Detection of facial features

Paola Campadelli, Elena Casiraghi, Raffaella Lanzarotti

Dipartimento di Scienze dell’Informazione
Università degli Studi di Milano
Via Comelico, 39/41 20135 Milano, Italy
{campadelli, casiraghi, lanzarotti}@dsi.unimi.it

Abstract. An algorithm for the automatic features detection in 2D color images of human faces is presented. It first identifies the eyes by means of a template matching technique, a neural network classifier, and a distance measure. It proceeds localizing lips and nose using a non-linear edge detector and color information. The method is scale-independent, works on images of either frontal, rotated or slightly tilted faces, and does not require any manual setting or operator intervention.

1 Introduction

Several applications like face recognition [7] [16], model-based coding of video sequences [1], 3D face reconstruction [14], facial expression analysis [17], and intelligent man-machine interfaces [1] require the localization and precise description of facial features (eyes, nose and lips).

Such is not an easy machine-vision task due to the high inter-personal variability (gender, race, ...), the intra-personal changes (pose, expression, ...), and the acquisition conditions (lighting, image resolution, ...).

Some techniques reported in literature determine feature points manually [17] [4]. Attempts to automate this phase have been done; such works can be distinguished in color-based and shape-based. The first class of methods tries to characterize the face and each feature with a certain combination of colors [6]. This is a low-cost approach, but not very robust. The shape-based approaches look for specific shapes in the image adopting either template matching (with deformable templates [19],[1], [18] or not [3]), graph matching [5], [16], snakes [13], or the Hough transform [8]. These methods work well only under restricted assumptions (regarding the head position and the illumination conditions) and they are computationally expensive.

We have proposed [9] an algorithm using both shape and color which has two hierarchical processing modules: the first identifies four sub-images, each tightly containing one of the features of interest; the latter module is specialized in localizing fiducial points on the found features with high accuracy. At that stage the most critical step was the identification of the eyes sub-images. We present here a solution based on a neural network classifier and the evaluation of the distances of pair of gray-level images.
The method we present works on images of face foregrounds. We acquire color images with homogeneous and light-colored background, and frontal and diffuse illumination. Faces can be either in frontal position or rotated around the head vertical axis of 30° at most and tilted laterally of about 10°. In any case, the completeness of the facial images is requested: no occlusions, no wearing glasses, no beard and closed mouth.

The paper is organized as follows: in section 2 image pre-processing and an initial feature localization are presented; in section 3 we describe the eyes localization using both a neural network and a distance measure; in section 4 lips and nose localization are introduced; in section 5 results on a database of 130 color images are reported and discussed.

2 Preprocessing and initial features localization

The original color images have a wide variety of resolutions ranging between 480x640 to 960x1280 pixels. For computational efficiency the sub-images identification is done on down-sampled images ranging between 150x170 to 300x340 pixels.

We first cluster the gray-level image into three clusters through the clustering algorithm presented in [2]. The lightest gray-level represents the background, the intermediate the skin and the darkest represents both the features and other dark pixels of the image (for example the hair).

To localize the features of interest the largest region, S, with pixels of the intermediate gray-level (corresponding to the skin) is found. Then all the pixels surrounded by pixels of S and belonging to the darkest gray-level are identified and set to 1; all the others are set to 0. What we obtain is the feature image [Fig.1].

![Feature images](image1.png)

**Fig. 1.** Features images

We observe that, besides the features of interest, also few regions, such as those corresponding to shadows, hair, or ears are set to 1. Further processings are therefore required to discard not interesting regions and to isolate the features of interest in separated sub-images.

We proceed localizing the eyes, then, in order to look for the lips and the nose, the attention can be concentrated upon a restricted image area.
3 Eyes localization

In order to determine the set of rows (eyes band) which contains the eyes, we apply the template matching to the features image [Fig.1], searching the two eyes. The difficulty is that we are not looking for an object with a fixed shape [Fig.2, first line]. For this reason we adopt a binary template [Fig.2, second line] which models the two eyes in a very rough way. It consists of two symmetric blobs placed side by side each being large enough to overlap the region corresponding to an eye in the features image. A single template has been used for all the images which are of significant different size, thus showing a desirable scale-independence property.

