Facial Feature Detection using Distance Vector Fields

Stylianos Asteriadis [1], Nikos Nikolaidis *, Ioannis Pitas

Department of Informatics, Aristotle University of Thessaloniki, Box 451, Thessaloniki, GR 54124, Greece

Abstract

A novel method for eye and mouth detection and eye center and mouth corner localization, based on geometrical information is presented in this paper. First, a face detector is applied to detect the facial region, and the edge map of this region is calculated. The distance vector field of the face is extracted by assigning to every facial image pixel a vector pointing to the closest edge pixel. The x and y components of these vectors are used to detect the eyes and mouth regions. Luminance information is used for eye center localization, after removing unwanted effects, such as specular highlights, whereas the hue channel of the lip area is used for the detection of the mouth corners. The proposed method has been tested on the XM2VTS and BioID databases, with very good results.

Key words: facial features detection, distance map, distance vector field (DVF).

* Corresponding author, tel: +30-2310-998566, fax: +30-2310-998453
Email addresses: stiast@image.ntua.gr (Stylianos Asteriadis), nikolaid@aiia.csd.auth.gr (Nikos Nikolaidis), pitas@aiia.csd.auth.gr (Ioannis Pitas).

[1] Current address: School of Electrical and Computer Engineering, National Tech-

1 Introduction

Numerous papers have been published recently on face and facial feature localization, notably on eye and mouth detection, since these tasks are essential for a number of important applications like face recognition, human-computer interaction, facial expression recognition, surveillance, etc. Facial feature detection techniques can be categorized in two major classes: In the first category, a face detection step has to be performed before facial feature detection, in order to provide the facial region where feature detection will be performed. The second class of approaches is looking for facial features over the entire image. In the latter case, constraints regarding facial geometry and facial feature size are usually taken into account, thus making this kind of approaches more prone to errors and, thus, limiting their application to images where the depicted faces are within certain size limits. Here, we shall concentrate on algorithms of the first class.

In [1], six facial features (left and right eye and eyebrow, nose and mouth) are detected, following the detection of human faces on images. Face detection is done using skin color segmentation in the YIQ color space, followed by a refinement step based on a genetic algorithm. Subsequently, image enhancement through histogram equalization and noise reduction are employed. An intensity threshold is then found and used to detect the facial features, since they appear darker than other face parts (cheeks, forehead, etc.). In this way, a binary image is produced, where the desired facial features constitute the foreground. Morphological dilation is used to remove small holes on the face. The resulting binary map is used to tag the facial features. Each one of six

nical University of Athens, Athens, Greece

facial elements is assigned a unique tag, depending on its location on the face area. In [2], eyes, nose, mouth and chin regions are searched for. Two thresholds are applied on the image edge map for the extraction of the head and face boundaries. Projections along the x and y axis are subsequently applied on the binary edge image of the face region for the detection of eye, nose and mouth regions. An ellipse is also used for the final detection of the chin line, which is, sometimes, difficult to localize. In cases where the above procedure results in facial features that are very close to each other and cannot be distinguished, the eyes are detected using Gabor filters and the other features are subsequently localized by using heuristic rules derived from face geometry. In [4], a sequence of bottom hat morphological operations is used to find dark regions (which usually correspond to eyes and lips) in a candidate elliptical face region and geometrical constraints are employed for the detection of candidate facial features. Further heuristic rules (corresponding to the expected face size, maximum and minimum dimensions of eyes and lips and aspect ratios of eyes and lips) are applied [5,6] and, if the features fulfill them, they are labelled as potential eye or lips regions. Using these labelled features, feature triplets (eyes/lips) are established and additional constraints are employed to decide if they are indeed facial features or not. In more detail, eye and mouth triplets that are far from the center of the face ellipse are removed. Also, triplets where the orientation of the line that connects the two eyes differs significantly from that of the ellipse minor axis, are removed.

In [7], facial features are detected following the detection of skin color- related elliptic facial regions. Red and Green components, divided by the intensity are used to form a luminance invariant vector for each pixel. These vectors are exploited to evaluate the probability that a pixel belongs to a face or not,

based on a look-up table created from training images of such components. The face images found are normalized in size and the intensity map is used to locate facial features. The first and second derivatives of a Gaussian in x and y directions $(G_x, G_y, G_{xx}, G_{xy}, G_{yy})$ are convolved with the facial image and the result forms a five-dimensional vector (local appearance descriptor) for each pixel. A set of training images is used to obtain such Gaussian derivatives which are clustered using the K-means algorithm. The obtained clusters are exploited in order to distinguish between hair, salient facial features (eyes, nose, mouth, chin) and other skin regions. Further geometrical analysis is used to decide whether a connected facial image region belongs to a particular facial feature. In [8], a multi-stage approach is used to locate 17 features on a face. These features are interesting points around the eyes, the mouth, the nose, the eyebrows and the chin. First, the face is detected using the boosted cascaded classifier algorithm by Viola and Jones [3]. The same classifier is trained using facial feature patches to detect facial features, thus a different detector is constructed for each feature. A novel shape constraint, the Pairwise Reinforcement of Feature Responses (PRFR) is used to improve the localization accuracy of the detected features. More specifically, a pairwise distribution is defined as the distribution of the true location of feature i given the best match for feature detector j in the reference coordinate system defined by the face region. The ensemble of true feature locations and detector matches on a set of training images is used to compute relative histograms H_{ij} which approximate these distributions. Using these histograms and a set of the most probable locations for each feature provided by the corresponding detector helps to improve detection. Further detection refinement is achieved by using the Active Appearance Model (AAM) algorithm using four values for each pixel: the normalized gradients in the x and y directions, its "edgeness" and

”cornerness”. In [9], a three stage technique is used for eye center localization. The Hausdorff distance between edges of the image and an edge model of the face is used to detect the face area. At a second stage, the Hausdorff distance between the image edges and a more refined model of the area around the eyes is used for more accurate localization of the upper area of the head. Finally, a Multi-Layer Perceptron (MLP) is used for finding the exact pupil locations. In [10], the authors use Generalized Projection Functions (GPF) to locate the eye area and, subsequently, the eye centers, starting from a coarse eye area estimate provided by the algorithm proposed in [11]. The GPFs are linear combinations of functions which consider the mean and variance of facial intensities along image rows and columns. GPFs were used to locate the bounding box of each eye. The eye center was then found by taking the center of this bounding box. In [12], the face is initially detected using a skin-color segmentation algorithm. Morphological sequences of dilation and erosion follow and the face contour is localized by applying an ellipse fitting algorithm. In that work, eyes and nostrils were searched for in certain areas of the detected face, as the darkest regions of their neighborhoods. Geometrical constraints were also applied in order to choose for the best pair of eyes or nostrils. In the case of mouth detection, integral projection functions and edge operators were used.

