A 3D geometric approach to face detection and facial expression recognition

Matteo Gaeta$^{1,2}$
Gerardo Iovane$^{1,3,*}$
Enver Sangineto$^{1,4}$

$^1$Dipartimento di Ingegneria dell’Informazione e Matematica Applicata (DIIMA)
Università degli Studi di Salerno
via Ponte Don Melillo
84084, Salerno
Italy

$^2$Centro di Ricerca in Matematica Pura e Applicata (CRMPA)
c/o DIIMA, Università degli Studi di Salerno
via Ponte Don Melillo
84084, Salerno
Italy

$^3$Centro di Eccellenza su Sistemi e Metodi per l’Apprendimento e
la Conoscenza (CEMSAC)
Università degli Studi di Salerno
via Ponte Don Melillo
84084, Salerno
Italy

$^4$Centro di Ricerca in Matematica Pura e Applicata (CRMPA)
Sezione “Roma Tre”, c/o DIA
Università degli Studi “Roma Tre”
via della Vasca Navale 79
00146 Roma
Italy

*E-mail: iovane@diima.unisa.it

Journal of Discrete Mathematical Sciences & Cryptography
Vol. 9 (2006), No. 1, pp. 39–53
© Taru Publications
Face detection and facial expression recognition are research areas with important application possibilities. Although the two problems are usually dealt with different approaches, we show in this paper how the same recognition process can be used to recognize both a generic “class-face” in a given, possibly complex image, and a specific facial expression.

The approach we propose is based on two steps. In the former we use alignment techniques in order to overlap the 3D representations of the main face components with the 2D image elements. In the latter we compare the candidate groups of localized components with a set of structural models, each of which representing a facial expression. Expression-independent face detection is achieved using the same approach with a model built generalizing over a set of face examples with different expressions.

**Keywords**: Face detection, facial expression recognition.

### 1. Motivations and goals

The rapidly growing interest in application fields such as automatic video surveillance, “intelligent” Human-Computer Interaction (HCI), automatic or semi-automatic video summarization/annotation, etc., has brought in the last ten years to an increasing effort of the Computer Vision and Pattern Recognition communities in studying face and face expression recognition problems for digital images.

The task of identifying the face of a given individual in an image is usually split in two main steps: the localization of candidate regions in the image in which likely there is a face (the *face detection* problem) and the comparison of such regions with a data base of known faces (*face recognition*). The former phase is necessary in order to select the patch of the image containing a face which is then used by the latter phase for identification issues. Moreover, face detection has other important motivations in itself. For instance, detecting human people in video sequences is usually performed by detecting faces (when visible) [11]. As observed by Yang et al. [22], face detection is an interesting challenge for Computer Vision because it well summarizes in a real domain all the main issues related to one of the most important and difficult problem of Computer Vision: *object recognition*. In fact, the “objects” of the face class have a great within-class variability, a face is a non-rigid object, there are segmentation problems related to the unknown position(s) of the (possibly
present) face(s) inside the image and so on [11, 22].

In this paper we deal with face detection by proposing a geometric approach for the localization of a face prototype in a digital image. Using a geometric description of a face model allows us to efficiently deal with one of the most important problems in face detection (as well as in face recognition): the treatment of the translation, scale and rotation variability of the appearance of a face. In fact, a face can be represented in an image with respect to different views: frontal, semi-frontal, profile and so on and usually this is dealt with by using different models, each of which approximates a possible view (viewer centered representation). Conversely, we use a single three dimensional model (object centered representation) which is matched with the image data regardless of the face position.

Other important problems which make difficult face detection are: a possible cluttered background, partial occlusion of the face(s), light conditions variability, changing of the face appearance due to different emotive expressions. The last problem is related to facial expression recognition, with important possible applications, for instance, in HCI. Facial expression recognition concerns the recognition of the affective states (happiness, sadness, disgust, fear, etc.) of an individual.

