responsible for the fusion of their results. Figure **??** shows the two sources of information (texture and shape) used by the system.

1.3 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

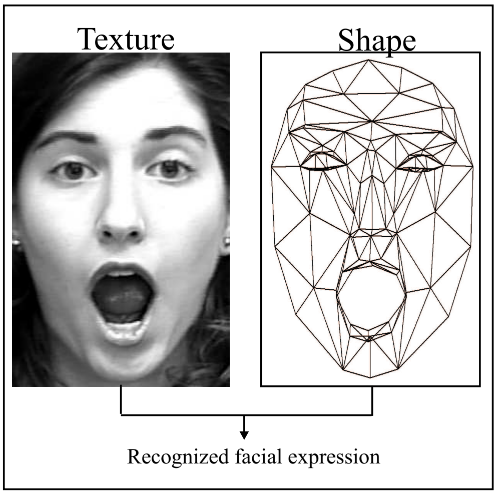

Fig. 1.2. Fusion of texture and shape.

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 Y. Thus, $\mathcal{Y}$ consists of images where the depicted facial expression obtains its greatest intensity . Each image $\mathbf{y} \in \mathcal{Y}$ belongs to one of the 6 basic facial expression classes$\{\mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_6\}$ with $\mathcal{Y} = \bigcup_{r=1}^{6} \mathcal{Y}_r$. Each image $\mathbf{y} \in \Re_+^{K \times G}$ of dimension $F = K \times G$ forms a vector $\mathbf{x} \in \Re_+^F$. The vectors $\mathbf{x} \in \Re_+^F$ will be used in our algorithm.

The algorithm used was the DNMF algorithm, which is a extension of the Non-negative Matrix Factorization (NMF) algorithm. The NMF algorithm algorithm is an object 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.

1.3.1 The Non-negative Matrix Factorization Algorithm

A facial image $\mathbf{x}_j$ after the NMF decomposition can be written as $\mathbf{x}_j \approx \mathbf{Z}\mathbf{h}_j$, 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 vectors. Vectors $\mathbf{h}_j$ can also be considered as the projections vectors of the original facial vectors $\mathbf{x}_j$ on a lower dimensional feature space .

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 :

$$\mathbf{X} \approx \mathbf{Z}\mathbf{H}. \tag{1.1}$$

where M is the number of dimensions taken under consideration (usually $M \ll F$).

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

$$\min_{\mathbf{Z},\mathbf{H}} D_N(\mathbf{X}||\mathbf{Z}\mathbf{H}) \text{ subject to} \tag{1.2}$$

$$z_{i,k} \geq 0,\ h_{k,j} \geq 0,\ \sum_i z_{i,j} = 1,\ \forall j.$$

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

1.3.2 The Discriminant Non-negative Matrix Factorization Algorithm

In order to incorporate discriminants constraints inside the NMF cost function (**??**), 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:

$$\mathbf{S}_w = \sum_{r=1}^{6} \sum_{\rho=1}^{N_r} (\boldsymbol{\eta}_\rho^{(r)} - \boldsymbol{\mu}^{(r)})(\boldsymbol{\eta}_\rho^{(r)} - \boldsymbol{\mu}^{(r)})^T \tag{1.3}$$

whereas the between scatter matrix is defined as:

$$\mathbf{S}_b = \sum_{r=1}^{6} N_r(\boldsymbol{\mu}^{(r)} - \boldsymbol{\mu})(\boldsymbol{\mu}^{(r)} - \boldsymbol{\mu})^T. \tag{1.4}$$

The discriminant constraints are incorporated by requiring $\mathrm{tr}[\mathbf{S}_w]$ to be as small as possible while $\mathrm{tr}[\mathbf{S}_b]$ is required to be as large as possible.

$$D_d(\mathbf{X}||\mathbf{Z}_D\mathbf{H}) = D_N(\mathbf{X}||\mathbf{Z}_D\mathbf{H}) + \gamma\mathrm{tr}[\mathbf{S}_w] - \delta\mathrm{tr}[\mathbf{S}_b]. \tag{1.5}$$

