ELSEVIER

Image and Vision Computing 18 (2000) 987–1003

image AND vision COMPUTING

www.elsevier.com/locate/imavis

Unsupervised and adaptive Gaussian skin-color model

L.M. Bergasa*, M. Mazo, A. Gardel, M.A. Sotelo, L. Boquete

Departamento de Electrónica Escuela Politécnica, Universidad de Alcalá, Campus Universitario s/n. 28805 Alcalá de Henares, Madrid, Spain

Received 4 June 1999; revised 13 March 2000; accepted 31 March 2000

Abstract

In this article a segmentation method is described for the face skin of people of any race in real time, in an adaptive and unsupervised way, based on a Gaussian model of the skin color (that will be referred to as Unsupervised and Adaptive Gaussian Skin-Color Model, UAGM). It is initialized by clustering and it is not required that the user introduces any initial parameters. It works with complex color images, with random backgrounds and it is robust to lighting and background changes. The clustering method used, based on the Vector Quantization (VQ) algorithm, is compared to other optimum model selection methods, based on the EM algorithm, using synthetic data. Finally, real results of the proposed method and conclusions are shown.

Keywords: Skin segmentation; Color clustering; Unsupervised learning; Bayesian methods; Gaussian mixture models

1. Introduction

The use of human face tracking systems within an image sequence has an increasing importance in the last years in applications such as: videoconferences with improved visual sensation [1], face identification in security systems [2], panoramic displays controlled with the gaze in virtual reality systems [3], lip readers [4], aid to the mobility of handicapped people [5,6], etc. To achieve this different techniques are applied, as for example, deformable templates to locate and track eyes and mouth in grey images [7]. Other authors as [1] use the pixel count of image edges, their integral projection and circular deformable templates to accomplish the eyes and mouth tracking. Baluja and Pomerleau [8], from The Carnegie Mellon University, apply an ALVINN neural network for this purpose. Cipolla [9], from the Cambridge University, locates certain points in the image employing a family of scalable Gabor filters and groups them in face candidates using geometric and grey level features. Using a probabilistic approach it locates the face with greatest probability among all candidates. On the other hand Yang, Waibel and Stiefelhagen [3,10] segment the face using a stochastic model of the skin color with some a priori model parameters calculated off-line and within this object they locate eyes and mouth. Heinzmann and Zelinsky [5] apply templates matching in grey and color images with the aid of a specific hardware called M.E.P. from Fujitsu to locate facial features such as: eyes, mouth, eyebrows, etc. Crowley and Coutaz [11] employ three processes, all in parallel: winking detection, histogram matching in normalized color and correlation to calculate the gaze direction. Through a confidence factor it knows at each moment the process to apply.

Color segmentation of the user face skin is a good method for doing face tracking because it is robust, easy to adapt to different light conditions and different users and performs in real time. On the other hand, color segmentation methods don't require many parameters. Within this, an interesting approach consists on applying statistic techniques, through which the different parts, of an image (classes) are well characterized by statistics measures of low order such as: mean, variance, correlation of functions or spectral power density. In this way the segmentation problem of an image is turned into a statistical optimization problem. That produces greater precision in the characterization of the classes in the image.

The segmentation techniques of images based on stochastic models may be supervised or unsupervised. The design of an autonomous segmentation system implies the use of unsupervised techniques. However the low reliability of some methods or the high complexity of others are the reasons why they are not typically used in real time segmentation, being a current topic of research [12,13]. The main problem found in unsupervised segmentation is the model adjustment according to the image histogram. A methodical and general solution hasn't been found yet.

* Corresponding author. Tel.: +34-91-885-6569-40; fax: +34-91-885-6591.

E-mail address: bergasa@depeca.alcala.es (L.M. Bergasa).

0262-8856/00/$ - see front matter
PII: S0262-8856(00)00042-1

In this article, we present a method for the calculation of a stochastic adaptive model of the color distributions of the skin on a normalized color space in an unsupervised way with a low computational cost. We will refer to this algorithm as Unsupervised and Adaptive Gaussian Skin-Color Model (UAGM).

This property makes the movement estimation easier since only an adjustment of the model is needed to estimate it. However, the color is not a physical phenomenon but a perception of the spectral characteristics of an electromagnetic radiation in the visible spectrum captured by the retina [15]. The skin color as a feature to track human faces presents several problems: (1) the color of the face obtained by the camera is influenced by factors such as light conditions, movement of objects, etc.; (2) different cameras produce different colors for the same person and under the same lighting conditions; and (3) the skin color changes from one person to other. It is necessary to solve the noted problems in order to use the color feature.

The authors of this work have accomplished a study of the distribution of human color skin in different color spaces (RGB, normalized RGB, HSI, SCT, YQQ [16]) and came to the conclusion that the best space for this application is normalized RG.

$$r = \frac{R}{R+G+B} \qquad g = \frac{G}{R+G+B} \tag{1}$$

In this space, the human skin color forms a compact class, the color differences between different people are reduced working with chromaticities, eliminating the intensity. Under certain light conditions, the skin color distribution can be modeled by a Gaussian function in "rg" space [10]. In spite of this, the segmentation may yield considerable errors if it works with a universal a priori model, being as well dependent on both lighting conditions and the camera used. To solve this, a personalized model is proposed for each user that is capable of detecting the skin class of that person, in an unsupervised way. It works properly with random backgrounds, making the model adaptive. In this way the previously three outlined problems are solved.

This article has been structured as follows. Section 2 describes the UAGM algorithm. Section 3 shows a comparison between the clustering method here proposed, based on the Vector Quantization algorithm (VQ), and other optimum models selection methods, based on the EM algorithm, using a series of synthetic data. It is demonstrated that equal or better results are obtained with the proposed method, being easier to apply and having smaller computational cost. Also some empirical segmentation results with real data for the UAGM method are shown. Finally, in Section 4 the conclusions on the proposed method are drawn.

2. UAGM method

Let $\mathbf{X}$ be a finite set of pixels of an image, $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, with each pixel defined by its color components "rg" $\mathbf{x}_j = (x_{jr}, x_{jg})$. For simplicity, a 2D image has been indexed as a 1-D array of length N. Let K be the number of classes into which the N components of $\mathbf{X}$ must be classified. We consider a model of K components (M_K) where each model is defined by a vector of parameters $\theta_K \in \Re^d$. Assume the probability of each class $P(\omega_i)$ is a priori known, and that the probabilistic structure of each class will be considered a Gaussian function. Within a Bayesian approach, denoting by $P(X|\omega_i, \theta_i)$ the probability that a pattern pertaining to class i takes the value $\mathbf{X}$, the probability of $\mathbf{X}$ for the statistics of all classes $\theta = (\theta_1, \theta_2, \ldots \theta_k)$ will be

$$P(X|\theta) = \sum_{i=1}^{K} P(X|\omega_i, \theta_i) P(\omega_i) \tag{2}$$

The unsupervised classification goal will be to estimate vector θ applying the maximum likelihood method, consistent in estimating the vector of parameters $\hat{\theta}$ that maximizes the probability $P(X|\theta)$. Making use of standard techniques [17] the following equations are obtained:

$$\begin{aligned} P(\omega_i|x_j, \hat{\theta}_i) &= \frac{P(x_j|\omega_i, \hat{\theta}_i)\hat{P}(\omega_i)}{\sum_{i=1}^{K} P(x_j|\omega_i, \hat{\theta}_i)\hat{P}(\omega_i)} \\ &= \frac{|\hat{C}_i|^{-1/2} \exp\{-\frac{1}{2}(x_j - \hat{m}_i)^{\mathrm{T}} \hat{C}_i^{-1}(x_j - \hat{m}_i)\} P(\hat{\omega}_i)}{\sum_{i=1}^{K} |\hat{C}_i|^{-1/2} \exp\{-\frac{1}{2}(x_j - \hat{m}_i)^{\mathrm{T}} \hat{C}_i^{-1}(x_j - \hat{m}_i)\} P(\hat{\omega}_i)} \end{aligned} \tag{3}$$

$$\hat{P}(\omega_i) = \frac{1}{N} \sum_{j=1}^{N} P(\omega_i|x_j, \hat{\theta}_i) \tag{4}$$

$$\hat{m}_i = \frac{\sum_{j=1}^{N} P(\omega_i|x_j, \hat{\theta}_i) x_j}{\sum_{j=1}^{N} P(\omega_i|x_j, \hat{\theta}_i)} \tag{5}$$

$$\hat{C}_i = \frac{\sum_{j=1}^{N} P(\omega_i|x_j, \hat{\theta}_i)(x_j - \hat{m}_i)(x_j - \hat{m}_i)^{\mathrm{T}}}{\sum_{j=1}^{N} P(\omega_i|x_j, \hat{\theta}_i)} \tag{6}$$

for $i \in [1, K]$, where $\hat{m}_i$ is the mean of class i, $\hat{C}_i$ is the covariance matrix for class i, $\hat{P}(\omega_i)$ is the a priori probability for class i and $P(\omega_i|x_j, \hat{\theta}_i)$ is the probability that datum x_j belongs to class i.

