Algorithms Fusion for Face Localization

R. BELAROUSSI

L. PREVOST

M. MILGRAM

Institute of Intelligent Systems and Robotics—PRC
University Pierre and Marie Curie

Face localization is a face detection problem where the number of people is known. We present a comparison between different algorithms fusion methods dedicated to the localization of faces in color images. Data to combine result from an appearance model supported by an auto-associative network, an ellipse model based on Generalized Hough Transform, and a skin color model. We introduce and compare several fusion methods like the Bayesian classifier with parametric or non-parametric technique, a fuzzy inference system, and a weighted average. Given an input image, we compute a kind of probability map on it using a sliding window. The face position is then determined as the location of the absolute maximum over this map. Improvement of basic detectors localization rates is clearly shown and prevalence of the weighted average is reported.

Manuscript submitted December 29, 2004; revised April 2, 2006.

Refereeing of this contribution was handled by Associate Editor Alexander Toet.

Authors' address: University Pierre and Marie Curie–Paris VI, LISIF-PARC, BC252, 4 place Jussieu 75252 Paris cedex 05 France, E-mail: (maurice.milgram {lionel.prevost}@upmc.fr).

1. INTRODUCTION

Face detection in an image has become a very important issue for many applications such as biometric, presence detection, video-conferencing, visiophony, indexation, car driver monitoring, virtual reality, lips reading, gaze tracking. Because of the high variability of the pattern to be detected, face detection without any hypothesis is a tough task [38]. Fixed camera and known background, use of motion information [6], strong hypothesis on the face location [20], scale or pose [33], special background for an easy extraction of the silhouette [24] or special lighting conditions (reflected infrared [9] or thermal infrared [11]): face detection applications often start with making assumptions. The face localization issue [4, 17, 20, 33] can be regarded as a face detection problem knowing the number of faces in the image. The location of the faces in the image—position and extent—is searched. The face localization issue is addressed in the present paper. It is not simpler without additional assumption.

A wide variety of works have been reported in face detection, much more than for face localization. Structural and holistic approaches, common in Pattern Recognition, are applied. Structural approaches try to detect facial landmarks (eyes, mouth, nose, head contour) and combines the results using models [3] or constellation analysis [2]. [3] built a generic model of the face through a joint distribution of parts (features models) positions. [12] brings a matching algorithm for pictorial structures (models of parts and connection between parts) applied to representation of an articulated human body. In [37] a hierarchical knowledge-based method finds face candidates at a low resolution and verifies presence of eyes and mouth at a high resolution. [39] uses deformable templates using a radiometrical model of eyes. For each facial feature, a statistical (GMM) model of Gabor filters responses is built in [17]. Features are detected over the whole image, then similarity with a constellation model is computed on a scanning-window, resulting in a coarse face localization. Then a cascade of two boosted SVM gives the accurate location of the face. In [2] component classifiers (SVM) are trained over selected parts of the face (bridge of the nose, nose, eyebrows, eyes, cheek, mouth): a constellations analysis performs face/non-face classification on a scanning window at several scales. [4] implements a similar approach at three scales with a skin/mouth color segmentation pre-processing. SVM are also used to model eyes and mouth in [33] at only one scale (corresponding to face's size).

Holistic approaches of face detection process a subimage of the input image into a feature vector (momentum, projection, gray level, wavelet…). These approaches estimate the classifier parameters on a training set, usually using a boosting procedure. Parameters can be weights of neural networks [16], [29], of weak classifiers [36] as described in Section 6.3 or terms of

a covariance matrix (statistical classifier) [34]. As in many detection issues, it is almost impossible to define the opposite class, the non-face patterns, which drives researchers to choose the model-based approach. A model does not require counter examples [13], which may seems an advantage but actually decreases classifier efficiency: generalization in a high dimension space (221 for 13×17 sub-images) is tough without knowing where are the patterns that might be confused. Another way is to design a combination of several detectors (classifiers). [13] and [14] did it to perform face detection. In [13] uses a conditional mixture of constraint generative models (Diabolo see Section 4) trained on different ranges of face orientation. Product and sum rules are used in [14] to combine two detectors based on edge orientation (edge orientation matching and Generalized Hough Transform) and one based on gray levels (Sparse Network of Winnows). Classifier combination has also been used in character [27, 28] and face recognition [7].

Our approach makes co-operate holistic and structural approaches: it is quite different but related to [14]. Generalization capability of a single classifier is limited, especially in a high dimension space. A more reliable decision can be obtained by combining output of several experts [27]: the face localization issue is divided in sub-problems easier to deal with. Various information is extracted from the same image using different kind of detectors. Some try to model global features while the others concentrate on structural features. Each face cues are searched by a relevant expert: elliptical shape, global appearance and skin color. Cooperation between experts exploits their complementarities and can also handle conflicts between sources.

An auto-associator network appearance based model and an ellipse detector are based on the image gradient's direction. A luminance-free skin color model is also implemented. The combination of these three detectors is done via various methods that are compared: Bayesian classifier, fuzzy inference system and neural networks.

Section 2 describes the skin color model, Section 3 details the ellipse model, and Section 4 deals with the appearance based model. Several combination strategies are presented in Section 5. Comparison of the combinations is detailed in Section 6 along with our experimental results and the contribution of the combination to the face localization problem. The last section is devoted to conclusions and prospects.

2. SKIN COLOR MODEL

Skin color classification aims at determining whether a color pixel has the color of flesh or not. Such a classification should overcome difficulties like different skin tones (white, pink, yellow, brown, black…) and scene illuminations, and the fact that background pixels can have the same color as a flesh type.

2.1. Color Spaces Definition

Two color spaces are investigated for skin color classification: HSV and YCbCr. These spaces are commonly used [26] in image processing for they are expected to be more robust to lighting condition by separating chrominance (color information) and luminance (grayscale levels) information. In a video signal, color images encoding separates the luminance and chrominance information: this way television standards (NSTC, Pal, Secam) ensured backward compatibility with black and white television. Chrominance is the color information to be added to the grayscale information to obtain a color image in red, green and blue primary colors. Chrominance information is widely used for skin color classification as it is expected to be a common cue between different skin tones contrarly to the luminance. Skin color classifiers based on chrominance tend to be more robust to different lighting conditions.

RGB conversion to YCbCr is linear (see (1))

$$\begin{aligned} \mathrm{Y} &= 0.299\mathrm{R} + 0.587\mathrm{G} + 0.114\mathrm{B} \\ \mathrm{Cb} &= 0.564(\mathrm{B} - \mathrm{Y}) + 128 \\ \mathrm{Cr} &= 0.713(\mathrm{R} - \mathrm{Y}) + 128. \end{aligned} \tag{1}$$

Y channel is the luminance, Cb and Cr channels represent chrominance. We used the definition of [19], it uses an RGB model that fits the phosphor emission characteristics of older cathode ray tubes. Y, Cb and Cr values range from 0 to 255. Variants of this definition that fit the phosphor emission characteristics of newer tubes and other modern display equipment can be found. YPbPr, YUV, YIQ are same or similar color spaces.

HSV space is a non-linear transformation of RGB space (see (2)): colors are defined by hue (H channel), saturation (S channel) and luminance (V channel)

$$\mathrm{V} = \max(\mathrm{R},\mathrm{G},\mathrm{B}), \qquad \mathrm{S} = 255\frac{\mathrm{V} - \min(\mathrm{R},\mathrm{G},\mathrm{B})}{V}$$

$$\mathrm{H} = \begin{cases} 30\dfrac{\mathrm{G}-\mathrm{B}}{\mathrm{S}} & \text{if} \quad \mathrm{V} = \mathrm{R} \\ 30\dfrac{\mathrm{B}-\mathrm{R}}{\mathrm{S}} + 90 & \text{if} \quad \mathrm{V} = \mathrm{G} \\ 30\dfrac{\mathrm{R}-\mathrm{G}}{\mathrm{S}} + 120 & \text{if} \quad \mathrm{V} = \mathrm{B} \end{cases}. \tag{2}$$

S ranges from 0 to 255, and represents the grayness of the color: the lower the saturation of a color is the more faded it appears (a monochrome color corresponds to S = 0). H values are defined modulo 180 from red (H = 0) through yellow, green, cyan, blue, and magenta, and returns to red (H = 180). Similar color spaces include HSB, HLS, and HIS.

2.2. Skin Color Pixels Classification

A recent comparison of different skin color classification algorithms can be found in [26]. Linear [1, 5,

TABLE I
Training and Validation Sets for Skin Segmentation

Dataset	No. Images	Skin Pixels	Non-Skin Pixels
Training	500	18.2 million	120.9 million
Test	550	23 million	136.6 million

TABLE II
Confusion Matrix of the CbCr Fixed Range Skin Classifier

Classification Class	Skin	Non-Skin
Skin	77%	23%
Non-skin	17%	83%

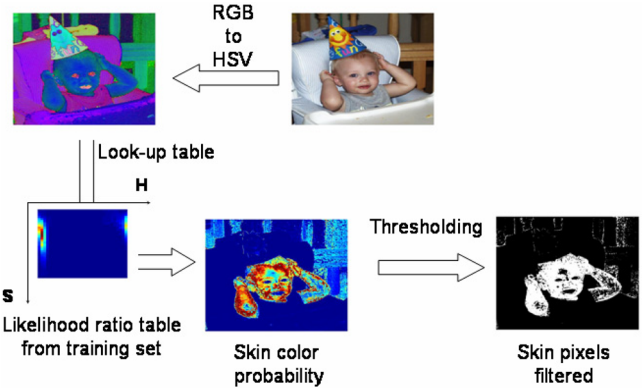


Fig. 1. Back project of the histogram ratio.

6] and Bayesian classifiers [21, 26] are proposed and compared in the present paper.

1050 images of the ECU database described in Section 6.1 are used for training and assessment of skin segmentation methods presented in the following subsection.

Repartition of the two sets is summarized in Table I. These images are not used for face localization tests (Section 6.2 and 6.3).

2.2.1. Rectangular Boundary in CbCr Plane

Linear classification uses a piecewise linear decision boundary in the Cb-Cr plane. The following fixed-range in Cb and in Cr is used to define skin color pixels:

$$\mathrm{Cb} \in [100\ \ 130] \qquad \text{and} \qquad \mathrm{Cr} \in [135\ \ 165].$$

These thresholds were experimentally tuned using images with people. Skin being characterized by specific chrominance information, the filter can be applied to any ethnic skin color but our threshold is not universal because the chrominance component is actually related to the luminance value Y [18]. In poor or bright illumination condition the filtered components are spurious and in some cases no skin at all is filtered: this skin detector is coarse but simple and we use it as a reference for comparison with other skin classifiers.