Both face detection and facial expression recognition (as well as face recognition) have been dealt with in a large amount of works, usually classified in two main categories: those focusing on a statistical characterization of the face pattern as a whole, also called holistic approaches, and those searching for structural relationships among the face components (e.g., see [22], Section 2.5 or [5, 7]). Generally speaking, the first category of approaches works as follows. First of all, the input image is scanned using a rectangular image window $W$ located in the image in different positions and with different sizes. The aim of this step is to select candidate portions of the image in which the next analysis phases will be performed. Each image window $W$ is then represented by a vector obtained by the concatenation of the window’s pixels, e.g., row by row. If $W$ is a $n \times m$ rectangle, it is represented by means of the vector $v$ of size $N = n \cdot m$ containing the gray scale (or the color) values of every pixel of $W$. The vector $v$ is a point in an $N$-dimensional space and then statistical pattern recognition techniques can be applied in order to classify punctual representations of patterns. Usually $v$ is projected into a subspace of reduced dimensions $N'$ ($N' \ll N$), obtained by means of linear transformations of the original
space which maintains the most discriminant information with respect to a given training set of images while reducing the space dimensionality. Typical adopted transformations are: Principal Component Analysis (PCA) [8] and the Fisher’s Linear Discriminant Analysis (FLD) [4], while the final space is usually called face space. If $v'$ is the projection of $v$ into the face space, in the final step the system decides if $v'$ is an instance of either the “face” or the “non-face” class depending on the distance between $v'$ and the clusters of points representing, respectively, the “face” and the “non-face” classes. Such clusters have been built during the training phase using a set of image examples. Typical adopted distances are the Euclidean or the Mahalanobis distance, while clustering of points representing face examples and non-face examples is usually realized with a supervised Machine Learning approach in which the training set is split in two sets of positive and negative examples of face images.

An example of this approach is the work of Turk and Pentland [19], who use the PCA as a compression technique. In the (off-line) training phase, the points of the face space representing the examples of face images of the training set are clustered together. The same is done for the images with no face. On-line the vector $v'$ obtained by the current image window is compared with both the face and the non-face clusters and is classified choosing the minimum distance. Yang et al. in [21] use the FLD as data compression technique showing better results with respect to the PCA. Sung and Poggio [18] use a set of Gaussian functions to represent 6 clusters for the “face” class and other 6 clusters for the “non-face” class. The image window is compared with all the 12 clusters using both the Euclidean and the Mahalanobis distance. The 24-dimensional vector $d$ so obtained is finally classified using a Multilayer Perceptron. In [16] Rowley et al. use a multiple classifier instead of a single neural network while Feraud et al. propose an autoassociative neural network with 5 layers [10]. Ma and Khorasani [15] use a feed forward neural network and Bartlett et al. [3] a Support Vector Machine to classify different facial emotions.

Approaches based on structural relationships among face components usually consist of two main phases. In the first phase the most important face components (nose, eyes, eye-brows, mouth, etc), are searched for in the input image. Hence, in this case a single component (e.g., the nose) rather than the whole face is searched for in the image
using pattern classification techniques. In the second step, the components reliably identified are grouped together computing the likelihood they belong to a face by means of statistical information about typical distances among the face elements. Statistical information is collected off-line in the training phase. Example of this category are the work of Schneiderman and Kanade who represent spatial relationships among face components using a set of histograms [17] or the work of Leung et al. [14] who use a graph representation. Cardaci et al. [7] use a structural approach and a graph representation for facial expression recognition.

A common problem of both global and structural approaches to face detection and expression recognition is the need to deal with different appearances of a face due to different viewpoints. Almost all the existing approaches treat this problem using a set of different models each of which approximates a different view of the face with a viewer centered approach. In this article we propose an object centered representation of a face model based on a 3D description of the face components’ prototypes which allows the system to uniformly deal with both face detection and facial expression recognition in a view point-independent fashion. We use alignment techniques in order to efficiently and accurately localize face components in a given image. Alignment techniques [2, 9, 13] are used in Computer Vision in order to recognize rigid objects. Although a face is a non-rigid (deformable) object, most of its components can be regarded as they were rigid by using suitable prototypes which approximate their typical shape.