A novel technique for eyes and mouth detection and eye center and mouth corner localization is proposed in this paper. The proposed technique is an improved variant of the technique presented in [13] by the same authors, which used a different approach for eye detection and did not include mouth detection/localization. The coarse detection of facial characteristics is done by using only geometrical information, thus, avoiding illumination problems and

facial image intensity variations. The detected face is normalized to a certain size along the x and y axes and a distance vector field is created by assigning to each image pixel a vector pointing to the closest edge. The eyes and mouth regions are detected by finding regions inside the face, whose distance vector fields resemble the distance vector fields of eye and mouth templates extracted from sample eye and lip images. Facial intensity and color information is then used within the detected eye and mouth regions in order to accurately localize the eye centers and mouth corners. Our technique has been tested on the XM2VTS and BioID databases with very good results. Comparisons with other state of the art methods verify that our method achieves superior performance.

The structure of the paper is as follows. In Section 2, the idea behind using the distance vector fields is discussed. In Section 3, the method used to localize eye and mouth areas, as well as eye centers and mouth corners on a face image is described. The experimental evaluation procedure (database, distance metrics, etc) and the obtained results are described in Section 4. A comparison of the proposed method with other approaches in the literature is also included in the same section. Conclusions follow in section 5.

2 Distance Vector Fields

For a binary 2-D image containing "object" and "background" pixels, the distance vector field (DVF), also known as vector distance field or vector distance map, is a vector field that is created by assigning to each background

pixel (i, j) the vector $\mathbf{v}$ pointing to the closest object pixel (k, l). Specifically:

$$\mathbf{v}(i, j) = [k - i, l - j]^T, (i, j) \in B \tag{1}$$

$$(k, l) = \arg \min_{m,n \in O} D((i, j), (m, n)) \tag{2}$$

where D is some metric and O, B are the sets of object and background pixels respectively.

Distance vector fields were introduced in [14] as an efficient means for calculating the true Euclidean distance map (also known as distance field) of an image, i.e. the map (image) whose pixels store the Euclidean distance of the corresponding image pixel from the closest object pixel. This gave rise to a family of distance transform algorithms, known as vector distance transform (VDT) algorithms (see for example [15]). It should be noted, however, that, in these algorithms, DVFs were used only as an intermediate step for the calculation of the distance field of an image. Distance vector fields are, also, sometimes used in graphics and animation [16].

In this paper, DVFs will be used to describe the geometry of facial features. Each pixel in a facial area is assigned a vector pointing to the closest edge pixel, extracting in this way, a distance vector field describing the geometry of facial regions. A schematic representation of the distance vector field of a face is shown in Figure 1.

Thus, instead of the facial image intensity values, we generate and use in the proposed algorithm a DVF, whose dimensions are equal to those of the image and where each pixel is characterized by the vector described above. This vector encodes, for each pixel, information regarding its geometric relation with

neighboring edges and, thus, is relatively insensitive to intensity variations or poor lighting conditions. Obviously, the DVF contains more information than the distance map and one can utilize this additional information for successful facial feature detection. The above distance vector field can be represented as two scalar maps (images) representing the values of the horizontal and vertical vector components for each pixel. The closest edge point for each pixel, which is necessary in order to calculate the distance vector field can be derived from the distance map of the edge image. The following procedure was used in our case: For each pixel (i, j) its 8-neighborhood on the distance map is considered and the pixel with the smallest distance map value is selected as the current pixel. The procedure continues recursively by finding the smallest distance map value in the 8-neighborhood of the current pixel until a pixel with value zero on the distance map is reached. If this is at location (k, l), the vector in (1) is assigned to the pixel.

Figures 2(a),2(b),2(c) depict faces detected in three images, Figures 2(d)-2(f), 2(g)-2(i), 2(j)-2(ℓ), show the Canny edge map, and absolute values of the horizontal and vertical component maps of the distance vector field in the detected face area, respectively. The distance map used for the calculations is depicted in Figures 2(m)-2(o). The distance transform described in [17] has been used for the calculation of the distance map throughout this paper.

3 Eye and mouth region detection

Prior to eye and mouth region detection, face detection is applied on the face images. The face is detected using the Boosted Cascade method [3]. This method uses the Adaboost algorithm to select and combine a set of appropriate

facial image features that resemble Haar basis functions, so as to train efficient classifiers. A combination of such successively more complex classifiers in a cascade allows the early rejection of non-face regions, thus, allowing for more computation to be spent on more promising areas.

3.1 Eye Region Detection

The proposed eye region detection method can be outlined as follows: First, the Canny edge detector [18] is applied on the output of the face detector and the distance vector field of the face area is calculated. For eye region detection, distance vector fields of candidate regions are compared to mean horizontal and vertical vector component maps (templates) for the two eyes. Figure 3 shows an example of an eye region, the corresponding distance map and horizontal and vertical vector component maps, as described above.

In more detail, the face area found by the face detector is scaled (up or down) to 150×120 pixels. This size proved satisfactory for our experiments, as the basic geometrical information needed for the eye region detection is retained. The Canny edge detector with a high threshold of 50 and a low threshold of 20 [18] was applied on the detected facial region. These values allow to detect the most prominent edges, such as those of the eyes and eyebrows. Subsequently, the distance vector field of the face region is extracted.

In order to detect the eye areas, regions R_k of size $N \times M$ within the detected face are examined and the corresponding distance vector fields are compared with the mean vector field (template) extracted from a set of right or left eye images. Different vector field templates were created for the right/left eyes.

Fifty eight manually selected right and left eye images, scaled to dimensions N=26 and M=26 pixels, were used to extract the mean vector component maps. The right eye mean distance vector field is shown in Figure 4. All these images were chosen so that the upper image boundary was placed a bit higher than the eyebrow whereas the left and right boundaries were placed just beyond the eye corners.