![Fig. 2. First line: examples of "clustered" eyes; Second line: Eyes Template](image)

Among the positions with the highest cross-correlation, we maintain the 10 which satisfy also the following symmetry condition: the cross correlation between half the template (one blob) and the sub-images on the left and on the right of the found position are compared and the position is rejected if the results are not similar enough, that is their ratio is lower than 0.7. Some examples of the behavior of this rule are shown in figure 3: both the blue and the red pixels have high cross-correlation, but the blue ones satisfy also the symmetry condition; the red pixels are rejected.

On all the images of our database most of the points calculated in this way are positioned in the eyes band (on average 8 out of 10) [Fig.3]. This allows us to select the band easily and with high reliability (100% of hits).

![Fig. 3. Template matching results and identified eyes band](image)
The features are usually well separated in the eyes band, therefore we can isolate them in different sub-images using the vertical projection. More than two distinct peaks, due to shadows, hair and ears, are usually present in the projection; for each of them we take the corresponding gray level piece of image and use a neural network trained by standard back-propagation to recognize the eyes from the other features. We use gray-level images instead of the eye features found by the clustering algorithm since with this second representation the eyes often loose their peculiar shape.

3.1 Eyes detection by neural networks

At this stage the goal is to classify all the possible sub-images extracted from the eyes band in two classes: eyes or non-eyes; example of them are shown in figure 4. As we can notice, their dimension are quite different (ranging from 20x20 to 50x50) and images representing the same feature have different appearances making the automatic classification problem more complex.

![Fig. 4. Some example of Input to the neural network](image)

First, we have to reduce all the sub-images to the same dimension. We thus search the minimal input size which gives to the network enough information to perform a good classification. To this end, we have experimentally compared the performances of networks trained with different input representations. We did 3 kinds of experiments: on down-sampled images (E1), on images compressed by the wavelet transform [10] (E2), and on the coefficients of the wavelet transform (E3).

More precisely, for the experiment E1, we down-sample the original images to the size of 4x4, 8x8, 16x16 and 32x32. For the experiments E2 and E3 we adopt the Haar transform, having tried also the Daubechies and the Symmetric without observing any improvement; we consider respectively the images obtained anti-transforming from the wavelet domain and the wavelet coefficients themselves. In both cases we consider the data at different scales corresponding to images of dimensions 4x4, 8x8, 16x16 and 32x32.

We experimented different architectures for each input size, varying the number of hidden layers and the number of neurons in them. In the following we report only the best trade off between results and network complexity.

We consider network architectures with one hidden layer with different number of neurons depending on the size of the input; the output neurons are always two in order to classify an image as eye or non-eye. More precisely we consider 4
architectures: $A_1$ with 16 and 5 neurons, $A_2$ with 64 and 10 neurons, $A_3$ with 256 and 10 neurons, and $A_4$ with 1024 and 30 neurons respectively.

The images are assigned to training and test sets in two ways, $S_1$ and $S_2$. In $S_1$ the images are randomly selected, so that eyes of the same subject are allowed to appear in both training and test set. In $S_2$ no subject can appear in both training and test set; tests are performed with novel faces. In both cases the training sets are composed of 200 images, and the test set are composed of 300 images; moreover eyes and non-eyes images are present in the same proportion. We notice that, building training and test set according to the rule $S_2$ the networks have more difficulties and for these reason we report only results obtained in this condition.

In table 1 we report the number of errors obtained by networks of different architectures $A_i$ on a test set of images reduced by the three methods $E_i$ described above.

We observe that networks $A_1$ and $A_4$ do not give good results; while $A_2$ and $A_3$ results are quite similar. We choose the architecture $A_2$ which introduces a lower number of free parameters. Moreover, we notice that there is no significant difference in the performances of architectures with the same number of neurons whose input are either down-sampled images, or wavelet compressed images or the wavelet coefficients (with the exception of $A_1$ which gives bad result anyway). For this reason we use (8x8) down-sampled images since down-sampling is less computationally expensive.

<table>
<thead>
<tr>
<th></th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>$A_3$</th>
<th>$A_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_1$</td>
<td>39</td>
<td>24</td>
<td>23</td>
<td>30</td>
</tr>
<tr>
<td>$E_2$</td>
<td>34</td>
<td>25</td>
<td>23</td>
<td>29</td>
</tr>
<tr>
<td>$E_3$</td>
<td>31</td>
<td>22</td>
<td>24</td>
<td>31</td>
</tr>
</tbody>
</table>

Table 1. Network results

Integrating the classification procedure in the face image processing, and applying it to the 130 face images, we obtained that 15 images had at least a misclassification. More precisely in 12 of them only 1 error has been done, in the other 3 cases 2 errors have been done which do not compensate each other. We can only detect an error if we detect 0, 1, or 3 eyes per face-image. When the error is caught further processings are necessary (see next paragraph).