Once the face components have been (possibly) localized in the image we use the parameters’ values of the geometric transformations resulting by the alignment phase in order to represent the mutual spatial relations among different components. Spatial relations are compared with statistical values collected in the training phase and represented as edge labels of a graph describing the whole face \( G_F \). Thus, we use a structural approach with an object centered representation of the face components. Moreover, once a face has been detected and localized in a given image, spatial relationships among the recognized components are used in order to recognize the facial expression type. To this aim, spatial relationships among the face components of the input image are compared with the graphs \( G_{E_1}, \ldots, G_{E_k} \) representing \( k \) types of different expressions. Each \( G_{E_i} (i \leq i \leq k) \) is a specialization of \( G_F \) and has been
built in the training phase selecting image examples containing only the 
i-th expression type.

The article is organized as follows. In Section 2 we show the (standard) pre-processing techniques we apply to each image in order to extract its edge map and other shape features. In Section 3 we show the alignment techniques used in order to recognize and localize each face component. In Section 4 we present our proposal to deal with the varying spatial relations of the face components and we show how this information can be used also for expression recognition. Finally, in Section 5 we discuss the advantages and disadvantages of our proposal and we conclude.

2. Feature extraction

In this section we show the pre-processing operations performed on a given image before the shape analysis process. First of all we use standard algorithms for edge detection and thresholding [6]. Then we prune those edge pixels belonging to possible thick textured areas using the texture filters proposed in [1]. Intuitively, the filters remove those edge pixels surrounded by a region with a high intensity of points and with a great variance of the edge orientation. The result of this phase is shown in Figure 1. From now on we indicate with $I$ the edge map of the input image after all the filtering processes.

The edge pixels of $I$ are then merged in order to obtain line segments which are perceptively uniform. The used merging criteria are: pixels proximity and local orientation uniformity. First of all we merge edge pixels which are adjacent stopping in line junctions. After that, we merge the line segments so obtained looking to their endpoints [13]. For each segment $s_i$ we look for the segment $s_j$ which minimizes the function $D(s_i, s_j)$ defined as follows:

$$D(s_i, s_j) = |e_1 - e_2|((\alpha + \beta \theta),$$

where $e_1$ is one of the two endpoints of $s_i$ and $e_2$ is one of the endpoints of $s_j$, $|e_1 - e_2|$ is the Euclidean distance between $e_1$ and $e_2$ and $\theta$ is defined as below:

$$\theta = 180 - (\theta_1 - \theta_2),$$

$\theta_1$ and $\theta_2$ being, respectively, the orientations of the tangent vectors in $e_1$ and $e_2$. If $s_j$ is the segment minimizing $D(s_i, s_j)$ with respect to $s_i$ then $s_j$
is the best match for $s_i$. If, vice versa, also $s_j$ is the segment minimizing $D(s_j, s_i)$ with respect to $s_j$ (i.e., $s_i$ is the best match for $s_j$), then $s_i$ and $s_j$ is a “mutual favorite pair” of segments. The segment merging process is performed iteratively, choosing at each step the mutual favorite pairs of segments as suggested in [13] and stopping when no more merging is possible.

Finally, we search for anchor points [13, 20]. Anchor points are those points in which the curvature of the line segment is high and they will be used in Section 3 in order to control the matching between $I$ and the 3D models of the face components. If $p_i$ is the $i$-th point of a given segment $s$, we can approximate the local orientation of $s$ in $p_i$ by:

$$T_{p_i} = p_i - d - p_i + d,$$  

(3)

where $d$ is a fixed number of pixels defining a neighborhood of $p_i$. If $\psi_{p_i}$ is the angle of the slope of the vector $T_{p_i}$, then the curvature of $s$ in $p_i$ can be approximated by:

$$k_{p_i} = \frac{d\psi}{ds},$$  

(4)

where $ds$ is the length of the segment from $p_{i-d}$ to $p_{i+d}$ and $d\psi = \psi_{p_{i+d}} - \psi_{p_{i-d}}$. For more details, refer to [13, 11]. The output of this phase is the set of image anchor points: $A_I = \{p_1, \ldots, p_n\}$. In Figure 3 we show all the on-line phases of the recognition process. Feature extraction is the first step. Figure 2(a) shows the anchor points, each couple of anchor points delimiting a line segment extracted from the edge map of Figure 1(b).