The similarity between the distance vector field $\mathbf{v}$ of a candidate image region and that of the templates is calculated using the following distance measure:

$$E_{L_2} = \sum_{i \in R_k} \|\mathbf{v}_i - \mathbf{m}_i\|_2 \tag{3}$$

where $\| \cdot \|_2$ denotes the L_2 norm, $\mathbf{v}_i$ are the vectors of the DVF of the candidate region and $\mathbf{m}_i$ the corresponding vectors of the mean distance vector field (template) of the eye we are searching for (right or left one). The candidate region on the face that minimizes E_{L_2} is marked as the region of the left or right eye. To make the algorithm faster, we utilize the knowledge of the approximate positions of eyes on a face. Thus, we search for the eyes only in a zone on the upper part of the detected face. Moreover, the right and left eye are searched at the right and left parts of this zone, respectively.

3.2 Mouth Region Detection

An approach similar to the one described above for eye region detection is followed for mouth region detection. The outcome of the eye centers localization (described in subsection 3.3) can be used to define the region where the mouth is to be searched. More specifically, it was proven experimentally that the mouth region can be searched for in a zone with the following characteristics:

Its upper and lower boundaries are at a distance d_{up}=1.2d, d_{low}=*1.8d* lower than the midpoint between the eye centers, where d is the found inter-ocular distance. The upper boundary of candidate mouth regions should always be within these boundaries. The middle, in the horizontal direction, of the mouth region is allowed to be a few pixels away from the vertical axis passing from the midpoint between the eye centers.

For the extraction of the mean DVF of the mouth, 16 mouth images were used. These images were scaled to the same dimensions $N_m \times M_m$, with N_m=13 and M_m=36. An example of a mouth image, used for the calculation of the mean distance vector field, its edge image and the corresponding horizontal and vertical coordinate maps can be seen in Figure 5. The two coordinate maps of the mean vector field can be seen in Figure 6.

The mouth region was detected using a procedure similar to the one used for eye detection. That is, the DVF of the candidate regions in the area described above were compared to the mean DVF (template). However, since lip and skin color are, in many cases, similar and since beard (when existent) might occlude or distort the lip shape, lip localization is more difficult than eye localization. For this reason, an additional factor is included in (3). This factor is the inverse of the number of edge pixels of the horizontal edge map evaluated within the candidate mouth area. Thus, the mouth region detected by the algorithm is the candidate region that minimizes:

$$E_{L_2}^{mouth} = \sum_{i \in R_k} \|\mathbf{v}_i - \mathbf{m}_i\|_2 + \frac{w}{I_{R_k}^{HE}}, \tag{4}$$

where w is a weight and $I_{R_k}^{HE}$ is the number of edge pixels in the horizontal edge map of candidate region R_k. The new term was added because, due to the

elongated shape of the lips, the corresponding area is characterized by a large concentration of horizontal edges. Thus, this factor, which favors areas with horizontal edges, helps in discriminating between mouth/non-mouth regions. The additional factor is weighted so that its mean value in the search zone is equal to the mean value of the term $\sum_{i \in R_k} \|\mathbf{v}_i - \mathbf{m}_i\|_2$ in this zone. In more detail, the weight w in (4) is calculated as follows:

$$w = \frac{\sum_{k=1}^{L} \sum_{i \in R_k} \|\mathbf{v}_i - \mathbf{m}_i\|}{\sum_{k=1}^{L} (I_{R_k}^{HE})^{-1}} \tag{5}$$

where L is the number of candidate mouth areas in the search zone.

After eye and mouth detection, the eye centers and mouth corners are localized within the found areas using the procedures described in the following sections.

3.3 Eye Center Localization

The eye area found using the procedure described in section 3 is scaled back to the dimensions $N_{eye} \times M_{eye}$ it had in the initial image. Moreover, before eye center localization, a pre-processing step is applied. Since reflections (specular highlights), that affect the results in a negative way, frequently appear on the eye, a reflection removal step is implemented. Such highlights are usually bright areas consisting of no more than a few pixels. The reflection removal step proceeds as follows: The eye area (Figure 7(a)) is first converted into a binary image (Figure 7(b)) through thresholding. This is done using the threshold selection method proposed in [19] that aims at maximizing the intraclass variance between the black and the white pixels on the resulting binary image.

Subsequently, all the white connected components of the resulting binary eye image that occupy less than 1% of the image pixels are considered as highlight areas (see Figure 7(c) depicting the binary image having these areas removed) and the intensities of the pixels in the grayscale image that correspond to these areas are substituted by the average luminance of their surrounding pixels. The result is an eye area with most highlights removed (Figure 7(d))

The eye center localization is performed in three steps, each step refining the results obtained in the previous one. By inspecting the eye images generated by the algorithm described in the previous section, one can observe that the eyes reside at the lower central part of the detected eye area, while the upper part comprises of the eyebrow and the left and right parts consist of skin or parts of the eyeglass frame. Thus, the eye center is searched within an area that covers the lower 60% of the eye region and excludes the right and left parts of this region (15% of the right and left part of the region are excluded). The information in this area comes from the eye itself and not from the eyebrow or the eyeglasses, as can be seen in Figure 8. The three steps of eye center localization are the following:

Step 1: Since, at the actual eye center position, there is significant luminance variation along the horizontal and vertical axes, the images $D_x(x, y)$ and $D_y(x, y)$ of the absolute discrete intensity derivatives along the horizontal and vertical directions are evaluated (Figure 9):

$$D_x(x, y) = |I(x, y) - I(x - 1, y)| \tag{6}$$

$$D_y(x, y) = |I(x, y) - I(x, y - 1)| \tag{7}$$

The contents of the horizontal derivative image are subsequently projected on

the vertical axis and the contents of the vertical derivative image are projected on the horizontal axis:

$$D_{px}(y) = \sum_x D_x(x, y) \tag{8}$$

$$D_{py}(x) = \sum_y D_y(x, y) \tag{9}$$

The four positions $x_i, y_i, i = 1 \ldots 4$ along the horizontal and vertical axes that correspond to the four largest values of the $D_{py}(x)$ and $D_{px}(y)$ respectively are selected. These positions correspond to the horizontal and vertical lines crossing the strongest edges. Subsequently, the pixel (x, y) where $x = median(x_1, x_2, x_3, x_4)$ and $y = median(y_1, y_2, y_3, y_4)$ defines an initial estimate of the eye center (Figure 10(a)).

