

A CONNECTIONIST COMPUTATIONAL METHOD FOR FACE RECOGNITION

FRANCISCO A. PUJOL ^a, HIGINIO MORA ^{a,*}, JOSÉ A. GIRONA-SELVA ^a

^aDepartment of Computer Technology
University of Alicante, 03690, San Vicente del Raspeig, Alicante, Spain
e-mail: {fpujol, hmora}@dtic.ua.es, jags20@alu.ua.es

In this work, a modified version of the elastic bunch graph matching (EBGM) algorithm for face recognition is introduced. First, faces are detected by using a fuzzy skin detector based on the RGB color space. Then, the fiducial points for the facial graph are extracted automatically by adjusting a grid of points to the result of an edge detector. After that, the position of the nodes, their relation with their neighbors and their Gabor jets are calculated in order to obtain the feature vector defining each face. A self-organizing map (SOM) framework is shown afterwards. Thus, the calculation of the winning neuron and the recognition process are performed by using a similarity function that takes into account both the geometric and texture information of the facial graph. The set of experiments carried out for our SOM-EBGM method shows the accuracy of our proposal when compared with other state-of-the-art methods.

Keywords: pattern recognition, face recognition, neural networks, self-organizing maps.

1. Introduction

In recent years, there has been intensive research carried out to develop complex security systems involving biometric features. Automated biometric systems are being widely used in many applications such as surveillance, digital libraries, forensic work, law enforcement, human computer intelligent interaction, and banking, among others. For applications requiring high levels of security, biometrics can be integrated with other authentication means such as smart cards and passwords. In relation to this, face recognition is an emerging research area and, in the next few years, it is supposed to be extensively used for automatic human recognition systems in many of the applications mentioned before.

One of the most popular methods for face recognition is elastic graph bunch matching (EBGM), proposed by Wiskott *et al.* (1997). This method is an evolution of the so-called dynamic link architecture (DLA) (Kotropoulos and Pitas, 1997). The main idea in elastic graph matching is to represent a face starting from a set of reference or fiducial points known as landmarks. These fiducial points have a spatial coherence, as they are connected using a graph structure. Therefore, EBGM represents faces as facial graphs with nodes at those facial landmarks (such

as eyes, the tip of the nose, etc.). Considering these nodes, geometric information can be extracted, and both distance and angle metrics can be defined accordingly.

This algorithm takes into account that facial images have many nonlinear features (variations in lighting, pose and expression) that are not generally considered in linear analysis methods, such as linear discriminant analysis (LDA) or principal component analysis (PCA) (Shin and Park, 2011). Moreover, it is particularly robust when out-of-plane rotations appear. However, the main drawback of this method is that it requires an accurate location of the fiducial points.

Artificial neural networks (ANNs) are one of the most often used paradigms to address problems in artificial intelligence (Bańka *et al.*, 2014; Kayarvizhy *et al.*, 2014; Tran *et al.*, 2014; Kumar and Kumar, 2015). Among the different approaches of ANNs, the self organizing map (SOM) has special features for association and pattern classification (Kohonen, 2001), and it is one of the most popular neural network models. This technique is suitable in situations where there is an inaccuracy or a lack of formalization of the problem to be solved. In these cases, there is no precise mathematical formulation of the relationship between the input patterns (Azorín-López *et al.*, 2014).

The SOM makes use of an unsupervised learning

*Corresponding author

process where the distribution of a set of patterns is learned without any class information (Loderer *et al.*, 2014). This network is able to emulate the ability of the human brain to project the input data to a position in the map using a neighborhood of neurons (En-Naimani *et al.*, 2014). That is, the topographic order of training samples can find clusters if the dimensionality of the network is smaller than the number of training samples. The neighborhood of neurons can map similar features to nearby positions in the feature map. This fact can be especially useful when applied to a set of face landmarks, as in the EBGM method.

Consequently, in this paper we will use a connectionist model to improve the efficiency of the EBGM algorithm. To do this, an SOM is applied in the construction of the database of facial graphs in an adaptive learning process. First, the fiducial points will be extracted automatically and, after that, faces will be grouped (or clustered) into M classes, each one corresponding to a different person. The main contributions of our paper can be summarized as follows:

- A modified version of the original EBGM method is introduced. In this work, fiducial points are obtained automatically by using an edge detector. The similarity function is composed of weighted geometric and texture distances.
- A self-organizing map framework for the recognition process is presented. The SOM will deal with the facial graphs obtained from the feature extraction process, will cluster similar facial graphs from the training set, and then will recognize new input images from the test database. There is no previous work that combines EBGM with an SOM framework.
- An RGB fuzzy skin detector is applied for the face detection process. Each color plane is modeled using fuzzy sets. This detector achieves very good detection rates and proves to be a suitable technique for segmenting skin in various environment conditions.

This paper is organized as follows. Section 2 describes the EBGM method and summarizes some related work. Section 3 considers the design of an RGB fuzzy system for detecting faces. Section 4 explains a modified proposal of an EBGM-based face recognition method and the formal framework to define it. Section 5 introduces the neural network approach with a self organizing map for recognition. Section 6 describes the experiments carried out. Finally, conclusions and some future works are discussed in Section 7.

2. EBGM algorithm and related work

In this section, the EBGM algorithm is described and, afterwards, some recent, related works are discussed.

2.1. Elastic bunch graph matching method.

Elastic bunch graph matching is a feature-based face identification method. It derives a bunch of jets for each training image and uses the jets to represent the graph node. To form a bunch graph, a collection of facial images is marked with node locations at defined positions of the head. These node locations are called landmarks and are obtained by a semi-automatic process. When matching a bunch graph to an image, the jet extracted from the image is compared with all jets in the corresponding bunch attached to the bunch graph, and the best matching one is selected.

Jets are defined as Gabor coefficients in a landmark location computed by convolving a set of Gabor wavelet filters around each landmark location. The jets of all the training images are collected in a data structure called a bunch graph. The bunch graph has a node for every landmark on the face and every node is a collection of jets for the corresponding landmark. The main steps for face recognition by EBGM are outlined below (Rattani *et al.*, 2006):

1. Select the landmarks on the training face images to create the face models. The selection is performed manually.
2. Convolve these points with a Gabor wavelet to construct the Gabor jets J . The local appearance around a fiducial point \vec{x} will be coded by using the convolution of the input image $I(\vec{x})$ with a Gabor filter $\psi_m(\vec{x})$ so that

$$\psi_m(\vec{x}) = \frac{\|\vec{k}_m\|}{\sigma^2} \exp\left(\frac{\|\vec{k}_m\|\|\vec{x}\|}{2\sigma^2}\right) \times \left[\exp\left(i\vec{k}_m\vec{x}\right) - \exp(-0.5\sigma^2)\right], \quad (1)$$

where the width of the Gaussian is controlled by the parameter $\sigma = 2\pi$ and \vec{k}_m is the wave vector:

$$\vec{k}_m = \begin{pmatrix} k_{m_x} \\ k_{m_y} \end{pmatrix} = \begin{pmatrix} k_\nu \cos \varphi_\mu \\ k_\nu \sin \varphi_\mu \end{pmatrix}, \quad (2)$$

$$k_\nu = 2^{-\frac{\nu+2}{2}}\pi, \quad \varphi_\mu = \mu\frac{\pi}{8}.$$

A jet J will have 40 coefficients, where $\nu = 0, 1, \dots, 4$ correspond to 5 different frequencies and $\mu = 0, 1, \dots, 7$ are 8 different orientations of the Gabor kernels.

3. Create a data structure called the bunch graph corresponding to facial landmarks that contains a bunch of model jets extracted from the face model.

4. Then for every new image to be recognized:
 - (a) Estimate and locate the landmark positions with the use of the bunch graph.
 - (b) Calculate the new jets displacement from the actual position by comparing it with the most similar model jet.
 - (c) Create a new facial graph containing each landmark position and jet values for that landmark position.
5. Similarly, for each new image, estimate and locate the landmarks using a bunch graph. Then the features are extracted by convolving with the Gabor filters followed by the creation of the facial graph. The matching score is calculated on the basis of the similarity between the facial graphs of the images in the database and the one in a new input image.

2.2. Related work. EBGGM has been used for face recognition in the last few years. Most of the methods based on EBGGM use Gabor wavelets for feature extraction (Shen and Bai, 2006). These features are represented by a grid of points geometrically adjusted to the features extracted. The recognition is based on the wavelet coefficients, which are calculated for the nodes of a 2D elastic graph representing the grid containing the landmarks. This method combines a local and a global representation through the processing of a Gabor filter with several scales and several directions (jets), of a point set called fiducial points located in specific regions of the face. The location of the fiducial points is the most complex task of this method. These points depend on the lighting conditions, the expression and the pose of the face.