The explicit values of $\hat{P}(\omega_i), \hat{m}_i, \hat{C}_i$ cannot be obtained from these equations. It is a set of non-linear equations that do not yield a unique solution and requires an iterative

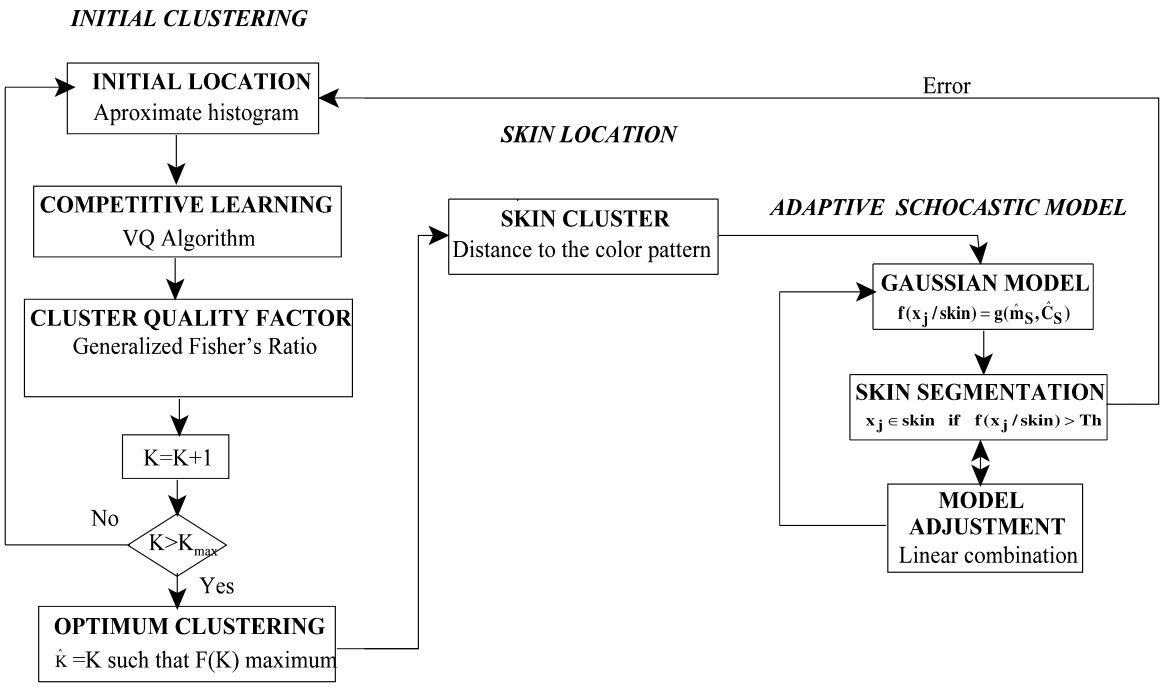

Fig. 1. Stages of the UAGM algorithm.

procedure to be solved. Expectation-Maximization (EM) [12] has been the algorithm used to resolve this problem.

The estimation of K is known as the cluster validation problem. One possible solution to this problem is to treat K as a parameter to be estimated along with the model parameters θ. This approach results in a maximum-likelihood (ML) estimate of K which has been shown to be strongly biased toward the maximum number of states considered. This bias reflects the roughly monotonically increasing behavior of the intensity log-likelihood as a function of K [13]. An alternative solution for the problem accomplishes a segmentation for different candidate classes (K), according to the applied method, and fix a cost function that permits to determine the optimum K. Following this line there are different methods with distinct cost functions (FHV, Evidence density, MDL, MML, GMM, etc.). These will be briefly explained in point 3 making a comparison to the UAGM method.

The a posteriori likelihood $P(\omega_i|x_j, \hat{\theta}_i)$ is simplified to reduce the parameters estimate calculation, assuming the following hypothesis: (1) *Normalization is not performed*. Pixels are classified into classes according to their probabilities to belong to one of such classes. Before comparing, the probabilities should be normalized using the same normalizing factor. That is the reason why normalization is not necessary. (2) *The probabilities for each a priori class are equal*. The equiprobability hypothesis is assumed as the content of the image is not known before hand. (3) *The covariance matrices for each one are equal*. It is intended to carry out a rough segmentation of the image to estimate the main colors. Skin should belong to one of these colors as it looks as a big blob in the image. (4) *Variances are equal and crosscovariances are assumed to be zero*, denoting this, independence between the components of the class. It was empirically demonstrated for skin class.

The a posteriori probability (Eq. (3)) becomes the following discriminate function:

$$df(\omega_i|x_j, \hat{\theta}_i) = (x_j - \hat{m}_i)^{\mathrm{T}}(x_j - \hat{m}_i) = \|x_j - \hat{m}_i\|^2 \tag{7}$$

This way the statistical estimation problem is reduced to a clustering one by the Euclidean distance where one must estimate the statistics ($\hat{m}_i$) of each class and the number of classes or existing colors ($\hat{K}$) in an image. Though the precision of the segmentation is reduced applying the hypothesis proposed, it is enough to detect the main colors of an image and, therefore, give a good estimate of the skin color.

To estimate ($\hat{m}_i$) a local competitive learning is applied based on Euclidean distance employing the Vector Quantization method (VQ) proposed by Kohonen [18]. To reduce the problem of local learning an initialization method will be used based on an approximate histogram. To estimate the number of classes ($\hat{K}$) the resulting clustering will be evaluated comparing to the topology of the colors distribution through a cost function, that attempts to minimize the internal deviation between the pixels belonging to the same class and maximize the distance between the different classes. A number of classes between 2 and a maximum ($K_{máx}$) will be tested, so that the value of K together with the maximum value of the cost function will give the number of the main colors ($\hat{K}$) in the image. The centroid location of the classes will represent the estimated means of the classes $\hat{m} =$

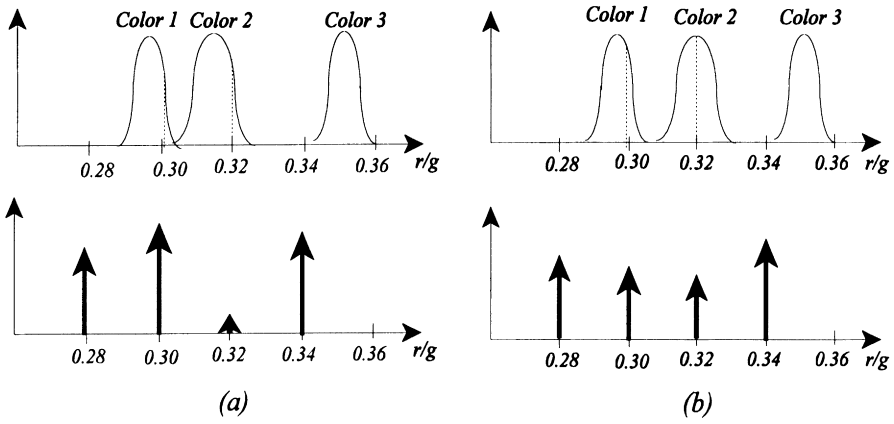

Fig. 2. Effect of the dispersion on several acumulators.

$(\hat{m}_1, \hat{m}_2, \ldots, \hat{m}_K)$. The skin class is located among these classes, calculating the distance to the pattern applying a Gaussian model on it. All those pixels for which the probability density function of the model yields a value greater than an adaptive threshold (Th) will be segmented as skin. Finally an adaptation of the model is accomplished through the estimate of its parameters, using a linear combination of those which are already known under the maximum likelihood criterion. In Fig. 1 a schema with the different phases of the process is shown.

Each phase in the clustering method is described in detail below.

2.1. Initial location

The aim of this phase is to give an initial estimation of the mean vectors of the classes that best approximate the color distributions of the histogram. Starting from these positions a local competitive learning algorithm will be applied to obtain the mean vector for each class. This type of learning makes the final positions of the vectors dependent on the initial estimate.

A good initial estimate is obtained applying an approximate histogram of the "rg" space. The mean vectors are located on the positions of greatest pixel concentrations in the histogram. These concentrations will be associated to the main colors of the image, among which is the skin color. The colors produce spatial Gaussian distributions in the histogram with a size equal to the working resolution (256×256 colors). However, these distributions become delta funtions choosing a smaller resolution of only $N \times N$ (much smaller than 256) chromaticities.

The user can choose the resolution of the approximate histogram, H, specifying the number of accumulators, N, for each color component. Therefore, each component is split into N intervals of size S equal to the dynamical range of the color component, P, divided by the number of accumulators, N, ($S = P/N$). The approximate histogram will be an $N \times N$ matrix where each accumulator is initialized to zero. For each image pixel, $\mathbf{x} = (x_r, x_g)$:

$$H(f_{\text{acum}}(x)) = H(f_{\text{acum}}(x)) + 1 \tag{8}$$

$$f_{\text{acum}}(x) = \left(\text{truc}\left(\frac{x_r}{S}\right), \text{truc}\left(\frac{x_g}{S}\right)\right) \tag{9}$$

where truc( ) indicates the floor rounded value of integer division.

The histogram approximation implies an $S \times S$ color resolution so that colors that are separated in the histogram less than this value, will fall in the same accumulator and will be considered as a unique color. The image main colors are obtained this way. For this application, we use a 50×50 accumulators matrix that provides a resolution of 0.02, for a dynamical range of 1 in the color components (r,g $\in$ [0,1]). This resolution is more than sufficient for our purposes as it implies to detect 2500 different chromaticities independent on luminance. On the other hand the skin class has a maximum evaluated variance of 0.0175. Most of the pixels belonging to one class will fall within the same cell. Color variances of other objects having the same or smaller values have been evaluated.

The algorithm locates the initial positions of the K vectors under evaluation on the positions of the K greatest accumulators obtained from the approximate histogram. These positions are not the exact centers of the color distributions but provide a good initial approximation of them with a maximum error of $\pm P/2N$.

A problem that can appear is the dispersion of a color distribution in several accumulators depending on where the mean of the distribution falls. In Fig. 2(a) this effect is noted assuming that distributions mostly fall in an accumulator and therefore the dispersion effect is negligible. In Fig. 2(b) a great dispersion of Color 2 is appreciated that causes the distribution over two accumulators. Nevertheless this effect is corrected during the training and determination phase of the optimum number of objects as a number of vectors equal to the main color distributions are located on the real centers of the Gaussians.

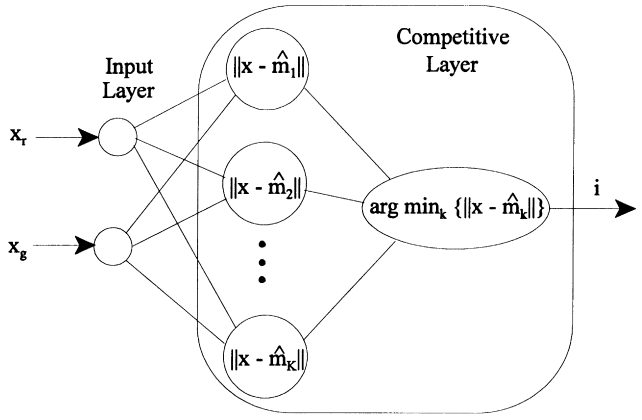

Fig. 3. Competitive learning.