Step 2: Using the fact that the eye center is in the middle of the largest dark area in the region, the previous result can be further refined. The darkest column in a $0.4N_{eye} \times 0.15M_{eye}$ pixels area around the initial estimate (Figure 10(b)) is found and its position is used to define the x coordinate of the refined eye center:

$$x = \arg\min_x \sum_y I(x, y) \tag{10}$$

In a similar way the darkest row in a $0.15N_{eye} \times 0.4M_{eye}$ area (Figure 10(c)) around the initial estimate is used to locate the vertical position of the eye center (Figure 10(d)):

$$y = \arg\min_y \sum_x I(x, y) \tag{11}$$

The sizes of the search areas were empirically decided.

Step 3: Since the iris is darker than the area around it, the darkest region of size $0.25N_{eye} \times 0.25M_{eye}$ is searched for in a $0.4N_{eye} \times M_{eye}$ area around the pixel found at the previous step and the middle of this area gives the final estimate of the eye center, as can be seen in Figure 10(e).

3.4 Mouth Corner Localization

For mouth corner localization, color information can be used. The existence of beards and the difficulty to differentiate between skin and lip areas in grayscale images led us to the use of the hue component of mouth regions in order to find mouth corners, since the hue values of the lips are distinct from those of the surrounding area. More specifically, the lip color is reddish and, thus, its hue values are concentrated around 0°, in the range [340°,20°]. Figure 11 shows results of mouth region detection and the corresponding hue component. It is obvious that, even in the case of difficult to distinguish lip regions, or even in the case where mouth region detection results are not good (i.e. when the mouth is not centrally located in the detected area), the hue component can be used very efficiently to distinguish the mouth. In order to detect the mouth corners, the pixels of the hue component are classified into two classes using the automatic thresholding approach of [19]. The class whose mean value is closer to 0° is declared as the lip class. Due to the angular nature of hue, the following definition of distance between two angular values a, b was utilized [20]:

$$d(a, b) = \pi - |\pi - |a - b|| \tag{12}$$

In certain cases, small connected components with low hue values might be erroneously assigned to the lip class by the thresholding. These components should be removed before proceeding with mouth corner localization. More specifically, all foreground components of size less than 15% of the mouth region are removed. An example of such a case is shown in Figure 12.

After the steps described above, the actual mouth corner localization is performed by scanning the binary image and looking for the rightmost and leftmost pixels belonging to the lip class. The result of mouth corner localization is shown in Figure 12(d) .

4 Experimental evaluation

The proposed method has been tested on the CDS001 dataset of the XM2VTS database [21], which has been used for testing in many facial feature detection papers. This dataset contains 1180 head and shoulder color images of 295 persons (four frontal images per person), each image being of dimensions 720×576 pixels and depicting a single person. The four recording sessions took place one month apart from each other. In more than one third of the images, people with eyeglasses are depicted. All images were acquired under controlled illumination conditions and the background was uniform. Ground truth for eye centers and mouth corners is provided. A few sample images from the database can be seen in Figure 13.

Our method has also been tested on the BioID database [22], which contains 1521 grayscale, frontal facial images of dimensions 384×286, acquired under various lighting conditions in a complex background. The database contains

tilted and rotated faces, people that wear eye-glasses and, in a few cases, people that have their eyes shut or pose various expressions. Thus, it is considered as one of the most challenging databases for the facial feature detection task. Examples of images from the BioID database are shown in Figure 14. Ground truth for eye positions is provided in the database. It should be noted here that the BioID database could not be used in our case for mouth corner localization, since it includes only greyscale images whereas our mouth corner localization technique requires color images.

Out of a total of 1180 images, only 3 faces failed to be detected when using the face detector [3] on the XM2VTS database. However, in many cases, more than one faces were erroneously detected. In this case, the proposed feature detection technique was used to eliminate the falsely detected faces. More specifically, for each candidate face detected, the sum of the distance metric (3) for the left and right eye and the distance metric (4) for the detected mouth was evaluated and the candidate with the smallest sum was retained, while the other candidate regions were rejected. By doing so, all 31 falsely detected faces were discarded. Similarly, out of a total of 1521 images, 18 faces failed to be detected on the BioID database. The simple approach outlined above was used to discard the erroneously detected faces (false alarms). A typical example of such a case is shown in figure 15.

For eye region detection, success or failure was declared depending on whether the ground truth for both eye centers was in the found eye regions. Mouth region detection was considered successful if both ground truth mouth corners were inside the region found. For the eye center localization, the correct detection rates were calculated through the following criterion, introduced in

[9]:

$$m_{e2} = \frac{max(d_{e1}, d_{e2})}{s_e} < T \tag{13}$$

In the previous formula, d_{e1} and d_{e2} are the distances between the eye centers in the ground truth and the eye centers found by the algorithm, and s_e is the distance between the two ground truth eye centers. A successful detection is declared whenever m_{e2} is lower than threshold T. The same formula was also used for mouth corner localization, with d_{m1} and d_{m2} being the distances between the ground truth mouth corners and the mouth corners found by the algorithm, and s_m the distance between the ground truth mouth corners:

$$m_{m2} = \frac{max(d_{m1}, d_{m2})}{s_m} < T \tag{14}$$

In order to measure the overall facial feature localization accuracy, an error metric which is analogous to the criterion used in [8] was adopted. This metric is equal to the average of the distances of each found feature, from the corresponding ground truth feature, normalized by the inter-ocular distance of the ground truth data:

$$m_{me4} = \frac{1}{4s_e}(d_{e1} + d_{e2} + d_{m1} + d_{m2}) < T \tag{15}$$

For all three cases, the success rate is defined as the percentage of images where a successful detection has been performed according to (13), (14), (15). Some results of the proposed method can be seen in Figure 24. It should be noted that all the successful detection figures (percentages) presented in the subsections below take into account the success rate of the face detector. In other words, cases where a face is not detected from the face detector are considered as failures for the facial feature detectors. If the results have been

evaluated on correctly detected faces only, the successful detection figures would have been somewhat higher.