An alternative method proposed by Monzo *et al.* (2010) is the application of the histogram of orientation gradients (HOG) instead of using Gabor filters to locate features. The HOG descriptor is a statistic measure where the orientations of all the image gradients around a reference point are taken into account. This algorithm provides invariance in terms of location and orientation.

Recently, a combination of EBGGM with PCA and soft biometrics is used to conduct a study on the influence of age variations in face recognition (Guo *et al.*, 2010). Additionally, some new versions of EBGGM focus on fast versions of the algorithm in order to make it feasible for real conditions; thus, a parallel version of EBGGM for fast face recognition using the MPI (message passing interface) is presented by Chen *et al.* (2013). Khatun and Bhuiyan (2011) proposed a neural network based face recognition system using Gabor filter coefficients, where the recognition used a hybrid neural network with two networks, a bidirectional associative memory (BAM) for

dimensionality reduction and a multilayer perceptron with backpropagation algorithm for training the network.

In the work of Mitra *et al.* (2011), a data mining approach to improve the performance of EBGGM when using a large database was proposed, based on an entropy decision tree with the most important features in the face recognition process. Finally, Sarkar (2012) combined skin detection with EBGGM so as to obtain an accurate recognition, since skin segmented images remove background noise and reduce errors in identifying Gabor features.

As mentioned before, calculating the precise location of the fiducial points is not straightforward. In the original EBGGM algorithm, a fixed number of features were established. They corresponded to specific face characteristics, such as the pupils or the corners of the mouth. As a result, a facial model graph is obtained and the fiducial points are manually selected for each image in the database. Another way to locate the features is based on a uniformly distributed grid of points that deforms and conforms to a pattern, such as the contours identified by an edge detector (Canny, Sobel, MLSEC, etc.) (Espí *et al.*, 2008; González-Jiménez and Alba-Castro, 2007).

Some advances have been made recently for the detection of the fiducial points in faces. Among others, Belhumeur *et al.* (2011) used a Bayesian model with very accurate results, whereas a method based on regression forests that detects 2D facial feature points in real-time was presented by Dantone *et al.* (2012). Moreover, Baltrusaitis *et al.* (2012) proposed a probabilistic landmark detector that learns non-linear and spatial relationships between the input pixels and the probability of a landmark being aligned. Finally, Jin *et al.* (2013) developed a Hough voting-based method to improve the efficiency and accuracy of fiducial points localization.

To sum up, from this revision, two conclusions emerge: first of all, there is still a great interest among many research groups in using and improving the original EBGGM method for face recognition; moreover, most of these investigations are focused on adapting EBGGM to be used in real-time conditions with an accurate location of the landmarks or fiducial points for faces. It is clear that there is still much to be done in this field, and no previous works on the explicit application of self-organizing maps to EBGGM have been found.

3. Skin-color face detection

Before our recognition algorithm is applied, faces must be located using some detection method. Many recent proposals are based on the underlying idea of representing the skin color in an optimal color space (such as RGB, YIQ or HSV) by means of the so-called skin cluster (Yang *et al.*, 2004). Thus, color information is an efficient tool

for identifying facial areas if the skin color model can be properly adapted for different lighting environments. Moreover, as color processing is much faster than processing other facial features, it can be used as a preliminary process for other face detection techniques (Hsu et al., 2002). Another advantage of skin color detection is that color is invariant to partial occlusion, scaling, rotations, translation, and face orientation. This fact is particularly useful in face detection techniques.

For our proposals, the RGB color system has been chosen. The RGB space corresponds closely to the physical sensors for colored light such as the cones in the human eye or red, green and blue filters in most color charge-coupled device (CCD) sensors. In the work of Pujol et al. (2008), a fuzzy RGB skin color detector was proposed. Let us summarize its main features and adapt it to face detection.

Given a color image I of size $W = n \times m$ pixels, where each pixel is defined by a color vector c in a color space C , so that $c(p) = c_1(p), c_2(p), \dots, c_l(p), \forall p \in I$, the histogram of $C, H(C)$ is defined as a $q \times l$ array $H(C) = \{f_1, f_2, \dots, f_l\}$, such that each f_i is the frequency vector on the image I , using q bins, of the color component c_i , for $i = 1, 2, \dots, l$.

Therefore, the value of each bin is the number of pixels in image I having the color c_i . If $H(C)$ is normalized by W , then $H(C)$ takes the color space C into the interval $[0, 1]$; that is, $H(C)$ represents the probability distribution of each colour c_i to be present in image I . According to Zadeh's theory (Zadeh, 1965; Piegat, 2005), a fuzzy set is a pair (A, m) where A is a set and $m: A \rightarrow [0, 1]$. This can be applied to the color histogram, where the fuzzy set can be defined as the pair (C, H) , where C is the color space and $H: C \rightarrow [0, 1]$ is the normalized histogram. For each $c \in C, H(c)$ is the grade of membership of c , so that $c \in (C, H) \iff c \in C$ AND $H(c) \neq 0$.

As a result, the membership functions for the skin color in each RGB plane can be modeled using a Gaussian function, such that

$$\mu_{SKIN_i}(c_i) = \beta_i \exp\left(-\frac{(c_i - \alpha_i)^2}{2\sigma_i^2}\right), \quad (3)$$

where $i = \{R, G, B\}; \{c_R, c_G, c_B\} \in [0, 255]; \beta_i = \max H(c_i), \sigma_i^2$ is the variance of each fuzzy set c_i and $\alpha_i = \arg \max_{c_i} H(c_i)$.

For the background pixels, i.e., the non-skin pixels in the image, let us consider a variation of the model introduced by Murthy and Pal (1990), which identifies the fuzziness in the transition region between the object (in this case, the skin) and the background classes. Thus, the membership value of a point to the object is determined by applying an S -function and a Z -function to the each color plane, so that:

$$\mu_{NSKIN_i}^S(c_i) = \begin{cases} 0 & \text{for } c_i \leq a_{S_i} \\ 2\left(\frac{c_i - a_{S_i}}{\gamma_{S_i} - a_{S_i}}\right)^2 & \text{for } a_{S_i} \leq c_i \leq b_{S_i} \\ 1 - 2\left(\frac{c_i - \gamma_{S_i}}{\gamma_{S_i} - a_{S_i}}\right)^2 & \text{for } b_{S_i} \leq c_i \leq \gamma_{S_i} \\ 1 & \text{for } \gamma_{S_i} \leq c_i, \end{cases} \quad (4)$$

$$\mu_{NSKIN_i}^Z(c_i) = \begin{cases} 1 & \text{for } c_i \leq a_{Z_i}, \\ 1 - 2\left(\frac{c_i - a_{Z_i}}{\gamma_{Z_i} - a_{Z_i}}\right)^2 & \text{for } a_{Z_i} \leq c_i \leq b_{Z_i}, \\ 2\left(\frac{c_i - a_{Z_i}}{\gamma_{Z_i} - a_{Z_i}}\right)^2 & \text{for } b_{Z_i} \leq c_i \leq \gamma_{Z_i}, \\ 0 & \text{for } \gamma_{Z_i} \leq c_i, \end{cases} \quad (5)$$

where the values b_{S_i}, b_{Z_i} are the cross-over points of the fuzzy sets defined by $\mu_{NSKIN_i}^S(c_i), \mu_{NSKIN_i}^Z(c_i)$, respectively, for $i = \{R, G, B\}$; i.e., the membership value of $c_i = b_{S_i}$ (or b_{Z_i}) that is equal to 0.5; $\{c_R, c_G, c_B\} \in [0, 255]$; $a_{S_i} = \gamma_{Z_i} = \arg \max_{c_i} H(c_i)$; $a_{Z_i} = 0.5 \gamma_{Z_i}$; and $\gamma_{S_i} = 1.5 a_{S_i}$. The results for the models of the fuzzy skin and non-skin classes are shown in Fig. 1; three classes are found for each color fuzzy set.