This classifier results in a one point ROC curve (see Fig. 3): Table II is the confusion matrix obtained over the validation set.

2.2.2. Statistical Classifiers

The Bayesian decision rule is a popular method in statistical pattern classification [10]. A color pixel $\vec{X}$ is classified as a skin pixel if its likelihood ratio is higher than a threshold:

$$\frac{P(\vec{X} \mid \text{skin})}{P(\vec{X} \mid \text{non-skin})} \geq \tau \tag{3}$$

$P(\vec{X} \mid \text{skin})$ and $P(\vec{X} \mid \text{non-skin})$ are the conditional probability density functions (denoted pdf in this paper) of respectively skin and non-skin color. τ is the decision threshold. A given τ value results in a confusion matrix: ROC curve of the classifier is obtained by varying the threshold τ.

The computation of the pdfs is done using the histogram technique. In [31] face color is tracked using this technique. Statistical repartition of skin pixels in HS plane (or CbCr plane) is calculated in a 2D histogram. Scaling the histogram results in $P(\vec{X} \mid \text{skin})$. Same operation is done with non-skin pixels to evaluate $P(\vec{X} \mid \text{non-skin})$.

Ratio of the two histogram results in a likelihood ratio table [31]: skin probability of a color pixel featured by (H, S) values is then computed by look-up table. Back projecting the histogram ratio onto the HSV (or YCbCr) image results in a skin color probability image as shown in Fig. 1.

H and S channels (respectively Cb and Cr channels) feed the 2D skin and non skin histograms. 32 bins per channel are allocated. [21] found that 32 bins are optimal whereas [26] concludes that larger histogram leads to finer pdfs estimation and better performances when training samples are sufficient. As explained in [26], when training set is not large enough, a larger histogram results in a noisier pdf compared to a smaller histogram size. Subsampling their original training set, they found that the 256-bin histogram is more sensitive to the number of training samples compared to the 32-bin histogram. And even with a huge sample number the larger histogram is just a few percent more efficient than a 32-bin histogram, justifying our choice.

In Fig. 2 it appears that skin and non-skin pixels are pretty well-separated in the HS plane. On the opposite, skin and non-skin pixel distributions in CbCr plane are clearly overlapping. Therefore model the non-skin distribution brings poor improvement in HS plane and dramatically increases classification performance in CbCr plane as shown by the ROC curves plotted in Fig. 3.

Assuming the non-skin pixel distribution is uniform, the decision rule in (3) is simplified. A color pixel is classified as skin color if

$$P(\vec{X} \mid \text{skin}) \geq \tau. \tag{4}$$

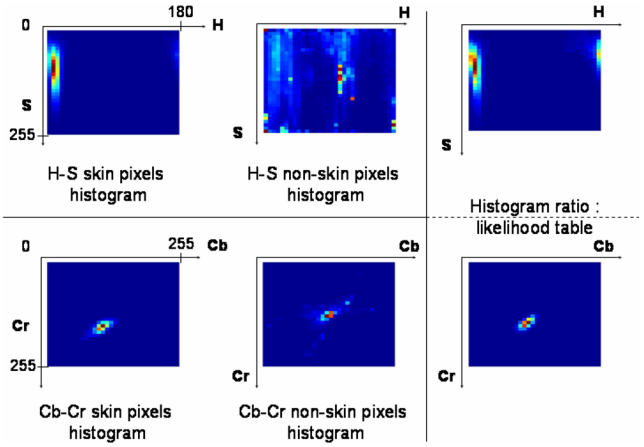


Fig. 2. Skin, non-skin and ratio histogram in HS plane (first line) and CbCr plane (last line): hot colors correspond to high values.

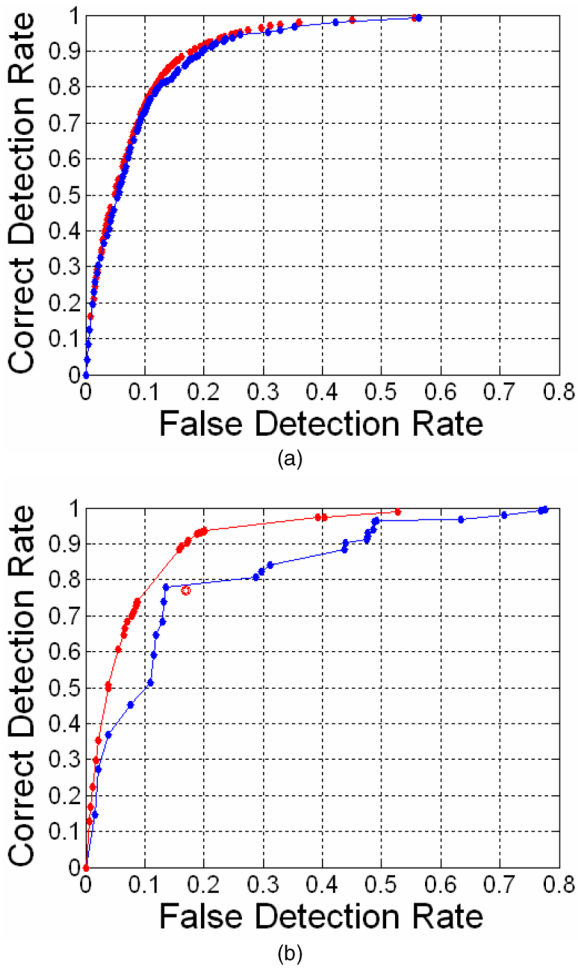


Fig. 3. ROC of skin color classifier in HS space (a) and CbCr space (b).

A classifier based on (4) uses the statistical repartition of skin pixels regardless of non-skin pixels distribution.

Classification performance is represented with the Receiver Operating Characteristic (ROC) curve: skin segmentation performance for a given decision threshold τ is measured in terms of correct detection rate and false detection rate. Correct detection rate is the proportion of skin pixels correctly classified whereas the false detection rate is the proportion of non-skin pixels classified as skin pixels. The ROC curve is obtained by calculating these rates for all coherent τ values.

ROC curve of the classification based on Cb-Cr statistical models of skin and non skin (decision rule (3)) color pixels is plotted in red in Fig. 3(b). Skin classifier based on statistical repartition of skin color pixels (decision rule (4)) in the Cb-Cr plane is plotted in blue in the same figure. The CbCr fixed range classifier ROC point is plotted in red.

ROC curve of CbCr skin model is highly irregular whereas classification that use the likelihood ratio is quite satisfactory compared to state of the art reported by [26]. In [26] the best classification performance is obtained by a Bayesian classifier (decision rule (3)) in the RGB space with the histogram technique: for a false detection rate of 10% a correct detection rate of 82% is reached whereas our classifier correct detection rate is 75% for the same false detection rate. Moreover it appears that modeling the non-skin distribution in the CbCr space is crucial: a classifier only based on the statistical repartition of skin pixels is not really efficient with a correct decision rate of 50% for 10% of false detection.

In Fig. 3(a), ROC curve of the classifier that models both skin and non-skin distributions in the H-S plane is plotted in red. ROC curve of the classifier modeling only the skin distribution in the H-S plane is plotted in blue.

Modeling non-skin distribution in the HS plane only brings a slight improvement of skin color classification performance compared to a classification based on the skin distribution alone. Moreover, modeling the non-skin distribution is not a satisfying approach as non-skin color cannot be defined: such a distribution completely depends on the non-skin training database. ROC of the likelihood ratio classifier in Cb-Cr plane is a bit better than ROC of the classifier based on skin color repartition in H-S plane but it is also more irregular and requires to compute the non-skin pixel distribution in the Cb-Cr plane.

Therefore, we selected the Bayesian classifier in HS space based on the skin color repartition as our skin detector for the multi-scale segmentation of the face in Section 6.3.

The CbCr fixed range classifier is also used for combiners comparison presented in Section 6.2 for its simplicity.

2.3. Skin Detector

For combination purpose (Section 5) each sub-window of the original image is featured by a single value. A retinal approach is implemented after the skin color pixels classification stage. A sliding window of fixed size (13×17) scans the skin filtered image and

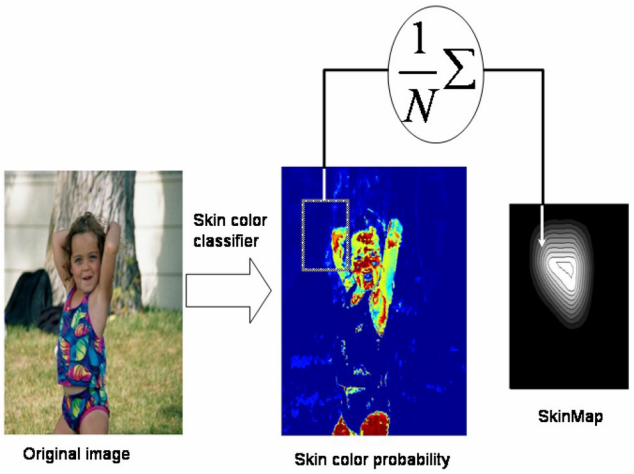


Fig. 4. SkinMap: proportion of skin pixels array.

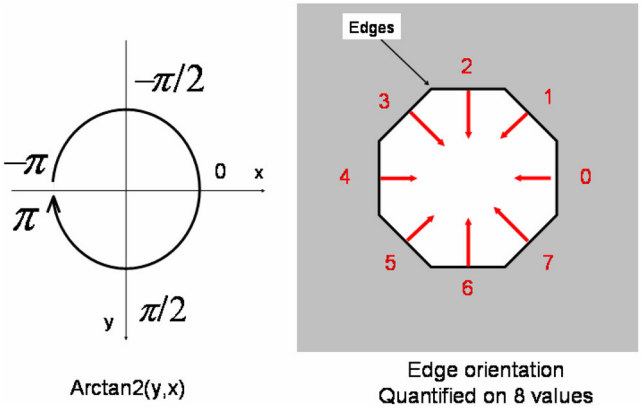


Fig. 5. Four-quadrant inverse tangent and quantification of edge orientation.

calculates the mean skin pixels probability at every position [32] as shown in Fig. 4.

The resulting array is named "SkinMap" and represents the face sub-image probability.