![Figure 1](image_url)

(a) A gray-level image showing three faces  
(b) The edge map of (a) after the filters’ application
3. Alignment of the face components

In a well-known work [13] Ullman and Huttenlocher propose to approximate a perspective projection with an orthographic projection plus a scale factor. This approximation is not critical for compact objects such as a face, and it is still good for all those objects that are not deep with respect to the distance from the viewer (otherwise a constant scale factor would not be sufficient for the entire object).

In these conditions, the authors show that, given three non-collinear points $a_m, b_m$ and $c_m$ of the 3D model $M$ of the object to recognize and three non-collinear points $a_i, b_i$ and $c_i$ in the image plan $I$, there is a unique (up to a reflection) transformation $T$ of $M$ into $I$ such that:

$$T(a_m) = a_i,$$

$$T(b_m) = b_i,$$

$$T(c_m) = c_i,$$

where $T$ is given by a rotation, a translation and a scaling of $M$ followed by an orthographic projection into $I$. This result is exploited for automatic (rigid) object recognition tasks. The alignment algorithm they propose is based on an exhaustive matching of all the possible triples of $M$ and $I$. 

Figure 2

(a) The anchor points and the line segments extracted from the edge map of Figure 1(b). Different line segments are represented with different gray values and anchor points delimiting the segments with black dots. (b) The 3D model approximating a human head and its anchor points. (c) The map Valid of the valid points of Figure 2(b)
(hypothesis phase) in order to hypothesize a possible transformation \( T \) and a subsequent test phase consisting in projecting all the points of \( M \) into \( I \) using \( T \) and checking how many of them are close to image elements.

We use this method to recognize the face components in the image. The image components we use are: the whole head, the eyes, the mouth and the nose (5 components). Each component is approximated by a 3D shape composed of a set of simple, connected surfaces. Let \( M_C \) be the 3D description of the component \( C \). \( M_C \) is the set of the edge points of \( C \) represented in a fixed 3D coordinate frame. The edges of \( M_C \) are the surface discontinuities of the shape used to model \( C \). Figure 2(b) shows the shape model of the head.

We manually select off-line a set \((A_{M_C})\) of anchor points for \( M_C \), chosen among its highest curvature points (e.g., see Figure 2(b)). Let \( A_{M_C} = \{q_1, \ldots, q_m\} \). The alignment algorithm (derived from [13]) to recognize and localize \( C \), if present, in \( I \) is the following.

\[
\text{Alignment}(M_C, I, A_I, A_{M_C})
\]

1. Let \( \text{Valid} \) be a Boolean array representing the neighborhood of the edge points of \( I \).
2. For each triple \((a_1, a_2, a_3) \in A_{M_C}\) do:
   1. For each triple \((b_1, b_2, b_3) \in A_I\) do:
      1. Compute \( T \) corresponding to \((a_1, a_2, a_3)\) and \((b_1, b_2, b_3)\).
      2. \( \text{Match} := \#\{p : p \in T(M_C) \land \text{Valid}(p)\}/\#T(M_C) \geq th \).
      3. If \( \text{Match} \) then exit with success and return \( T \).
3. Exit with failure.

In Step 1, we build a Boolean matrix \( \text{Valid} \) of the same dimension of \( I \) representing all the points of the image close to any edge point. \( \text{Valid} \) can be easily built by means of a dilatation operation of the elements of \( I \) (see Figure 2(c)). In Step 5 we indicate with \( \#A \) the cardinality of the set \( A \). Moreover, \( T(M_C) \) is the set of points of \( M_C \) projected into \( I \) using \( T \). The Boolean variable \( \text{Match} \) is true when the cardinality of the points of \( T(M_C) \) which are close to any point of \( I \), normalized with respect to the cardinality of the whole \( T(M_C) \), is greater than a given threshold \( th \). If this happens, the procedure ends ignoring the remaining triples not yet unified and returns with success. The values of the parameters of \( T \) are
4. Variable spatial relations

As the most biological entities, also human face is a non-rigid object, i.e. an object which can undergo to non-rigid geometric transformations of its shape (deformations) without changing its nature. In fact, the relative positions of the main face components (nose, eyes, eye-brows, mouth, etc) can change due to face muscles stressing and blending, for instance caused by different face expressions. Moreover, mutual spatial relationships of face components usually vary in a given population of individuals.