Now, given an input image I , for any pixel $p \in I$, its color components are fuzzified, according to the parameters defined in Eqns. (3)–(5), and using some of the three color spaces considered above. Then the

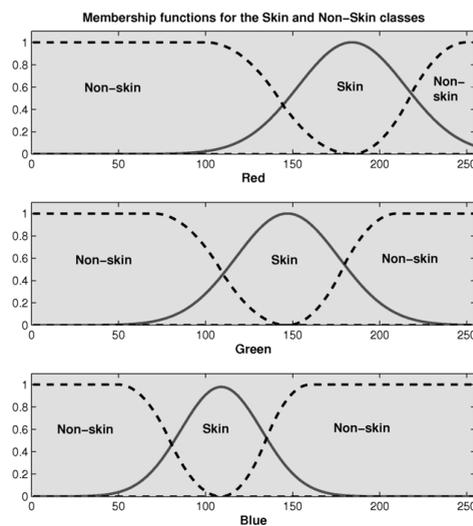


Fig. 1. Modeling the skin and non-skin classes for the RGB color space.

inferencing system processes each pixel and, by using the available knowledge in the form of IF-THEN rules, it identifies and classifies skin color pixels in the output image. This procedure results in the assignment of one output fuzzy set for each rule. A total of 12 rules for each color space were extracted for our system. The min-max inferencing technique was used, where the output membership function of each rule is clipped off at a height corresponding to the rule premises computed degree of truth. The combined fuzzy output membership function is built by combining the results of all the fuzzy rules. If an output fuzzy set is activated by more than one rule, the maximum of all activations is considered in the construction of the combined output membership function.

The Mamdani method was chosen as the defuzzification procedure, which means that the fuzzy sets obtained by applying each inference rule to the input data were joined through the add function; the output of the system was then computed as the centroid of the resulting membership function.

After the defuzzifying process, the system creates an output binary image where pixels are segmented as skin or non-skin pixels. The skin localization process is achieved by applying an 8-connected blob algorithm (Di Stefano and Bulgarelli, 1999). Then, after identifying skin regions, the algorithm detects face regions by applying the well-known Viola-Jones face detector (Viola and Jones, 2004). Combining the skin segmentation process with the Viola-Jones algorithm increases the efficiency of face detection. Once detected, faces will be normalized to the same size for the recognition process.

4. Proposal of an EBGM-based face recognition method

Before developing a connectionist model, in this section a new version of the original EBGM method described previously is presented.

4.1. Problem formulation. Let us define first a conceptual framework of the face recognition problem to formalize the method proposed in this paper. The recognition system is restricted to a set of individuals that compose the collection of subjects of interest. Let \mathbb{H} be the set of users h_i where the face recognition process is performed:

$$\mathbb{H} = \{h_1, h_2, \dots, h_n\}. \quad (6)$$

Let \mathcal{F} be an ideal facial recognition function. Let I and γ be an input image containing a face and the set of all the images in a database, respectively. Therefore, function $\mathcal{F} : \gamma \rightarrow \mathbb{H} \cup \{\emptyset\}$ is able to identify, $\forall I \in \gamma$, if any input subject of the set \mathbb{H} is found or not in image I :

$$\mathcal{F}(I) = \begin{cases} h_i, & \text{if } I \text{ has a face of person } h_i \in \mathbb{H}, \\ \emptyset, & \text{otherwise.} \end{cases} \quad (7)$$

Artificial face recognition methods try to implement the above ideal function using computer algorithms to approximate its behavior. In a general case, the main goal is to find a function that provides a distance value. This distance measures the degree of similarity between a new input face to be identified using some recognition method and the training set of faces previously learned by the system. This similarity function is calculated according to a distance function S :

$$S : \gamma \times \mathbb{H} \rightarrow \mathbb{R}^+ \cup 0. \quad (8)$$

Thus, the following equation defines the approximation function to the ideal recognition function $\widehat{\mathcal{F}} : \gamma \rightarrow \mathbb{H} \cup \{\emptyset\}$, so that $\forall I \in \gamma$:

$$\widehat{\mathcal{F}}(I) = \begin{cases} h_i, & \text{if } S(I, \mathcal{F}(I)) < \tau, \\ \emptyset, & \text{if } S(I, \mathcal{F}(I)) \geq \tau, \end{cases} \quad (9)$$

where h_i represents a certain user in \mathbb{H} and τ is an acceptance threshold to be determined.

4.2. Recognition method. From the formulation above, this work proposes an implementation of function \mathcal{F} in Eqn. (9) based on a modified version of the EBGM method.

Therefore, faces are represented using a facial graph that includes both geometric and textural information. The facial graph is defined as a pair $\{V, A\}$, where V refers to the set of vertices or nodes and A to the set of edges. Each vertex corresponds to a fiducial point and encodes the corresponding vector of jets and its location, that is, $V_i = \{J_i, P_i(x, y)\}$. Each edge A_{ij} encodes information on the distance and angle between the two nodes (i, j) it connects, so that $A_{ij} = \{d_{ij}, \theta_{ij}\}$.

For each node, a 2-dimensional histogram $hist_i$ is constructed. In this histogram, the information about the distance $D = \{d_{i1}, d_{i2}, \dots, d_{in}\}$ and the angle $\Theta = \{\theta_{i1}, \theta_{i2}, \dots, \theta_{in}\}$ from node i to the other nodes in the graph will be stored. Consequently, the histogram $hist_i$ consists of k bins corresponding to x distance-intervals $\times y$ angle-intervals. Thus, the k bins in histogram $hist_i$ are uniformly constructed in a log-polar space. Each pair $(\log(d_{ij}), \theta_{ij})$ increases the corresponding histogram bin. The procedure to obtain the fiducial points and build the facial graph is outlined as Algorithm 1.

A graphical example of how the fiducial points are extracted can be found in Fig. 2.

From the fiducial points of the face obtained by the previous algorithm, the facial graph is configured.

Algorithm 1. Obtaining the fiducial points from faces.

Step 1. Pre-process images: Skin face detection, image size normalization and grayscale conversion.

Step 2. Apply an edge detector. In this work, the well-known Canny edge detector (Canny, 1986) is used.

Step 3. Create a grid of $W_x \times W_y$ points, where nodes are uniformly distributed.

Step 4. Adjust each node's position to the nearest point in the edges obtained in Step 2.

Step 5. Calculate the distances and angles from each final node to the rest of nodes.

Figure 3 shows visually how the graph is adjusted from the initial homogeneous grid according to Algorithm 1.

A Gabor jet J is now constructed. Following Wiskott's approach (Wiskott et al., 1997), a vector of 40 complex components will be calculated. A jet J is then obtained considering the magnitude parts only. The position of each of the nodes in the two facial graphs to be compared is known, as each vertex V_i encodes this information: $V_1 = \{J_1, P\}$, $V_2 = \{J_2, Q\}$, where $P = \{p_1, p_2, \dots, p_n\}$ and $Q = \{q_1, q_2, \dots, q_n\}$ are the vectors with the positions of each of the fiducial points for both faces.

As a consequence, as in the original EBGM method, in order to match two facial graphs, $G_1 = \{V_1, A_1\}$ and $G_2 = \{V_2, A_2\}$, both geometric and texture information will be used. Thus, three similarity functions are proposed

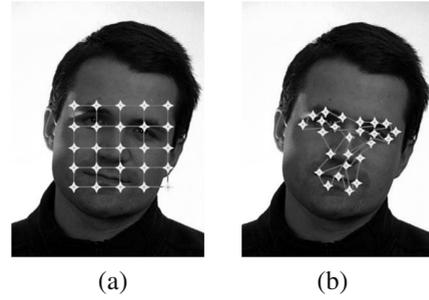


Fig. 3. Facial graph construction for a 5×5 grid: initial position (a), final position of nodes (b).

in this work: the *match cost* (MC) function, the *norm vector* (NV) function and the *Gabor feature match* (GFM) function. These functions are based on the ones proposed by González-Jiménez and Alba-Castro (2007).

Regarding the geometric information, our goal is to compute the distance between the two graphs as a sum of the matching errors between corresponding fiducial points (MC), together with a term measuring the magnitude of the aligning transformation (NV).

Taking into account the histograms previously computed with the geometric information of the nodes, it is natural to use the χ^2 test statistic in order to calculate the distance between two facial graphs. Consequently, MC is calculated by adding the matching costs for each node in the input facial graph G_1 with its corresponding node in the stored facial graph G_2 :

$$MC(G_1, G_2) = MC(P, Q) = \frac{\sum_{i=1}^n \sum_k \frac{[hist_{p_i}(k) - hist_{q_i}(k)]^2}{hist_{p_i}(k) + hist_{q_i}(k)}}{\|P\| \cdot \|Q\|}, \quad (10)$$

where $\|P\|, \|Q\|$ are the norms of vectors P and Q .

