Nonlinear Color Space and Spatiotemporal MRF for Hierarchical Segmentation of Face Features in Video

Marc Liévin and Franck Luthon, *Member, IEEE*

***Abstract*—This paper deals with the low-level joint processing of color and motion for robust face analysis within a feature-based approach. To gain robustness and contrast under unsupervised viewing conditions, a nonlinear color transform relevant for hue segmentation is derived from a logarithmic model. A hierarchical segmentation scheme is based on Markov random field modeling, that combines hue and motion detection within a spatiotemporal neighborhood. Relevant face regions are segmented without parameter tuning. The accuracy of the label fields enables not only face detection and tracking but also geometrical measurements on facial feature edges, such as lips or eyes. Results are shown both on typical test sequences and on various sequences acquired from micro- or mobile cameras. The efficiency of the method makes it suitable for real-time applications aiming at audiovisual communication in unsupervised environments.**

***Index Terms*—Face analysis, hue, liptracking, logarithmic color space, Markov random field, motion detection, segmentation.**

I. Introduction

FACE ANALYSIS is an active research area nowadays due to the wide range of possible applications [1]: videophone, videoconferencing, special effects for movies, synthetic talking faces in human computer interface (HCI), synthetic clone-assistant for e-learning, face recognition and identification, communication for disabled people, MPEG compression, video indexing. Due to unpredictable environment conditions, image analysis techniques are not currently able to yield robust and accurate enough results for automatic face feature extraction. Viewing conditions encompass lighting variations (shadows), varying scale and pose, changes in speaker's face (skin color, eyeglasses, hair, beard, make-up), camera type (3-CCD, mono-CCD, mobile camera, webcam) and unknown background. Any robust application in face analysis has to take these real-world conditions into account. Combining color and motion information has become a standard approach to deal with such situations.

Based on those considerations, our feature-based approach is intended to be robust to real-world viewing conditions. For that purpose, two modeling tools are introduced: a nonlinear color transform computed by applying a logarithmic isomorphism on the *RGB* space, and a spatiotemporal Markov random field (MRF) model that integrates both hue information and temporal changes. A hierarchical algorithm with iterative labeling provides a robust labeling of each face feature.

The paper is organized as follows. After an overview of related works in Section II, Section III presents the logarithmic color transform. The spatiotemporal MRF prior model is detailed in Section IV. Section V describes the hierarchical segmentation algorithm that combines hue and motion detection for statistical clustering of face regions. Experimental results and dedicated postprocessing stages corresponding to real-world applications are shown in Section VI, namely for face detection and tracking, face feature segmentation, liptracking and eye contour extraction.

II. Related Works

In this section, we give a brief overview of current techniques for face analysis and we comment on their pros and cons, in particular their robustness with respect to viewing conditions and their ability to extract face features.

Image-based methods have proved to be relevant for face analysis [2]. Neural networks provide a convenient tool for picture classification or face detection in pictures [3], [4]. Generally applied on gray level images, neural approaches need a huge learning database for the training phase and are sensitive to lighting variations. Color processing actually reduces the sensitivity to lighting conditions [5].

Statistical analysis may be used to detect specific patterns in the face, i.e., features that are considered constant over time [6]. MRF modeling is also efficient for face detection [7]. Such approaches need to be combined with pattern recognition techniques to gain temporal stability and robustness to viewing conditions. Genetic algorithms propose a new approach able to detect lip, nose, eyes, and eyebrows from a face [8]. However, these methods require parameters that are dependent on viewing conditions.

Lipreading has been studied far longer before face analysis. Basic image processing modeling gives good results under constrained views [9]. Authors introduced preprocessing stages and dedicated tools (active contours, deformable templates, point distribution models). Nowadays, the last generation of such algorithms provides methods for face analysis [10].

As face synthesis improves in quality, 3-D face models have been used to locate and estimate real faces by 3-D/2-D registration [11], [12]. Unfortunately, the complexity of face analysis and the inverse mathematical problem often lead to suboptimal solutions without feature extraction for lipreading or expression analysis. Moreover, simulating real-world environment in synthetic scenes is still a challenging issue [13].

Manuscript received January 8, 2002; revised May 28, 2003. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jianying Hu.