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

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

$$h_{k,j}^{(t)} = \frac{T_1 + \sqrt{T_1^2 + 4(2\gamma - (2\gamma + 2\delta)\frac{1}{N_r})h_{k,j}^{(t-1)}}}{2(2\gamma - (2\gamma + 2\delta)\frac{1}{N_r})}$$

$$\frac{\sum_i z_{i,k}^{(t-1)} \frac{x_{i,j}}{\sum_l z_{i,l}^{(t-1)} h_{l,j}^{(t-1)}}}{2(2\gamma - (2\gamma + 2\delta)\frac{1}{N_r})}. \tag{1.6}$$

where T_1 is given by:

$$T_1 = (2\gamma + 2\delta)(\frac{1}{N_r} \sum_{\lambda, \lambda \neq l} h_{k,\lambda}) - 2\delta\mu_k - 1. \tag{1.7}$$

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

$$\acute{z}_{i,k}^{(t)} = z_{i,k}^{(t-1)} \frac{\sum_j h_{k,j}^{(t)} \frac{x_{i,j}}{\sum_l z_{i,l}^{(t-1)} h_{l,j}^{(t)}}}{\sum_j h_{k,j}^{(t)}} \tag{1.8}$$

and

$$z_{i,k}^{(t)} = \frac{\acute{z}_{i,k}^{(t)}}{\sum_l \acute{z}_{l,k}^{(t)}}. \tag{1.9}$$

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:

$$\acute{\mathbf{x}}_j = \mathbf{Z}_D^{\dagger} \mathbf{x}_j \tag{1.10}$$

For the projection of the facial image $\acute{\mathbf{x}}_j$, one distance from each center class is calculated. The smallest distance defined as:

$$r_j = \min_{k=1,\ldots,6} \|\acute{\mathbf{x}}_j - \boldsymbol{\mu}^{(r)}\| \tag{1.11}$$

is the one that is taken as the output of the DNMF system.

1.4 Shape information extraction

The geometrical information extraction is done by a grid tracking system, based on deformable models [**?**]. 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.

The shape information used from the j 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 the first and last frame [**?**]:

$$\mathbf{d}_j^i = [\Delta x_j^i \Delta y_j^i]^T \quad i \in \{1, \ldots, K\} \quad \text{and} \quad j \in \{1, \ldots, N\} \tag{1.12}$$

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:

$$\mathbf{g}_j = [\mathbf{d}_j^1 \quad \mathbf{d}_j^2 \quad \ldots \quad \mathbf{d}_j^K]^T. \tag{1.13}$$

Let $\mathcal{U}$ be the video database that contains the facial videos, that are clustered into 6 different classes $\mathcal{U}_k,\ k = 1, \ldots, 6$, each one representing one of 6 basic facial expressions. The feature vectors $\mathbf{g}_j \in \Re^F$ labelled properly with the true corresponding facial expression are used as an input to a multi class SVM and will be described in the following section.

1.4.1 Support Vector Machines

Consider the training data:

$$(\mathbf{g}_1, l_1), \ldots, (\mathbf{g}_N, l_N) \tag{1.14}$$

where $\mathbf{g}_j \in \Re^F \quad j = 1, \ldots, N$ are the deformation feature vectors and $l_j \in \{1, \ldots, 6\} \quad j = 1, \ldots, N$ are the facial expression labels of the feature vector. The approach implemented for multiclass problems used for direct facial expression recognition is the one described in [**?**] that solves only one optimization problem for each class (facial expression). This approach constructs 6 two-class rules where the $k-$th function $\mathbf{w}_k^T \phi(\mathbf{g}_j) + b_k$ separates training vectors of the class k from the rest of the vectors. Here ϕ is the function that maps the deformation vectors to a higher dimensional space (where the data are supposed to be linearly or near linearly separable) and $\mathbf{b} = [b_1 \ldots b_6]^T$ is the bias vector. Hence, there are 6 decision functions, all obtained by solving a different SVM problem for each class. The formulation is as follows:

$$\min_{\mathbf{w},\mathbf{b},\boldsymbol{\xi}} \qquad \frac{1}{2} \sum_{k=1}^{6} \mathbf{w}_k^T \mathbf{w}_k + C \sum_{j=1}^{N} \sum_{k \neq l_j} \xi_j^k \tag{1.15}$$

subject to the constraints:

$$\begin{aligned} \mathbf{w}_{l_j}^T \phi(\mathbf{g}_j) + b_{l_j} &\geq \mathbf{w}_k^T \phi(\mathbf{g}_j) + b_k + 2 - \xi_j^k \\ \xi_j^k \geq 0, \quad j = 1, \ldots, N, &\quad k \in \{1, \ldots, 6\} \backslash l_j. \end{aligned} \tag{1.16}$$

where C is the penalty parameter for non linear separability and $\boldsymbol{\xi} = [\ldots, \xi_i^m, \ldots]^T$ is the slack variable vector. Then, the function used to calculate the distance of a sample from each center class is defined as:

$$s(\mathbf{g}) = \max_{k=1,\ldots,6} (\mathbf{w}_k^T \phi(\mathbf{g}) + b_k). \tag{1.17}$$

That distance was considered as the output of the SVM based shape extraction procedure. A linear kernel was used for the SVM system in order to avoid search for appropriate kernels.

1.5 Fusion of texture and shape information

The application of the DNMF algorithm on the images of the database resulted in the extraction of the texture information of the facial expressions depicted. Similarly, the classification procedure performed using the SVM system on the grid following the facial expression through time resulted in the extraction of the shape information .

More specifically, the image $\mathbf{x}_j$ and the corresponding vector of geometrical displacements $\mathbf{g}_j$ were taken into consideration. The DNMF algorithm, applied to the $\mathbf{x}_j$ image, produces the distance r_j as a result, while SVMs applied to the vector of geometrical displacements $\mathbf{g}_j$, produces the distance s_j as the equivalent result. The distances r_j and s_j were normalized in $[0, 1]$ using Gaussian normalization. Thus, a new feature vector $\mathbf{c}_j$, defined as:

$$\mathbf{c}_j = [r_j \quad s_j]^T. \tag{1.18}$$

containing information from both sources was created.

1.5.1 Radial Basis Function (RBF) Neural Networks (NNs)

A RBF NN was used for the fusion of texture and shape results. The RBF function is approximated as a linear combination of a set of basis functions [?]:

$$p_k(\mathbf{c}_j) = \sum_{n=1}^{M} w_{k,n} \phi_n(\mathbf{c}_j) \tag{1.19}$$

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 implements a Gaussian function:

$$\phi_n(\mathbf{c}_j) = exp[-(\mathbf{m}_n - \mathbf{c}_j)^T \mathbf{\Sigma}_n^{-1}(\mathbf{m}_n - \mathbf{c}_j)] \tag{1.20}$$

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

Each pattern $\mathbf{c}_j$ is considered assigned only to one class l_j. The decision regarding the class l_j of $\mathbf{c}_j$ is taken as:

$$l_j = \underset{k=1,\ldots,6}{\operatorname{argmax}} \, p_k(\mathbf{c}_j) \tag{1.21}$$

The feature vector $\mathbf{c}_j$ was used as an input to the RBF NN that was created. The output of that system was the label l_j that classified the sample under examination (pair of texture and shape information) to one of the 6 classes (facial expressions).

1.6 Experimental results

In order to create the training set, the last frames of the video sequences used were extracted. By doing so, two databases were created, one for texture extraction using DNMF and another one for shape extraction using SVMs. The texture database consisted of images that corresponded to the last frame of every video sequence studied, while the shape database consisted of the grid displacements that were noticed between the first and the last frame of every video sequence.

The databases were created using a subset of the Cohn-Kanade database that consists of 222 image sequences, 37 samples per facial expression. The leave-one-out method was used for the experiments [**?**]. For the implementation of the RBF NN, 25 neurons were used for the output layer and 35 for the hidden layer.