2.2. Competitive learning

To adjust the cluster centers a competitive learning algorithm is used based on Euclidean distance proposed by Kohonen and denoted as VQ ("Vector Quantization"). This algorithm uses a reduced set of vectors (called neurons) to approximate a large volume of information. To apply the VQ algorithm, the number of vectors of the approximation (K) and an estimate of the initial positions of these vectors must be known. The first parameter will be evaluated between a minimum of 2 (as each image is assumed to have at least a face and a background) and a maximum value (K_{max}). For each value of K the learning phase is accomplished to further evaluate the resulting clustering through the cost function.

The initial locations of the vectors are determinant in this method since they can largely influence on the accuracy of the final result. In Refs. [19,20] a discussion on this topic is presented. In our case the local learning problems of VQ method are not solved using random data initialization but the proposed method, assuring a good estimate for the neurons.

Given the finite set of normalized pixels of an image $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ and a number of mean vectors located on the positions defined by $\hat{m}$:

$$\hat{m} = (\hat{m}_1, \hat{m}_2, \ldots, \hat{m}_K); \quad \hat{m}_i = (\hat{m}_{ir}, \hat{m}_{ig}) \tag{10}$$

The VQ method gives the best approximation to the probability density function, $f(\mathbf{x})$, of the stochastic variable $\mathbf{x} \in R^2$ making use of a finite number of vectors $\hat{m}$, called neurons. A two layers system is used: an input layer and a competitive layer, as can be seen in Fig. 3.

Index **i** is obtained in an implicit way by a decision process of the form

$$i = \arg\min_k \{\|x - \hat{m}_k\|\} \tag{11}$$

where $\| \|$ denotes the Euclidean norm.

To calculate the best approximation of **X**, Kohonen defines a quadratic mean error of the quantification function, as it is shown in Eq. (12). The minimum of this function will give the set of vectors $\hat{m}$ that best approximates **X**. The gradient descent technique is employed to find the minimum, obtaining recurrent Eq. (13) to move the neurons in the color space.

$$E = \int \|x - \hat{m}_i\|^2 f(x)\, \mathrm{d}x \tag{12}$$

$$\hat{m}_i(t+1) = \hat{m}_i(t) + \gamma(t)[x(t) - \hat{m}_i(t)] \tag{13}$$

where $\gamma(t)$ is the learning step varying in the range [0,1]

A training subset $\mathbf{X}_L = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L\}$ is randomly taken from the input samples **X**. The Euclidean distance to the vectors $\hat{m}$ is calculated for each sample $\mathbf{x}_j$. The vector showing the minimum Euclidean distance to the sample will be the winner. This vector is moved an amount proportional to the distance that separates the pixel from the vector. This process is iteratively repeated until the vectors are moved less than an a priori empirically defined threshold. The quantity that pixels are moved is controlled by parameter $\alpha(t)$ and decays with time.

2.3. Clustering quality factor

Through the previous learning process the best possible approximation of the K vectors color distribution is obtained. A classification of the pixels in the vectors' prototypes is accomplished below, using the Euclidean distance to the vectors:

$$x_j \in \alpha_i \text{ if } i = \arg\min\{\|x_j - \hat{m}_k\|\}\ 1 \le k \le K;\ \forall x_j \in X \tag{14}$$

A quality factor is needed to evaluate the adjustment between the number of classes and the color distribution. The maximum value of the factor corresponds to the optimum number of classes and therefore the optimum clustering. Pixels between classes are distributed in such a way that some measure of internal similarity of the class is maximized and therefore the divergence between the different classes is also maximized.

Prototype vectors (or mean values for each class) are defined by Eq. (15) where M_k is the number of pixels that belong to cluster kth.

$$\hat{m}_k = \frac{1}{M_k} \sum_{i=1}^{M_k} x_i \quad 1 \le k \le K \tag{15}$$

The mean pattern vector for all classes will be

$$\hat{m}_0 = \frac{1}{M} \sum_{i=1}^{M} x_i = \sum_{k=1}^{K} \hat{m}_k \tag{16}$$

where M indicates the total number of pixels to classify.

The within-cluster scatter matrix and the between-cluster

scatter matrix can be regarded as

$$S_{\mathrm{W}} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{M_k}\sum_{i=1}^{M_k}[x_i - \hat{m}_k][x_i - \hat{m}_k]^{\mathrm{T}} \tag{17}$$

$$S_{\mathrm{B}} = \frac{1}{K}\sum_{k=1}^{K}[\hat{m}_k - \hat{m}_0][\hat{m}_k - \hat{m}_0]^{\mathrm{T}} \tag{18}$$

In the last matrix all classes have the same weight when defining the mean variance with respect to the pattern. It makes that classes with very few pixels decrease the effect of classes with a great number of pixels. In this application we intend to locate the image main colors. A modification has been introduced on this matrix so that the average variance is weighted for each class with respect to the global pattern, through the number of pixels of the class. Therefore, a modified between-cluster scatter matrix has been applied, defined by the following equation:

$$S_{\mathrm{B}} = \frac{1}{KM}\sum_{k=1}^{K}M_k[\hat{m}_k - \hat{m}_0][\hat{m}_k - \hat{m}_0]^{\mathrm{T}} \tag{19}$$

The within-cluster and between-cluster scatter matrices depend on how the pixels are distributed on the different classes. Thus, for homogeneous classes S_{W} matrix decreases while S_{B} matrix increases since the variance between classes is greater. Matrix S_{W} is minimized and, conversely, matrix S_{B} is maximized to achieve the greatest similarity between pixels belonging to one class. Therefore, the main goal will be to increase the ratio, i.e. to augment the variance between classes with respect to the internal variance of each class.

One of the most widely used criteria maximizes the trace of $S_{\mathrm{W}}^{-1}S_{\mathrm{B}}$, according to Eq. (20). This criterion is known as Hotelling or generalized Fisher ratio [21]. For a given number of classes K the cost function (F_K) will be the sum of the eigenvalues $\lambda_1,\lambda_2,\ldots,\lambda_K$ of matrix $S_{\mathrm{W}}^{-1}S_{\mathrm{B}}$.

$$F_K = \mathrm{tr}[S_{\mathrm{W}}^{-1}S_{\mathrm{B}}] = \begin{bmatrix} \lambda_1 & & \neq 0 \\ & \lambda_2 & \\ & \vdots \quad \vdots & \\ \neq 0 & & \lambda_K \end{bmatrix}$$

$$= \sum_{i=1}^{K}\lambda_i; \quad 1 \le K \le K_{max} \tag{20}$$

Thus, the optimum number of classes will be the maximum in the cost function (F). The degree of success achieved under this criterion depends on how the patterns to classify are grouped. The final results will be right if these patterns form compact groups, i.e. they are well separated and features for each pattern are independent. We work under the assumption that the previous conditions are met, since color distributions have compact shapes, they are relatively separated and the dependence between characteristics (r and g) is low.

2.4. Skin segmentation

Once the main colors that compose the image are classified, the skin class must be located among them. A class (not pixel) discriminant method is used, consistent with computing the Euclidean distance between the centers of the image clusters ($\hat{m}_k$) and a pattern cluster representing the color prototype for human skin (m_{pattern}). The closest class to this prototype will be considered as the skin class, according to the following equation:

$$\hat{m}_{\mathrm{S}}^{(0)} = \min_k\left\{\|\hat{m}_k - m_{\mathrm{pattern}}\|\right\} \tag{21}$$

To improve the segmentation of this class, the color distribution of skin class pixels is modeled through a 2D Gaussian function, $N(m_{\mathrm{S}}^{(0)}, C_{\mathrm{S}}^{(0)})$, where $m_{\mathrm{S}}^{(0)}$ stands for the position of the skin class prototype and $C_{\mathrm{S}}^{(0)}$ is the covariance matrix of the color components "rg" of the pixels classified as skin (M_{S}). This Gaussian function provides the probability that a pixel belongs to the skin class:

$$f(x_j|\mathrm{skin}) = \frac{1}{2\pi|\hat{C}_{\mathrm{S}}|^{1/2}}\exp\left[-\tfrac{1}{2}(x_j - \hat{m}_{\mathrm{S}})^{\mathrm{T}}\hat{C}_{\mathrm{S}}^{-1}(x_j - \hat{m}_{\mathrm{S}})\right] \tag{22}$$

A threshold (Th) is established so that if $f(x|\mathrm{skin})$ is greater than the chosen threshold the pixel is considered to belong to the skin class.

2.5. Model adjustment

A linear combination of the previous model parameters will be used to predict the new ones. Let $\mathbf{X}_{\mathrm{S}}$ be the set of pixels belonging to the skin class, modeled by a 2D normal function. If $\mathbf{Y}_{\mathrm{S}} = \mathbf{B}\mathbf{X}_{\mathrm{S}}$ is a linear transformation of $\mathbf{X}_{\mathrm{S}}$, where $\mathbf{B}$ is a $(m \times p)$ real matrix of rank m, with $m \le p$, then $\mathbf{Y}_{\mathrm{S}}$ is also a 2D normal distribution.