3. ELLIPSE DETECTOR BASED ON GENERALIZED HOUGH TRANSFORM

3.1. Edge Orientation Field

Edge orientation information is processed by an appearance-based model (so called Diabolo see Section 4) and an ellipse detector (Generalized Hough Transform).

Evaluation of the orientation of the gradient on the edges requires a low pass filtering of the image: see Fig. 5. Gradient field is estimated using Roberts masks (2×2), so that horizontal gradient is calculated by $I_x = I_{\text{filtered}} \otimes [1 \ \ -1]$ and vertical gradient with $I_y = I_{\text{filtered}} \otimes \begin{bmatrix} 1 \\ -1 \end{bmatrix}$.

Then the gradient magnitude $= \sqrt{I_x^2 + I_y^2}$ is threshold to define edge pixels. For the generalized Hough transform, a global threshold is applied over the whole input image. Orientations of these edge pixels are then

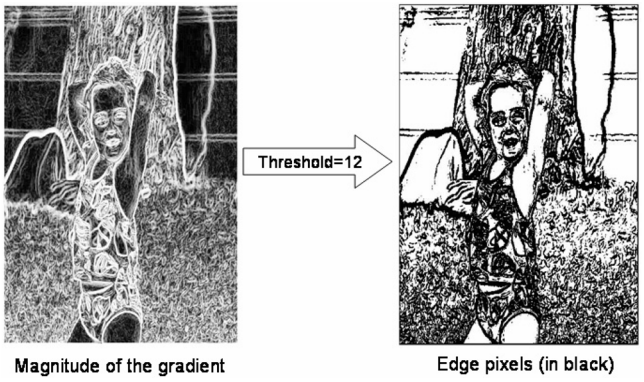


Fig. 6. Threshold of magnitude field defines edge pixels.

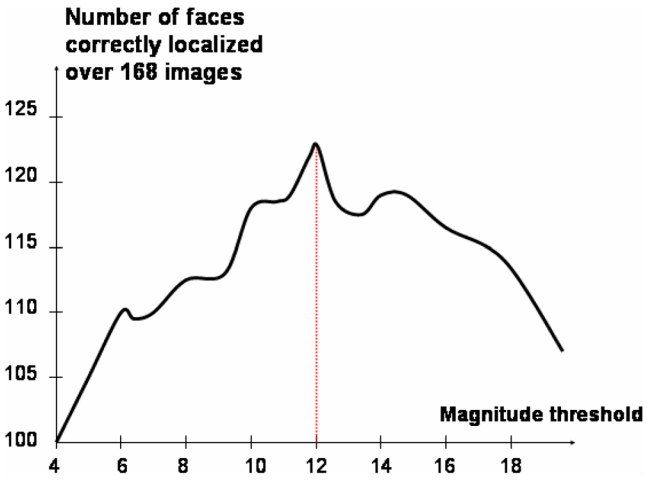


Fig. 7. Magnitude threshold to define edge is tuned to maximize GHT performance over 168 images.

quantized on $N = 36$ values:

$$\text{orien} = \text{round}\left(\frac{N}{2\pi}\arctan 2(I_y, -I_x)\right) \mod N \quad (5)$$

where arctan 2 is the four-quadrant inverse tangent. This function is depicted in Fig. 5 with an ellipse's edge orientation quantified on $N = 8$ values (Freeman chain code) using (5).

For the Generalized Hough Transform edges are defined as pixels with a magnitude greater than a threshold equal to 12: see Fig. 6.

This threshold was tuned over 168 training images containing only one person. This training corpus is not overlapping with the test set used in Section 6. For a given image, a Generalized Hough Transform is computed (see next section) and the maximum of the accumulator is defined as the location of the face. Using ground truth we evaluate the number of faces correctly localized versus threshold, see Fig. 7.

3.2. Ellipse Detector Based on Generalized Hough Transform

The elliptical shape of the face is searched using a Generalized Hough Transform: faces are modeled as vertical ellipses with a specific eccentricity.

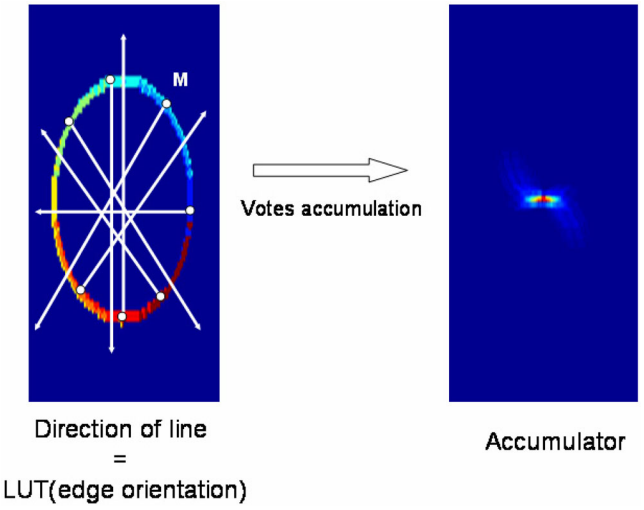


Fig. 8. Generalized Hough Transform in case of an ellipse: half-line votes accumulation.

The gray level dynamic of the input image is first linearly adjusted between 0 and 255. This operation proved to be better than performing histogram equalization. Orientation of the gradient over the whole gray level image is then determined. Then a Generalized Hough Transform (GHT) is performed on the resulting orientation map: the HT constitutes a popular method for extracting geometrical properties [10, 32]. When the edge orientation is used and when it is applied to non parametric curves, the HT becomes the Generalized HT. Each edge pixel votes for all possible location of the shape (actually for the location of the barycentre). For ellipse detection, there is a simplified structure for the GHT based on the geometrical properties of ellipses.

The method consists in casting votes for a half-line starting at each boundary pixel M with an orientation determined by the edge one. The method consists in casting votes for a line through each boundary pixel with an orientation indexed in a look-up table by the edge orientation. We suppose that we know the orientation of the ellipse. So for each point M, a simple look-up table specifies the angle between the tangent Mt (to the boundary) and the radius MO (O is the centre of an ellipse passing through M). Faces are modeled as vertical ellipses with a specific eccentricity so we can build up our look-up table to cast votes from each edge pixel, knowing its gradient orientation. Fig. 8 illustrates an ellipse case: some half-lines are drawn. Each pixel of a line increment a vote array which is the accumulation of all lines votes.

Accumulator maximum corresponds in the image to the position most likely to be the center of an upright ellipse with a horizontal minor axis $a = 8$, and a vertical major axis $b = 10$. Fig. 9 illustrates an example of such an accumulation by Generalized Hough Transform in case of a cluttered scene. Finally, the accumulator is scanned with a 13×17 sliding window and at each position a weighted average of the number of vote is

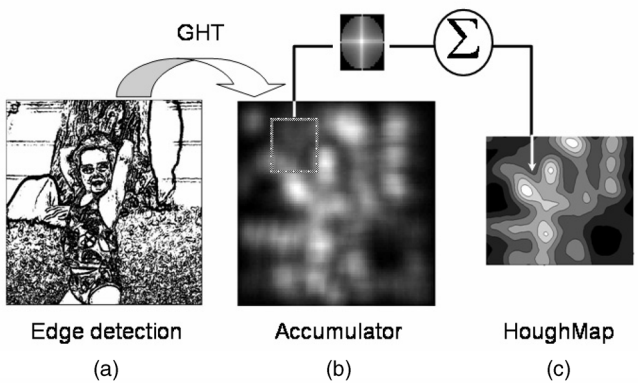


Fig. 9. Edge detection (a), Generalized Hough Transform accumulator (b) computed over gradient orientation of the edge and resulting HoughMap (c).

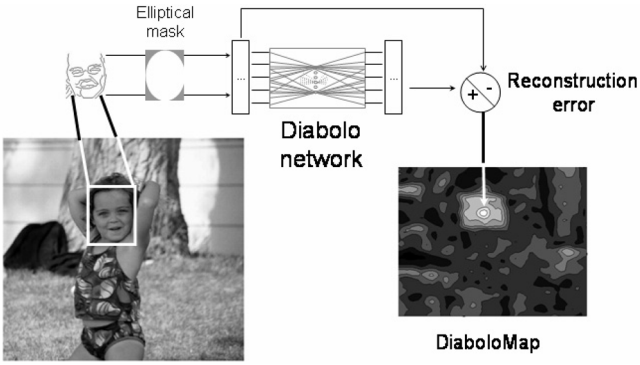


Fig. 10. DiaboloMap: array of reconstruction errors calculated at all positions of the image.

calculated as shown in Fig. 9: the resulting array is named "HoughMap."

4. APPEARANCE-BASED MODEL OF THE FACE

The Diabolo is an auto-associator network: its number of output equals its number of input. It is trained to reconstruct an output identical to its input, and only face examples constitute the training database. It implements a specialized compression for its hidden layer has much less units than input or output does. So a non-face image should be badly compressed and the reconstruction error (square root of the mean square error between the input and the calculated output) would be higher than for a face image. The Diabolo was successfully used for handwritten character recognition [30], face detection [13] and compression [8].

As represented in Fig. 10, reconstruction error is computed on a fixed size window sliding over the entire image. The resulting array is called "DiaboloMap": clear color correspond to small reconstruction error.

Diabolos we implemented have one hidden layer. Hidden neurons have sigmoid activation function, and output neurons have linear activation function, see Fig. 11. The training set is made of face images (see Table III). Faces are various in terms of pose, lighting conditions and skin tones.

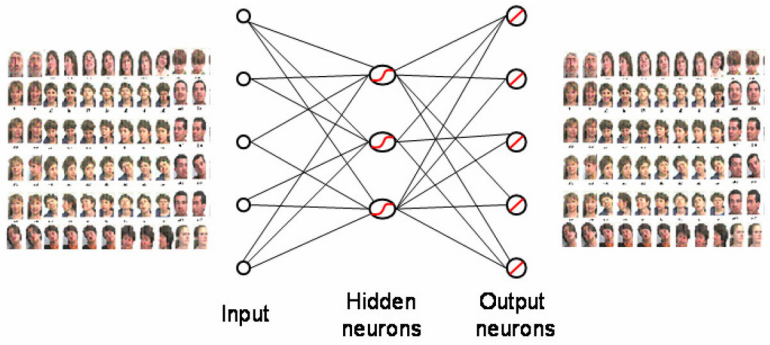


Fig. 11. Architecture of a Diabolo: target is equal to input, training set is made of faces examples.