4.1 Eye detection and eye center localization

Correct eye region detection percentages for the XM2VTS and BioID databases are listed in the column denoted "Eye regions" of Tables 1 and 2, respectively. It is obvious that the detection rates are very good both for people not wearing eyeglasses and those who do. The columns labelled "Eye Centers" in the same Tables present correct eye center localization results for threshold values T=0.25 and T=0.1 in (13). As mentioned in [9], a threshold value T=0.25 means that the maximum allowable deviation from the actual eye center positions is half the width of an eye, while T=0.1 means that the maximum error allowed cannot be more than 10% of the inter-ocular distance. Results verify that the proposed method can achieve very good eye center localization even in the challenging BioID database where, as expected, results are worse than in the XM2VTS database. The results are, in general, very good even for people wearing glasses. This is due to the fact that the basic shape of the eye region remains unchanged, even if the eyebrows are occluded because of the glasses frame. In such cases, the geometry of the eyebrows is replaced by the upper part of the frame of the glasses, which makes eye region detection easy. The lower part of the eye glasses frame is usually too far from the eyes to affect the distance vector field of the area and, thus, distort the results.

Furthermore, the success rates for various values of the threshold T, for the entire databases, as well as for the two subsets of people not wearing eyeglasses and those who do are depicted in Figures 16 and 17 for the XM2VTS

and BioID, respectively. It can be observed that, even for very small thresholds T (i.e. for very strict criteria), success rates remain very high, especially for the XM2VTS database. For example, as can be seen in Figure 16(a), the maximum distance of the detected eye centers from the real ones does not exceed 5% (T=0.05) of the inter-ocular distance in 92.4% of the cases in the XM2VTS database, which means that the algorithm can detect eye centers very accurately. Figures 18 and 19 show the distribution of errors, i.e. the histogram of the error values m_{e_2}, as they were defined in (13), for the XM2VTS and BioID databases, respectively. The range of these values has been quantized to 1000 bins. The mean value of m_{e2} for XM2VTS is 0.027, that is, the mean maximum error $max(d_{e1}, d_{e2})$ for both eyes is only 2.7% of the actual inter-ocular distance s_e. The corresponding figure for the BioID is 0.063.

4.2 Mouth detection and mouth corner localization

The mouth region was correctly detected in 96.1% of the cases in the XM2VTS database. The mouth corner localization success rates in the same database for T=0.25 and T=0.1 in (14) are 97.2% and 80.6% respectively. Figure 20 shows the success rates of mouth corner localization for various T. It is obvious that the method has very good performance in detecting the mouth and localizing its corners. The fact that success rates in this case are lower than in the case of eye center localization can be attributed to the fact that the mouth corner error $max(d_{m1}, d_{m2})$ is normalized by the mouth corner distance s_m, which is smaller than the inter-ocular distance s_e. Consequently, the error m_{m2} in (14) obtains larger values than the error m_{e2} in (13).

Figure 21 shows the histogram of the error values m_{m2} in (14). The mean

value of m_{m2} is 0.083, i.e. the mean maximum error $max(d_{m1}, d_{m2})$ for both mouth corners is 8.3% of the actual distance between the mouth corners.

4.3 Aggregate results

The ability of the system to detect simultaneously all three regions of interest (left/right eye and mouth) and localize all four characteristic points (eye centers and mouth corners) was also experimentally tested in the XM2VTS database. No aggregate results could be obtained for the BioID database since our mouth corner localization method requires color images. For feature region detection, success was declared when all three regions were correctly detected. In the experiments, all three feature regions were successfully detected at 95.48% of the cases. Since, in most of the cases, erroneously detected regions were very close to the characteristic points, the localization of the characteristic points was even more satisfying, giving a success rate of 98.5% for T=0.25 in (15). Figure 22 shows the results for various thresholds.

Figure 23 shows the histogram of the error value m_{me4}. The mean value of m_{me4} is 0.054, i.e. the mean average error for all four characteristic points is 5.4% of the inter-ocular distance.

4.4 Comparison with other methods

The method has been compared with other existing methods that were tested by the corresponding authors on the same databases for the eye center localization task (Tables 1, 2). Unfortunately, no mouth corner detection method tested on the XM2VTS database has been found. Moreover, eye detection and

localization results for our method are provided only for the BioID database, since our method requires color information for mouth detection. All figures presented for other methods have been obtained from the corresponding papers. Some of these figures have been derived from plots included in these papers. These approximate figures are preceded by the "$\sim$" symbol.

In the XM2VTS, for T=0.25 our method achieves an overall detection rate of 98.85%, which is essentially equivalent with the results obtained by the method in [8] ($\sim$ 99%) while Jesorsky *et al* in [9] achieve 98.4%. The proposed method is significantly superior than the other methods for stricter criteria, i.e. for smaller values of the threshold T. For T=0.1, both [9] and [8] achieve a success rate of 93%, while the proposed method localizes the eye centers successfully in 98.14% of the cases.

In the BioID database, for T=0.25, our method achieves an overall detection rate of 96%, i.e. equal to the detection rate in [8] while all three other methods achieved inferior results. For T=0.1, our method achieves a detection rate of 89.42%, and is surpassed only by the method in [8] that achieves 96%. The rest of the methods under comparison achieve detection rates significantly lower than the proposed method. The fact that our method scores lower detection rates for T=0.1 can be largely attributed to the fact that the BioID database contains two subjects (each being depicted into approximately 50 images) where the method continuously fails to achieve m_{e2} values lower than 0.1, due to the thick eye-glasses worn by these subjects.

As far as computational complexity is concerned, the time required from our method to detect eye/mouth regions and localize eye centers and mouth corners was on average 260ms per facial image for both databases whereas the

time for executing the face detection step on the XM2VTS database was 230ms, on a Pentium M processor running at 1.60GHz. It should be noted that no optimization of any sort has been performed on the code. Moreover, by further restricting the size of the areas where eyes and mouths are searched for, one can reduce dramatically the amount of time needed for facial features localization while keeping success rates high.

The method in [9] reports 23.5ms on a Pentium 3, 850Mhz for the coarse localization of the face region and 7ms for the refinement that detects exact positions of facial points. The authors do not clarify whether these execution times refer to the XM2VTS or the BioID database. The method in [8] takes 1.4sec on a Pentium 3, 500Mhz for the whole procedure (face detection and localization of 17 facial features) on BioID images, while facial feature detection alone needs more than 800ms on the same processor. No computational complexity results are provided in [10] and [11].