### 3.2 Error correction using distance between images

We suppose that, among the sub-images extracted from the eyes band, the two ones representing the eyes are the most similar. Thus, we have experimentally compared three methods to check the similarity. Two of them are based on distance measures between images ([11], [21]), the other is a matching technique described in [12]. The best result has been obtained with the modified Hausdorff
distance [11] defined for binary images and adapted here to gray-scale ones. The distance $D_{m,h}(A, B)$ between two images $A$ and $B$ is described by the expression:

$$D_{m,h}(A, B) = \max \left( \frac{1}{N_a G} \sum_{a \in A} d_{p,i}(a, B), \frac{1}{N_b G} \sum_{b \in B} d_{p,i}(b, A) \right)$$

(1)

where $N_a$ and $N_b$ are the number of pixels in image $A$ and image $B$ respectively, $G$ is the number of the gray levels.

d$_{p,i}$ is the point-to-image distance function defined by the following expression:

$$d_{p}(a, B) = \min_{b \in W_i} (d_{p,p}(a, b))$$

(2)

where $W_i$ is a window centered in the pixel $b \in B$ corresponding to the pixel $a \in A$ and $d_{p,p}(a, b)$ is:

$$d_{p,p}(a, b) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2}$$

(3)

where $z_i$ is the gray level of pixel $i$, and $(x_i, y_i)$ its position.

On the 15 images on which the neural network fails, the similarity evaluated by $D_{m,h}(A, B)$ corrects all but two errors.

4 Lips and nose localization

Once determined the eyes’ bounding boxes, we move to the lips and nose localization using both gray level and color information. On the gray-level image a non-linear edge detector is applied [20]. The detector uses local statistical information: a square window centered around a pixel $P$ is divided into two sub-windows of equal size in four different ways [Fig.5]. For each sub-division $i$ the following function $D_i$ (diversity) is evaluated:

$$D_i = \alpha \Delta m - (1 - \alpha) \Delta \sigma$$

where $\Delta m = |m_a - m_b|$, $\Delta \sigma = |\sigma_a - \sigma_b|$, $m_a$, $m_b$, $\sigma_a$, $\sigma_b$ are respectively the mean and the standard deviation in the regions $a$ and $b$, and $\alpha$ is a constant ($0 \leq \alpha \leq 1$).

![Fig.5. Windows](image)
The maximum diversity $D(P) = \max_{i=1, A} \{D_i\}$ is assigned to $P$. We use this method with a $7 \times 7$ window size positioned on a pixel out of 3. The lips and the nose are roughly characterized by pixels with high horizontal and vertical diversity respectively.

To better localize the lips, we combine color information in the Cr plane with the output of the edge detector.

The nose is localized in the remaining portion of the image in the region where the horizontal diversity is different from 0.

\begin{figure}[h]
\centering
\includegraphics[width=0.6\textwidth]{fig6.jpg}
\caption{Subdivisions}
\end{figure}

We have thus identified four sub-images [Fig.6]. Given the sub-images, the fiducial points are detected with high accuracy on frontal faces [9]. Further improvements are required for rotated faces.

\section{Results and discussion}

The method described has been experimented on 130 color images of very different scales acquired in different illumination conditions. They represent either frontal or rotated faces of Caucasian women and men. We asked to the people to have a neutral expression keeping the mouth closed and the eyes opened. We have not dealt with the case of men with beard.

The proposed method localizes the sub-images representing the features of interest with high confidence on both frontal and rotated images: on our database it fails on two images only.

A final consideration has to be done about the program running time. The algorithm has been developed in IDL, an interpreted language; its running time, on a Pentium III, 800MHz, 256Mb of RAM is, on average, of 5 seconds. More precisely, if the network manages to determine which are the two eyes sub-images, the algorithm takes 4 seconds; on the contrary, if it is necessary to process the distance measures, the running time is increased of 0.7 seconds for each comparison. We observe that most of the computational time is due to the execution of the clustering algorithm. The time can be certainly reduced developing the algorithm in a compiled language and optimizing the code.
References

7. J. Huang, C. Liu, and H. Wechsler. Eye detection and face recognition using evolutionary computation. In Wechsler et al. [15].