TABLE III
Training and Cross-Validation Sets of Diabolos

Set	Training	Cross-Validation
No face images	1602	178

The training database is divided into two sets: one for neural networks training, one for the cross-validation and assessment of the best architecture. The cross-validation face samples are extracted from 167 images: amongst these images, 126 images contain only one person and are used as an assessment set to optimize the inputs coding.

Face examples are used to learn parameters (weights) of the neural networks. Training is done using a gradient descent with adaptative learning rate stopped by cross-validation. Gradient descent algorithm is a standard backpropagation in which the network weights are moved along the negative of the gradient of the cost function. The cost function implemented here is the sum over training examples of the square reconstruction error between target and simulation (output calculated by the MLP).

Networks are trained for pattern model: target is equal to the input. Before training the MLP weights must be initialized: a different initialization leads to different weights, therefore to different networks. For a given neural net architecture, several initializations must be tested in order to avoid the network to fall in a local minimum of the performance function different to its global minima.

The Diabolo is fed with a specific coding of edges orientation. Gradient field orientation is quantized on $N = 36$ values as defined in Section 3.1 by (5). Edges are defined by a local magnitude threshold depending on the search sub-window. The threshold is defined over each 13×17 sub-windows of the input image, so that 20% of the pixels are then regarded as edge: an example is given in Fig. 12.

A global threshold over the whole image (face + background) would result in a strong smoothing of the face. A local threshold keeps facial features visible when the sear window is over the face, but it also emphasizes edges over non-face subwindow, which results in lot of false alarms if the face location is defined

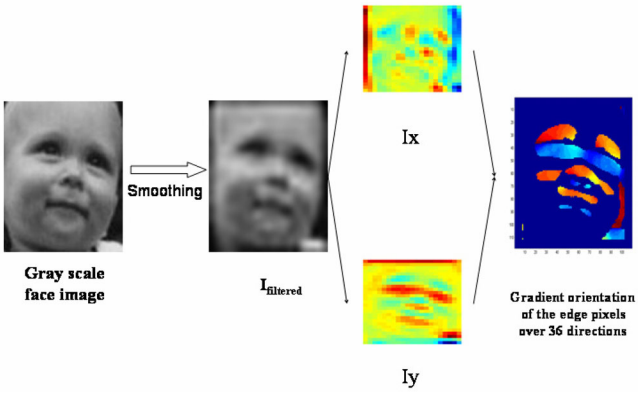


Fig. 12. Estimation of gradient field and edges orientation.

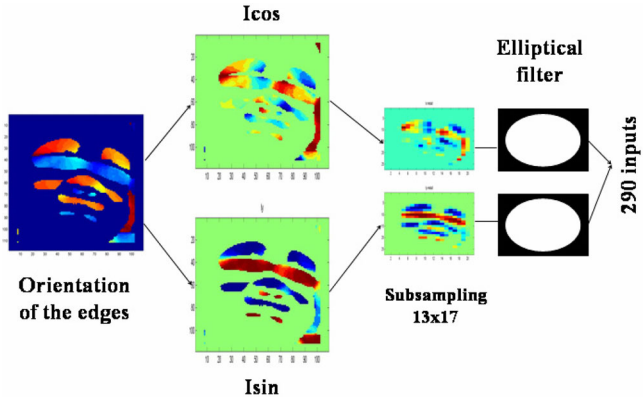


Fig. 13. Training example pre-processing.

as location of the smallest reconstruction error of the whole image (minimum of DiaboloMap).

Each pixel is described by two features $(I_{\cos}, I_{\sin})$: $I_{\cos}(i,j) = \cos((2\pi/N) \cdot \text{orien}(i,j))$ and $I_{\sin}(i,j) = \sin((2\pi/N) \cdot \text{orien}(i,j))$ for the edge pixels, where orien refer to (5); (0,0) is allocated to the non-edge pixels. An elliptical mask filters the interior part of the face as shown in Fig. 13.

We compared that coding with two others: feeding the Diabolo with grayscale face image or with the gradient field as illustrated in Fig. 14.

For the three selected input coding the best Diabolo architecture (i.e. optimal number of hidden neuron) correspond to the best face localization rate. Face localiza-

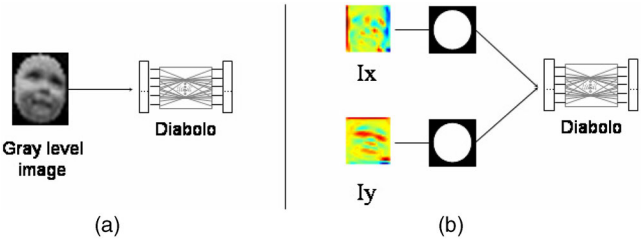


Fig. 14. Other input coding: grayscale image (a) and gradient field (b).

TABLE IV
Performances Versus Input Coding

Input Coding	Gray Levels	Gradient Field	Orientations Coding
Localization rate	27%	18%	40%

tion rate is evaluated over an assessment set: 126 images from which the cross-validation set was extracted that contain only one face. For each image a DiaboloMap is built as in Fig. 10 and face location is defined as the position of the minimum (smaller reconstruction error over the whole image). Face localization performance of the Diabolo fed with the three kind of inputs are given in Table IV: gradient orientation coding reaches the higher localization rate (40%), followed by the gray level coding of inputs (27%) and the gradient field (18%).

This comparison was done using a 21×27 retina to build the DiaboloMaps. We also investigated 17×22 and 13×17 retina: for the selected pre-processing of the training examples (see Fig. 13) the optimal retina size is 13×17. It is the best size for face localization purpose and also for computational effort. Finally the optimal Diabolo architecture is made of 290 inputs and outputs, and 18 hidden neurons. Note the dimension reduction from 442 ($2 \times 13 \times 17$ elements in $I_{\cos}$ and $I_{\sin}$) to 290 due to the elliptical filtering of inputs.

Interior part of faces is used to train the network using an elliptical mask to reduce border effects and in order not to model the elliptical shape of the face. The Diabolo is trained to model facial features: mouth and eyes, mainly. This approach is different from a face detector based on neural network which takes the face contour into account: this enhances the face detection rate. Our approach aims at compute face contour and facial features separately. This way, redundant information between the appearance-based model and the ellipse model are reduced.

5. COMBINATION OF THE SOURCES FOR FACE LOCALIZATION PURPOSE

5.1. Overview of the Combination Approach

We have implemented three holistic detectors for a color image, which result in three maps: DiaboloMap, HoughMap, and SkinMap. When each detector alone failed to model facial features, the combination of the three sources can achieve this task very well. The combination can also handle conflicts between sources.

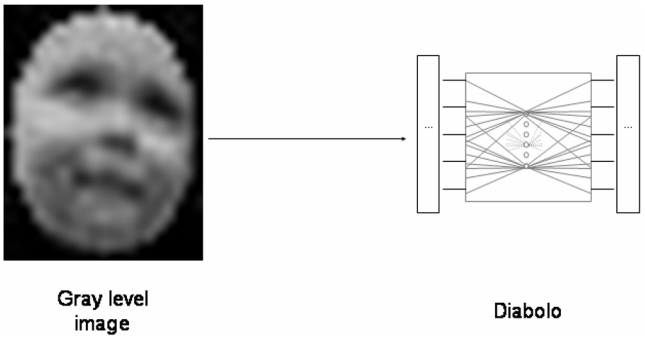


Fig. 15. Overview of the face localization system.

TABLE V
Training Set of Combiners

Class	Face	Non face
No. Samples	19.579	482.783

For that purpose, each detector map is linearly adjusted onto $[-1\ 1]$. Using the three detectors, a search window at position (i, j) in the original image is then featured by $I_{i,j}$ = [H D S].

Several architectures exist for data fusion, we can divide them into three kinds: serial (or sequential), parallel and hybrid (mixing sequential and parallel, with feed-back or interaction…). Our face localization system has a parallel architecture (see Fig. 15).

Combination rules are various, depending on the application: mean, weighted sum, product or maximum of experts outputs, majority vote, fuzzy rules, neural networks, or neuro-fuzzy inference for example.

Several algorithms have been proposed for combining our three detectors: parametric and non-parametric combination strategies are described in this sub-section. The next section is dedicated to their comparison.

Table V summarizes the number of face and non-face samples used for training combiners: these data were extracted from the cross-validation images used to stop Diabolo training.

5.2. Bayesian Classifier: Parametric and Non-Parametric Approaches

For combination purpose the input data of the Bayesian classifier are the normalized response of our three detectors. A sub-image featured by a 3D vector $\vec{X}$ = [H D S] is classified as a face if

$$\frac{P(\vec{X} \mid \text{face})}{P(\vec{X} \mid \text{non-face})} \geq \tau \tag{6}$$

where $P(\vec{X} \mid \text{face})$ and $P(\vec{X} \mid \text{non-face})$ are respectively the conditional probability density function (pdf) of the

face and non-face class. τ is the decision threshold usually estimated over a training set. As the application presented in this paper is face localization and not face detection, no estimation of τ was done. The face location shall be the one that maximize the value of the likelihood ratio (left hand side of (6)):

$$\text{Face location} \leftrightarrow \max\left(\frac{P(\vec{X} \mid \text{face})}{P(\vec{X} \mid \text{non-face})}\right) \qquad (7)$$

Parametric and non-parametric estimations of the class-conditional pdf are implemented.

The histogram technique is a non-parametric method. For each class a 3D histogram is computed using the training examples. Due to the small amount of face examples, it has only five bins per dimension: 5^3 being equal to 125, a mean of about 160 examples per bin is available. We combine the two histograms obtained into one histogram which bins values are the ratio of the bins frequency of the two preceding histograms (face/non-face). Resulting histogram values are then scaled into [0 255]. When a test image is processed three maps are calculated corresponding to our three face models over a sliding search window at each position of the image (see Fig. 15). For each position of the test image a 3D vector is computed and a back-projection of the histogram is done by a look-up table operation. This back-project is the FusionMap illustrated in Fig. 15, face location should correspond to the position of its maximum value.