The second function, NV, is calculated by adding the norm of the vector of differences among the matched nodes:

$$NV(G_1, G_2) = NV(P, Q) = \sum_{i=1}^n \|\vec{p_i c_P} - \vec{q_i c_Q}\|, \quad (11)$$

where

$$c_P = \sum_{i=1}^n p_i, \quad c_Q = \sum_{i=1}^n q_i.$$

The texture information given by the Gabor jets from each node will be used to define the third similarity function, GFM. Thus, for each node $p_i \in P$, a jet J_{p_i} is calculated. Let R contain the Gabor jets of all the nodes in a facial graph, $R = \{J_{p_1}, J_{p_2}, \dots, J_{p_n}\}$. The function

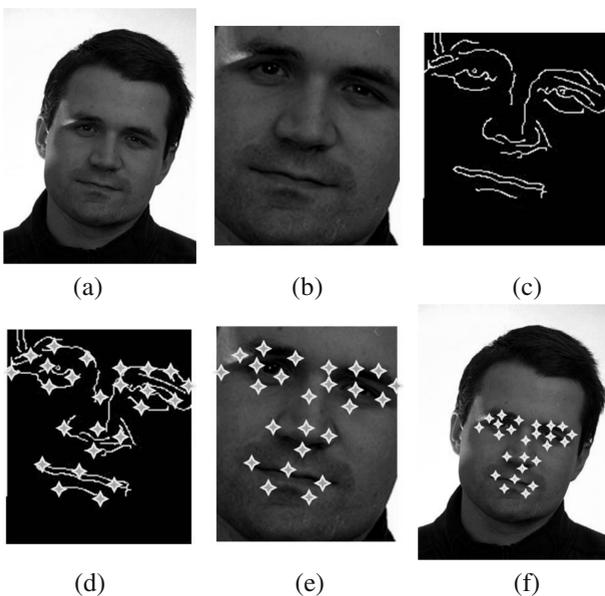


Fig. 2. Process of obtaining the fiducial points: original image (a), detected face (b), canny edge detector (c), fiducial points over edges (d), fiducial points over detected face (e), fiducial points over the original image (f).

GFM between two facial graphs is calculated as follows:

$$\text{GFM}(G_1, G_2) = \text{GFM}(R_1, R_2) = \frac{1}{n} \sum_{i=1}^n \langle R_{1i}, R_{2i} \rangle, \quad (12)$$

where $\langle R_{1i}, R_{2i} \rangle$ is the normalized dot product between the i -th jet in R_1 and the i -th jet in R_2 . As mentioned before, only the magnitude of the Gabor coefficients in the jets is considered.

Finally, from the expressions defined in Eqns. (10)–(12), the final similarity function (called the *global distortion* (GD) function) is defined by combining the results:

$$\begin{aligned} \text{GD}(G_1, G_2) &= \lambda_1 \text{MC}(G_1, G_2) \\ &+ \lambda_2 \text{NV}(G_1, G_2) + \lambda_3 \text{GFM}(G_1, G_2), \end{aligned} \quad (13)$$

where $\lambda_1, \lambda_2, \lambda_3$ are coefficients to be obtained experimentally and $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

The three functions in GD are normalized to the range $[0, 1]$. This normalization is performed with the maximum values for each component function using the distance between the input facial graph and the facial graphs stored in the database. Subsequently, the final GD function will be

$$\begin{aligned} \text{GD}(G_1, G_2) &= \frac{\lambda_1 \text{MC}(G_1, G_2)}{\max(\text{MC}(G_1, G_2))} \\ &+ \frac{\lambda_2 \text{NV}(G_1, G_2)}{\max(\text{NV}(G_1, G_2))} \\ &+ \frac{\lambda_3 \text{GFM}(G_1, G_2)}{\max(\text{GFM}(G_1, G_2))}. \end{aligned} \quad (14)$$

Consequently, facial images belonging to the same person will result in $\text{GD} \simeq 0$, and facial images of different people will have a GD close to 1.

5. Improving the EBGm algorithm with a neural network approach

The self-organizing map (SOM) is a neural network technique that implements a nonlinear projection from a high-dimensional space onto a low-dimensional array of neurons. That mapping tends to preserve the topological relationship of the inputs and, as a consequence, the visual image of this map depicts clusters of input information and their neighbor relationships on the map (Kohonen, 2001; Yin, 2008; Goćławski *et al.*, 2012). This kind of neural network in unsupervised learning is widely used to learn better representations of the input. The effectiveness of SOMs for recognition problems has been shown in many previous works. Next, we describe the SOM formalization for the proposed facial identification process.

An SOM is defined at any time by a collection of neurons, their position on the map and the weight of each neuron. The neurons are connected to adjacent neurons by a neighborhood relation. This fact sets up the topology or the structure of the map. The topological configurations of neurons are generally rectangular or hexagonal grids (Yin, 2008). For an SOM of M neurons, the set $W = \{w_1, w_2, \dots, w_M\}$ stores the weight information, where w_i is the weight vector associated to neuron i and has the same dimension as the input. The set W evolves according to the self-organizing map algorithm. The neurons positions, defined by their weight vector, configure a topological mapping of the input space.

Let $X \in \mathbb{R}^k$ be the input vector of the SOM. When applied to EBGm, this vector consists of the Gabor jets extracted from the face graph. Let $\Psi : \gamma \rightarrow \mathbb{R}^k$ be the function that obtains the geometrical and textural features of the face to construct vector X . Then

$$\forall I \in \gamma, \Psi(I) = X \in \mathbb{R}^k. \quad (15)$$

The classification of a new input face is obtained when running the SOM algorithm on the feature vectors previously calculated by using Ψ on a face image database. A clustering process can be applied prior to the classification process (Costa, 2010). However, the representation of clusters in a 2D region is usually not a simple problem, since the input data are usually high dimensional. Let $G_\Psi = \{g_1, g_2, \dots, g_M\}$ be the set of groups obtained when the clustering process is performed. Thus,

$$\forall I \in \gamma, \text{SOM}(\Psi(I)) = X \in G_\Psi, \quad (16)$$

where the SOM function is the one that classifies an input feature vector in the map as one of the groups generated by the clustering process.

Consequently, the self-organizing map builds the facial graph database from the training image dataset. Specifically, a two-dimensional $N_x \times N_y$ SOM neural network was used. The number of neurons of the map was determined experimentally in order to establish the minimum size that maximizes the efficiency in the identification.

To analyze and extract the features from each image, the facial graph obtained from our modified proposal of EBGm was taken into account. From these data, the SOM uses the extracted features, i.e., the facial graphs, as inputs. A threshold u is then considered. It consists of the maximum distance that characterizes the clusters organized into the 2D SOM in the training process. In this case, the SOM applies a classification process from the set of training facial graphs and generates the set of clusters G_Ψ , where each cluster corresponds to one of the individuals to be identified.

As described in Section 4, the facial graph includes a set of nodes corresponding to the set of facial landmarks, each of them containing both geometric and texture information. Figure 4 shows the facial graph representation of the face image of one individual as an input matrix array. Each circle represents a fiducial point. Thus, the geometric information is encoded using both the position of the i -th fiducial point ($P_i(x, y)$) and its corresponding histogram $hist_i$. Then, the texture information is represented by its Gabor jet.

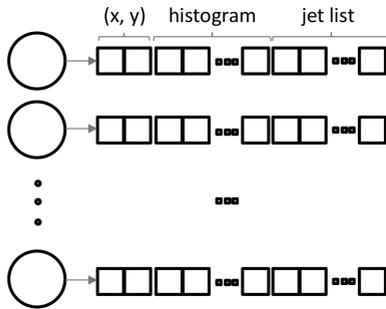


Fig. 4. SOM input data from the facial graphs.

From a practical point of view, the two-dimensional matrix in Fig. 4 can be arranged as a one-dimensional vector, just by concatenating all the rows in the matrix. Algorithm 2 shows the learning algorithm of our SOM-EBGM approach.

After this training process, a map of $N_x \times N_y$ neurons is obtained. The neurons are organized according to their similarity with respect to the input data. That is, the neurons that are near each other and in the same cluster (g_k) have information about user k , as shown in Fig. 5.

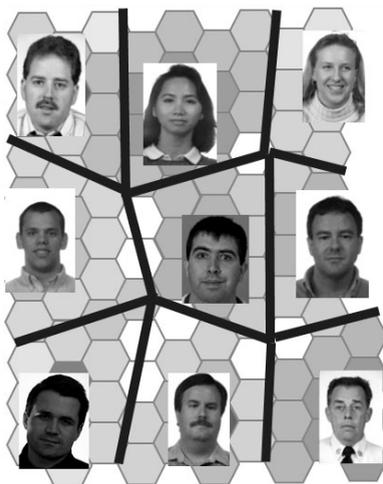


Fig. 5. Clustering process for the SOM-EBGM algorithm.

