Face Detection Using Integral Projection Models*

Ginés García-Mateos[1], Alberto Ruiz[1], and Pedro E. Lopez-de-Teruel[2]

[1] Dept. Informática y Sistemas
[2] Dept. de Ingeniería y Tecnología de Computadores
University of Murcia, 30.170 Espinardo, Murcia (Spain)
{ginesgm,aruiz}@um.es
pedroe@ditec.um.es

Abstract. Integral projections can be used to model the visual appearance of human faces. In this way, model based detection is done by fitting the model into an unknown pattern. Thus, the key problem is the alignment of projection patterns with respect to a given model of generic face. We provide an algorithm to align a 1-D pattern to a model consisting of the mean pattern and its variance. Projection models can also be used in facial feature location, pose estimation, expression and person recognition. Some preliminary experimental results are presented.

1 Introduction

Human face detection is an essential problem in the context of perceptual interfaces and human image processing, since a fixed location assumption is not possible in practice. It deals with determining the number of faces that appear in an image and, for each of them, its location and spatial extent [1]. Finding a fast and robust method to detect faces under non-trivial conditions is still a challenging problem.

Integral projections have already been used in problems like face detection [2, 3] and facial feature location [4]. However, most existing techniques are based on max-min analysis [2, 3], fuzzy logic [4] and similar heuristic approaches. To the best of our knowledge, no rigorous study on the use of projections has been done yet. Besides, the use of projections constitutes a minor part in the vision systems. Our proposal is to use projections as a means to create 1-dimensional face models.

The structure of this paper is the following. In Section 2, we show how projections can be used by themselves to model 3-D objects like faces. The face detection process is presented in Section 3. Section 4 focuses in the key problem of projection alignment. Some preliminary experimental results on the proposed model are described in Section 5. Finally, we present some relevant conclusions.

* This work has been supported by the Spanish MCYT grant DPI-2001-0469-C03-01.

T. Caelli et al. (Eds.): SSPR&SPR 2002, LNCS 2396, pp. 644-653, 2002.

2 Modeling Objects with Integral Projections

Integral projections can be used to represent the visual appearance of a certain kind of object under a relatively wide range of conditions, i.e., to model object classes. In this way, object analysis can be done by fitting a test sample to the projection model. We will start this section with some basic definitions on integral projections.

2.1 One-Dimensional Projections

Let *i(x, y)* be a grayscale image and *R(i)* a region in this image, i.e., a set of contiguous pixels in the domain of *i*. The horizontal and vertical integral projections of *R(i)*, denoted by $P_{HR(i)}$ and $P_{VR(i)}$ respectively, are discrete and finite 1-D signals given by

$$P_{HR(i)} : \{x_{min},...,x_{max}\} \rightarrow \mathbf{R} \;;\; P_{HR(i)}(x) := |Rx(i)|^{-1} \sum_{y \in Rx(i)} i(x,y) \;; \tag{1}$$

$$P_{VR(i)} : \{y_{min},...,y_{max}\} \rightarrow \mathbf{R} \;;\; P_{VR(i)}(y) := |Ry(i)|^{-1} \sum_{x \in Ry(i)} i(x,y) \;; \tag{2}$$

where

$$x_{min} = \min_{(x,y)\in R(i)} x;\; x_{max} = \max_{(x,y)\in R(i)} x;\; y_{min} = \min_{(x,y)\in R(i)} y;\; y_{max} = \max_{(x,y)\in R(i)} y; \tag{3}$$

$$\mathrm{Rx(i)} = \{\mathrm{y} / \forall\ \mathrm{y}, (\mathrm{x}, \mathrm{y}) \in \mathrm{R(i)}\}\ ;\ \mathrm{Ry(i)} = \{\mathrm{x} / \forall\ \mathrm{x}, (\mathrm{x}, \mathrm{y}) \in \mathrm{R(i)}\}\ . \tag{4}$$

The sets $\{x_{min}, ..., x_{max}\}$ and $\{y_{min}, ..., y_{max}\}$ are called the domains of the horizontal and vertical integral projection, denoted by $Domain(P_{HR(i)})$ and $Domain(P_{VR(i)})$ respectively. Similarly, we can define the projection along any direction with angle α, $P_{\alpha R(i)}$, as the vertical projection of region *R(i)* rotated by angle α. Applied on faces, vertical and horizontal projections produce typical patterns, as those in Fig. 1.