A parametric approach models both skin and non skin class-conditional pdf by a unimodal Gaussians. The face location is then defined as the position of the maximum of the logarithm of the likelihood ratio:

$$(\vec{X} - \vec{\mathrm{M}}_{\text{face}})^T \Sigma_{\text{face}}^{-1} (\vec{X} - \vec{\mathrm{M}}_{\text{face}}) - (\vec{X} - \vec{\mathrm{M}}_{\text{non-face}})^T \Sigma_{\text{non-face}}^{-1} (\vec{X} - \vec{\mathrm{M}}_{\text{non-face}})$$

where the parameters of the Gaussian (Σ, M) are the mean and covariance matrix of each class computed over the training set. If $\vec{X}^i = (\mathrm{H}^i \ \mathrm{D}^i \ \mathrm{S}^i)^T$ is the ith example out of N_{faces} of the face training set:

$$\vec{\mathrm{M}}_{\text{face}} = \frac{1}{N_{\text{faces}}} \sum_{i=1}^{N_{\text{faces}}} \vec{X}^i$$

is the mean faces vector and

$$\sum_{\text{face}} = \frac{1}{N_{\text{faces}}} \sum_{i=1}^{N_{\text{faces}}} (\vec{X}^i - \vec{\mathrm{M}}_{\text{face}}) \cdot (\vec{X}^i - \vec{\mathrm{M}}_{\text{face}})^T$$

is the covariance matrix of the face class.

Other parametric functional forms of the pdf were investigated. The simplest is a unimodal Gaussian of the face class: this assumes that the non-face class is uniformly distributed. In this case the face location is defined as the maximum of the square Mahalanobis distance to the mean center of face training examples. Mixture of Gaussians were also tested but led to very poor results. Due to the small amount of training data available, this method is out of scope in this paper.

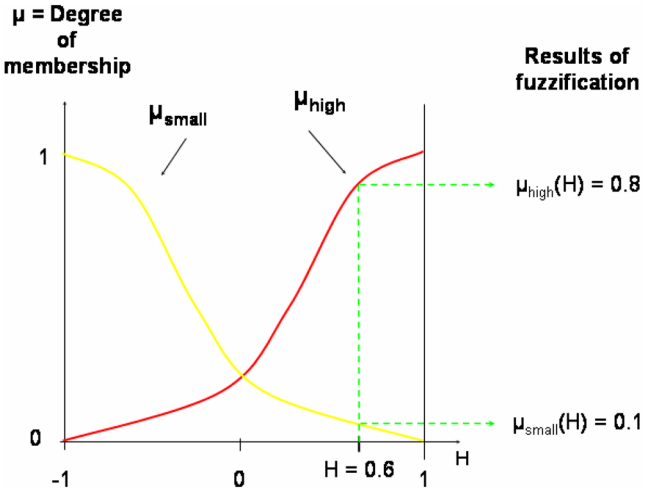


Fig. 16. Membership functions of the class "H high" and "H small".

5.3. Fuzzy Inference System

A face sub-image should be featured by a small Diabolo reconstruction error D, a high number of GHT votes H and a high proportion of skin pixels S. A classical set approach would define a threshold on each face model values. $\mathrm{H}_{\text{high}} = \{\mathrm{H} \mid \mathrm{H} > \text{thresh}\}$ the set of high H (for instance) values and $\mathrm{H}_{\text{small}} = \{\mathrm{H} \mid \mathrm{H} < \text{thresh}\}$ the set of small H values would be separated by this sharp boundary: a H value slightly under that threshold is then considered as small which make little sense. The fuzzy logic approach is more flexible by admitting partial membership to a class [40]. It is also coherent with natural language by introducing the degree of membership of H value in the class "high" and "small":

$$\mathrm{H}_{\text{high}} = \{\mathrm{H}, \mu_{\text{high}}(\mathrm{H})\} \quad \text{and} \quad \mathrm{H}_{\text{small}} = \{\mathrm{H}, \mu_{\text{small}}(\mathrm{H})\}.$$

In Fig. 16 a value of $\mathrm{H} = 0.6$ belongs the "high" class at 80% and the "small" class at 10%.

S value is the normalized proportion of skin in the sub-image: as H, high values of S correspond to high probability of the sub-image to contain a face. D value is the normalized Diabolo reconstruction error: the smaller it is, the higher is the probability of the sub-image to be a face one. For these three sources, two class are defined with respect to their value: high and small. As shown in Fig. 16, the membership functions for these classes are Gaussian functions centered respectively in $+1$ and -1.

To combine our three sources, a fuzzy inference system of Mamdani type [22] was built. A fuzzy inference system requires fuzzifying inputs, to formulate a set of linguistic rules and logical operators, and to aggregate results of the fuzzy rules. Three output class are defined as fuzzy sets: non-face, unknown, and face patterns.

Each output set is defined by a Gaussian membership function centered in 0 (non-face), $+0.5$ (unknown) or $+1$ (face), as shown in Fig. 17.

Considering only the ellipse model (H value), a simple statement can be formulated: if H is high then

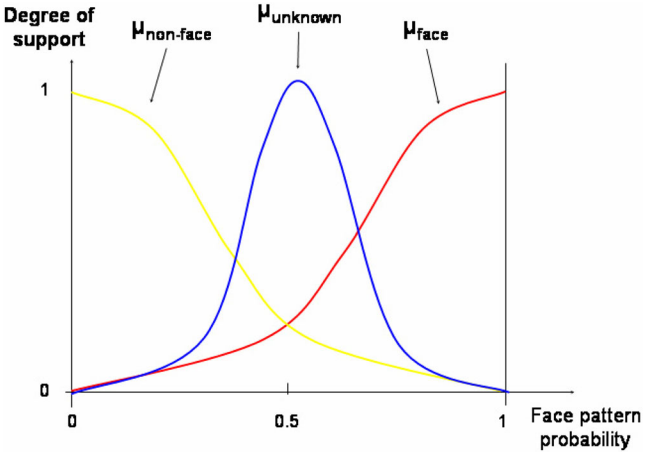


Fig. 17. Output fuzzy sets membership functions.

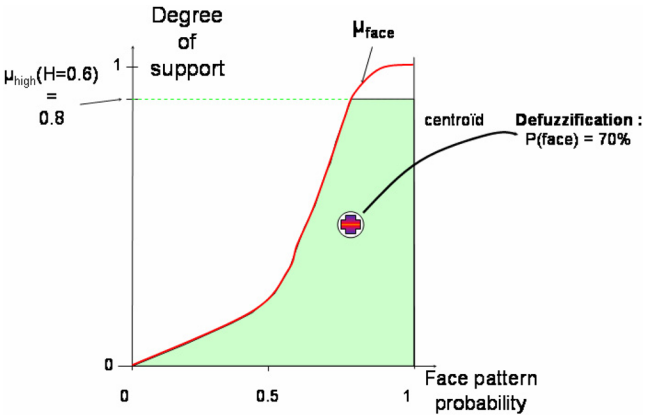


Fig. 18. Implication method: "then" operation.

sub-image is a face. Consequent of this fuzzy rule assigns a fuzzy set to the output which membership function is a truncation of the "face" set depending on the degree of support and according to the implication method (i.e. the mathematical definition of "then"). Degree of support in this particular statement only involving H value is the degree of membership in the "H high" class. The "then" operator results in a membership function equal to the minimum between the degree of support and the output fuzzy set membership function (the green area in Fig. 18 showing the case of H = 0.6).

Finally a decision can be made out of the resulting function by resolving a single value representing the probability of the sub-image to be a face pattern. A typical defuzzification method is the calculation of the center of the area under the curve (centroïd).

Now consider a statement with multi-part antecedent: if H is small or D is high or S is small then sub-image is unknown. The "or" fuzzy operation is mathematically defined as maximum of the three calculated degree of membership: this minimum is the degree of support for the output "unknown" set. In Fig. 19 a sub-image is featured by [H D S] = [0.6 0.8 −0.2]: for each source, a degree of membership is calculated. The "or" operation resolves them to a single number: the higher

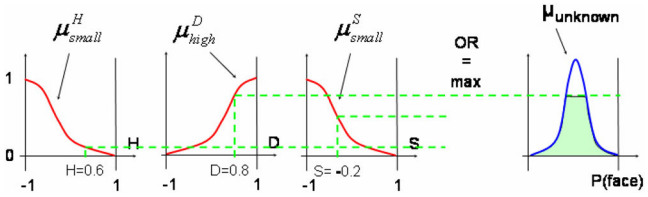


Fig. 19. Application of fuzzy operator "or".

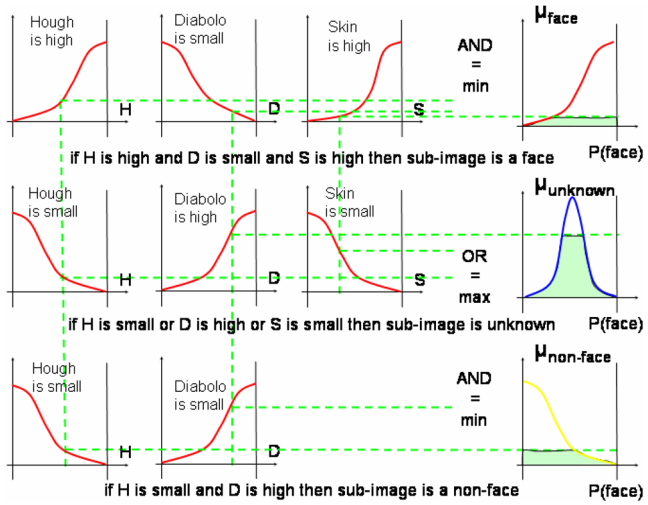


Fig. 20. Fuzzy inference diagram representing the rules.

value is kept as degree of support for the rule shaping the "unknown" fuzzy set.

One rule by itself leads to a very poor localization rate. We found experimentally that the three following fuzzy rules are optimal for face localization purpose:

—if H is high and D is small and S is high then sub-image is a face,

—if H is small or D is high or S is small then sub-image is unknown,

—if H is small and D is high then sub-image is a non-face.

The "and" operator is defined as the minimum of the degrees of membership. The rules are given the same weight, and order of the rules is unimportant as they are evaluated in parallel as shown in Fig. 20.

One can notice that the skin detector is not taken into account in the last rule: our skin color model is not elaborated enough and this is also noted with a weighted average combination (see next section).

Aggregation of the output fuzzy sets consists in calculating a membership function as the maximum of the three consequent membership functions calculated before (see Fig. 21).

This membership function is finally defuzzified by calculating the centroid of it, which provide a single number: the probability that the input sub-image is a face one.

This process is applied at all position of the original image to construct the "fuzzy" FusionMap used to define face location.