In addition, if the network is properly trained with images of different facial expressions, each group can

Algorithm 2. Learning algorithm for SOM-EBGM.

Step 1. Randomize the map’s neuron weight vector.

Step 2. Obtain the facial graph bunch by using the modified EBGM on an input face.

Step 3. Every neuron is examined to calculate the most similar weights with respect to the input data. The winning neuron c is obtained as the one with the shortest distance to the input:

$$c = \arg \min_{1 \leq i \leq N_x N_y} \{ \|w_i(t) - X(t)\| \}, \quad (17)$$

where $\| \cdot \|$ is the Euclidean distance, $X(t)$ and $w_i(t)$ are the input and the weight vector of neuron i at iteration t , respectively.

Step 4. The radius ρ of the winning neuron is then updated according to a neighborhood function:

$$\rho(t) = \rho(0) \exp\left(-\frac{t}{\beta}\right), \quad (18)$$

where $\rho(0)$ is the neighborhood radius at the first iteration and β is a constant.

Step 5. The weights of each neighboring node (found in the previous step) are adjusted to make them similar to the input vector according to this learning function:

$$w_i(t+1) = w_i(t) + \alpha(t) \exp\left(-\frac{\|r_c - r_i\|^2}{2\rho^2(t)}\right) \times (X(t) - w_i(t)), \quad (19)$$

where r is the coordinate position of the neurons on the map and $\alpha(t)$ is the learning rate, which decreases monotonically with t .

Step 6. Repeat steps 2–5 for T iterations.

have subgroups for different expressions or poses of the same individual. For example, the distribution of the winning neurons (hit histogram) shown in Fig. 6 illustrates this idea: each group concentrates most of the hits in a few neurons. Note that the other winning neurons of the group share many of the facial features of the individual and the position of its fiducial points (eyes, nose, mouth, etc.). However, they have changes that represent features corresponding to different facial expressions.

Therefore, with an accurate training and a complete set of test images, the proposed method can be robust against expression variations. This trained map will be used for face recognition.

The calculation of the winning neuron is performed by using the similarity function defined in Eqn. (14). The neuron that minimizes the result of the GD function is the

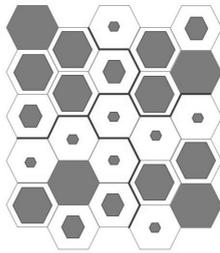


Fig. 6. Hit histogram detail.

winning neuron, and then the cluster where this neuron is located indicates who is the person to whom the face belongs.

The computational complexity of this approach can be analyzed from various points of view: on the one hand, the cost of training the SOM network and, on the other hand, the cost of recognition once the network has been trained. The time cost of network training is proportional to the number of faces that are used for training. However, a training delay does not affect the recognition complexity since that processing can be performed offline. On the other hand, regarding the computational complexity of the recognition function, the procedure consists of finding which neuron in the SOM is closest to the face to be recognized. In this process, the search is performed only with the excited neurons of the SOM. This way, the proposed method compacts the information of the user faces and, consequently, the size of the database of the registered faces is reduced considerably. Therefore, the amount of facial comparisons in the recognition process is no greater than the size of the SOM. The time cost of each comparison will be the time to compute the GD function for each pair of facial graphs. To do this, the histograms and facial graph of the input face must be first obtained using the described methods. This cost is similar to individual matching using the standard EBGm algorithm.

As the self-organizing map has been built and trained, the following section describes the results of some experiments completed for our model and compares them with other existing methods.

6. Experiments

6.1. Experimental setup. The recognition scheme was tested with three different databases:

- FERET database (Phillips *et al.*, 2000). The version used in this work is Color FERET. It contains 11,338 pictures of 994 different individuals. Images are distributed into different sets: fa, fb, fc, dup1 and dup2. Images stored in fb were taken a few moments of time after the ones in fa, so in most cases some changes in the expression of the model can be

noticed. Images in fc pictures were taken on the same day as fa pictures but with different cameras and illumination. Then, dup1 pictures were taken on different days than fa pictures but within a year. Dup2 pictures were taken at least one year later than fa pictures.

- Extended Yale B database (Georghiades *et al.*, 2001) (more precisely, the cropped Extended Yale B version (Lee *et al.*, 2005)) was taken into account. The database consists of 2,414 frontal-face images of 38 subjects captured under 64 different lighting conditions, which were manually aligned, cropped, and then resized to 168×192 pixels.
- Labeled Faces in the Wild (LFW) database (Huang *et al.*, 2007). It contains 13,233 images of 5,749 different individuals. Of these, 1,680 people have two or more images in the database. The database is designed as a benchmark for the problem of unconstrained automatic face verification with face images containing large variations in pose, age, expression, race and illumination.

A subset of 200 users extracted from the set fa of the FERET database were used to train the model, find the best value of the parameters and configure the SOM to maximize the recognition rate (tests 1, 2 and 3). Once the SOM had been configured, the recognition rate of the method was obtained by using the other sets of images and has been compared with other methods (test 4). Figure 7 shows this process.

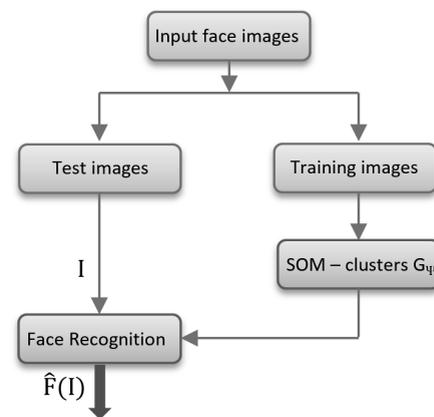


Fig. 7. SOM-EBGM recognition process.

The tests were performed using Matlab® with a 2 GHz Intel Core i5 and 8 GB memory. The training process had been run until the improvement in the recognition rate between two iterations was insignificant.

6.2. Results on face detection. In order to use our fuzzy skin detector, first the normalized histogram for

the RGB color space for the training set, $H(C) = \{f_R, f_G, f_B\}$, must be calculated. The training was performed using the set of 200 images from the Color FERET database mentioned before, extracting only the skin information, and using different ethnic groups and changing lighting conditions.

As mentioned in Section 3, after defining a skin color detection system, we will need to determine if the resulting pixels belong to a face or not. To do this, the Viola–Jones detector is used. Some examples of the results of the skin face detector are shown in Fig. 8.



Fig. 8. Results on face detection. From left to right: original image, fuzzy RGB skin detector, face detection.

A comparison of our detector with other related works is shown next. Thus, the Viola–Jones method (Viola and Jones, 2004) is selected first. Then, recent Sarkar’s proposal (Sarkar, 2012) is considered, where a skin face color detector is proposed, but the HSV color system is used instead of RGB. The results on the detection for all the three methods can be seen in Fig. 9, where some images from the FERET and LFW databases are considered.

In addition, the three methods have been compared by calculating the detection rate (DR), the false positive rate (FPR) and the false negative rate (FNR), defined as follows:

$$DR(\%) = \frac{N_S}{N_F} \times 100, \tag{20}$$

$$FPR(\%) = \frac{N_{FP}}{N_{NF}} \times 100, \tag{21}$$

$$FNR(\%) = \frac{N_{FN}}{N_F} \times 100. \tag{22}$$

In these equations, N_F is the total number of skin color pixels, N_S is the number of correctly detected skin color pixels, N_{NF} is the total number of non-skin

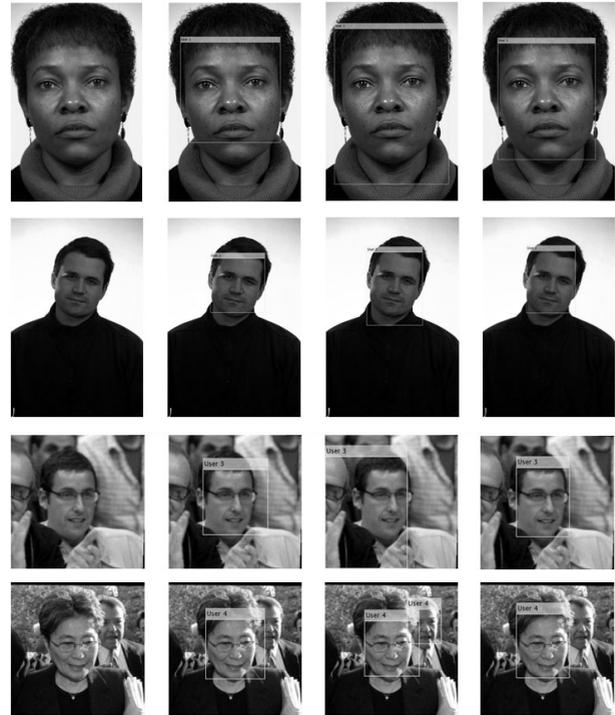


Fig. 9. Comparison of face detectors. From left to right: original image, Viola and Jones detector, Sarkar’s detector, fuzzy RGB skin detector.

color pixels, N_{FP} is the number of non-skin pixels that are detected incorrectly as skin color, and N_{FN} is the number of skin color pixels that are detected incorrectly as non-skin color pixels. Table 1 shows the results of these measures using the three methods considered.