A stochastic Gaussian model has been used, defined by its mean ($\hat{m}_{\mathrm{S}}^{(0)}$) and covariance ($\hat{C}_{\mathrm{S}}^{(0)}$), to perform the skin segmentation of a static image. In an image sequence, it can be considered that the estimated value of the statistics will be a linear combination of the last $\mathbf{z}$ values calculated in the previous iterations, i.e.

$$\hat{m}_{\mathrm{S}}^{(p+1)} = \sum_{l=0}^{z-1}\alpha_l R_l \tag{23}$$

$$\hat{C}_{\mathrm{S}}^{(p+1)} = \sum_{l=0}^{z-1}\beta_l R_l \tag{24}$$

where $\hat{m}_{\mathrm{S}}^{(p+1)}$ is the estimated mean vector at instant $(p+1)$; $\mathbf{R}_i$, are the previous mean vectors and $\alpha_l \le 1$ are the weighting coefficients used to calculate the estimated mean, $l = 0,\ldots,z-1$. $\hat{C}_{\mathrm{S}}^{(p+1)}$ Is the estimated covariance matrix at instant $(p+1)$; $\beta_l \le 1$ are the weighting coefficients for the estimated covariance; and $\mathbf{S}_{\mathrm{l}}$ are the previous covariance matrices. Weighting coefficients determine how the previous statistics influence on the estimation of the current

statistics. The maximum likelihood criterion will be used to find the best set of coefficients for optimal prediction. The probability function obtained upon applying the estimated model to the M_S skin class pixels in the 2D normalized color space, will be the product of the probabilities for each pixel.

$$L = \prod_{k=1}^{M_S} f(x_k) = \frac{1}{(2\pi)^{M_S}|\hat{C}_s|^{1/2M_S}} \times \exp\left[-\frac{1}{2}\sum_{k=1}^{M_S}(x_k - \hat{m}_s)^T \hat{C}_s^{-1}(x_k - \hat{m}_s)\right] \quad (25)$$

Applying simple mathematical transformations and properties related to the matrix trace, the following equation can be derived.

$$\log L = -M_S \log(2\pi) - \tfrac{1}{2}M_S \log|\hat{C}_S| - \tfrac{1}{2}M_S \operatorname{tr} \hat{C}_S^{-1}C_S - \tfrac{1}{2}M_S(m_S - \hat{m}_S)^T\hat{C}_S^{-1}(m_S - \hat{m}_S) \quad (26)$$

The first partial derivative of $\log L$, with respect to the weighting coefficients, is calculated in order to obtain the maximum probability. A numerical iterative technique proposed by Anderson [22] is used to solve this equation, as in general, there does not exist an explicit solution. Basically, coefficients $\alpha_l^{(i)}$ and $\beta_l^{(i)}$ are calculated in an iterative and independent way, where the superscript (i) denotes for the ith iteration at step (p). The iterative process implies the parameters calculation in the following order: α_l, $\hat{m}_S$, C_S, β_l, $\hat{C}_S$, $1 = 0,\ldots,z-1$. The process is stopped if $\max(|\beta_l^{(i)} - \beta_l^{i-1}|, 1 = 0,\ldots,z-1) \le \epsilon$, where ϵ is an error parameter empirically defined. The algorithm for the ith iteration will be:

$$\alpha_j^{(i)} = \left(\sum_{l=0}^{z-1} R_j^T(\hat{C}_S^{(i-1)})^{-1}R_l\right)^{-1} R_j^T(\hat{C}_S^{(i-1)})^{-1}m_S\, j = 0,\ldots,z-1 \quad (27)$$

$$\hat{m}_S^{(i)} = \sum_{l=0}^{z-1}\alpha_l^{(i)}R_l \quad (28)$$

$$C_S^{(i)} = \frac{1}{M_S}\sum_{l=1}^{M_S}(x_l - m_S)(x_l - m_S)^T + (x_l - \hat{m}_S^{(i)})(x_l - \hat{m}_S^{(i)})^T \quad (29)$$

$$\sum_{l=0}^{z-1}\operatorname{tr}(\hat{C}_S^{(i-1)})^{-1}S_j(\hat{C}_S^{(i-1)})^{-1}S_l\beta_l^{(i)} = tr(\hat{C}_S^{(i-1)})^{-1}S_j(\hat{C}^{(i-1)})^{-1}C_S^{(i)}\ j = 0,\ldots,z-1 \quad (30)$$

$$\hat{C}_S^{(i)} = \sum_{l=0}^{z-1}\beta_l^{(i)}S_l \quad (31)$$

The model is applied on a new image $(p+1)$, using the estimated statistics at pth iteration, the segmentation is accomplished and the parameters are again estimated for the next image. It has been empirically observed that the segmentation improves using an adaptive threshold proportional to the trace of the estimated covariance matrix, as it is shown in the following equation:

$$\mathrm{Th} = K_{\mathrm{Th}}\operatorname{tr}[\hat{C}_S] \quad (32)$$

3. Results

At this point, a comparative study between the proposed method to find the optimum number of classes and methods from other authors is presented, using a series of synthetic data referenced in [14].[1] Practical results obtained with the UAGM method are also analyzed, applied to a real case of skin segmentation.

3.1. Comparison with other methods

In this section we provide a brief description of the different methods chosen to establish the comparison. Readers are referred to Refs. [14,17,23–25] for further information.

1. *Fuzzy Hypervolume* (FHV). This method is outlined in Ref. [23] and looks at models with the lowest total volume, defined via

$$V(K) = \sum_{k=1}^{K}\sqrt{|C_k|} \quad (33)$$

where C_k is the covariance matrix of class k.

2. *Evidence density*. This technique is explained in Ref. [24] and uses the FHV measure to penalize the log of the likelihood $(L(X|\theta) = \ln P(X|\theta))$ at the maximum likelihood solution

$$\rho(K) = \frac{L(X|\hat{\theta}_K)}{V(K)} \quad (34)$$

3. *Minimum description length* (MDL). This method was developed by Rissanen [25]. It is based on the selection of the model order that minimizes a length function formed by a combination of data and model parameters.

$$\mathrm{MDL}(K) = -L(X|\hat{\theta}_K) + \tfrac{1}{2}N_p(K)\ln N \quad (35)$$

where $N_p(K)$ is the number of parameters in the K Gaussian model, and N is the number of data points.

4. *Minimum message length* (*MML*). Created by Wallace and Freeman, and further extended in Ref. [17]. The MML expression used is a given in Ref. [17].

[1] IEEE Transactions on Pattern Analysis and Machine Intelligence, November 1998.

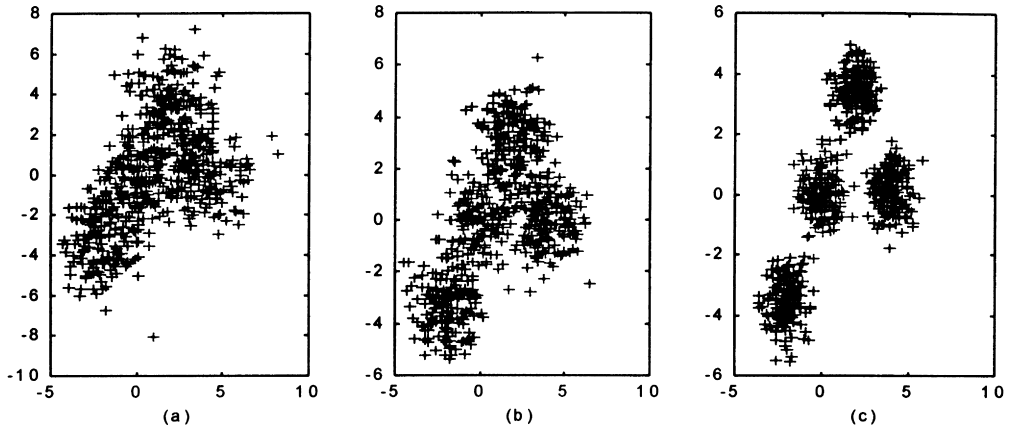

Fig. 4. Data for experiment 1: (a) $\sigma = 1.2$; (b) $\sigma = 1.0$; (c) $\sigma = 0.66$.

$$\mathrm{MML}(K) = K\sum_{i=1}^{d} \ln(2\sigma^2_{\mathrm{pop}(i)}) - \ln(K-1)! + \frac{N_\mathrm{p}}{2}\ln\kappa(N_\mathrm{p}) - \ln K! + \sum_{i=1}^{d}\sum_{k=1}^{K} \ln\frac{\sqrt{2}N_k}{\sigma^2_{k,i}} + \frac{1}{2}\ln N - \frac{1}{2}\sum_{k=1}^{K}\ln P(k) - L(X|\hat{\theta}) + \frac{N_\mathrm{p}}{2} \tag{36}$$

where d is the dimension of the data set X, $\sigma_{\mathrm{pop}(i)}$ is the population variance of the ith measurement, $\kappa(N_\mathrm{p})$ is the optimum lattice quantification constant in an N_p dimensional space, $\sigma_{k,\mathrm{i}}$ is the maximum likelihood solution for the standard deviation of the ith measurement in the kth class, $P(k)$ is the relative abundance of class k, N_k is the number of data items belonging to class k (hence $N_k = P(k) \times N$) and $L(X|\hat{\theta})$is the log of likelihood at the maximum likelihood solution. This equation is derived from the asumption that, a priori, each component of the mean vector, $m_{k,i}$ for the K Gaussians, have a flat distribution in the range $(-\sigma_{\mathrm{pop}(i)}, \sigma_{\mathrm{pop}(i)})$. Likewise the standard deviations for each of the Gaussians, $\sigma_{k,i}$, are assumed to have a flat distribution in the range $(0,\sigma_{\mathrm{pop}(i)})$.

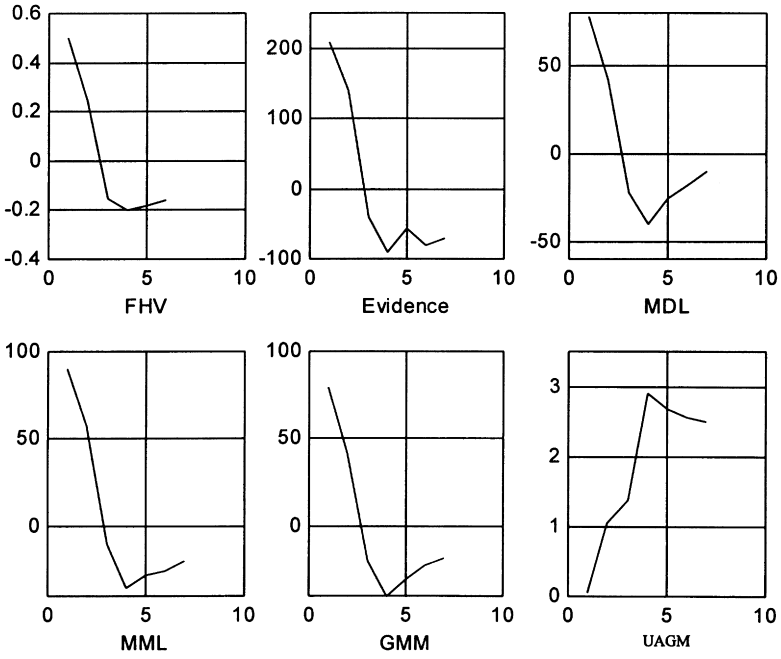

Fig. 5. Results for experiment 1, $\sigma = 0.66$.