M. Liévin is with the Surgical Systems Lab., Research Center Caesar, 53175 Bonn, Germany.

F. Luthon is with the Computer Science Research Laboratory, University of Pau and Adour Province, Bayonne, France.

Digital Object Identifier 10.1109/TIP.2003.818013

Low-level image properties provide also relevant information for face analysis. Motion information may be used to track faces or lips. Motion estimation techniques, like optical flow, are also used for face expression modeling and analysis [14], [15]. As a global method, optical flow cannot separate each face feature movement.

To gain independence from lighting conditions, color processing (instead of luminance) has become of standard use for face tracking in image sequences [16]. To take account of the specific face color distribution, two color-based approaches are common in the literature: the first one uses color angle methods for illumination invariant recognition. The second one estimates the skin distribution (or locus) in an appropriate color space [17]. Unfortunately, in noisy conditions (weak lighting, mono-CCD camera), angular transforms give poor results and color modes are often mixed together [18]. The *YCrCb* color space is widely used in face analysis for estimating the skin color locus [19]. Nonetheless, lighting compensation is still mandatory for robust face detection [20].

In conclusion, face analysis is often limited to ellipsoid or rectangular region detection [21]. This first localization is then used to run additional algorithms in constrained viewing conditions for face feature extraction, e.g., mouth and eye. Combining several approaches may lead to cumbersome frameworks and complex parameter tuning. Chromatic information is undoubtedly relevant for face analysis in video but standard transforms still remain sensitive to lighting conditions and very noisy in shadow areas, so that some authors even preclude the use of color [22], relying only on contours and motion. For automatic face feature segmentation further investigations in this field are therefore required, which is the goal of this paper. Moreover, we do believe that processing video sequences instead of static images is of precious help for face featrure extraction since it allows to integrate temporal information in addition to spatial information.

III. Nonlinear Color Transform

A. YCrCb and HSI Color Spaces

The *YCrCb* video format is a linear combination of red, green, and blue components (*RGB*) used as TV standard. The coefficients are standard-specific (cf. Rec. 601 and 709). For simplify and without loss of generality, we consider the following transform:

$$\begin{cases} Y = 0.3R + 0.6G + 0.1B \\ C_r = R - Y \\ C_b = B - Y \end{cases} \tag{1}$$

which may be written in vector form: $[Y, C_r, C_b]^t = T \cdot [R, G, B]^t$ where t denotes transposition and T is the matrix of chromatic coefficients.

Since color distributions based on linear combination of *RGB* often require a learning database or strong priors in order to be well estimated, angular transforms are sometimes preferred for color segmentation. The *HSI* color space (hue, saturation, intensity) is a typical nonlinear transform of the *RGB* space

$$\begin{aligned} I &= \frac{R+G+B}{3} \\ H &= \frac{k\pi}{2} - \arctan\frac{2R-G-B}{\sqrt{3}(G-B)} \quad \text{with} \\ k &= \begin{cases} 1 & \text{if } G > B \\ 3 & \text{otherwise} \end{cases} \\ S &= 1 - \frac{\min(R,G,B)}{I}. \end{aligned} \tag{2}$$

It has proven to be suited for face and lip processing since it is more related to psychovisual perception, and the red hue distribution actually little depend on the speaker's skin color [2], [1]. Yet, low-cost video systems for face analysis often use mono-CCD cameras that yield poor results with angular color transforms due to noisy conditions.

B. Original LUX Color Space

In order to be robust to lighting conditions, we work with a nonlinear color space based on a logarithmic transform. It is inspired both by biological considerations, i.e., cone distribution in the fovea and nonlinear transduction of cones followed by bipolar cell differencing in the retina (cf. [23, Fig. 6]) and by a mathematical framework, namely the logarithmic image processing (LIP) model, known to yield impressive contrast enhancement [24].

The LIP theory was developed for gray level images. The LIP model is basically defined in the continous case by three equations: a transform f from the intensity space (variable x) to the space of tones (variable y), an isomorphism ϕ (valid $\forall y \in]-\infty \ldots M[$) from the space of tones onto a logarithmic space (variable $\bar{x}$) and an inverse isomorphism ϕ^{-1} (trivial expression not given here, the interested reader is referred to [25] for details)

$$f : x \to y = f(x) = M\left(1 - \frac{x}{x_0}\right) \tag{3}$$

$$\phi : y \to \bar{x} = \phi(y) = -M \ln\left(1 - \frac{y}{M}\right) \tag{4}$$

where $x \in]0 \ldots x_0]$ is a continuous gray level, $x_0 \in]0 \ldots M]$ is the maximum transmitted light and M is the dynamic range of gray levels (typ. $M = 256$ for 8-bit coding).