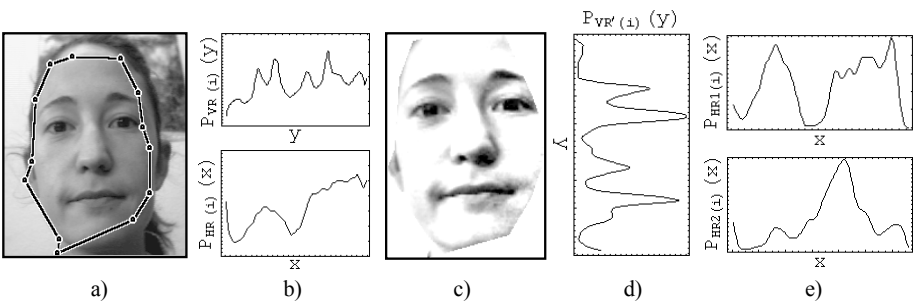

Fig. 1. Vertical and horizontal integral projections. a) A face region *R* found using skin color analysis. b) Vertical (up) and horizontal (down) projections of *R*. c) Segmentation and intensity equalization of *R*, to produce *R'*. d) Vertical projection of *R'*. e) Horizontal projection of the upper (*R1*) and lower (*R2*) halves of *R'*

Integral projections give marginal distributions of gray values along one direction, so they usually involve a loss of information. An approximate reconstruction of $R(i)$ can be easily computed from $P_{HR(i)}$ and $P_{VR(i)}$. Let us suppose $i(x,y)$ is normalized to values in [0, 1], then the reconstruction is: $\hat{i}(x,y) = P_{HR(i)}(x) \cdot P_{VR(i)}(y)$, $\forall (x,y) \in R(i)$.

Modeling Faces

The following questions have to be dealt with when using a model of projections:

- How many projections are used to model the object class.
- For each of them, which angle and which part of the region is projected.
- What projected pixels represent, e.g., intensity, edge-level, color hue.

As we have mentioned, an approximate reconstruction can be computed from the projections and, somewhat, the similarity with respect to the original image indicates the accuracy of the representation. For instance, three reprojections of a face image using different numbers of projections are shown in Fig. 2.

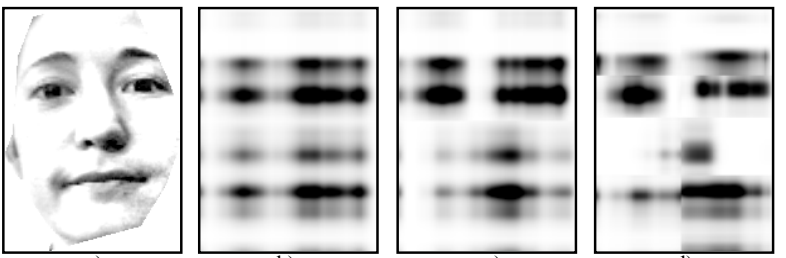

Fig. 2. Face image reconstruction by reprojection. a) Original face image, segmented and intensity equalized. b)-d) Reconstruction using vertical (*vip*) and horizontal (*hip*) integral projections: b) 1 *vip*, 1 *hip*; c) 1 *vip*, 2 *hip*; d) 2 *vip*, 4 *hip*

Obviously, the accuracy of the reprojection increases with the number of projections used. However, using a high number of projections involves an important loss of robustness and efficiency. Thus, we have chosen the representation of 1 vertical and 2 horizontal projections, shown in Fig. 2, which gives admissibly results.

To model the variability of the face class, we propose a gaussian-style representation of the one-dimensional signals. That is, for each point j in the domain of the signal, the mean value $M(j)$ and the variance $V(j)$ are computed. Summing up, the face model consists of the following 1-D signals:

- $M_{V,FACE}$, $V_{V,FACE}$: $\{1, ..., f_{max}\} \rightarrow \mathbf{R}$. Mean and variance of the vertical projection of the whole face region, respectively.
- $M_{H,EYES}$, $V_{H,EYES}$: $\{1, ..., e_{max}\} \rightarrow \mathbf{R}$. Mean and variance of the horizontal projection of the upper part of the face, from forefront to nose (not included).
- $M_{H,MOUTH}$, $V_{H,MOUTH}$: $\{1, ..., m_{max}\} \rightarrow \mathbf{R}$. Mean and variance of the horizontal projection of the lower part of the face, from nose (included) to chin.