For these reasons, first of all we separately detect in the image the main face components, each of which can be approximately regarded as a rigid object and hence dealt with a 3D “traditional” alignment approach (see the previous section). Then, we assembly the face components recognized in $I$ looking for those groups of components whose mutual spatial relations are consistent with a relational model previously built. Statistic information concerning the spatial relationships among different face components is represented by means of a graph. Each node of the graph is associated with a face component. Each edge between two nodes $\nu$ and $\upsilon$ represents the possible coordinate transformations from a reference frame centered in $\nu$ and a reference frame centered in $\upsilon$. In the following subsections we respectively show how the graph is built during the training phase and how it is used in the on-line recognition process.

![Flow diagram of the on-line phases of the recognition process](image)

**Figure 3**

The flow diagram of the on-line phases of the recognition process

4.1 Graph building

During the training phase, face image examples are used in order to collect statistic information about the spatial relationships holding among the face components. Suppose $F$ is a face example composed of $N$ components: $F = \{C_1, \ldots, C_N\}$, where each $C_r = (P_r, \alpha_r, \beta_r, \gamma_r)$, is the spatial representation of the rigid $r$-th component of the face and:
FACE DETECTION AND RECOGNITION

- \( P_r = (x_r, y_r, z_r) \) is the position of the center of mass of \( C_r \) (translation parameters with respect to a fixed reference frame),
- \( A_r, B_r \) and \( G_r \) are the rotational parameters of \( C_r \) with respect to the world-coordinate frame (\( \alpha_r, \beta_r, \gamma_r \in [0, 360] \)).

We do not represent scale parameters for reasons that will be clarified later. For every couple \( C_i = \langle P_i, \alpha_i, \beta_i, \gamma_i \rangle, C_j = \langle P_j, \alpha_j, \beta_j, \gamma_j \rangle \in F \), and with axes along the \( \alpha_i, \beta_i, \gamma_i \) directions to a reference frame centered in \( P_i \) and axes oriented along the \( \alpha_j, \beta_j, \gamma_j \) directions:

1. \( \delta x_{ij} = x_i - x_j \), (being \( P_i = (x_i, y_i, z_i) \) and \( P_j = (x_j, y_j, z_j) \));
2. \( \delta y_{ij} = y_i - y_j \);
3. \( \delta z_{ij} = z_i - z_j \);
4. \( \delta \alpha_{ij} = \alpha_i - \alpha_j \);
5. \( \delta \beta_{ij} = \beta_i - \beta_j \);
6. \( \delta \gamma_{ij} = \gamma_i - \gamma_j \),

where the differences in Steps (4)-(6) are computed modulo 2\( \pi \). We represent \( F \) using a totally connected graph \( G(F) \) in which every node is associated with an element of \( F \) and every arch \((i, j)\) is labeled with \( T_{ij} \). \( G(F) \) represents a specific face example of the training set. What we need now is to generalize \( T_{ij} \) for a generic face. This is achieved as follows.

Suppose the training set is composed of \( E \) face examples \( F_1, \ldots, F_E \), respectively represented by the graphs \( G(F_1), \ldots, G(F_E) \). The face model prototype generalizing all the \( E \) examples is the graph \( G(F_1, \ldots, F_E) \) in which each node represents a face component and each arch \((i, j)\) is labeled with \( H_{ij} \). \( H_{ij} \) is the convex hall of the set of points:

\[
\{(\delta x_{ij}, \delta y_{ij}, \delta z_{ij}, \delta \alpha_{ij}, \delta \beta_{ij}, \delta \gamma_{ij})|, \ldots, (\delta x_{ij}, \delta y_{ij}, \delta z_{ij}, \delta \alpha_{ij}, \delta \beta_{ij}, \delta \gamma_{ij})_E\},
\]

(8)
every point belonging to the Relative Configuration Space \( RC = \mathbb{R}^3 \times [0, 360] \).