Here, we extend the LIP model to handle colors (i.e., *YCrCb*) as well. We thus derive a new color space called *LUX* (for **L**ogarithmic h**U**e e**X**tension). For that purpose, only the composition function $\Phi = \phi \circ f$ is of pratical interest. The isomorphism Φ provides a logarithmic transform normalized by the maximum transmitted light x_0

$$\Phi : x \to \bar{x} = M \ln\frac{x_0}{x} \tag{5}$$

$$\Phi^{-1} : \bar{x} \to x = x_0 \exp -\frac{\bar{x}}{M}. \tag{6}$$

Since $(R, G, B) \in [0, M[\times[0, M[\times[0, M[$ in the discrete case, we take *rgb* (with $r = R + 1$ etc.) so that we stick to the interval $]0, M]$ as required by the LIP theory. Similarly, we will note *lux* (with $l = L + 1$, etc.) the transformed variables.

The diagram below helps understand how we build the *LUX* color space

$$\begin{tikzcd}
(R,G,B) \arrow[r, "(+1)"] \arrow[d] & (r,g,b) \arrow[r, "\Phi"] \arrow[d, "\Psi"] & (\bar{r},\bar{g},\bar{b}) \arrow[d, "T"] \\
(L,U,X) & (l,u,x) \arrow[l] & (\bar{l},\bar{u},\bar{x}) \arrow[l, "\Phi^{-1}"']
\end{tikzcd} . \quad (7)$$

The isomorphism Φ transforms the $[r,g,b]$ vector into its logarithmic counterpart $[\bar{r},\bar{g},\bar{b}]$ on which the linear matrix T is applied, yielding vector $[\bar{l},\bar{u},\bar{x}]$

$$\begin{cases} \bar{l} = 0.3\bar{r} + 0.6\bar{g} + 0.1\bar{b} \\ \bar{u} = \bar{r} - \bar{l} \\ \bar{x} = \bar{b} - \bar{l}. \end{cases} \quad (8)$$

Let $r_0g_0b_0$ be the maximal values of *rgb*. Then, an explicit formulation of (8) gives

$$\bar{l} = M \ln\left[\left(\frac{r_0}{r}\right)^{0.3}\left(\frac{g_0}{g}\right)^{0.6}\left(\frac{b_0}{b}\right)^{0.1}\right]. \quad (9)$$

Denoting $l_0u_0x_0$ the maximal values of *lux* and using (5), the nonlinear transform Ψ which is the composition of $\Phi^{-1} \circ T \circ \Phi$ is directly given by

$$\begin{aligned} l &= l_0 \frac{1}{r_0^{0.3}g_0^{0.6}b_0^{0.1}} r^{0.3}g^{0.6}b^{0.1} \\ u &= u_0 \left(\frac{g_0^{0.6}b_0^{0.1}}{r_0^{0.7}}\right) \frac{r^{0.7}}{g^{0.6}b^{0.1}} \\ x &= x_0 \left(\frac{g_0^{0.6}r_0^{0.3}}{b_0^{0.9}}\right) \frac{b^{0.9}}{g^{0.6}r^{0.3}}. \end{aligned} \quad (10)$$

For practical use, (10) contains too many unknowns. Therefore, we assume that each component r_0, g_0, b_0 is close to the maximal intensity I_0 (first order approximation). This hypothesis is valid when the camera is calibrated for full range on the white values. A second approximation is that the maximal luminance I_0 is close to the dynamic range M. This strong assumption corresponds to an automatic contrast correction. Moreover, we impose: $l_0 = u_0 = x_0 = M$ in order to keep the same dynamic range. Then (10) reduces to

$$\begin{aligned} l &= r^{0.3}g^{0.6}b^{0.1} \\ u &= Mr/l \\ x &= Mb/l \end{aligned} \quad (11)$$

So far, this logarithmic model works only for positive values in the range of $]0 \ldots M]$. To take account of the possibly negative values of the chromatic components in (8), we have to consider also the opposite formulae

$$\begin{aligned} \bar{u}_- &= \bar{l} - \bar{r} \\ \bar{x}_- &= \bar{l} - \bar{b}. \end{aligned} \quad (12)$$