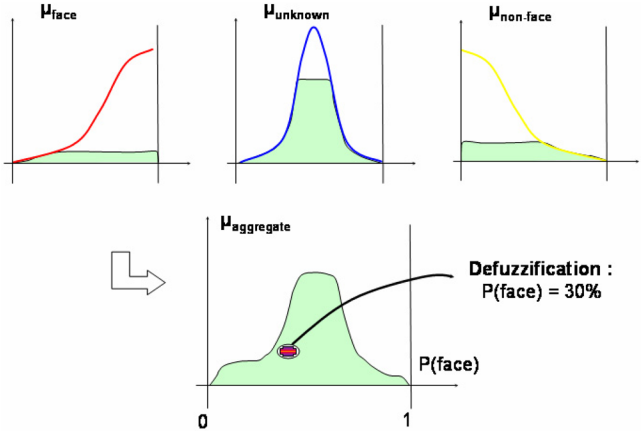


Fig. 21. The aggregate output fuzzy set.

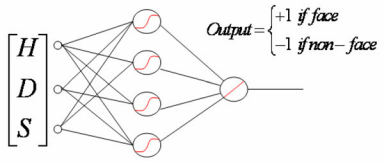


Fig. 22. Combining the three detectors with a multi-layer perceptron (MLP).

5.4. Weighted Average and Multilayer Perceptron

We investigate neural combination of the three face models: the three sources are the inputs of the multilayer perceptron (MLP). The MLP hidden neurons have sigmoid activation function, and the output neuron has a linear activation function as described in Fig. 22.

The training database is divided in two set: one for neural networks training, the other for assessment of the best architecture. 12713 face examples and 341316 non-face examples are used as training examples to learn parameters (weights) of the MLPs. Training is done using a gradient descent with adaptative learning rate stopped by cross-validation. The cost function implemented here is the sum over training examples of the square difference between target and network output.

The network is trained for pattern classification: target is $+1$ when the input [H D S] corresponds to a face and -1 else. Before training the MLP weights must be initialized: for a given neural net architecture (i.e. number of hidden neurons), several initializations are tested.

During the test phase the MLP output is a value of the interval $[-1 \ +1]$. Network output is calculated at all location of the image, which produces the "neural" FusionMap: face location is the position of the maximum of this map. The optimal MLP architectures is searched over 50 images (not used during training) containing only one face. MLPs with different number of hidden neurons, and different initialization of the weights are trained then assessed over this set. This exhaustive search leads us to the conclusion that the best architecture correspond to one output neuron. Actually, a growing number of hidden cells do not dramatically decreases the localization rate: for numbers of hidden neurons less than 3 the rates are quite the same order. The natural approach is to choose the simplest architecture for the MLP. That is to say the best neural combination is a weighted average of the inputs:

$$\text{FusionMap}_{i,j} = \text{a.H}_{i,j} + \text{b.D}_{i,j} + \text{c.S}_{i,j}$$

where a = 0.2280, b = −0.2620, and c = 0.1229.

One can notice the weight of the S input: as in the preceding section, it is half the weight of Hough or Diabolo response. This is due to the fact that the skin color model is pretty coarse.

This weighted average is compared to a simple average (same weight for the inputs: a = b = c = 1) in the next section.

6. EXPERIMENTAL RESULTS

In order to compare the combination strategies we used the ECU face database [26]: we compare the face localization rate of the algorithms on a test set of color images not used during training. Each of these image contains only one person, and the rectangle bounding the face is the same size over the whole set. A face is considered as correctly localized or not using the face ground truth and verification of a human operator. A correct localization of the face contains the eyes, the mouth, and is well-centered on the face.

6.1. ECU Face Detection Database

The ECU face and skin detection database was created in Edith Cowan University [26]. It has three sets of images particularly useful in our study (see Fig. 23). The first set is made of original color images. The second set is the corresponding ground-truth location of the faces. The third set is the ground-truth of skin pixels.

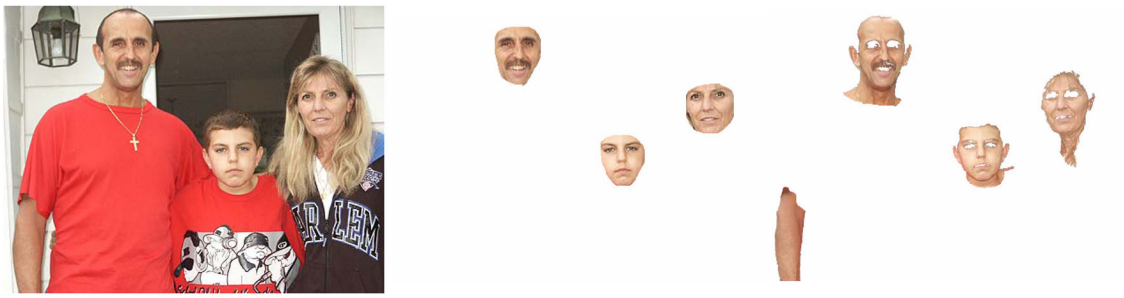
Almost all the images are taken from the Web, and were selected to have a wide variety of illumination conditions, background (mostly complex), face poses (upright, pan, tilted) and skin tones. It is widely depicted in [26].

Our test uses a set of 1353 images non overlapping with the training and cross-validation corpus. Each test image contains only one person.

6.2. Combiners Comparison

In the preceding section, different combination algorithms have been proposed. They include Bayesian classifier with parametric (unimodal Gaussian model of face and non-face) and non parametric techniques (histogram), fuzzy inference system, neural combination and weighted average.

It is important to outline the contribution of combination, and a reference for face localization rates.



(a) (b) (c)

Fig. 23. Sample from the ECU database: (a) original image, (b) face ground truth, (c) skin ground truth.

If we only consider the appearance-based model alone, the face location is defined as the minimum of Diabolo reconstruction error over the whole image, as a face image should be better reconstructed than a non-face image. Under this consideration, 656 faces out of 1353 are correctly localized: the localization rate is **48.5%**. Such a poor rate is explained in Section 4: the Diabolo is trained on examples of the interior part of the face, so we can see it as an eyes and mouth model. As an eye or a mouth detector it results in many false alarms in a cluttered scene [17]: non-face pattern is not compiled in the Diabolo. Moreover, edges are defined over each sub-window which makes appear patterns in a non-flat sub-image. And even if the Diabolo response shows a local minimum over the face area, lower minima can be found in unexpected area of the image.

Using the ellipse model alone, the face location is defined as the maximum number of vote given by the Generalized Hough Transform: 903 faces are correctly localized. In this case the face localization rate is higher: **67%**. The GHT is a cumulative approach more efficient than the appearance-based model. Missed faces of the test set correspond to an ellipse localized in a complex background with a lot of edge pixels from which a lot of vote were forecast to the accumulator.

The Bayesian classifier with the histogram technique reaches a rate of only **22%**. That means that the face and non-face distribution are strongly interleaved in the "H-D-S" space (see Fig. 24). The fuzzy approach is more efficient with a face localization rate of **72%**; it brings an improvement of 5% compared to the ellipse detector alone.

A classification based on the modeling of the face by a unimodal 3D Gaussian gives a poor **5%** of success. It means that the unimodal Gaussian center of the face class is not far enough from the non-face examples (see Fig. 25). When the three detectors respond strongly over the face region, it results in a feature vector HDS close to $A = [1 \ -1 \ 1]$ (area outlined by the red ellipse in Fig. 24(a). On the opposite, a lot of non-face sub-image are featured by a point close to point $B = [-1 \ 1 \ -1]$ in HDS (blue ellipse in Fig. 24(b) which corresponds to a non-face pattern for the three basics detectors. These two points should be correctly classify with a high confidence. But the face Gaussian center (Fig. 25) is at an equal Euclidian distance to these points: even with the covariance matrix of the face model it is not possible to discriminate samples from the two class. Results dramatically change if we use discriminant classification with both Gaussian distribution of face and non face. Indeed unimodal 3D-Gaussian of face and non-face Bayesian classification achieves **84%**. The non-face class Gaussian center is close to the non-face HDS point clouds as we can see on Fig. 25.

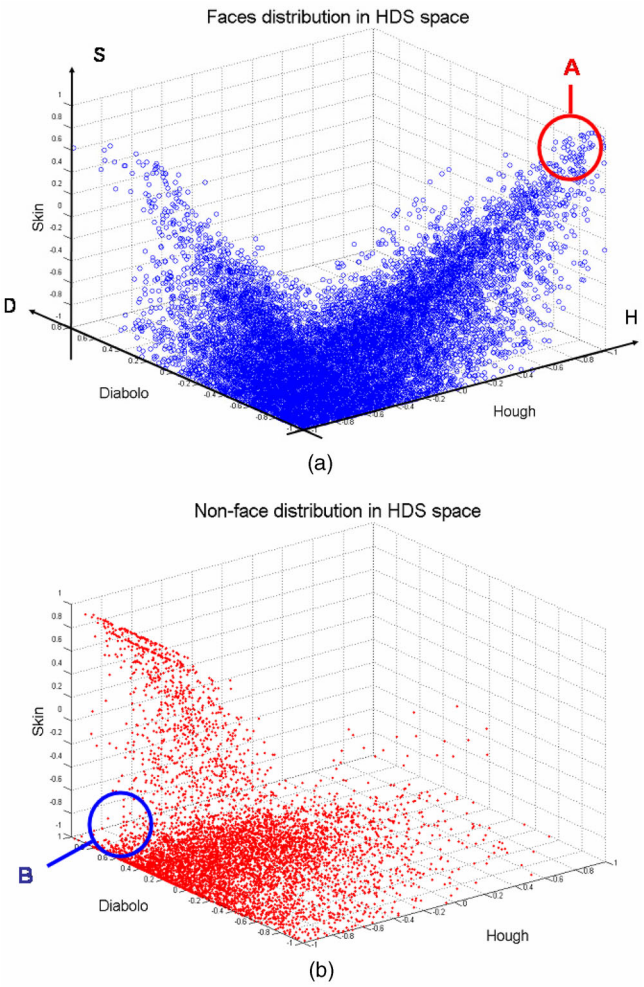


Fig. 24. Faces (a) and non-faces (b) distributions in the HDS space.