3 Face Detection Using Projections

In a general sense, object detection using models consists of fitting a known model into an unknown pattern. If a good fitting is found, the object is said to be detected. However, as the location, scale and orientation of the object in the image is unknown, either selective attention [3, 5, 6] or exhaustive multi-scale searching [7, 8], are needed. We use the selective attention mechanism described in [3], based on connected components of skin-like color. This process was used in the experiments to extract the face and non-face candidate regions, which are the input to the face detection algorithm using the projection models. The algorithm is shown in Fig. 3.

Algorithm: Face Detection Using a Projection Model

Input

i: Input image

M = ($M_{V,FACE}$, $V_{V,FACE}$, $M_{H,EYES}$, $V_{H,EYES}$, $M_{H,MOUTH}$, $V_{H,MOUTH}$): Face model

Output

n: Number of detected faces

$\{R_1, ..., R_n\}$: Region of the image occupied by each face

1. Segment image *i* using connected components of skin-like color regions.

2. For each candidate region *R(i)* found in step 1, do.

2.1. Compute $P_{VR(i)}$, the vertical integral projection of *R(i)*, taking the principal direction of *R(i)* as the vertical axis.

2.2. Align $P_{VR(i)}$ to ($M_{V,FACE}$, $V_{V,FACE}$) obtaining $P'_{VR(i)}$.

2.3. If a good alignment was obtained in step 2.2, compute $P_{HR1(i)}$ and $P_{HR2(i)}$, the horizontal integral projections of the upper and lower parts of *R(i)* respectively, according to the results of the alignment $P'_{VR(i)}$.

2.4. Align $P_{HR1(i)}$ to ($M_{H,EYES}$, $V_{H,EYES}$) obtaining $P'_{HR1(i)}$, and align $P_{HR2(i)}$ to ($M_{H,MOUTH}$, $V_{H,MOUTH}$) obtaining $P'_{HR2(i)}$.

2.5. If good alignments were obtained in step 2.4, then *R(i)* corresponds to a face. Increment *n*, and make $R_n = R(i)$. The location of the facial features can be computed by undoing the alignment transformations in steps 2.2 and 2.4.

Fig. 3. Global structure of the algorithm for face detection using a projection model

First, the ($M_{V,FACE}$, $V_{V,FACE}$) part of the model is fitted into the vertical projection of the whole candidate region, which might contain parts of hair or neck. If a good alignment is obtained, the vertical locations of the facial components are known, so we can compute the horizontal projections of the eye and mouth regions, removing hair and neck. If both are also correctly aligned, a face has been found.

4 Projection Alignment

The key problem in object detection using projections is alignment. The purpose of projection alignment is to produce new derived projections where the location of the facial features is the same in all of them. Figs. 4a,b) show two typical skin-color regions containing faces, which produce unaligned patterns of vertical projections. After alignment, eyes, nose and mouth appear at the same position in all the patterns.

In this section, we describe one solution to the problem of aligning 1-D patterns, or signals, to a model consisting of the mean signal and variance at each point, as introduced in Section 2.2. Note that in the algorithm for face detection, in Section 3, the goodness of alignment is directly used to classify the pattern as face or non-face. In general, any classifier could be used on aligned patterns, thus making clear the difference between preprocessing (alignment) and pattern recognition (binary classification face/non-face). In the following, we will suppose any classifier can be used.

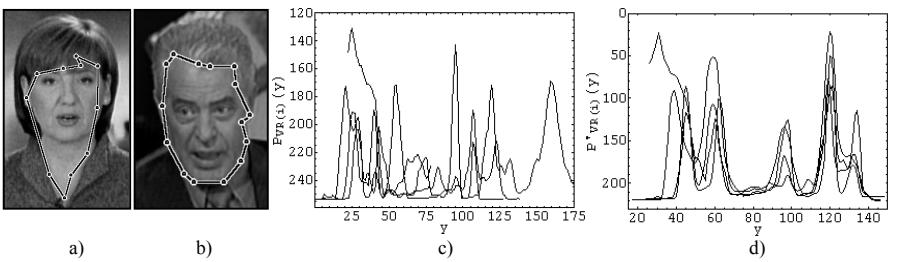

Fig. 4. Alignment of projections. a)-b) Two typical face regions, producing unaligned projections. c) Unaligned vertical projections of 4 faces. d) The same projections, after alignment

4.1 Alignment Criterion

Theoretically speaking, a good alignment method for detection should produce a representation of face patterns invariant to lighting conditions, pose, person and face expression. Let us suppose we have a set of projections $P=P_{FACE} \cup P_{NON\text{-}FACE}$, and a set of alignment transformations $A=\{a_1, ..., a_m\}$. The best alignment is the one that produces the best detection ratios, that is, a high number of detected faces and a low number of false-positives. Instead, we will work with a more practical criterion.