The accuracy achieved when only DNMF was applied was equal to 86,5%, while the equivalent one when SVMs along with shape information were used was 93,5%. The obtained accuracy after performing fusion of the two information sources was equal to 98,2%. By fusing texture information into the shape results certain confusions are resolved. For example, some facial expressions involve subtle facial movements. That results in confusion with other facial expressions when only shape information is used. By introducing texture information, those confusions are eliminated. For example, in the case of anger, a subtle eyebrow movement is involved which cannot probably be identified as movement, but would most probably be noticed if texture is available. Therefore, the fusion of shape and texture information results in correctly classifying most of the confused cases, thus increasing the accuracy rate.

The confusion matrix [**?**] has been also computed. It is a $n \times n$ matrix containing information about the actual class label l_j, $j = 1, .., n$ (in its columns) and the label obtained through classification o_j, $j = 1, .., n$ (in its rows). The diagonal entries of the confusion matrix are the numbers of facial expressions that are correctly classified, while the off-diagonal entries correspond to misclassification. The confusions matrices obtained when using DNMF on texture information, SVM on shape information and when the proposed fusion is applied are presented in Table **??**.

Table 1.1. Confusion matrices for DNMF results, SVMs results and fusion results, respectively.

$lab_{cl} \backslash lab_{ac}$	anger	disgust	fear	happiness	sadness	surprise
anger	13	0	0	0	0	0
disgust	10	37	0	0	0	0
fear	4	0	37	0	0	1
happiness	2	0	0	37	0	0
sadness	7	0	0	0	37	5
surprise	1	0	0	0	0	31

$lab_{cl} \backslash lab_{ac}$	anger	disgust	fear	happiness	sadness	surprise
anger	24	0	0	0	0	0
disgust	5	37	0	0	0	0
fear	0	0	37	0	0	1
happiness	0	0	0	37	0	0
sadness	8	0	0	0	37	0
surprise	0	0	0	0	0	36

$lab_{cl} \backslash lab_{ac}$	anger	disgust	fear	happiness	sadness	surprise
anger	33	0	0	0	0	0
disgust	2	37	0	0	0	0
fear	0	0	37	0	0	0
happiness	0	0	0	37	0	0
sadness	2	0	0	0	37	0
surprise	0	0	0	0	0	37

1.7 Conclusions

A novel method for facial expression recognition is proposed in this paper. The recognition is performed by fusing the texture and the shape information extracted from a video sequence. The DNMF algorithm is applied at the last frames of every video sequence corresponding to the greatest intensity of the facial expression, extracting that way the texture information. Simultaneously, a SVM system classifies the shape information obtained by tracking the Candide grid between the first (neutral) and last (fully expressed facial expression) video frame. The results obtained from the above mentioned methods are then fused using RNF NNs. The system achieves an accuracy of 98,2% when recognizing the six basic facial expressions.

1.8 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. P. Ekman, and W.V. Friesen, "Emotion in the Human Face," *Prentice Hall*, 1975.
2. T. Kanade, J. Cohn, and Y. Tian, "Comprehensive Database for Facial Expression Analysis," *Proceedings of IEEE International Conference on Face and Gesture Recognition*, 2000.
3. B. Fasel, and J. Luettin, "Automatic Facial Expression Analysis: A Survey," *Pattern Recognition*, 2003.
4. 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*, accepted for publication, 2005.
5. D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *Advances in Neural Information Processing Systems*, vol. 13pp. 556−562, 2001.
6. I. Kotsia, and I. Pitas, "Real time facial expression recognition from image sequences using Support Vector Machines," *IEEE International Conference on Image Processing (ICIP 2005)*, 11-14 September, 2005.
7. V. Vapnik, "Statistical learning theory," 1998.
8. A. G. Bors and I. Pitas, "Median Radial Basis Function Neural Network," *IEEE Transactions on Neural Networks*, vol. 7, pp. 1351-1364, November 1996.