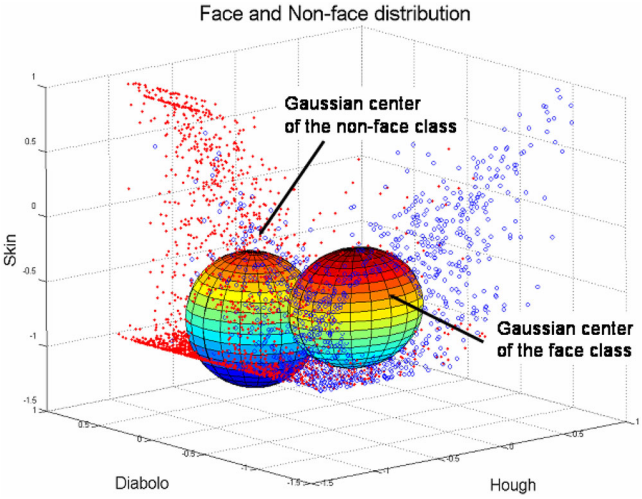


Fig. 25. Gaussian centers of the face and non-face distribution.

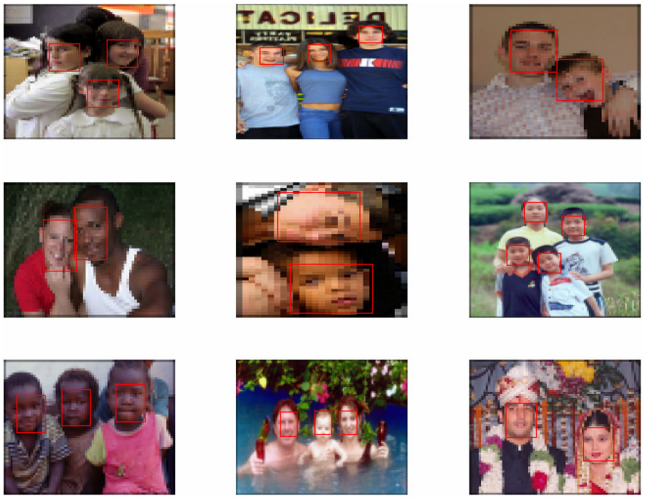


Fig. 27. Multiple faces localization: the number of faces is supposed to be known.

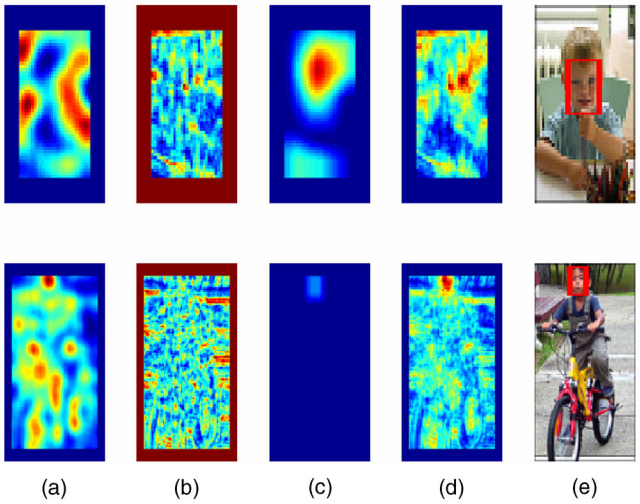


Fig. 26. (a) HoughMap (b) DiaboloMap (c) SkinMap (d) FusionMap and (e) the corresponding face localization on the original image.

Anyway, compare to all these methods, the weighted average performs the best. With a localization rate of **86%** it outperforms all the other approach. In order to measure the effect of the weights on the detection result, a simple average (i.e. all weights equal 1) is performed. With a rate of **80%** it performs well too, but less than the weighted average with the weights learned by gradient descent.

Amongst the multiple classifier systems, linear combiners are the most frequently used: a recent study can be found in [15] with a theoretical analysis based on the framework of [35]: the analysis of linear combiners is still a promising path of research.

Fig. 26 shows detectors response and their combination using the weighted average.

In the first example of Fig. 26, the ellipse detector failed to locate correctly the face, while the combination system did. In the second example, the SkinMap maximum is very low (0.23), but the combination (weighted average) brings a correct face location.

To validate the face localization rate of the weighted average combination, a second test was performed on 205 multiple faces images (non overlapping with the training and cross-validation corpus) containing a total of 482 faces. Number of people in each image is known in a face localization approach.

In single face images, face location is defined as the position of the maximum of FusionMap. In a N faces image (N is supposed to be known in a localization problem) the N highest maxima (with a sufficient distance to avoid overlapping detections) of FusionMap correspond to the location of the faces. 396 faces are correctly localized (**82**%). Some examples of correct localization are shown in Fig. 27.

For all tests of this sub-section the face size is also supposed to be known: this information can be retrieved if we know the distance between the person and the camera. Videos available at [25] were particularly interesting for this approach, showing people in front of their computer: face size does not vary widely along the image sequence.

The performance of the weighted average approach at a known scale on video sequences was tested on three videos sequences. In each image of the sequences only one person is present. Fig. 28 gives examples of the sequences, with the localization rates on each sequence.

6.3. Face Localization: Multiple Scale Approach

In the previous tests, face size is supposed to be known: it is the case when distance between the person and the camera is given. When this information is unavailable, a multi-scale approach of the weighted average combination is implemented. To localize faces of various sizes a pyramid of images is produced: the image is repeatedly subsampled with a classical [16, 36]

Fig. 28. Examples of face localization by our system on videos with the number of faces correctly localized. (a) Jamie sequence: 40 correct localization/43 images. (b) Ilkay sequence: 72 correct localization/80 images. (c) Geoff sequence: 24 correct localization/24 images.

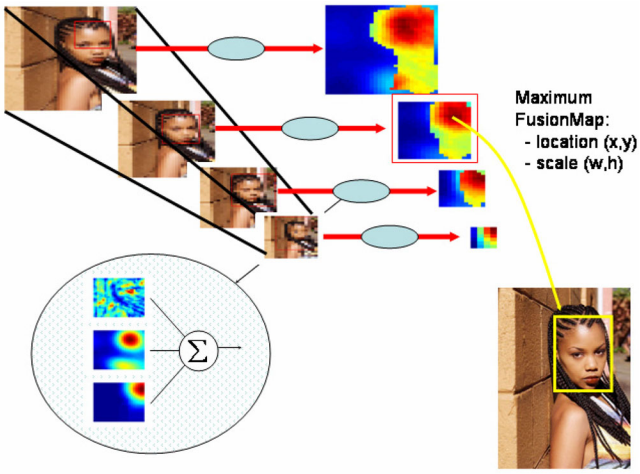


Fig. 29. Images Pyramid to deal with face size.

scale factor of 1.2. For each scale a FusionMap is built using a sliding window of a fixed size: face location probabilities are then compared across the different scales. Fig. 29 illustrates the images pyramid principle.

That multiscale approach is tested over 923 images containing one person with a face width superior to 100 pixels, so that the number of scale to scan is less than twelve. 537 faces are correctly localized: the face localization rate is **58%**. This rate is small compare to state of the art face detector [36]. We used the Haar face detector publicly available in [19]. A statistical model of the face, made of a cascade of boosted tree classifiers, is trained. The cascade is trained on face and non-face examples of fixed size 24×24. A 24×24

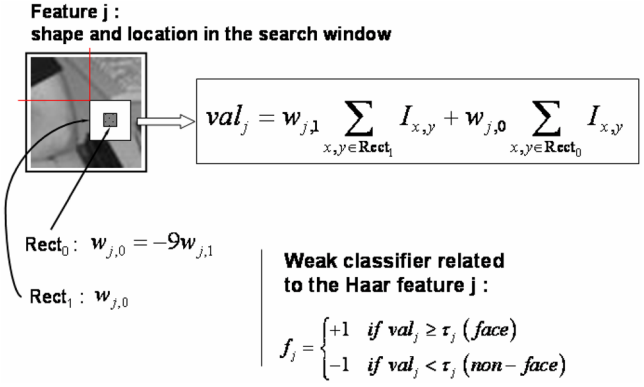


Fig. 30. A Haar-like feature is defined by its shape and its location relative to the 24×24 sliding window.

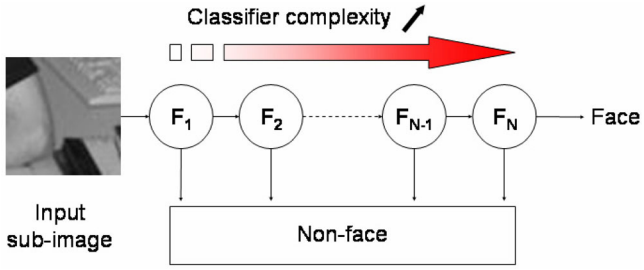


Fig. 31. Cascade of boosted classifiers.

sliding window scans the image and each sub-image is classified as face or non-face. To deal with face size the cascade is scaled with a factor of 1.2 by scaling the coordinates of all rectangles of Haar-like features. Hundreds of features are used as these shapes are applied at different position in the 24×24 retina: a feature is defined by its shape (including its size depending on a scale factor that defines the expected face size) and its location (see Fig. 30).

A simple decision tree classifier, referred to as "weak" classifier, processes the feature value. A complex classifier $F_k = \text{sign}(\sum_{i=1}^{n} c_i f_i)$ is iteratively computed as a weighted sum of weak classifiers using a boosting procedure. At each iteration a weak classifier parameters are trained and a weight c_j is assigned to the weak classifier relatively to its error on the training set. The trained weak classifier is then added to the sum and the training samples weights are updated in order to emphasize the misclassified ones to train the next weak classifier. Finally an attentional cascade is implemented: it is a cascade of boosted classifiers with increasing complexity. As shown in Fig. 31, the simplest classifiers comes first and is intended to reject majority of sub-window before calling more complex classifiers.

This face detector is robust to illumination condition but hardly work when face is too slanted. Fig. 32 illustrates the limitation of the detector: in the first row the face is correctly detected. In the second row the face moved slightly from the previous position and is not detected.