Combining (8) and (12) while adapting the dynamic range yields the final expression of the *LUX* components

$$\begin{aligned} L &= (R+1)^{0.3}(G+1)^{0.6}(B+1)^{0.1} - 1 \\ U &= \begin{cases} \frac{M}{2}\left(\frac{R+1}{L+1}\right) & \text{if } R < L, \\ M - \frac{M}{2}\left(\frac{L+1}{R+1}\right) & \text{otherwise.} \end{cases} \\ X &= \begin{cases} \frac{M}{2}\left(\frac{B+1}{L+1}\right) & \text{if } B < L, \\ M - \frac{M}{2}\left(\frac{L+1}{B+1}\right) & \text{otherwise.} \end{cases} \end{aligned} \quad (13)$$

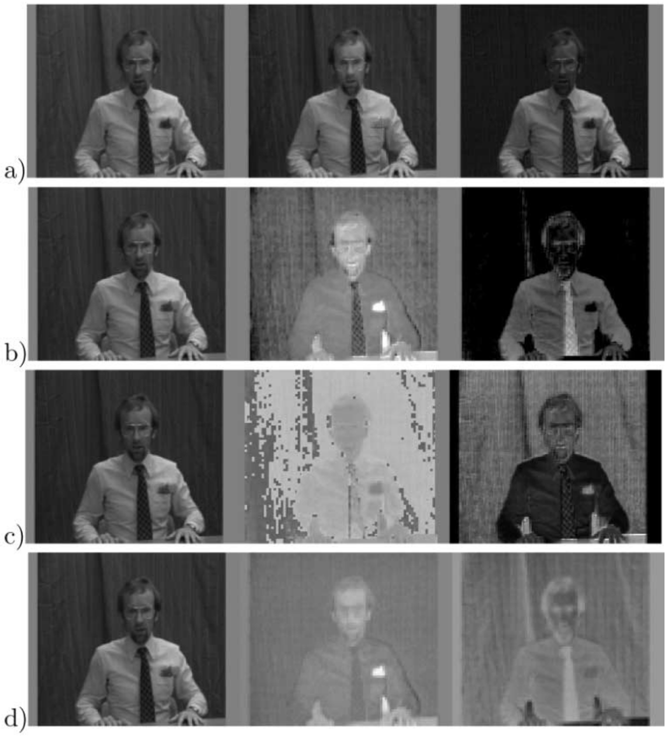

Fig. 1. Chromaticity results on Trevor sample frame: (a) *From left to right: GRB*; (b) *LUX*; (c) *IHS*; and (d) *YCrCb* (after histogram equalization on Cr and Cb to enhance the contrast).

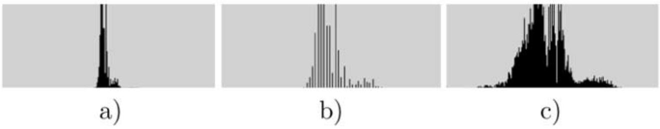

Fig. 2. Comparison of histograms on Trevor sample frame: (a) Cr without histogram equalization; (b) Cr after histogram equalization; and (c) U chroma computed with the proposed *LUX* transform.

Fig. 1 illustrates the robustness of the *LUX* transform in adverse background (brown curtain close to red locus) and the failure of the classical color transforms. As is well-known, the *RGB* components are strongly correlated [Fig. 1(a)]. The *LUX* space Fig. 1(b) reveals hue areas corresponding to face and hands, the background being clearly distinguishable. The hue computed from *HSI* is sensitive to light variations in the background [Fig. 1(c)]. One may observe that the saturation computed from *HSI* is little sensitive to light variations and provides good contrast for feature detection. However, saturation describes only the quantity of color but not its value. Therefore, it cannot be used alone for color segmentation (e.g., the tie and hands are not distinguished from the shirt). The *CrCb* components [Fig. 1(d)] provide less contrast than UX to distinguish between various color areas (face skin versus lips, shirt, and tie versus background). Although a histogram equalization was performed on Cr and Cb, the enhancement is only visual. In Fig. 2, a comparison of the histograms of red chrominances confirms the improvement obtained with U instead of Cr.