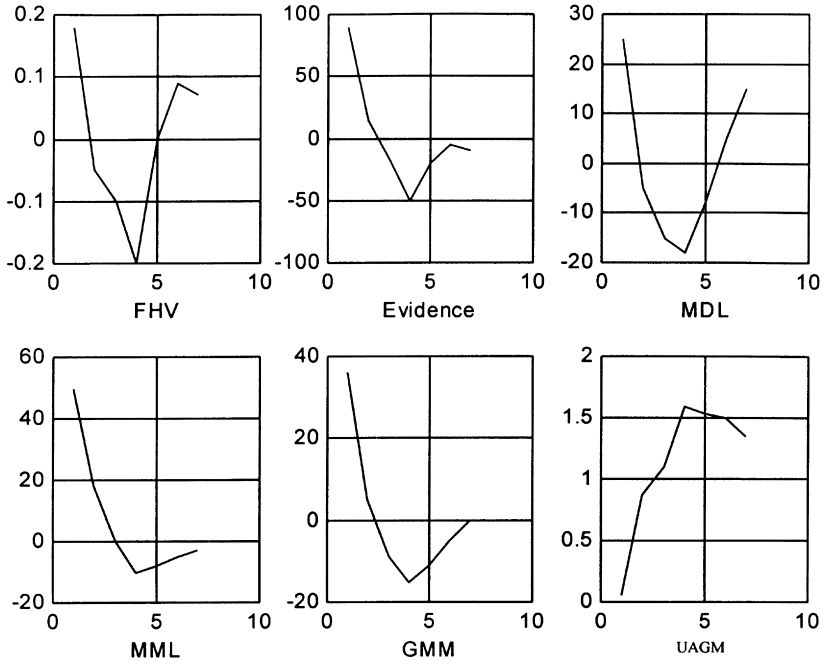

Fig. 6. Results of experiment 1, $\sigma = 1.0$.

5. *Gaussian mixture modeling* (GMM). Developed in Ref. [13]. It outlines that the evidence of a sample ($\ln P(X)$) depends on three factors.

$$\ln P(X) = L(X|\hat{\theta}) + f_{\text{post}}(H) + f_{\text{prior}}(\hat{\theta}, X) \tag{37}$$

where $L(X|\theta)$ is the log-probability under a Bayesian approach, f_{prior} is an a priori function and f_{post} is an a posteriori function that depends on the Hessian matrix of the GMM parameters. Thus, the estimated evidence of X is:

$$\ln P(X) = L(X|\hat{\theta}) - K\sum_{i=1}^{d} \ln(2\sigma^2_{\text{pop}(i)}) + \ln(K-1)!$$
$$+\frac{N_{\text{p}}}{2}\ln(2\pi) - \frac{1}{2}\left(\sum_{k=1}^{K-1}\ln\sum_{j=1}^{N}\left(\frac{\hat{P}(\omega_k|x_j,\theta_k)}{\hat{P}(\omega_k)} - \frac{\hat{P}(\omega_K|x_j,\theta_K)}{\hat{P}(\omega_K)}\right)^2\right.$$
$$\left.+2d\sum_{k=1}^{K}\ln(\sqrt{2}N\hat{P}(\omega_k)) - 2\sum_{k=1}^{K}\sum_{i=1}^{d}\ln\lambda_{k,i}\right) \tag{38}$$

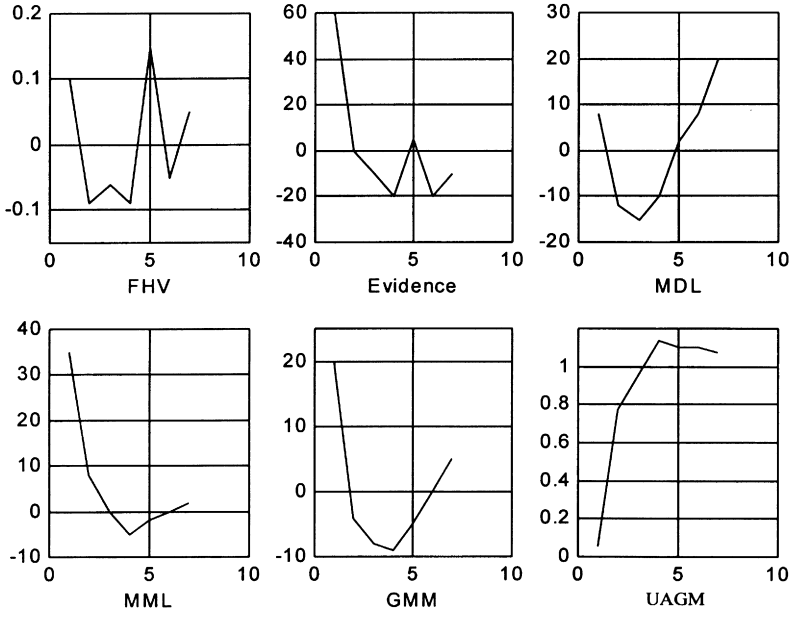

Fig. 7. Results of experiment 1, $\sigma = 1.2$.

where $\lambda_{k,l}$ are the eigenvalues of the covariance matrix. For simplicity, a normal assumption applied is to consider that the covariance matrices are diagonal. Thus, $\lambda_{k,l} = \sigma_{k,1}^2$. Notice that the a priori function is similar to the one used in the MML method, where the lattice constant has been substituted by its lowest limit. Linear interpolation is used for those values where the constant does not exist.

3.2. Experiment 1

To prove the properties of the previously commented methods, a simple classification problem has been used that resembles the problem of colors classification in an image. Data are generated using four Gaussian functions with the same standard deviation (σ) but different means:

$$m_1 = (0,0)^{\mathrm{T}} \qquad m_2 = (2,\sqrt{12})^{\mathrm{T}}$$

$$m_3 = (4,0)^{\mathrm{T}} \qquad m_4 = (-2,-\sqrt{12})^{\mathrm{T}}$$

$$\sigma = \{1.2, 1.0, 0.66\} \tag{39}$$

A hundred twenty samples are taken from each Gaussian, implying a total of 480 samples, and three different variance cases are evaluated (see Fig. 4), obtaining the results shown in Figs. 5–7. The results for the methods presented in Section 3.1 have been obtained after 10 iterations of the EM algorithm, employing a different random seed for each one. A maximum number of 10 iterations have also been taken in the UAGM method. For all methods, a number of classes (K) between 1 and 7 has been evaluated.

As can be observed, for the case of $\sigma = 0.66$ and $\sigma = 1.0$, all methods yield a correct result, identifying an optimum number of four classes corresponding to the four Gaussians of the experiment. For the case of $\sigma = 1.2$, only the MDL, MML, GMM and UAGM methods provide correct results. In all methods, the optimum result is provided by the minimum of the function, except in UAGM, where the optimum is given by the maximum of such function. Notice that there exists a clear correlation between MDL, MML and GMM methods, due to the intimate links existing between them [26,27]. On the other hand, it is interesting to also emphasize that the MML method appears to be more stable, under the same conditions.

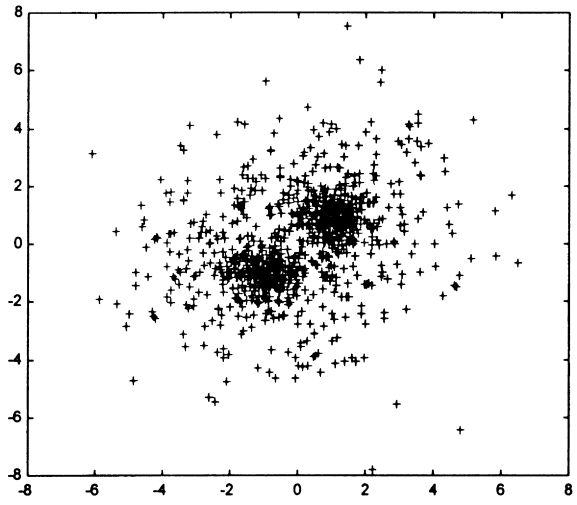

Fig. 8. Data for experiment 2.

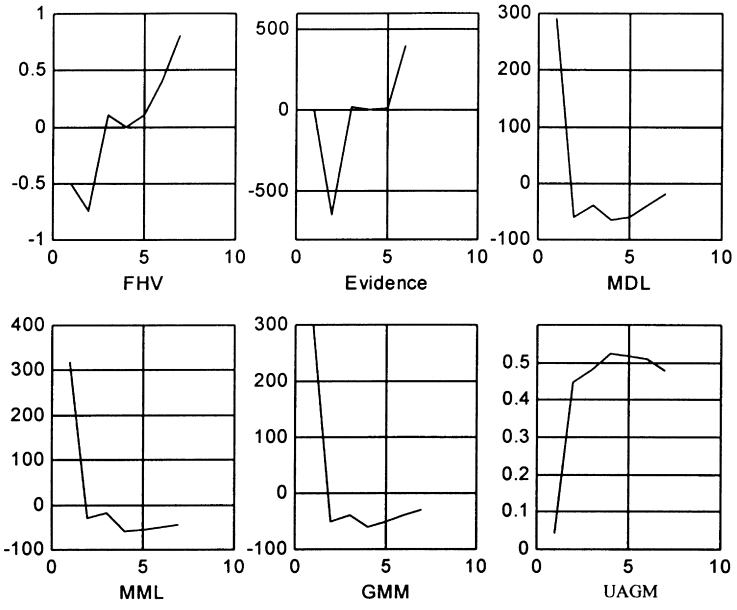

Fig. 9. Results for experiment 2.

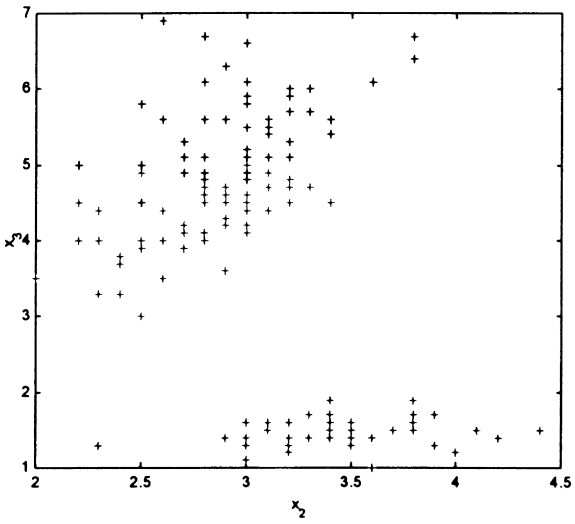

Fig. 10. Iris data.