Localization rate measures a face localizer performance: a false positive also correspond to a missed

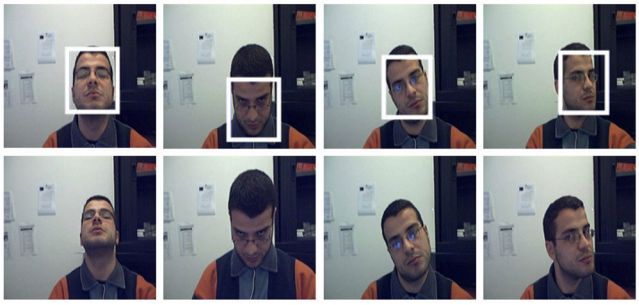


Fig. 32. Limits of the face detector.

face as only one location is searched in the image. A face detector is evaluated by its ROC performance: at least two scores are required, the detection rate (complementary of the missed rate) and the false positive rate. The cascaded face detector is more efficient than the weighted average combination. It detects 713 faces out of 923 (**77%**) with 78 false detections; 210 faces are missed. It is, with [16] the state-of-the-art in face detection. Its multi-scale approach is more efficient that the usual pyramid of images produced by down-sampling the original image: it scales the Haar filters, so that the search window contains a "high" resolution sub-image whatever the scale considered. In 210 images (23%) of the 923 test images, the face is missed. They correspond to faces highly rotated (pan, tilt or roll rotation) or occluded. On these particularly difficult images the weighted average localizer performs quite well with 90 faces correctly localized out of 210 (43%). It appears that our approach could be used as an alternative to the Haar detector when it fails to detect anyone in the scene. It potentially could decrease the missing rate by 43%.

7. CONCLUSION AND PROSPECTS

This paper aimed to present a significant contribution to the image fusion task with application to face localization. We have presented three different detectors: skin color, auto-associative multi-layer perceptron, and ellipse Hough Transform. We proposed three various combination schemes and compare them: Bayesian classifier, fuzzy logic and connexionist. An awesome improvement of localization rate is brought by the two last methods.

For the face detection/localization issue, several improvements are in progress: more sophisticated skin color models like ellipsoidal threshold, Gaussian density functions or mixture of Gaussians [38]. A more efficient appearance-based model is also elaborated, based on the Viola&Jones face detector [36]. For the combination part, it is not clear when and why a combination method outperforms the others: quantitative and qualitative investigations of classifiers output correlation effect on combiners performance are under study.

REFERENCES

[1] R. Belaroussi, L. Prevost and M. Milgram
Combining model-based classifiers for face localization.
In *Proceedings of the IAPR Conference on Machine Vision Application*, 2005, 290–293.

[2] S. M. Bileschi and B. Heisele
Advances in component based face detection.
In *Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, 2003, 149–156.

[3] M. C. Burl, M. Weber and P. Perona
A probabilistic approach to object recognition using local photometry and global geometry.
In *Proceedings of the European Conference on Computer Vision*, Vol. 2, 1998, 628–642.

[4] P. Campadelli, R. Lanzarotti, G. Lipori, and E. Salvi
Face and facial feature localization.
In *Proceedings of the International Conference on Image Analysis and Processing*, Vol. 3617 of Lecture Notes in Computer Science, 2005, 1002–1009.

[5] D. Chai and K. N. Nang
Locating facial region of a head-and-shoulders color image.
In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, 1998, 124–129.

[6] R. Choudhury Verma, C. Schmid and K. Mikolajczyk
Face detection and tracking in a video by propagating detection probabilities.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **25**, 10 (Oct. 2003), 1215–1228.

[7] J. Czyz, J. Kittler and L. Vandendorpe
Combining face verification experts.
In *Proceedings of the IAPR International Conference on Pattern Recognition*, Vol. 2, 2002, 28–31.

[8] D. DeMers and G. Cottrell
Non-linear dimensionality reduction.
In *Proceedings of the Conference on Neural Information Processing Systems*, Vol. 5, 1993, 580–587.

[9] J. Dowdall, I. Pavlidis and G. N. Bebis
Face detection in the near-IR spectrum.
Image and Vision Computing, **21**, 7 (July 2003), 565–578.

[10] R. O. Duda, P. E. Hart, and D. G. Stork
Pattern Classification.
New York: Wiley, 2001.

[11] C. K. Eveland, D. A. Socolinsky and L. B. Wolff
Tracking human faces in infrared video.
Image and Vision Computing, **21**, 7 (July 2003), 579–590.

[12] P. Felzenschwab and D. Huttenlocher
Efficient matching of pictorial structures.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2000, 66–73.

[13] R. Féraud, O. Bernier, J. Viallet, and M. Collobert
A fast and accurate face detector based on neural networks.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **23**, 1 (Jan. 2001), 42–53.

[14] B. Froba and W. Zink
On the combination of different template strategies for fast face detection.
In *Proceedings of the International Workshop on Multiple Classifier Systems*, 2001, 418–428.

[15] G. Fumera and F. Roli
A Theoretical and experimental analysis of linear combiners for multiple classifier systems.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **27**, 6 (June 2005), 942–956.

[16] C. Garcia and M. Delakis
Convolutional face finder: A neural architecture for fast and robust face detection.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **26**, 11 (Nov. 2004), 1408–1423.

[17] M. Hamouz, J. Kittler, J-K. Kamarainen, P. Paalanen, H. Kalviainen and J. Matas
Feature-based affine-invariant localization of faces.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **27**, 9 (Sept. 2005), 1490–1495.

[18] M. Hu, S. Worrall, A. H. Sadka and A. M. Kondoz
Automatic scalable face model design for 2D model-based video coding.
Signal Processing: Image Communication, **19** (2004), 421–436.

[19] Intel Corporation
Open Source Computer Vision Library—Reference Manual, 2001, homepage: http://developer.intel.com.

[20] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz
Robust face detection using the Hausdorff distance.
In *Proceedings of the IEEE International Conference on Audio- and Video-based Biometric Person Authentication*, 2001, 90–95.

[21] M. J. Jones and J. M. Rehg
Statistical color models with application to skin detection.
International Journal of Computer Vision, **46**, 1 (Jan. 2002), 81–96.

[22] E. H. Mamdani and S. Assilian
An experiment in linguistic synthesis with a fuzzy logic controller.
International Journal of Man-Machine Studies, **7**, (1) 1975, 1–13.

[23] S. J. McKenna, S. Gong, and Y. Raja
Modeling facial color and identity with gaussian mixtures.
Pattern Recognition, **31**, 12 (Dec. 1998), 1883–1892.

[24] K. Messer, J. Matas, J. Kittler, J. Luettin and G. Maitre
XM2VTSDB: The extended M2VTS database.
In *Proceedings of the International Conference on Audio- and Video-based Biometric Person Authentication*, 1999, 72–77.

[25] Microsoft Research Cambridge
Homepage: http://research.microsoft.com/vision/cambridge/i2i/DSWeb.htm.

[26] S. L. Phung, A. Bouzerdoum and D. Chai
Skin segmentation using color pixel classification: Analysis and comparison.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **27**, 1 (Jan. 2005), 148–154.

[27] L. Prevost and M. Milgram
Automatic allograph selection and multiple expert classification for totally unconstrained handwritten character recognition.
In *Proceedings of the IAPR International Conference on Pattern Recognition*, Vol. 1, 1998, 381–383.

[28] A. F. Rahman and M. C. Fairhust
Multiple classifier decision combination strategies for character recognition: A review.
International Journal on Document Analysis and Recognition, **5**, 4 (July 2003), 166–194.

[29] H. Rowley, S. Baluja and T. Kanade
Neural network-based face detection.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **20**, 1 (Jan. 1998), 23–38.

[30] H. Schwenk and M. Milgram
Transformation invariant auto-association with application to handwritten character recognition.
In *Proceedings of the Conference on Neural Information Processing Systems*, Vol. 7, 1995, 991–998.

[31] K. Schwerdt and J. L. Crowley
Robust face tracking using color.
In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 2000, 90–95.

[32] R. Séguier
A very fast adaptive face detection system.
In *Proceedings of the International Conference on Visualization, Imaging, and Image Processing*, 2004.

[33] F. Smeraldi and J. Bigun
Retinal vision applied to facial features detection and face authentication.
Pattern Recognition Letters, **23**, 4 (Feb. 2002), 463–475.

[34] K. K. Sung and T. Poggio
Example-based learning for view-based human face detection.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **20**, 1 (Jan. 1998), 39–51.

[35] K. Tumer and J. Ghosh
Analysis of decision boundaries in linearly combined neural classifiers.
Pattern Recognition **29**, 2 (Feb. 1996), 341–348.

[36] P. A. Viola and M. J. Jones
Rapid object detection using a boosted cascade of simple features.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 1, 2001, 511–518

[37] G. Yang and T. S. Huang
Human face detection in complex background.
Pattern Recognition, **27**, 1 (Jan. 1994), 53–63.

[38] M-H. Yang, D. Kriegman, and N. Ahuja
Detecting faces in images: A survey.
IEEE Transactions on Pattern Analysis and Machine Intelligence, **24**, 1 (Jan. 2002), 34–58.

[39] A. Yuille, P. Hallinan and D. Cohen
Feature extraction from faces using deformable templates.
International Journal of Computer Vision, **8**, 2 (Aug. 1992), 99–111.

[40] L. A. Zadeh
Outline of a new approach to the analysis of complex systems and decision processes.
IEEE Transactions on Systems, Man, and Cybernetics, **3**, 1 (Jan. 1973), 28–44.

Rachid Belaroussi received the M.S. degree in electronics from the University Pierre and Marie Curie and the engineer diploma of Ecole Supérieure de Physique et de Chimie Industrielle in 2002. He is pursuing a Ph.D. degree in computer vision at the LISIF laboratory. His research activities are focused on face detection and tracking in videos or still images, and include image processing, machine learning and neural networks.

Lionel Prevost received the Ph.D. degree in pattern recognition from the University Pierre and Marie Curie–Paris 6 in 1998. His research interests included machine learning and information fusion for handwriting recognition. He is currently associate professor at this university, in the Institute of Intelligent Systems and Robotics (Perception team). His current activities are in the fields of neural networks, pattern recognition, image and video analysis with application to face and facial feature localization and vehicle classification. He serves on the program committee of various conferences.

Maurice Milgram was born in 1948 in Paris and obtained the Agrégation of Mathematics in 1971 and a Ph.D. from UTC in probabilistic automata networks in 1981. He started as an assistant professor at Technology University of Compiègne (1973) then he became full professor (1983) at ENSEA and joined the Robotic Laboratory of the Pierre and Marie Curie University in Paris in 1986. He founded the PARC Laboratory, which is now part of the LISIF, in 1992. He has supervised about 30 Ph.D. thesis and 20 industrial contracts. His research interests concern pattern recognition and image processing.

His current research activities are focus on face localization/detection, gaze tracking, person tracking in image sequences, gesture recognition. His collaboration with public and private research laboratories includes recent contracts such as person localization for a smart airbag (Faurecia), automatic directional control of the vehicle's headlamp beams (Valeo), vehicle type recognition (LPR), biometry (Sagem), on-vehicle obstacle detection (PSA), baby gaze tracking (Necker Hospital).