C. Simplified Formulae for Skin Locus Detection

This paragraph proposes a simplification of the *LUX* transform specifically suited for skin detection in the context of real-time implementation. Indeed the computation cost,

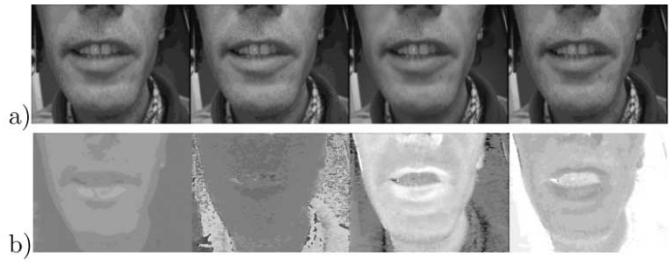

Fig. 3. Typical frame of speaker Benny sequence: (a) *From left to right:* the luminances Y, I, L, and G. (b) The chrominances Cr, H, U, and $\hat{U}$.

and hence the number of elementary operations, has to be minimized.

Since the hue of the face skin is mainly red (i.e., $R > L \Leftrightarrow \bar{r} < \bar{l}$), we may take only the contribution of $\bar{u}_-$ and define the red chroma as

$$U = \begin{cases} M\frac{L+1}{R+1} & \text{if } R > L, \\ M-1 & \text{otherwise.} \end{cases} \tag{14}$$

Taking a step further, the luminance formula L in the *LUX* space is simply a weighted geometrical mean of *RGB* (13). In practice, whatever the image, L gives very little difference compared to Y (or even G component, see Fig. 3(a)). Therefore, the following simplified formula might by used in place of (14):

$$\hat{U} = \begin{cases} 256 \times \frac{G}{R} & \text{if } R > G, \\ 255 & \text{otherwise.} \end{cases} \tag{15}$$

Already used empirically for skin detection [26], the ratio G/R is derived here from a simplification of the *LUX* transform, which provides some kind of theoretical background. This ratio is scaled by a multiplicative constant in order to adjust its range to the 8-bit quantization levels ($M = 256$). The conditional test in (15) accounts for the sign of the color difference.

Fig. 3 shows the results of the various transforms applied to a typical color image representing the lower part of a speaker's face: the C_r component is not enough discriminating for segmenting lips; the hue H yields a noisy map, whereas the logarithmic chroma U clearly exhibits the lip shape and other facial parts (tooth, nostrils). The logarithmic transform gains in contrast while being insensitive to illumination variations, thanks to its homomorphic nature. As expected, the simplified chroma $\hat{U}$ (depicted with inverted gray levels compared to U since it corresponds to the opposite formula $\bar{u}_-$) is very close to U which means that the proposed simplification is relevant when aiming at face and lip segmentation.

IV. Spatiotemporal MRF Prior Model

In order to build a probability map of face and facial feature presence at each time t, we propose to combine color estimates with motion observations (temporal changes in the video). Each pixel s will be attributed a label $l_s = (\kappa_s, \mu_s)$ that jointly reflects its hue class and its motion class

$$\begin{aligned} &\mu_s = 1 \quad \text{for } s \in \text{mobile class} \quad (0 \text{ if static}) \\ &\kappa_s = n \\ &\qquad \text{for } s \in \text{hue class number } n \quad (0 \text{ if unclassified}) \end{aligned} \tag{16}$$

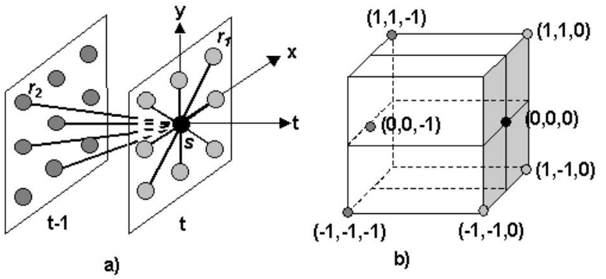

Fig. 4. (a) Spatiotemporal neighborhood $\eta(s)$: in black, the current pixel s; in light gray, the spatial neighbors r; in dark gray, the temporal neighbors r. (b) Euclidian structure: metric cube with coordinates of vectors $\overrightarrow{sr}$.

where μ is a Boolean variable indicating the motion class (mobile or static). κ is an integer indicating the hue class ($\kappa = 1 \ldots n \ldots N$). Note that we take $\kappa = 0$ for all the unclassified pixels (still belonging to the background).