In order to achieve good detection results, independently from the classifier used, the alignment should minimize the variance of aligned face patterns, denoted by P^a_{FACE}, and maximize the interclass variance of $\{P^a_{FACE}, P^a_{NON-FACE}\}$. However, estimating this interclass variance is a hard problem, since no finite set $P_{NON\text{-}FACE}$ can be representative enough of everything which is not a face.

Supposing the average projection of the infinite class of non-faces is a uniform signal, the variance between face and non-face classes can be estimated with the inner variance, or energy, of the mean signal $\overline{P^a_{FACE}}$. In this way, the goodness of an alignment transformation a can be estimated with the ratio

$$RATIO\,(a,P_{FACE}) := \frac{VARIANCE(P^a_{FACE})}{VARIANCE(\overline{P^a_{FACE}})}\,. \tag{5}$$

A lower value of (5) means a better alignment. In practice, we are interested in aligning patterns according to a projection model in the form (M: mean; V: variance) learnt by training. This involves that the average face projection (and, consequently, its energy) is computed in the training process. As a result, for a given pattern p, the alignment should minimize its contribution to (5), which can be expressed as

$$Distance\,(a,p,M,V) := \sum_{i\in Domain(M)} \frac{(p^a(i)-M(i))^2}{V(i)}\,. \tag{6}$$

4.2 Transformation Functions

A transformation is a function that takes a signal as an input and produces a derived signal. It is called an alignment, or normalization, if the transformed signals verify a given property. We are interested in parameterized functions, where the parameters of the transformation are calculated for each signal and model. It is convenient to limit the number of free parameters, as a high number could produce a problem of *over-alignment*: both face and non-face patterns could be transformed to face-similar patterns, causing many false-positives.

We will use the following family of parameterized transformation functions

$$t_{a,b,c,d,e} : (\{s_{min},\ldots,s_{max}\}\to\mathbf{R}) \longrightarrow (\{\left|\frac{s_{min}-e}{d}\right|,\ldots,\left|\frac{s_{max}-e}{d}\right|\}\to\mathbf{R})\,, \tag{7}$$

defined by

$$t_{a,b,c,d,e}(S)(i) := a + b{\cdot}i + c{\cdot}S(|\,d{\cdot}i+e\,|)\,. \tag{8}$$

As expressed in (8), the function $t_{a,b,c,d,e}$ makes a linear transformation both in value and in domain of the input signal S. It has five free parameters: (a, b, c) the value transformation parameters, and (d, e) the domain transformation parameters. Geometrically interpreted, (a, e) are translation parameters in value and domain, respectively; (c, d) are scale parameters, and b is a skew parameter that, in our case, accounts for a non-uniform illumination of the object.

4.3 Alignment Algorithm

For the alignment, we will use the family of transformation functions with the form $t_{a,b,c,d,e}$, defined in (7) and (8). We can obtain the objective function of alignment replacing p^a in (6) with (8). Thus, the optimum alignment of a signal S to a model (M, V) is given by the set of values (a, b, c, d, e) minimizing

$$Distance\,(a,b,c,d,e) := \sum_{i\in Domain(M)} \frac{(a+b{\cdot}i+c{\cdot}S(|\,d{\cdot}i+e\,|)-M(i))^2}{V(i)}\,. \tag{9}$$

Due to the form of the transformation, both in value and domain, standard optimization techniques can not be applied to minimize (9). Instead of it, we use an iterative two-step algorithm that, alternatively, solves for the domain parameters (d, e) and the value parameters (a, b, c). The algorithm is presented in Fig. 5.

Algorithm: Linear Alignment of a 1-D Pattern to a Mean/Variance Model

Input

S: Signal pattern to be aligned
M, V: Signal model, mean and variance respectively

Output

S': Optimum alignment of signal S to model (M, V)

1. Transformation initialization. Set up initial values for (a, b, c, d, e), e.g., locating two clearly distinguishable points in S. Obtain S' applying equation (8).

2. Repeat until convergence is reached or after MAX_ITER iterations:

2.1. Pattern domain alignment.

2.1.1. Assign each point i of S', in $Domain(S') \cap Domain(M)$, to a point $h(i)$ of M, in $Domain(M)$, with reliability degree $w(i)$.