3.3. Experiment 2

In this example, the behavior of the models is investigated upon data also generated by four Gaussians. However, in this experiment, Gaussians are paired such that each pair has a common mean. We set $\sigma_1 = \sigma_3 = 1$ and $\sigma_2 = \sigma_4 =$ 2.250 samples are taken from each function (see Fig. 8). We apply the same methodology as in previous example. The experiment results are shown in Fig. 9.

$$m_1 = m_2 = (1,1)^{\mathrm{T}}$$

$$m_3 = m_4 = (-1,-1)^{\mathrm{T}}$$

$$\sigma_1 = \sigma_3 = 1$$

$$\sigma_2 = \sigma_4 = 2 \tag{40}$$

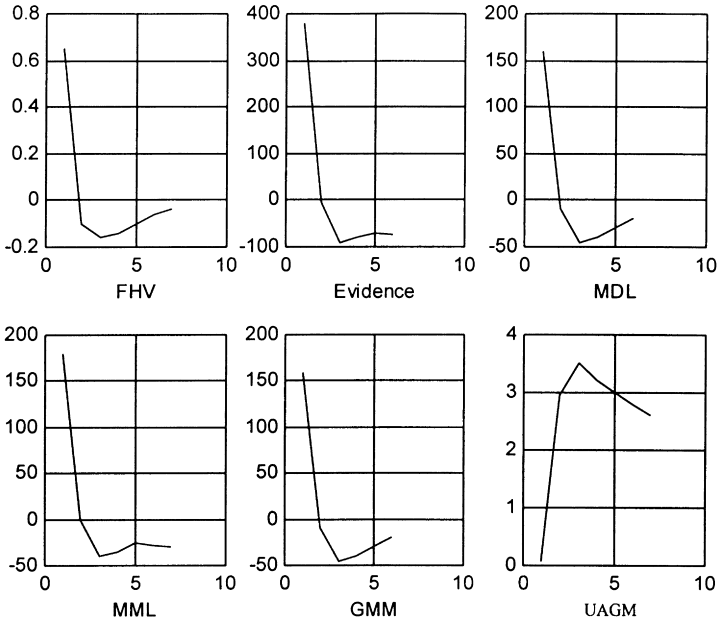

Fig. 11. Results for experiment 3.

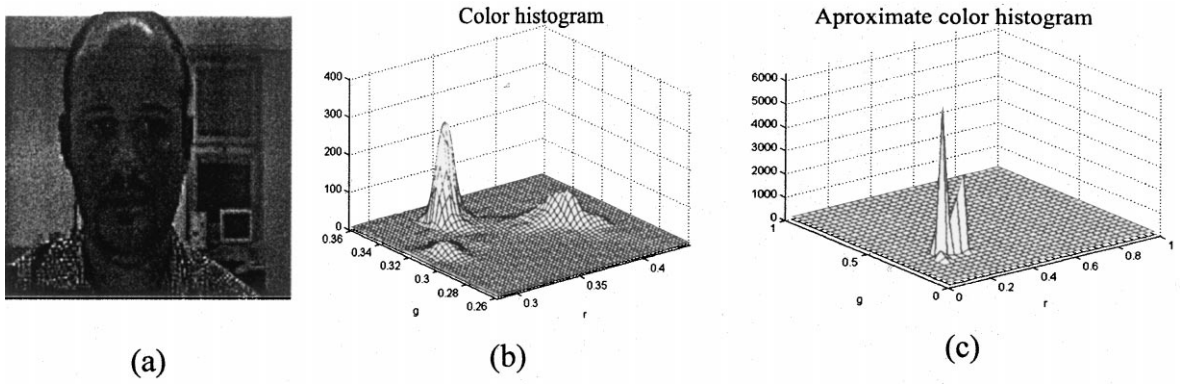

Fig. 12. Example of the approximate histogram obtained.

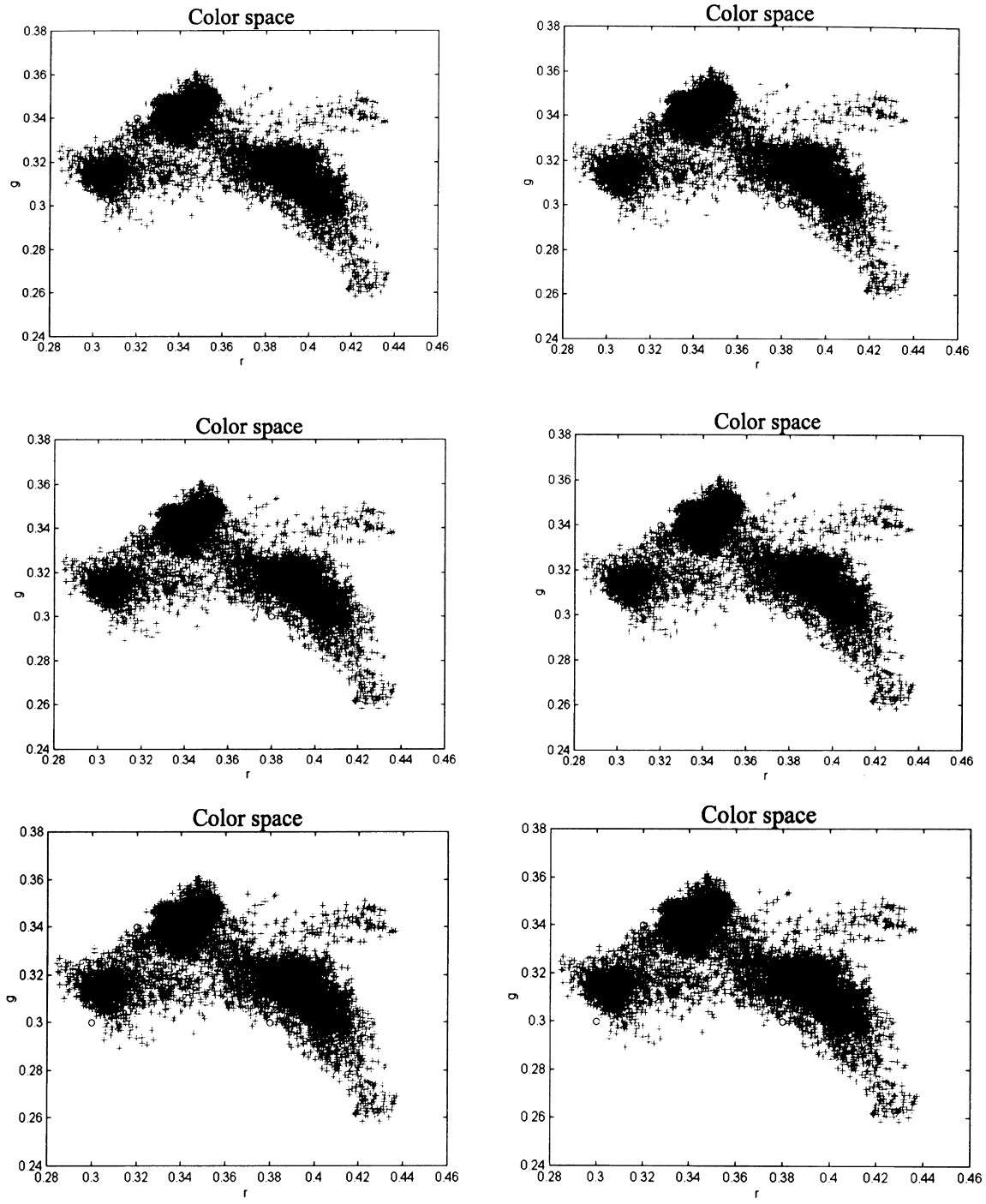

Fig. 13. Learning results for a number of neurons between 2 and 7, using 10% of samples.

The MML, MDL, GMM and UAGM methods give a correct result of four optimum classes, while the FHV and Evidence yield a wrong value of two. This experiment is not applicable to the outlined colors classification problem since it can not have two different distributions with the same mean.

3.4. Experiment 3 (Iris data)

Anderson's "Iris" data set is well known in analysis of classifiers. It consists of measurements from the plants' morphology. "Iris data sets" are formed by 50 samples of three classes of data: *Iris versicolor*, *Iris virginica* and *Iris*

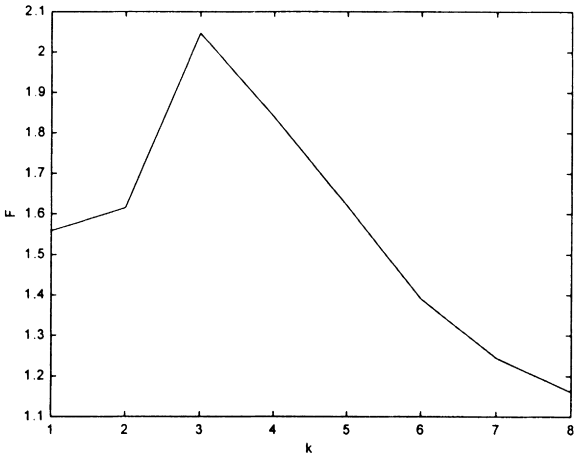

Fig. 14. Quality factor.

setosa. Each data is formed by four characteristics. Only characteristics (x_2, x_3) (Fig. 10) have been taken. The different methods have been applied, obtaining the results shown in Fig. 11.

As can be proven all methods provide a correct result of $K = 3$. Data classification has been accomplished using the UAGM method obtaining four errors in 150 data, implying a 97.5% of correct results. These results are similar to the ones exhibited by the GMM method (98% of hits), though a smaller calculation time is required.

3.5. Skin color segmentation

After analyzing the previous experiments, we can conclude that the results obtained using the UAGM algorithm are similar to those obtained with the MML, MDL and GMM methods. It is a much more simple method because the EM algorithm has been substituted by a competitive VQ learning method, solving the cluster validation problem by means of a cost function. Then it requires shorter calculation time, improving results provided by FHV and Evidence methods.

Next, it is presented a complete example of segmentation for a real image, using the UAGM algorithm. Fig. 12 depicts the color image to analyze, the histogram in the "rg" space and its approximate histogram. Vectors are initially located on the peaks of the histogram, to further perform the training phase through competitive learning.