Table 1. Comparison of some face detection methods.

Method	DR	FPR	FNR
Viola–Jones	94.8	10.7	7.9
Sarkar	87.5	18.8	13.3
Fuzzy skin	96.7	5.1	4.8

According to this comparison, our method leads to very accurate segmentation of face images and improves the results of the Viola–Jones face detector, which is the actual standard in many applications of face detection. Consequently, we believe that the developed fuzzy skin detector is a reliable face detector.

After the detection is completed, faces are rescaled to a size of 120×120 pixels and converted into grayscale for the recognition process. Note that for the cropped Extended Yale B database images only need to be resized, since faces are already detected and in grayscale.

6.3. Results on face recognition. The experimentation completed has a dual objective: firstly, the best

configuration for the SOM network, the facial graph and the weighting coefficients λ_i is found in order to achieve higher recognition rates. Afterwards, our proposal is compared with a set of related, state-of-the-art techniques.

Map test 1: Type and size. Table 2 shows the recognition rate, where M is the number of different users to be recognized in our system. The coefficients λ_i were considered as $\lambda_1 = 0.25, \lambda_2 = 0.25, \lambda_3 = 0.5$.

Table 2. Recognition rate (%): map type and size.

SOM size	Rectangular grid	Hexagonal grid
$M - 10 \times M - 10$	85.6	85.7
$M - 5 \times M - 5$	85.1	83.9
$M \times M$	92.3	95.8
$M + 5 \times M + 5$	91.1	91.5
$M + 10 \times M + 10$	90.1	91.3

The results show that the optimal size of the map corresponds to the number of individuals to be identified. Moreover, a hexagonal distribution of neurons provides slightly better results than a rectangular grid.

Test 2: Size of the facial graph. In this experiment, the relation between the size of the facial graph obtained from the SOM-EBGM method and the recognition rate is analyzed. Here λ_i were considered as $\lambda_1 = 0.25, \lambda_2 = 0.25, \lambda_3 = 0.5$ and the size of the SOM was $M \times M$. The results are shown in Table 3.

Table 3. Recognition rate (%): size of the facial graph.

Size	6×6	8×8	10×10
Recognition rate	91.3	92.1	94.4
Size	12×12	14×14	16×16
Recognition rate	95.2	93.9	90.1

From these results, it must be noticed that increasing the size of the facial graph increases the accuracy of the SOM-EBGM system until it reaches the maximum recognition rate when a 12×12 grid is used. However, adding more fiducial points to the graph reduces significantly the accuracy of our method. This is due to the fact that the correspondence between the two facial graphs to be compared is much less precise as the size of the graph increases. As a result, a 12×12 grid will be used for the rest of the experiments. If a faster system is needed, a 10×10 grid will be enough, since it reduces considerably the training cost of the SOM-EBGM method and the recognition rate only reduces 0.8%.

Test 3: Weighting coefficients λ_i of the similarity function. This test consists in systematically testing all combinations of weights and calculating the

correspondence obtained in each case. Table 4 shows only some of the most representative cases for the following configurations: (a) cases where only one of the functions is considered, (b) cases where one of the functions is excluded from the final calculation, (c) a uniform weighting case and, (d) cases where a function is weighted over the others. As mentioned before, $\lambda_1 + \lambda_2 + \lambda_3 = 1$, and the size of the SOM was $M \times M$, with a facial graph of 12×12 fiducial points.

Table 4. Recognition rate (%): selection of coefficients.

Case	Coefficients $\lambda_1, \lambda_2, \lambda_3$	Recogn. rate
(a)	$\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 1$	77.3
	$\lambda_1 = 0, \lambda_2 = 1, \lambda_3 = 0$	60.1
	$\lambda_1 = 1, \lambda_2 = 0, \lambda_3 = 0$	61.0
(b)	$\lambda_1 = 0.5, \lambda_2 = 0.5, \lambda_3 = 0$	70.3
	$\lambda_1 = 0.5, \lambda_2 = 0, \lambda_3 = 0.5$	72.9
	$\lambda_1 = 0, \lambda_2 = 0.5, \lambda_3 = 0.5$	84.3
(c)	$\lambda_1 = 1/3, \lambda_2 = 1/3, \lambda_3 = 1/3$	85.6
(d)	$\lambda_1 = 0.2, \lambda_2 = 0.7, \lambda_3 = 0.1$	83.7
	$\lambda_1 = 0.7, \lambda_2 = 0.2, \lambda_3 = 0.1$	83.5
	$\lambda_1 = 0.2, \lambda_2 = 0.2, \lambda_3 = 0.6$	89.9
	$\lambda_1 = 0.25, \lambda_2 = 0.25, \lambda_3 = 0.5$	96.5

From these results, it becomes clear that a combination of both textural and geometric information from the fiducial points gives better results, whereas taking into account only the Gabor jets (related to coefficient λ_3) or the geometric information (related to coefficients λ_1 and λ_2) does not result in a reliable recognition rate. In particular, when the global distortion (GD) function has a uniform combination of geometric features ($\lambda_1 = 0.25, \lambda_2 = 0.25$) and Gabor features ($\lambda_3 = 0.5$), the system achieves a higher accuracy. This result is consistent with our initial proposal and shows that all the three functions of similarity defined in Section 4 have great influence on the success of the face recognition process.

Test 4: Comparison with other methods. Having chosen the optimal parameters of the SOM-EBGM algorithm, a set of experiments to compare our method with other existing algorithms for face recognition must be completed. This test was performed with the trained SOM and the test sets of facial images. That is, we consider closed-set identification to classify a given face as belonging to one of M identities stored in the gallery.

Note that the LFW database is commonly used for benchmarking face verification. However, in this work we consider the closed set identification protocol defined by Best-Rowden *et al.* (2014) and Taigman *et al.* (2015). Thus, when using LFW, the gallery set includes 4,249 identities, each with only a single example, and the probe set includes 3,143 faces belonging to the same set of

identities. The performance is measured by the rank-1 identification accuracy, i.e., by a correct identification.

Firstly, three classical methods were taken into account: Wiskott's EBGM (Wiskott *et al.*, 1997) with 12×12 nodes, eigenfaces (PCA) (Turk and Pentland, 1991) with 20 eigenvectors, and Ahonen's local binary patterns (LBP) (Ahonen *et al.*, 2004) with a neighborhood of 8 points of radius 4 with respect to the central pixel in the neighborhood, as in the original paper.

In addition, there are numerous kernel-based algorithms proposed for the face recognition task (Zheng *et al.*, 2011; Gao and Fan, 2011; Gao *et al.*, 2012; Li *et al.*, 2014). For our proposals, the comparison includes the following relevant works:

- Kernel fuzzy discriminant analysis (KFDA) (Gao *et al.*, 2013a). This is a novel method for feature extraction and recognition. The KFDA method is obtained by combining the advantages of fuzzy methods in order to obtain a nonlinear model and a new learning method. A polynomial kernel $p = 6$ is selected with QR decomposition.
- Median null(S_w)-based (M-N(S_w)) method (Gao *et al.*, 2013b): This method proposes a two-stage linear discriminant analysis for face feature recognition. According to the experiments performed, this technique is more robust than the class sample average based traditional linear discriminant analysis models. We used 15 discriminant vectors when comparing it with our method.
- Patterns of oriented edge magnitudes histogram sequence (POEM-HS) (Vu and Caplier, 2012). The POEM feature descriptor is a robust and computationally inexpensive method. According to the authors, this method outperforms many other related descriptor-based techniques. For the experiments, we considered the optimal parameters defined by the authors: unsigned representation with three bins, with 10×10 pixel blocks, 7×7 pixel cells, and 6 neighbors per cell.
- Statistical local features with robust kernel representation (SLF-RKR) (Yang *et al.*, 2013). The model proposed used a multipartition max pooling technology to enhance the invariance of local features in face recognition, and a kernel model, which adopts a robust function to handle the occlusion in facial images. The number of histogram bins selected for each sub-block is 16 and the histogram intersection kernel is used as the kernel function.
- Linear discriminant regression classification (LDRC) (Huang and Yang, 2013). This approach embeds discriminant analysis into the linear regression

classification algorithm for achieving an optimal projection matrix that leads to a high discriminatory ability for classification.