2.1.2. Estimate parameters (d, e), as the linear regression parameters of the set (i, $h(i)$) taking into account the weights $w(i)$, for i in $Domain(S') \cap Domain(M)$.

2.1.3. Transform S' in domain, to obtain S'_d. That is, setting (a, b, c) = (0,0,1), make $S'_d(i) := S'(d \cdot i + e)$

2.2. Pattern value alignment.

2.2.1. Estimate value transformation parameters (a, b, c) as the values minimizing $\Sigma(a + b \cdot i + c \cdot S'_d(i) - M(i))^2 / V(i)$

2.2.2. Transform pattern S'_d to obtain the new S', using: $S'(i) := a + b \cdot i + c \cdot S'_d(i)$; $\forall\, i \in$ Domain (S') = Domain (S'_d)

Fig. 5. Structure of the algorithm to compute the optimum linear alignment of a 1-D pattern, or signal, to a mean/variance model of the signal

The algorithm is based on an assignment $h(i)$ of points in $Domain(S)$ with the corresponding points in $Domain(M)$. This assignment is computed as the most similar point to $S(i)$ around a local proximity in $M(i)$. The similarity is a combination of position and slope likeness. The reliability degree $w(i)$ is proportional to the maximum similarity and inversely proportional to the similarity of non-maximum.

5 Experimental Results

The purpose of the experiments described herein has been to assess the invariance and robustness of the aligned projection representation and its potential to discriminate

between faces and non-faces. In this way, the results indicate both the strength of the alignment algorithm and the feasibility of modeling faces with projections.

The test set consists of 325 face (R_{FACE}) and 292 non-face ($R_{NON\text{-}FACE}$) regions segmented from 310 color images, using color segmentation (see step 1, in Fig. 3). These images were captured from 12 different TV channels, with samples taken from news, series, contests, documentaries, etc. The existing faces present a wide range of different conditions in pose, expression, facial features, lighting and resolution. Some of them are shown in Fig. 7. The face model was computed using a reduced set of 45 faces, not used in the test set, and is shown in Fig. 6.

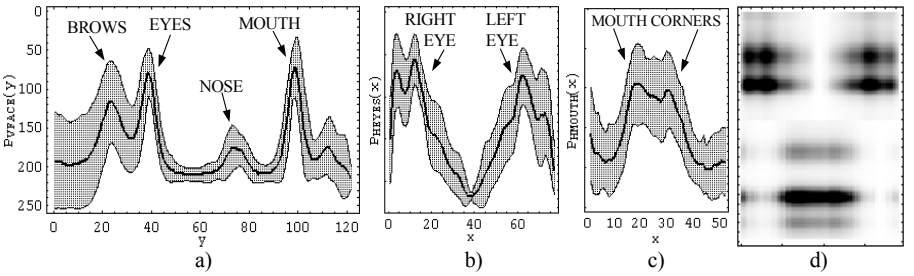

Fig. 6. Integral projection model of the face. a) $M_{V,FACE}$ and $V_{V,FACE}$. b) $M_{H,EYES}$ and $V_{H,EYES}$. c) $M_{H,MOUTH}$ and $V_{H,MOUTH}$. d) Reprojection of the face model