A prior model should properly define the interactions between the labeled pixels. MRFs are well suited for that purpose. Let us consider the *spatiotemporal* neighborhood structure η shown in Fig. 4(a). The label field is supposed to verify the main MRF property related to that neighborhood, namely the label l_s of the current pixel s depends only on the labels l_r of its neighbors $r \in \eta(s)$.

Given this neighborhood structure and considering only binary cliques of first order, the prior energy is classically expressed as a sum of potential functions V that model spatiotemporal interactions within the neighborhood $\eta(s)$ of pixel s

$$e_p(s) = \sum_{r \in \eta(s)} V(l_r, l_s). \tag{17}$$

Let $(\delta_x, \delta_y, \delta_t)$ be the coordinates of vector $\overrightarrow{sr}$ associated to the clique (s, r): $\delta \in \{-1; 0; 1\}$ Fig. 4(b). In [27], the potential functions were defined according to the basic idea that interaction gets weaker as the neighbor is farther. But instead of simply taking the inverse of the Euclidian distance between two neighbors s and r (norm of the vector), the coordinates of vector $\overrightarrow{sr}$ were divided by three potential parameters β_x, β_y for (x, y)-axis and β_t for temporal axis

$$V(l_r, l_s) = \frac{1}{\sqrt{\left(\frac{\delta_x}{\beta_x}\right)^2 + \left(\frac{\delta_y}{\beta_y}\right)^2 + \left(\frac{\delta_t}{\beta_t}\right)^2}}. \tag{18}$$

A spatial anisotropy: $\beta_x = 2 \cdot \beta_y$ was imposed to emphasize on horizontal configurations, which are of more importance for liptracking. Then, (18) reduces to

$$V(l_r, l_s) = \frac{\beta_x \beta_t}{\sqrt{\beta_t^2 \left(\delta_x^2 + 4\delta_y^2\right) + \beta_x^2 \delta_t^2}} \tag{19}$$

where spatial parameter $\beta_x > 0$ and temporal parameter $\beta_t > 0$ control spatial (resp. temporal) homogeneity.

Here, we modify the model in [27] by taking a neighborhood that is temporally causal (for real-time implementation purpose), and we extend the model to handle N different hue classes (instead of only 2). A unique potential function encompasses all configurations. The two elementary potentials $\beta_x(l_r, l_s)$ and $\beta_t(l_r, l_s)$ are no longer constant but as defined

TABLE I
ELEMENTARY POTENTIALS β_t AND β_x ($\times$ = DON'T MATTER). CROSSING BETWEEN ROWS AND COLUMNS EXPRESSES A LOGICAL AND BETWEEN CONDITIONS

$\beta_t(l_r, l_s)$	$\kappa_r = \kappa_s$	$\kappa_r \neq \kappa_s$
$\mu_s = 0,\ \mu_r = \times$	1	2
$\mu_s = 1,\ \mu_r = \times$	2	1
$\beta_x(l_r, l_s)$	$\kappa_r = \kappa_s$	$\kappa_r \neq \kappa_s$
$\mu_s = \mu_r$	1	2
$\mu_s \neq \mu_r$	2	2

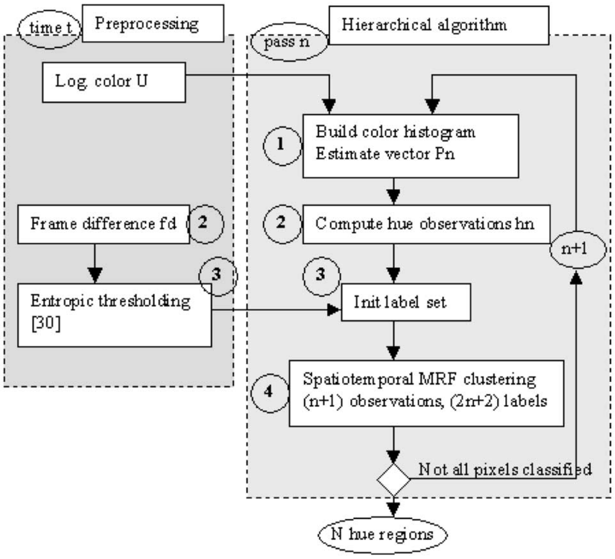

Fig. 5. Hierarchical segmentation framework.

in Table I. These potentials constrain the model respectively to spatial homogeneity of labels in frame t and temporal homogeneity of hue when no motion is detected between $t-1$ and t.