- Discriminant locality preserving projections (DLPP) based on L1-norm maximization (Zhong *et al.*, 2014): this work aims at learning a set of local optimal projection vectors by maximizing the ratio of the L1-norm-based locality preserving between-class dispersion and the L1-norm-based locality preserving within-class dispersion. For the experiments, the updating parameter β of DLPP-L1 is set to 0.01.
- Multilinear sparse principal component analysis (MSPCA) (Lai *et al.*, 2014). The optimal multilinear sparse principal component vectors are obtained from an iterative algorithm using singular value decomposition on tensor data instead of vectors. The authors claim that this algorithm may outperform the existing PCA-based subspace learning algorithms. The dimensionality selected for MSPCA in this case is 17×17 .

Finally, some recent variants of EBGM were considered as well. In particular, both the neural network face recognition system using Gabor filter coefficients proposed by Khatun and Bhuiyan (2011) (NN-EBGM) and a skin detector combined with EBGM (Sarkar, 2012) (SD-EBGM) were taken into account. 40 Gabor filters were used in both works instead of the number proposed by the authors. The results for all the algorithms involved are shown in Table 5.

Table 5. Comparison between methods. The performance is measured in each database by means of the rank-1 accuracy (%) for the closed set identification protocol.

Algorithm	FERET	Yale B	LFW
PCA	66.4	78.3	45.6
LBP	74.5	84.7	53.5
EBGM	80.9	82.2	51.1
KFDA	52.1	68.5	47.1
M-N(S_w)	66.8	72.7	58.3
POEM-HS	86.9	94.0	61.6
SLF-RKR	88.5	96.3	61.0
LDRC	89.0	99.0	60.5
DLPP	74.9	90.7	56.8
MSPCA	83.0	91.8	54.6
NN-EBGM	75.3	90.9	55.2
SD-EBGM	88.0	86.4	57.7
SOM-EBGM	90.2	99.2	62.9

From these results, our method achieves the best results for the FERET database over 90% of correct recognitions. It is about 1–2% better than other recent state-of-the art methods and more than 15% better than the previous neural network system applied to EBGM.

Regarding the Yale B database, our method achieves over 99% of correct identifications the best total performance compared with the other methods. This fact shows that our proposal is robust against illumination variations, such as the ones found in the Yale B database. Finally, for the LFW database, the overall results are not as accurate as in the other two databases, because of the great difficulty in achieving good identification rates when using a challenging unconstrained database like LFW. Nevertheless, our results outperform the other algorithms, as in the other databases.

As a conclusion, it is proved that our SOM-EBGM proposal can be used as an alternative of the original EBGM method for face recognition at a lower computational cost due to its ability of compacting face information.

7. Discussion and conclusions

This work presented a modified version of EBGM that makes use of an SOM to cluster and classify faces. From the results of the experiments, it was proved that applying connectionist techniques to build the face database from the facial graph improved the results compared with many other recent proposals.

Applying an SOM achieves a much more compact database than traditional methods do. Thus, in EBGM-based algorithms, the bunch graph for the matching process contains information of each graph from all the input images in the training set. Therefore, the more images in the training set, the larger the size of the graph database. In our approach, the information is compacted. Thus, the map size determines the number of graphs to be used in the matching phase. Consequently, the training can be applied to an extensive set of images without affecting the amount of memory required to store the whole database during the execution of the recognition algorithm.

Our future works are aimed at two goals. First, the experiments are being performed using open-set identification to verify the suitability of our method for real-world applications. On the other hand, the SOM framework proposed is being combined with other recent recognition methods to investigate more potential uses of our approach.

References

- Ahonen, T., Hadid, A. and Pietikainen, M. (2004). Face recognition with local binary patterns, *Proceedings of the 8th European Conference on Computer Vision, ECCV 2004, Prague, Czech Republic*, pp. 469–481.
- Azorín-López, J., Saval-Calvo, M., Fuster-Guilló, A. and Oliver-Albert, A. (2014). A predictive model for recognizing human behaviour based on trajectory representation, *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2014, Beijing, China*, pp. 1494–1501.
- Baltrusaitis, T., Robinson, P. and Morency, L. (2012). 3D constrained local model for rigid and non-rigid facial tracking, *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Providence, RI, USA*, pp. 2610–2617.
- Bańka, S., Dworak, P. and Jaroszewski, K. (2014). Design of a multivariable neural controller for control of a nonlinear MIMO plant, *International Journal of Applied Mathematics and Computer Science* **24**(2): 357–369, DOI: 10.2478/amcs-2014-0027.
- Belhumeur, P.N., Jacobs, D.W., Kriegman, D. and Kumar, N. (2011). Localizing parts of faces using a consensus of exemplars, *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Colorado Springs, CO, USA*, pp. 545–552.
- Best-Rowden, L., Han, H., Otto, C., Klare, B.F. and Jain, A.K. (2014). Unconstrained face recognition: Identifying a person of interest from a media collection, *IEEE Transactions on Information Forensics and Security* **9**(12): 2144–2157.
- Canny, J. (1986). A computational approach to edge detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**(6): 679–698.
- Chen, X., Zhang, C., Dong, F. and Zhou, Z. (2013). Parallelization of elastic bunch graph matching (EBGM) algorithm for fast face recognition, *Proceedings of the 2013 IEEE China Summit & International Conference on Signal and Information Processing, Beijing, China*, pp. 201–205.
- Costa, J.A.F. (2010). Clustering and visualizing SOM results, *Proceedings of the 11th International Conference on Intelligent Data Engineering and Automated Learning, IDEAL 2010, Paisley, UK*, pp. 334–343.
- Dantone, M., Gall, J., Fanelli, G. and Van Gool, L. (2012). Real-time facial feature detection using conditional regression forests, *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Providence, RI, USA*, pp. 2578–2585.
- Di Stefano, L. and Bulgarelli, A. (1999). A simple and efficient connected components labeling algorithm, *Proceedings of the International Conference on Image Analysis and Processing, ICIP 1999, Kobe, Japan*, pp. 322–327.
- En-Naimani, Z., Lazaar, M. and Ettaouil, M. (2014). Hybrid system of optimal self organizing maps and hidden Markov model for Arabic digits recognition, *WSEAS Transactions on Systems* **13**(60): 606–616.
- Espí, R., Pujol, F.A., Mora, H. and Mora, J. (2008). Development of a distributed facial recognition system based on graph-matching, *Proceedings of the International Symposium on Distributed Computing and Artificial Intelligence, DCAI 2008, Salamanca, Spain*, pp. 498–502.
- Gao, J. and Fan, L. (2011). Kernel-based weighted discriminant analysis with QR decomposition and its application face recognition, *WSEAS Transactions on Mathematics* **10**(10): 358–367.