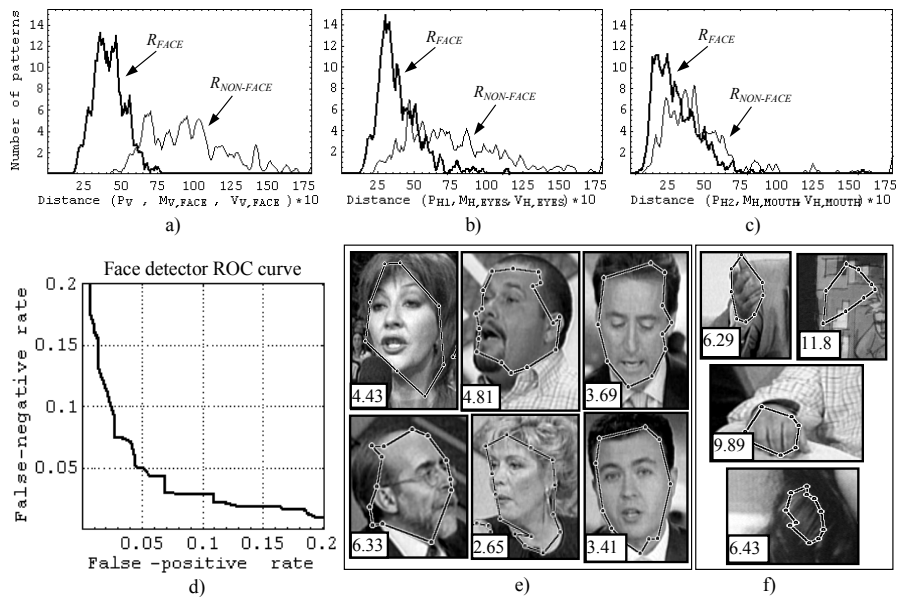

Fig. 7. Detection results. a-c) Distances of face and non-face aligned projections to the model: a) *V,FACE*; b) *H,EYES*; c) *H,MOUTH*. d) ROC curve of the face detector. e), f) Some face and non-face regions, respectively. The distances of P_V to ($M_{V,FACE}$, $V_{V,FACE}$) are shown below

As described in Section 3, detection is done according to the distance from the model to the aligned projections (P_{VR}, P_{HR1}, P_{HR2}) of each region R, using equation (9). The distances obtained for the test set are shown in Fig. 7. As expected, R_{FACE} projections yield lower distance values, although a certain overlapping exists. This overlapping is 11%, 39% and 67% for Figs. 7a), 7b) and 7c) respectively, so, by itself, the vertical projection of the whole face is the most discriminant component.

The results of the face detector, using different distance thresholds, are shown in the ROC curve in Fig. 7d). At the point with equal number of false-positives and false-negatives the detection ratio is 95.1%. A direct comparison with other methods is not meaningful, since the color-based attention process should also be taken into account. In previous experiments [3], this process showed an average detection ratio of 90.6% with 20.1% false-negatives (similar results are reported by other authors in [5]). Combined with the projection method, the faces detected are 86.2% with 0.96% of false-negatives. These results are comparable with some state-of-the-art appearance based methods (see [7] and references therein), detecting between 76.8% and 92.9% of the faces, but with higher numbers of false detections.

6 Conclusion and Future Work

This work constitutes, to the best of our knowledge, the first proposal concerning the definition and use of one-dimensional face patterns. This means that projections are used not only to extract information from max-min analysis or similar heuristic methods, but to model object classes and perform object detection and analysis. The preliminary experiments have clearly shown the feasibility of our proposal.

Somewhat, our approach could be considered equivalent to a 2-D appearance based face detection, where the implicit model is like the image shown in Fig. 6d). However, our method has several major advantages. First, working with 1-D signals involves an important improvement in computational efficiency. Second, the separation in vertical projection and then horizontal projections, makes the process very robust to non-trivial conditions or bad segmentation, without requiring exhaustive multi-scale searching. Third, the kind of projection model we have used, has proven an excellent generalization capability and invariance to pose, facial expression, facial elements and acquisition conditions.

References

1. Yang, M. H., Ahuja, N., Kriegman, D.: A Survey on Face Detection Methods. IEEE Trans. on Pattern Analysis and Machine Intelligence (to appear 2002)
2. Sobottka, K., Pitas, I.: Looking for Faces and Facial Features in Color Images. PRIA: Advances in Mathematical Theory and Applications, Vol. 7, No. 1 (1997)
3. García-Mateos, G., Vicente-Chicote, C.: Face Detection on Still Images Using HIT Maps. Third International Conference on AVBPA'2000, Halmstad, Sweden, June 6-8, (2001)

4. Yang, J., Stiefelhagen, R., Meier, U., Waibel, A.: Real-time Face and Facial Feature Tracking and Applications. In Proc. of AVSP'98, pages 79-84, Terrigal, Australia (1998)
5. Terrillon, J. S., Akamatsu, S.: Comparative Performance of Different Chrominance Spaces for Color Segmentation and Detection of Human Faces in Complex Scene Images. Vision Interface '99, Trois-Rivieres, Canada, pp.1821, (1999)
6. Gong, S., McKenna, S. J., Psarrou, A.: Dynamic Vision, From Images to Face Recognition. Ed. Imperial College Press (2000)
7. Rowley, H. A., Baluja, S., Kanade, T.: Neural Network-Based Face Detection. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 20, No. 1, pp. 23-38 (January 1998)
8. Moghaddam, B., Pentland, A.: Probabilistic Visual Learning for Object Detection. International Conference on Computer Vision, Cambridge, MA (1995)