The intuitive meaning of \( H_{ij} \) is the following. Suppose \( F_1, \ldots, F_E \) are the faces of the training set. For instance, they can represent the faces of different individuals and/or different expressions of the face of the same individual. In both cases we usually expect to have different values of the \( T_{ij} \) elements \((1 \leq i, j \leq N)\) for different faces of the set. Given a face
$F_k$ ($1 \leq k \leq E$) and two face components $C_i$ and $C_j$. $T_{ij}^k$ is the relation between $C_i$ and $C_j$ in $F_k$. The points $T_{ij}^1, \ldots, T_{ij}^k, \ldots, T_{ij}^E \in RC$ represent all the possible variations of $T_{ij}$ with respect to the training set $F_1, \ldots, F_E$. We represent the generalization of the set of points $T = \{T_{ij}^1, \ldots, T_{ij}^k, \ldots, T_{ij}^E\}$ by means of the convex hall of $T$, because it is the minimum convex connected region of RC containing all the points, hence the minimum (connected) possible generalization.

$G(F_1, \ldots, F_E)$ is a face model generalizing the spatial relationships among the face components observed in the training set $F_1, \ldots, F_E$. Since $G(F_1, \ldots, F_E)$ is built off-line this process is not included in the steps of Figure 3.

4.2 Verifying the geometric structure of the face

The recognition process is composed of two different phases. Given a test image, we first use the alignment techniques described in Section 3 to separately look for each face component. Then, for each set of recognized components $F' = \{C'_1, \ldots, C'_{N'}\}$ let $G(F')$ be the totally connected graph representing $F'$ and the spatial relationships of its components. We verify that $G(F')$ can be consistently matched with $G(F_1, \ldots, F_N)$. It is worth noticing that $N' \leq N$, since not necessarily all the face components need to be recognized in a given image. For this reason we look for a matching between $G(F')$ and a subgraph of $G(F_1, \ldots, F_N)$. Since the cardinality of the involved graphs is very small ($N, N' \leq 5$) and each node is labeled with its component type (nose, mouth, etc.), the matching procedure is quite trivial and not computational expensive. Moreover, the scale factor of a face component remains almost unchanged with respect to the other components, both in different facial expressions and in population of individuals. Thus, we do not need to store scale information in $G(F_1, \ldots, F_N)$. A simple test, checking that all the $C'_1, \ldots, C'_{N'}$ elements have approximately the same scale factor, is sufficient to deal with scale issues. After that, we compare the values of the edge labels of $G(F')$ with the labels of $G(F_1, \ldots, F_N)$. For every couple of elements $i$ and $j$ we compute the world coordinates of $C'_i$ and $C'_j$ using the values returned by the alignment procedure (Section 3). Then we check if the point $T'_{ij}$ of $G(F')$ belongs to the convex hall $H_{ij}$ represented in $G(F_1, \ldots, F_N)$. The successful recognition of a face depends on the number of consistent spatial relations verified. The verification process on a graph structure
is a quite “traditional” technique in Computer Vision (e.g., see [12]). In Figure 3 the last two steps correspond, respectively, to the creation of $G(F')$ and its matching with $G(F_1, \ldots, F_N)$.

This schema can be easily generalized for face expression treatment. In fact we only need to have separate models $G_H$, $G_S$, etc, for different emotive states (happiness, sadness, etc.). $G_H$ is obtained suitably selecting a training set composed of only happy face examples, while $G_S$ is built using only images of sad people, etc. (e.g., see [7]). $G_N$ corresponds to the neutral expression (all the muscles relaxed). Each graph so obtained represents the specific spatial relations among the face components statistically characterizing a given human expression. The recognition phases as well as the off-line graph building process remain unchanged.

5. Discussion and conclusions

As far as we know, alignment methods have never been used in face detection problems up to date. In this paper we have proposed to use a 3D prototype model of a face and an alignment technique for face detection. This approach has the advantage to cope with different views of a face without the need to iterate the detection process using different 2D models nor the need to scan the whole image with different image windows of different sizes and positions as it is usually the case of existing face detection systems. Moreover, since a face is a non-rigid object, we need to take into account that the spatial relations among the face components can change due to both inter-individual differences and different facial expressions. For this reason we approximate each face component with a rigid 3D model which is on-line aligned with the image, and then we compare groups of candidate components localized in the image with a graph representing the possible spatial variations. Finally, different facial expression can be dealt with by means of different graph models, each of which representing the possible spatial variations over a training set specialized on a particular expression.

References


Received November, 2005