- Gao, J., Fan, L., Li, L. and Xu, L. (2013a). A practical application of kernel-based fuzzy discriminant analysis, *International Journal of Applied Mathematics and Computer Science* **23**(4): 887–903, DOI: 10.2478/amcs-2013-0066.
- Gao, J., Fan, L. and Xu, L. (2012). Solving the face recognition problem using QR factorization, *WSEAS Transactions on Mathematics* **11**(8): 728–737.
- Gao, J., Fan, L. and Xu, L. (2013b). Median null(sw)-based method for face feature recognition, *Applied Mathematics and Computation* **219**(12): 6410–6419.
- Georgiades, A.S., Belhumeur, P.N. and Kriegman, D.J. (2001). From few to many: Illumination cone models for face recognition under variable lighting and pose, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**(6): 643–660.
- Gocławski, J., Sekulska-Nalewajko, J. and Kuźniak, E. (2012). Neural network segmentation of images from stained cucurbits leaves with colour symptoms of biotic and abiotic stresses, *International Journal of Applied Mathematics and Computer Science* **22**(3): 669–684, DOI: 10.2478/v10006-012-0050-5.
- González-Jiménez, D. and Alba-Castro, J.L. (2007). Shape-driven Gabor jets for face description and authentication, *IEEE Transactions on Information Forensics and Security* **2**(4): 769–780.
- Guo, G., Mu, G. and Ricanek, K. (2010). Cross-age face recognition on a very large database: The performance versus age intervals and improvement using soft biometric traits, *Proceedings of the 20th International Conference on Pattern Recognition, ICPR, Istanbul, Turkey*, pp. 3392–3395.
- Hsu, R.-L., Abdel-Mottaleb, M. and Jain, A.K. (2002). Face detection in color images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(5): 696–706.
- Huang, G. B., Ramesh, M., Berg, T. and Learned-Miller, E. (2007). Labeled faces in the wild: A database for studying face recognition in unconstrained environments, *Technical Report 07-49*, University of Massachusetts, Amherst, MA.
- Huang, S.M. and Yang, J.F. (2013). Linear discriminant regression classification for face recognition, *Signal Processing Letters* **20**(1): 91–94.
- Jin, X., Tan, X. and Zhou, L. (2013). 3D constrained local model for rigid and non-rigid facial tracking, *Proceedings of the 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, FG13, Shanghai, China*, pp. 1–8.
- Kayarvizhy, N., Kanmani, S. and Uthariaraj, R. (2014). ANN models optimized using swarm intelligence algorithms, *WSEAS Transactions on Computers* **13**(45): 501–519.
- Khatun, A. and Bhuiyan, M. (2011). Neural network based face recognition with Gabor filters, *International Journal of Computer Science and Network Security* **11**(1): 71–74.
- Kohonen, T. (2001). *Self-Organising Maps, 3rd Edition*, Springer-Verlag, Berlin/Heidelberg.
- Kotropoulos, C. and Pitas, I. (1997). Rule-based face detection in frontal views, *IEEE International Conference on Acoustics, Speech, and Signal Processing, Munich, Germany*, pp. 2537–2540.
- Kumar, A. and Kumar, S. (2015). An improved neural network based approach for identification of self & non-self processes, *WSEAS Transactions on Computers* **14**(28): 272–286.
- Lai, Z., Xu, Y., Chen, Q., Yang, J. and Zhang, D. (2014). Multilinear sparse principal component analysis, *IEEE Transactions on Neural Networks and Learning Systems* **25**(10): 1942–1950.
- Lee, K.-C., Ho, J. and Kriegman, D.J. (2005). Acquiring linear subspaces for face recognition under variable lighting, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(5): 684–698.
- Li, J.-B., Chu, S.-C. and Pan, J.-S. (2014). *Kernel Learning Algorithms for Face Recognition*, Springer, New York, NY.
- Loderer, M., Pavlovicova, J., Feder, M. and Oravec, M. (2014). Data dimension reduction in training strategy for face recognition system, *Proceedings of the IEEE 2014 International Conference on Systems, Signals and Image Processing, IWSSIP, Dubrovnik, Croatia*, pp. 263–266.
- Mitra, S., Parua, S., Das, A. and Mazumdar, D. (2011). A novel datamining approach for performance improvement of EBGM based face recognition system to handle large database, *Proceedings of the 1st International Conference on Computer Science and Information Technology, CCSIT 2011, Bangalore, India*, pp. 532–541.
- Monzo, D., Albiol, A. and Mossi, J.M. (2010). A comparative study of facial landmark localization methods for face recognition using hog descriptors, *Proceedings of the 20th International Conference on Pattern Recognition, ICPR, Istanbul, Turkey*, pp. 1330–1333.
- Murthy, C.A. and Pal, S.K. (1990). Fuzzy thresholding: Mathematical framework, bound functions and weighted moving average technique, *Pattern Recognition Letters* **11**(3): 197–206.
- Phillips, P.J., Moon, H., Rizvi, S.A. and Rauss, P.J. (2000). The FERET evaluation methodology for face recognition algorithms, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(10): 1090–1104.
- Piegat, A. (2005). A new definition of the fuzzy set, *International Journal of Applied Mathematics and Computer Science* **15**(1): 125–140.
- Pujol, F., Espí, R., Mora, H. and Sánchez, J. (2008). A fuzzy approach to skin color detection, *Proceedings of the 7th Mexican International Conference on Artificial Intelligence, MICAI 2008, Atizapan de Zaragoza, Mexico*, pp. 532–542.
- Rattani, A., Agarwal, N., Mehrotra, H. and Gupta, P. (2006). An efficient fusion-based classifier, *Proceedings of the Workshop on Computer Vision, Graphics and Image Processing, WCVGIP, Hyderabad, India*, pp. 104–109.

- Sarkar, S. (2012). Skin segmentation based elastic bunch graph matching for efficient multiple face recognition, *Proceedings of the 2nd International Conference on Computer Science, Engineering and Applications, ICCSEA 2012, New Delhi, India*, pp. 31–40.
- Shen, L. and Bai, L. (2006). A review on Gabor wavelets for face recognition, *Pattern Analysis and Applications* **9**(2): 273–292.
- Shin, Y.J. and Park, C.H. (2011). Analysis of correlation based dimension reduction methods, *International Journal of Applied Mathematics and Computer Science* **21**(3): 549–558, DOI: 10.2478/v10006-011-0043-9.
- Taigman, Y., Yang, M., Ranzato, M.A., and Wolf, L. (2015). Web-scale training for face identification, *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA*, pp. 2746–2754.
- Tran, H.L., Pham, V.N. and Vuong, H.N. (2014). Multiple neural network integration using a binary decision tree to improve the ECG signal recognition accuracy, *International Journal of Applied Mathematics and Computer Science* **24**(3): 647–655, DOI: 10.2478/amcs-2014-0047.
- Turk, M. and Pentland, A. (1991). Eigenfaces for recognition, *Journal of Cognitive Neuroscience* **3**(1): 71–86.
- Viola, P. and Jones, M.J. (2004). Robust real-time face detection, *International Journal of Computer Vision* **57**(2): 137–154.
- Vu, N.S. and Caplier, A. (2012). Enhanced patterns of oriented edge magnitudes for face recognition and image matching, *IEEE Transactions on Image Processing* **21**(3): 1352–1365.
- Wiskott, L., Fellous, J.-M., Kruger, N. and von der Malsburg, C. (1997). Face recognition by elastic bunch graph matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(7): 775–789.
- Yang, J., Fu, Z. and Tan, T. (2004). Skin color detection using multiple cues, *Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, Cambridge, UK*, pp. 632–635.
- Yang, M., Zhang, L., Shiu, S.K. and Zhang, D. (2013). Robust kernel representation with statistical local features for face recognition, *IEEE Transactions on Neural Networks and Learning Systems* **24**(6): 900–912.
- Yin, H. (2008). The self-organizing maps: Background, theories, extensions and applications, in J. Fulcher and L.C.J. Lakhmi (Eds.), *Computational Intelligence: A Compendium*, Springer-Verlag, Berlin, pp. 715–762.
- Zadeh, L.A. (1965). Fuzzy sets, *Information and Control* **8**(3): 338–353.
- Zheng, W., Lai, J., Xie, X., Liang, Y., Yuen, P. C. and Zou, Y. (2011). Kernel methods for facial image preprocessing, in P. Wang (Ed.), *Pattern Recognition, Machine Intelligence and Biometrics*, Springer, Berlin/Heidelberg, pp. 389–409.
- Zhong, F., Zhang, J. and Li, D. (2014). Discriminant locality preserving projections based on L1-norm maximization, *IEEE Transactions on Neural Networks and Learning Systems* **25**(11): 2065–2074.

Francisco A. Pujol received his B.Sc. degree in telecommunications engineering at the Polytechnic University of Valencia (Spain) in 1998, and the Ph.D. degree in computer science at the University of Alicante (Spain) in 2001. He is currently an associate professor at the Computer Technology Department of the University of Alicante. His research interests focus on biometrics, pattern recognition, computer vision and computer parallel architectures. He has several published papers on face recognition, computer parallel architectures for computer vision and robotics.

Higinio Mora received a B.Sc. degree in computer science engineering and a B.Sc. degree in business studies at the University of Alicante, Spain, in 1996 and 1997, respectively. He received a Ph.D degree in computer science from the University of Alicante in 2003. Since 2002 he has been a member of the faculty of the Department of Computer Science Technology at the same university, where he is currently an associate professor. His present areas of research interest include computer arithmetic and the design of floating points units and approximation algorithms related to VLSI design.

José A. Girona-Selva received the B.Sc. degree in computer science engineering from the University of Alicante, Spain, in 1996. Currently, he works in the Computer Department at the General Hospital of Alicante and is a Ph.D. student of computer science at the University of Alicante. His research areas include artificial neural networks and facial recognition.

Received: 17 May 2015

Revised: 31 October 2015

Re-revised: 25 January 2016

Accepted: 15 February 2016