V. HIERARCHICAL FACE SEGMENTATION

A coarse-to-fine approach is appropriate for face feature extraction since the head is composed of a main area: the face skin, and secondary areas representing each feature: upper lip, lower lip, left eye, right eye, nose, eyebrows. Moreover, the hue distribution estimation gains in accuracy when estimated hierarchically on smaller areas. We describe here a hierarchical iterated algorithm for segmenting N regions (e.g., N face features) where N is unknown. To detect face regions, motion information is combined with red hue.

In most of face analysis algorithms, face parameters are determined manually beforehand or by a learning stage. Here, the parameter vectors $P_n, (n = 1 \ldots N)$ corresponding to N hue clusters are automatically estimated. At each iteration (or pass), a new label corresponding to a new hue cluster is added during the segmentation process. One pass (iteration n) of the algorithm consists in the four sequential steps shown in Fig. 5.

A. Step1: Mode Estimation on Hue Histogram

A hue cluster is defined by vector $P_n = \{U_n, \Delta_n\}$, where U_n and Δ_n are the mean value and standard deviation of the chrominance distribution for all the pixels that are still unclassified at pass n. An algorithm similar to the one proposed in [28] is used for finding the main mode P_n

1) build a smooth histogram from the chroma distribution with a Gaussian kernel;
2) compute the first derivative and count the zero-crossings;
3) find the greatest mode U_n corresponding to a zero-crossing;
4) compute the second derivate to estimate Δ_n.

B. Step2: Computation of Observations

From the simplified color space defined in Section III-C, two observations are derived, taking values in the same range $[0 \ldots 255]$ as the 8-bit quantized image.

- The hue observation $h_n(s)$ corresponding to pass n is computed by filtering the chrominance $U(s)$ at pixel s with a parabola[1] while discarding the influence of outliers:[2]

$$h_n(s) = \left[256 - \left(\frac{U(s) - U_n}{\Delta_n}\right)^2\right] \times 1_{|U(s) - U_n| \leq 16\Delta_n}. \tag{20}$$

- The temporal observation is the frame difference $fd(s)$ computed on the luminance (either L, Y, I or even G as explained in III-C)

$$fd(s) = |Y_t(s) - Y_{t-1}(s)|. \tag{21}$$

C. Step 3: Initialization of the Label Set

It results from (16) that, during the segmentation process of the image at pass n, the label set is made of $2 \times (n+1)$ distinct labels that code hue and motion.

Initial label field is computed by binarizing the $n+1$ observations o, where o denotes either $fd, h_1, \ldots,$ or h_n. Two thresholds are used for that purpose.

- A threshold γ is associated to the hue distribution. We compute the ratio $\rho = |U(s) - U_n|/\Delta_n$ within a trust margin of 50%. From (20), this is reached when $\rho = \sqrt{256}/2 = 8$. The thresholded hue observation is then given by $h_n(s) > \gamma$ where $\gamma = 256 - \rho^2 = 192$.
- A threshold θ is used to suppress the camera noise without cutting significant temporal changes. This threshold may be tuned manually or estimated on-line using the entropy power of the frame difference as proposed in [30]. The binarized motion observation is then given by $fd(s) > \theta$. As can be seen from Fig. 6, motion detection will help and contribute in extracting moving feature edges.

[1] The parabola associated to cluster P_n is centered on U_n with a standard deviation Δ_n. This type of weighting function was already proposed by Coianiz to emphasize the desired red hue [29].

[2] The notation $1_{\text{condition}}$ denotes a binary function which takes the value 1 if the condition is true, 0 otherwise.

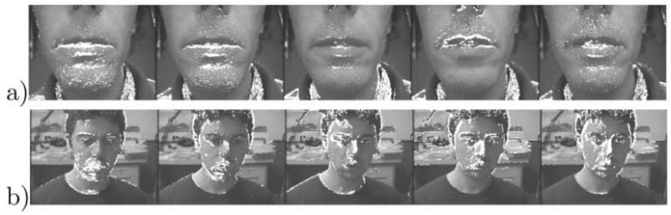

Fig. 6. Temporal changes detected by entropic thresholding: (a) micro-camera centered on the lower part of the face and (b) webcam sequence in office room.

D. Step4: Statistical Relaxation

Since those initial label fields are non homogeneous and noisy, a statistical relaxation is needed to segment accurately the face. The MRF framework with the prior model given in Section IV is adopted for that purpose.