The neurons positions are shown in Fig. 13, before training (circles) and after the training stage (crossings) for a number between 2 and 7 and a training set of 10% of the image pixels. As can be observed, the vectors are distributed on the greatest pixels density areas.

In Fig. 14 the classification quality factor for a number of classes between 2 and 7 is shown. As can be seen, the optimum number of classes is $K = 3$.

Fig. 15 illustrates the image pixels classification into the different classes, for the different evaluated configurations $(2 \leq K \leq 7)$. As can be noticed, face color is well distinguished from the rest for $K = 2$. Using $K = 3$, a new class appears for the shirt color. With $K = 4$, the background color is split into two classes, due to the existence of light and dark areas in the image. With $K = 5$, the skin color is split into two, one containing the reddest part (lips and face rednesss) and other for the rest. With $K = 6$, the shirt color is split into two and, finally, with $K = 7$ a new color appears for dark areas in the image, such as eyes and beard sides. The best approximation of the image main colors is obtained for $K = 3$: skin color, background and shirt.

The results obtained with this method are better than those presented in [28], since VQ learning algorithm, with optimal number of vectors, is performed instead of Self Organizing Maps (SOMs). SOMs are topology organizers in the sense that a number of neurons P is organized on a map, as a function of the input data topology. SOMs do not accomplish a clustering process. In fact, another algorithm is required to further perform the classification[27]. SOMs reduce the amount of information, but it still needs to be classified. Examples of this type are presented in Ref. [28], where a supervised classification is made, once the neurons are located in the Kohonen map. After that, pixels are classified employing the K-nearest neighbors technique. Other examples are given in Ref. [29,30] where a SOM is applied to organize data, and a multilayer perceptron performs the segmentation using supervised training.

In the UAGM method, neurons are located on the color space and their positions are the means of the Gaussian functions that model the histogram. Thus, the classification is accomplished in a direct way, without requiring any other method. On the other hand, the system is capable of calculating the number of functions that best models the histogram in an unsupervised way.

As can be seen in Fig. 15, the resulting clustering (for $K = 3$) is not perfect since colors in the image are present with a small number of pixels, that are not detected by the algorithm (the window). On the other hand, the mouth and part of the hair are also considered as skin color. Nevertheless, this method gives a good estimate of the image skin pixels improving the algorithm presented in Ref. [3], where the a priori skin class is off-line computed.

Fig. 16 depicts pixels clustering into the optimum number of calculated classes in the color space (a) and skin segmentation in the (x, y) space (b) for the example figure. The model is applied on the skin cluster so that all those pixels for which the function evaluation has a value greater than the threshold (Th) are segmented as skin.

Fig. 17 shows the pixels belonging to the skin class (red color) applying the clustering process and the model for different thresholds. In Fig. 18, the segmented

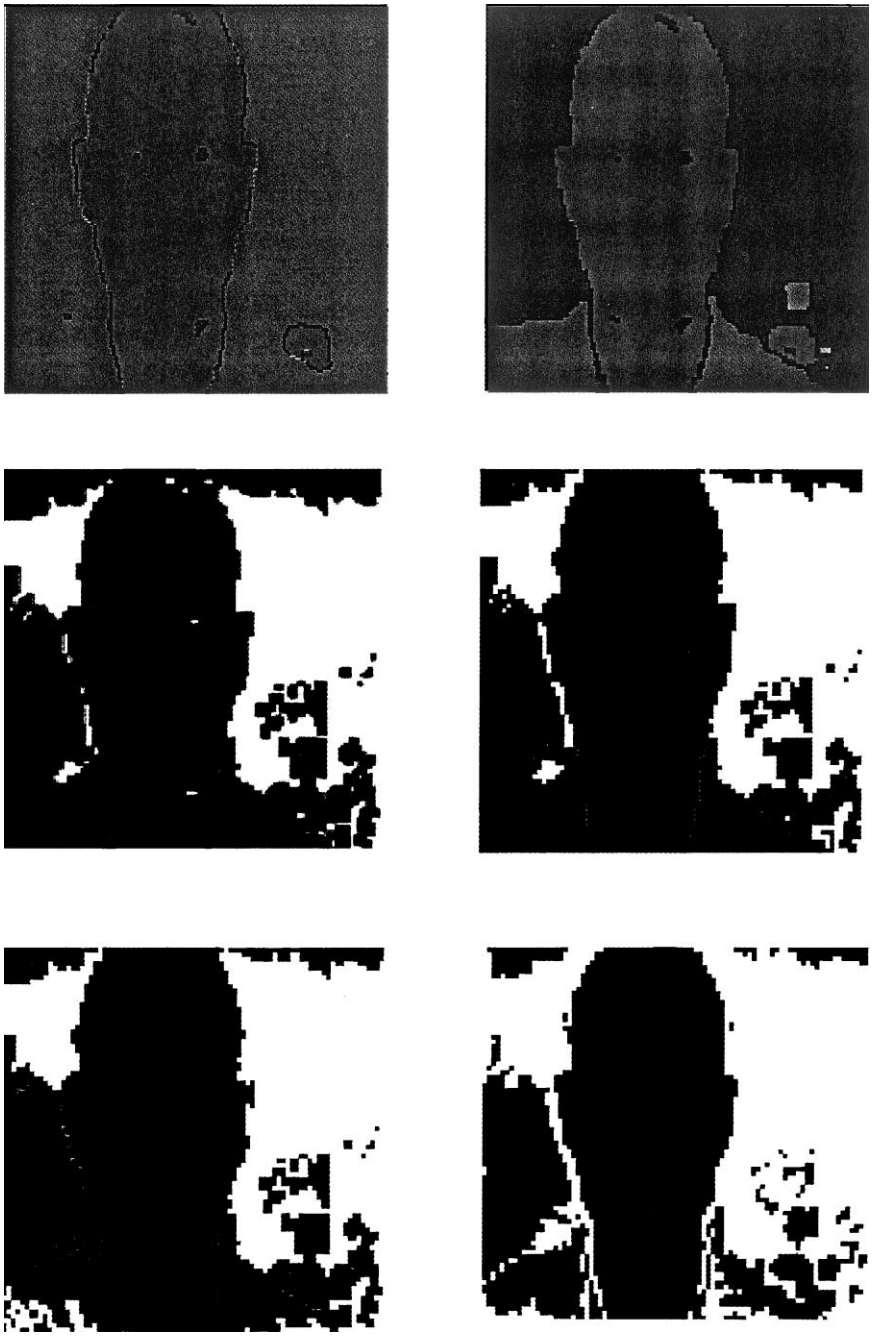

Fig. 15. Pixel clustering for the different evaluated classes.

images in the (x, y) space for different thresholds are depicted.

Once the skin is segmented for the first image of a sequence, the model is applied to the following ones, estimating their parameters and using an adaptive threshold. In Fig. 19, the evolution of mean and estimated covariance is shown for a sequence of 500 images, acquired without any kind of previous conditioning. Also, the evolution of the segmented area is shown as a function of time, where it is noted that the variation is very small, being its variance 1% of the total skin area.

4. Conclusions

We present a segmentation method of a person skin from any race, in real time, in an unsupervised and adaptive way, without introducing initial parameters.

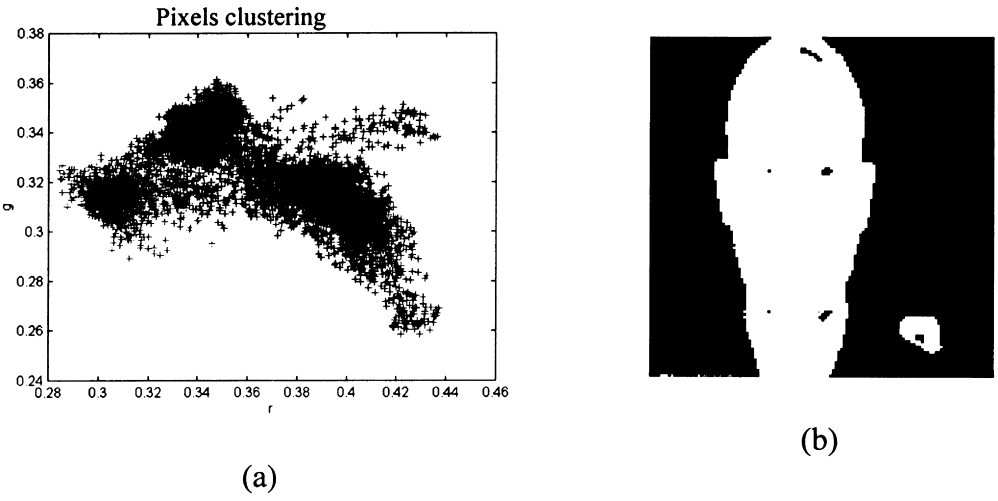

Fig. 16. (a) Pixels clustering in the rg space. (b) Skin cluster detection.

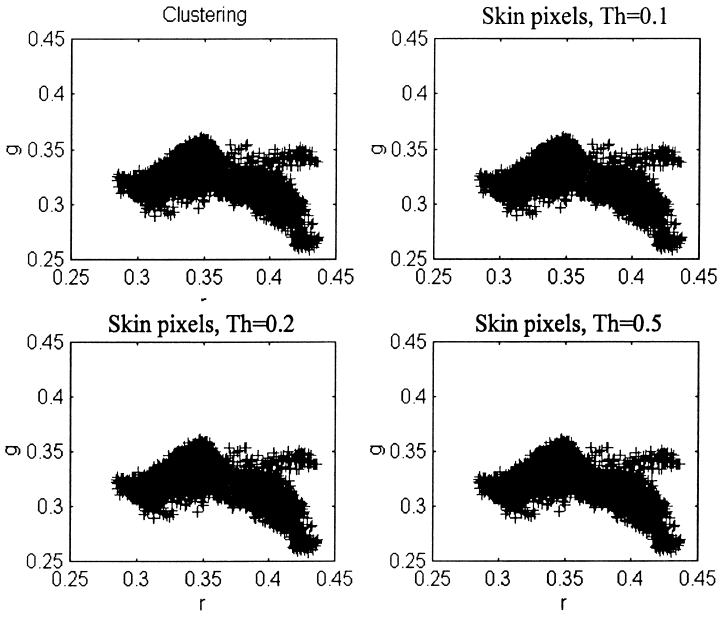

Fig. 17. Skin pixels for different thresholds.