5 Conclusions

A novel method for facial feature detection and localization was proposed in this paper. The method utilizes the distance vector field that is formed by assigning to each pixel a vector pointing to the closest edge, thus, encoding, the geometry of such regions. Distance vector fields employ geometrical information and thus can help to avoid illumination problems in the critical step of eye and mouth region detection. Once facial feature areas are detected, luminance and chromatic information is exploited for accurate localization of characteristic points, namely the eye centers and mouth corners within these areas. Eye center localization is based on the fact that this center resides in the

middle of a small patch with strong edges, as well as on the fact that the iris of the eye is the darkest area in the eye region. Furthermore, for mouth corner localization, the hue component, that can distinguish the lip region from adjacent areas, is utilized. The method proved to give very accurate results, failing only at extreme cases.

Acknowledgment

This work has been partially supported by the FP6 European Union Network of Excellence MUSCLE "Multimedia Understanding Through Semantic Computation and Learning" (FP6-507752).

References

[1] M.A. Bhuiyan, V. Ampornaramveth, S. Muto, H. Ueno, Face Detection and Facial Feature Localization for Human-Machine Interface, National Institute of Informatics Journal, 5 (3) (2003) 25-39.

[2] F.Y. Shih, Chao-Fa Chuang, Automatic extraction of head and face boundaries and facial features, Information Sciences Informatics and Computer Science: An International Journal, 158 (1) (2004) 117-130.

[3] P. Viola, M. Jones, Robust real-time face detection, International Journal of Computer Vision, 57 (2) (2004) 137-154.

[4] V. Perlibakas, Automatical detection of face features and exact face contour, Pattern Recognition Letters, 24 (16) (2003) 2977-2985.

[5] C.C. Han, H.Y.M. Liao, G.J. Yu, L.H. Chen, Fast face detection via

morphology-based pre-processing, Pattern Recognition, 33 (10) (2000) 1701-1712.

[6] S.C.Y. Chan, P.H. Lewis, A pre-filter enabling fast frontal face detection, 3rd International Conference on Visual Information and Information Systems (1999) 777-784.

[7] J.L. Crowley, N. Gourier, D. Hall, Facial Features Detection Robust to Pose, Illumination and Identity, International Conference on Systems Man and Cybernetics (2004) 617-622

[8] D. Cristinacce, T. Cootes, I. Scott, A multi-stage approach to facial feature detection, 15th British Machine Vision Conference (2004) 231-240.

[9] O. Jesorsky, K.J. Kirchberg, R. W. Frischholz, Robust face detection using the hausdorff distance, 3rd International Conference on Audio and Video-based Biometric Person Authentication (2001) 90-95.

[10] Z.H. Zhou, X. Geng, Projection functions for eye detection, Pattern Recognition, 37 (5) (2004) 1049-1056.

[11] J. Wu, Z.-H.Zhou, Efficient face candidates selector for face detection, Pattern Recognition, 36 (5) (2003) 1175-1186.

[12] J.-G. Wang, E.Sung, Frontal-view face detection and facial feature extraction using color and morphological operations, Pattern Recognition Letters 20(10) (1999) 1053-1068

[13] S. Asteriadis, N. Nikolaidis, A. Hajdu, I. Pitas, A Novel Eye detection Algorithm Utilizing Edge-related Geometrical Information, 14th European Signal Processing Conference (2006).

[14] P.E. Danielsson, Euclidean Distance Mapping, Computer Graphics and Image Processing, 14 (3) (1980) 227-248

[15] R. Satherley, M.W. Jones, Vector-City Vector Distance Transform, Computer Vision and Image Understanding, 82 (3) (2001) 238-254.

[16] G. Zachmann, E. Langetepe, Geometric data structures for computer graphics, Proc. of ACM SIGGRAPH, Transactions of Graphics (2003)

[17] H. Breu, J. Gil, D. Kirkpatrick, M. Werman, Linear Time Euclidean Distance Transform Algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence, 17 (5) (1995) 529-533.

[18] J. Canny, A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, 8 (6) (1986) 679-698.

[19] N. Otsu, A Threshold Selection Method from Gray-Level Histograms, IEEE Transactions on Systems, Man, and Cybernetics, 9 (1) (1979) 62-66.

[20] K.V. Mardia, Statistics of directional data, Academic Press (1972).

[21] K. Messer, J. Matas, J. Kittler, J. Luettin, G. Maitre, XM2VTSDB, The extended M2VTS database, 2nd International Conference on Audio and Video-based Biometric Person Authentication (1999) 72-77.

[22] The BioID face database:
http://www.bioid.com/downloads/facedb/facedatabase.html.

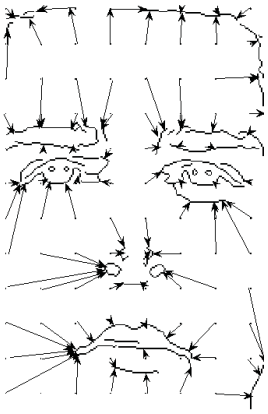


Fig. 1. Schematic representation of the distance vector field of a face.

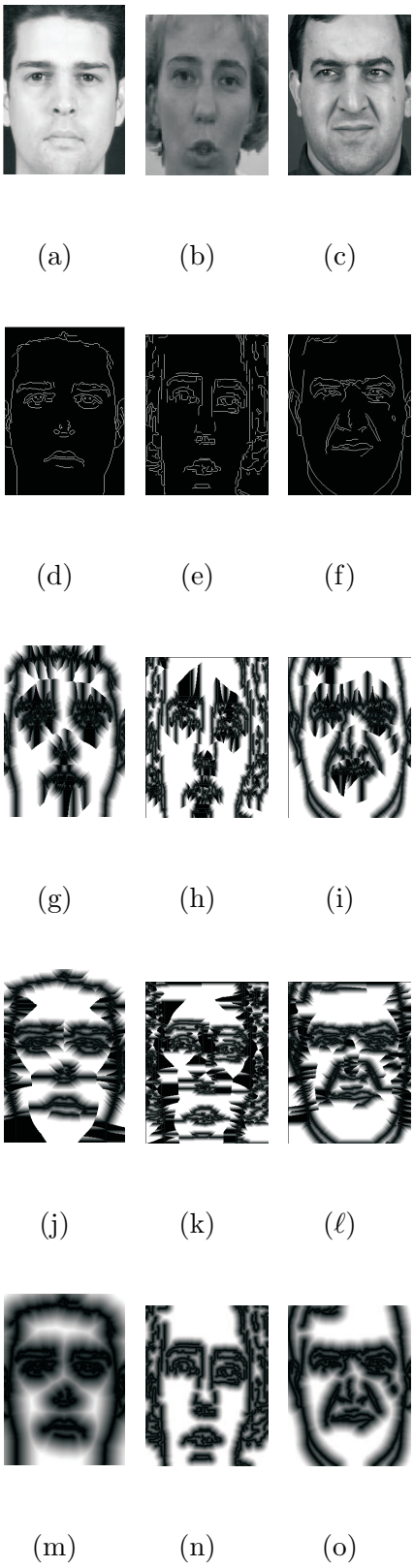
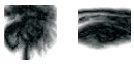


Fig. 2. (a-c) Detected face; (d-f) Canny edge map; (g-i) absolute values of horizontal components of distance vector field; (j-l) absolute values of vertical components of distance vector field; (m-o) distance map

(a) (b) (c) (d)

Fig. 3. Example of distance vector field extraction for an eye region a) eye image; b) distance map; c) vertical components of the vector field; d) horizontal components of the vector field.

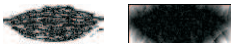


(a) (b)

Fig. 4. a) Mean horizontal component map of right eye; b) mean vertical component map of right eye.

(a) (b) (c) (d)

Fig. 5. Example of vector coordinate map extraction for the mouth region a) mouth area; b) canny edge map of mouth area; c) vertical coordinate map of the DVF; d) horizontal coordinate map of the DVF.

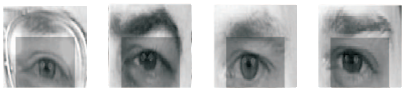


(a) (b)

Fig. 6. a) Vertical and b) horizontal coordinates for mouth mean distance vector field.

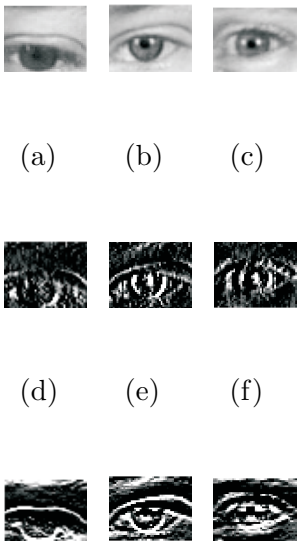
(a) (b) (c) (d)

Fig. 7. Light reflections removal procedure: a) eye region; b) binary eye region; c) binary eye region with small connected areas removed; d) grayscale image with reflections removed.



(a) (b) (c) (d)

Fig. 8. Examples of detected eye regions and eye centers search areas within them.



(a) (b) (c)

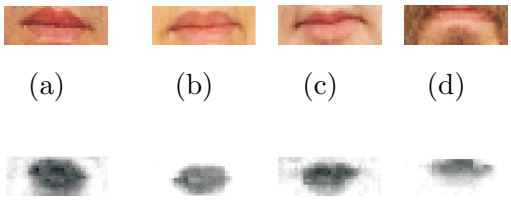
(d) (e) (f)

(g) (h) (i)

Fig. 9. Examples of eye center search areas (a-c) and their vertical (d-f) and horizontal (g-i) derivative images.

(a) (b) (c) (d) (e)

Fig. 10. Steps followed for the detection of the eye center a) Initial estimate of eye center; b),c) regions used to get refined estimates; d) estimate after first refinement; e) final eye center localization.



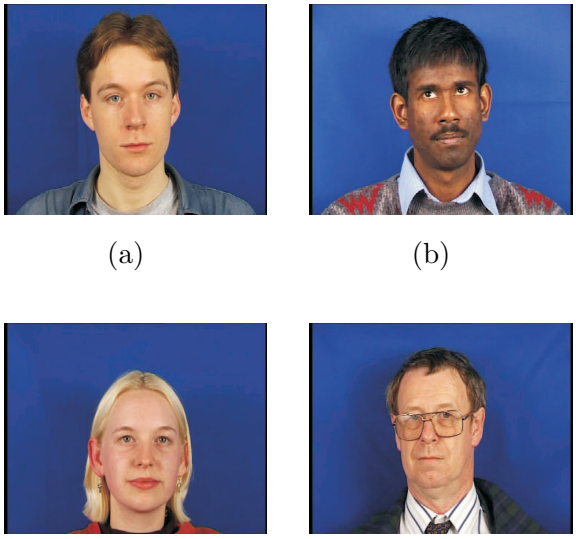
(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 11. Mouth regions detected (a-d) and the corresponding hue component values (e-h).

(a) (b) (c) (d)

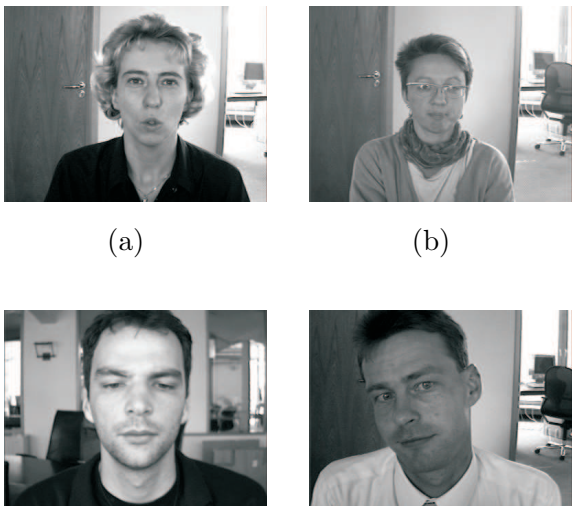
Fig. 12. a) Detected mouth region; b) binary image with small connected components that correspond to skin and have been falsely assigned to the foreground; c) binary image without the small connected components; d) mouth corners.



(a) (b)

(c) (d)

Fig. 13. Examples from the XM2VTS database.



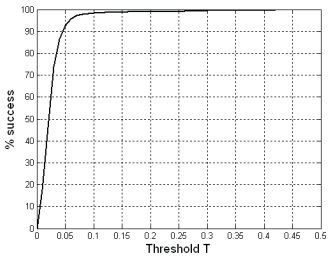
(a) (b)

(c) (d)

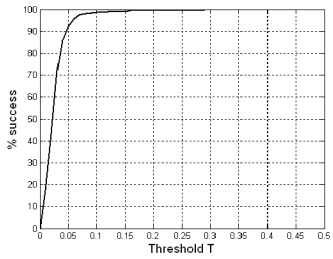
Fig. 14. Examples from the BioID database.