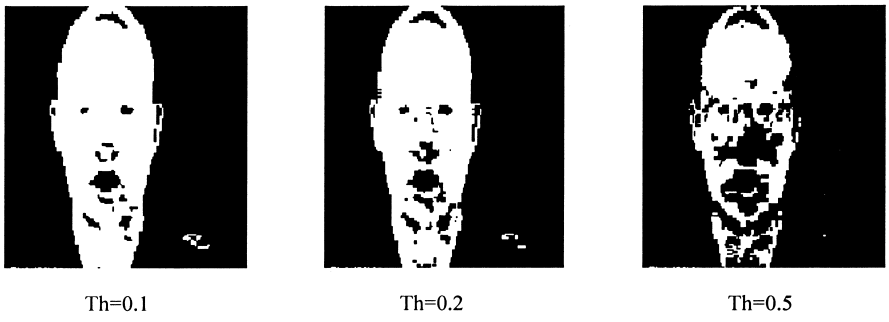

Fig. 18. Skin segmentation for different thresholds.

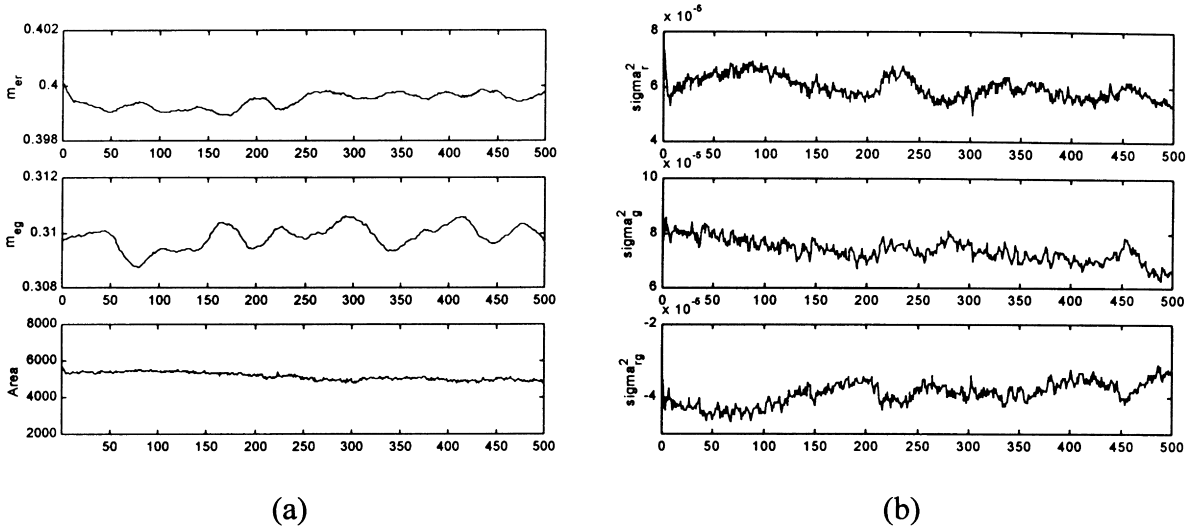

Fig. 19. Evolution of the model parameters as a function of time.

The system segments complex color images with random backgrounds correctly, and is robust to lighting and background changes. It has been demonstrated that, assuming certain hypothesis, the employed method is a particular case of the statistic general Bayesian approach, but without the convergence and complexity problems that arise in the same. The EM method has been substituted by a competitive VQ learning method, solving the cluster validation problem by means of a cost function, that is a slight modification of the generalized Fisher ratio. It has been demonstrated that the results obtained using this method, on a series of synthetic experiments, are at least as good as those exhibited by the EM algorithm and the model order selection techniques: FHV, Evidence density, MDL, MML and GMM. Several tests have been accomplished using real image sequences, obtaining optimum results. The algorithm has been tested with persons of different races (white, black and yellow), performing correctly in any case. The main drawback found in the method is that if a color in the background presents a chromaticity similar to the skin, it will also be segmented as skin. Anyway, if the object does not have any connection to the skin blob, the algorithm eliminates it and, therefore, it does not affect the estimated parameters. However, if some connection exists between the object and the skin blob, it will be considered as skin, introducing noise in the estimation process. On the other hand, minimum light conditions are required for correct operation. Above these minimum conditions the system is capable to adapt within a large range.

Acknowledgements

This work has been financed by the CICYT (Spanish Interministerial Science and Technology Commission) through the project TER96-1957-C03-01.

References

[1] De Silva, K. Aizawa, M. Hatori, Detection and tracking of facial features by using edge pixel counting and deformable circular template matching, IEICE Trans. Inf. & Syst. E78-D (9) (1995) 1195–1207.
[2] Kah-Kay Sun, Tomaso Poggio, Example-based learning for view-based human face detection. IEEE Trans. Pattern Anal and Machine Intelligence, 220 (1) (1998) pp 39–51.
[3] J. Yang, L. Wu, A. Waibel, Skin color modelling and adaptation. Technical Report CMU-CS-97-146, CS department, CMU, 1997.
[4] R. Meier, R. Stiefelhagen, J. Yang, A preprocessing of visual speech under real word conditions, Proceedings of European Tutorial & Research Work Shop on Audio-Visual Speech Processing, 1997 pp 123–129.
[5] H.A. Zelinsky, Robust real-time face tracking and gesture recognition, Proceedings of IJCAI'97, International Joint Conference on Artificial Intelligence, August 1997 Vol 2, pp 1525–1530.
[6] L.M. Bergasa, A. Gardel, M. Mazo, M.A. Sotelo, Face tracking using an adaptive skin color model, Third International ICSC Symposia on Intelligent industrial automation (IIA'99) and Soft Computing (SOCO'99), Genova, Italy, 1999, pp. 133–138.
[7] A.L. Yuille, P.W. Hallinan, D.S. Cohen, Feature extraction from faces using deformable templates, Int. J. Comput. Vision (1992) 99–111.
[8] B.D. Pomerleau, Non-intrusive gaze tracking using artificial neural networks, Advances in Neural Information Processing Systems, vol. 6, Morgan Kaufmann, Los Altos, CA, 1993.
[9] Y.R. Cipolla, Towards an automatic human face localization system, Department of Engineering, in: Proceedings of British Machine Vision Conference, vol. 2, Birmingham, October 1995, Springer, Berlin, pp 701–705.
[10] Y.A. Waibel, A real-time face tracker, Proccedings of WACV'96, Sarasota, Florida, USA. 1996 Technical Report CMU-CS-95-210, CS department, CMU, 1995.
[11] J.L. Crowley, J. Coutaz, Vision for Man Machine Interaction EHCI'95. Grand Targhee, August 1995 pp 341–346.
[12] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from

incomplete data via the EM algorithm, J. Royal Stat. Soc. 39 (1) (1997) 1–38.
[13] D.A. Lagan, J.W. Modestino, J. Zhang, Cluster validation for unsupervised stochastic model-bases image segmentation, IEEE Trans. Image Processing 7 (2) (1998) 180–195.
[14] S.J. Roberts, D. Husmeier, I. Rezek, W. Penny, Bayesian approaches to Gaussian mixture modeling, IEEE Trans. Image Processing 20 (11) (1998) 1133–1142.
[15] S.E. Umbaugh, Computer Vision and Image Processing, Prentice Hall, Englewood Cliffs, NJ, 1998.
[16] E. Littmann, H. Ritter, Adaptive color segmentation. A comparison of neural and statistical methods, IEEE Trans, Neural Networks, 8(1) (1997) pp 175–185.
[17] J.J. Oliver, R.A. Baxter, C.S. Wallace, Unsupervised learning using MML, Proceedings of the 13th International Conference on Machine Learning (ICML'96), San Francisco, 1996, pp. 364–372.
[18] T. Kohonen, Self Organizing Maps, Springer, Berlin, 1997.
[19] A. Gonzalez, M. Grana, A. D'Anjou, An analysis of the GLVQ algorithm, IEEE Trans. Neural Networks 6 (4) (1995) 1012–1016.
[20] N. Karayiannis, J. Bezdek, N. Pal, R. Hathaway, P. Pai, Repairs of GLVQ: a new family of competitive learning schemes, IEEE Trans. Neural Networks 7 (5) (1996) 1062–1071.
[21] P.M. Lee, Bayesian Statistic: An Introduccion, Arnold, Paris, 1994.
[22] T.W. Anderson, Asymptotically efficient estimation of covariance matrices with linear structure, Ann. Stat 1 (1) (1973) 135–141.
[23] I. Gath, B. Geva, Unsupervised optimal fuzzy clustering, IEEE Trans. Pattern Anal. Machine Intelligence 11 (7) (1989) 773–781.
[24] S.J. Roberts, Parametric and Non-Parametric Unsupervised Cluster Analysis, Pattern Recognition 30 (2) (1997) 261–272.
[25] J. Rissanen, Modeling by shortest data description, Automatica 14 (1978) 465–471.
[26] R.A. Baxter, J.J. Oliver, MDL and MML: simularities and differences, Technical Report TR 207, Department of Computer Science, Monash University, Clayton, Victoria 3168, Australia, 1994. Available on thewww from http://www.cs.monash.edu.au/~jono.
[27] J.J. Oliver, R.A. Baxter, Unsupervised learning using MML, Proceedings of the 13th International Conference on Machine Learning (ICML'96), San Francisco, 1996, pp. 364–372. Available on thewww from http://www.cs.monash.edu.au/~jono.
[28] J. Moreira, L. Da Fontoura Costa, Neural-based color image segmentation and classification using self-organizing maps, IX SIBGRAPI, October 1996 pp 47–54.
[29] N.W. Campbell, B.T. Thomas, T. Troscianko, Segmentation of natural images using self-organising feature maps, in: The British Machine Vision Conference, British Machine Vision Association, September 1996, pp. 223–232.
[30] N.W. Campbell, B.T. Thomas, Automatic selection of gabor filters for pixel classification, in: Sixth International Conference on Image Processing and its Applications, IEE, July 1997, pp. 761–765.