(a) (b)

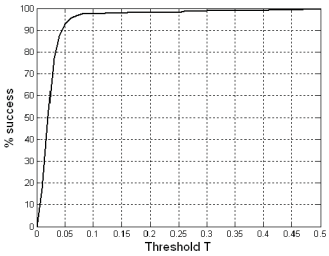
Fig. 15. Example of false alarm removal in the BioID database a) Initial face detection using the method in [3]. Two face regions are detected, including a false alarm; b) removal of falsely detected face region using the proposed facial feature detection method as a verification step.



(a)

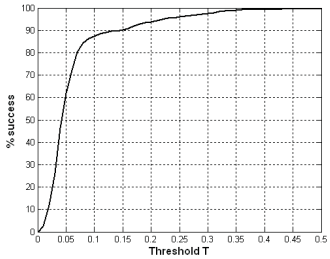


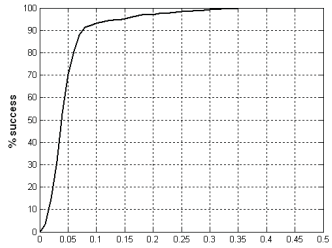
(b)



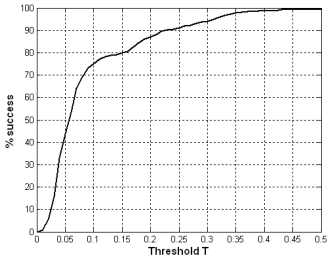
(c)

Fig. 16. Eye center localization for various thresholds T a) for the entire XM2VTS dataset; b) for images depicting people without eyeglasses; c) for images depicting people with eyeglasses.





(a) (b)



(c)

Fig. 17. Eye center localization for various thresholds T a) for the entire BioID dataset; b) for images depicting people without eyeglasses; c) for images depicting people with eyeglasses.

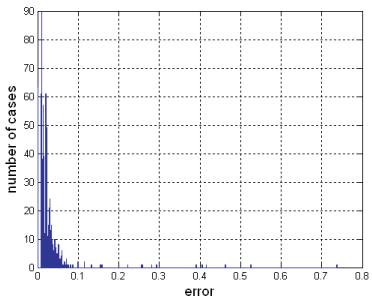


Fig. 18. Distribution of errors for eye center localization on the XM2VTS database.

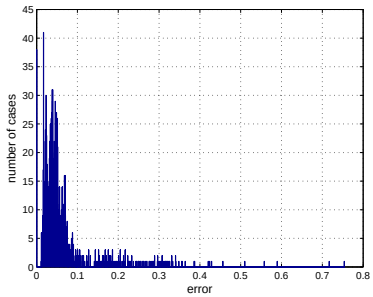


Fig. 19. Distribution of errors for eye center localization on the BioID database.

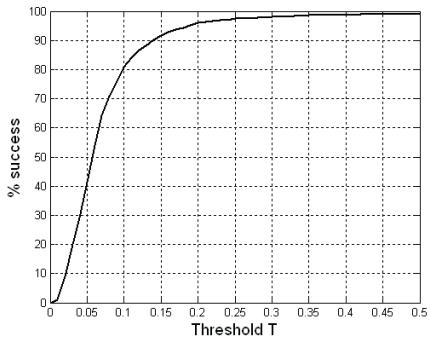


Fig. 20. Mouth corner localization for various thresholds T for the XM2VTS database.

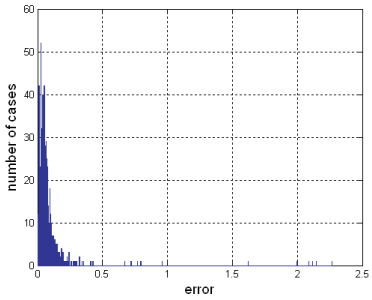


Fig. 21. Distribution of errors m_{m2} for mouth corner localization on the XM2VTS database.

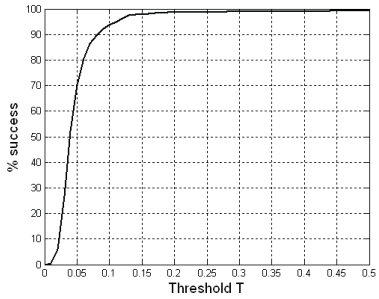


Fig. 22. Localization of all four characteristic points (eye centers and mouth corners) for various thresholds T for the entire XM2VTS database.

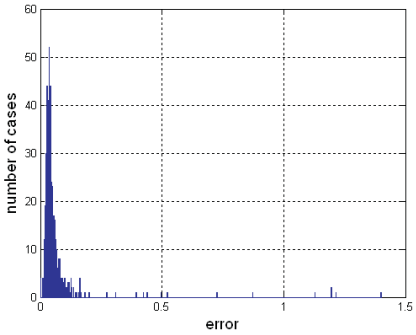


Fig. 23. Distribution of errors m_{me4} in the localization of all 4 characteristic points on the XM2VTS database.

Fig. 24. Successfully (a)-(f) and erroneously (g)-(h) detected facial features on XM2VTS images.

Table 1

Eye region detection and eye centers localization results on the XM2VTS database.

	Eye Regions	**Eye Centers, T=0.25**	**Eye Centers, T=0.1**
Proposed method, people without glasses	99.06%	99.33%	98.54%
Proposed method, people with glasses	98.7%	97.92%	97.44%
Proposed method, total	98.93%	98.85%	98.14%
Method in [9]	-	98.4%	$\sim$ 93%
Method in [8]	-	$\sim$ 99%	93%

Table 2

Eye region detection and eye centers localization results on the BioID database.

	Eye Regions	**Eye Centers, T=0.25**	**Eye Centers, T=0.1**
Proposed method, people without glasses	99.50%	98.74%	94.30%
Proposed method, people with glasses	98.3%	90.95%	75.93%
Proposed method, total	99.10%	96.00%	89.42%
Method in [9]	-	91.8%	79.0%
Method in [8]	-	$\sim$ 96%	96.0%
Method in [10]	-	94.81%	-
Method in [11]	-	94.5%	$\sim$ 53%