Maximizing the a posteriori probability (MAP criterion) of the label field given the observations is equivalent to minimizing a global energy function over the image grid S at time t [31]

$$E(S,t) = \sum_{s \in S} \left[\lambda . e_p(s) + e_{fd}(s) + \sum_{i=1}^{n} e_{h_i}(s) \right] \tag{22}$$

where e_p is the *prior energy* (corresponding to spatial and temporal *a priori* constraints) as defined by (17), and e_o (with o denoting any observation, i.e., $fd, h_1, \ldots, h_n$) represent the various *likelihood energies* (expressing the link between labels and the $n+1$ observations) of pixel $s \in S$. λ is a weighting coefficient for balancing the influence of the two terms of energy (typ. $\lambda = 20$).

The energies e_o are classically defined as [32]

$$e_o(s) = \frac{[o(s) - \psi_o(s)]^2}{2\sigma_o^2} \tag{23}$$

where ψ_o are data-link functions that depend on the (possibly local) estimates of the average value of observations. They are simply defined as two-valued functions

$$\psi_{fd}(s) = \begin{cases} [fd]_\theta & \text{if } \mu_s = 1, \\ 0 & \text{if } \mu_s = 0. \end{cases} \tag{24}$$

$$\psi_{h_n}(s) = \begin{cases} [h_n]_\gamma & \text{if } \kappa_s = n, \\ 0 & \text{if } \kappa_s = i \neq n \end{cases} \tag{25}$$

where $[o]_{\theta_o}$ represents the average value of observations above the threshold

$$[o]_{\theta_o} = \frac{\sum_{s \in S_o} o(s)}{\mathrm{Card}(S_o)} \quad \text{where } S_o = \{s \in S, o(s) > \theta_o\}. \tag{26}$$

A variance σ_o^2 is associated to each observation o. Both $[o]_{\theta_o}$ and σ_o^2 are estimated on-line. The iterative deterministic algorithm ICM (Iterated Conditional Modes) is implemented to compute the minimum energy at each site, starting from the binarized observations as initial label configuration. ICM is chosen because of its low computation cost but it may converge toward local minima. In our tests however, a stable minimum was always reached in practice after a few iterations on the field (less than 10). The relative variation of the global energy is used as stopping criterion: $\Delta E(S,t)/E(S,t) < \epsilon$ (typ. $\epsilon = 0.05\%$). One obtains homogeneous label fields. Fig. 7 shows final hue labels when the algorithm is applied to a moving face.

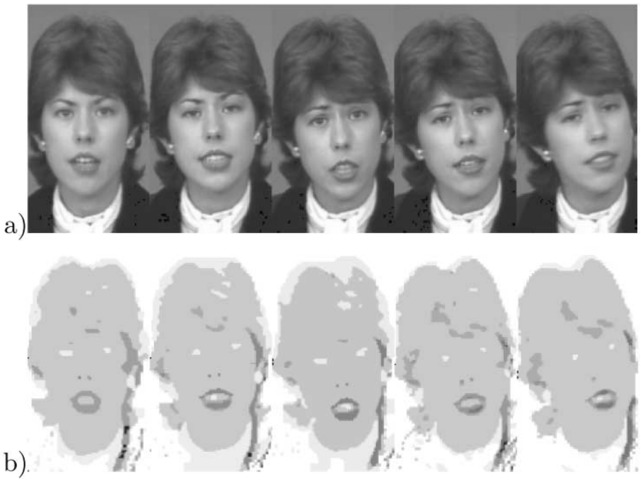

Fig. 7. Face Segmentation: (a) sample frames of color video sequence Claire and (b) label fields after relaxation, with gray levels coding the hue clusters.

E. Contour Postprocessing

1) Region of Interest Extraction: Localization of region of interest (ROI) is a prerequisite toward feature analysis and anthropometrical measurements. Any suboptimal (ad hoc) approach may be applied to estimate the ROI from the good segmentation results of the MRF labeling (binary masks). In this paper, results are shown with rectangular shapes. Each vertex location is computed automatically by using a gradient descent method applied on the ratio between the number of labeled points and the area of the region.

2) Unsupervised Active Contour: Active contours were primarily designed for interactive segmentation where the user guides, by external forces, the contour close to the desired solution [33]. Active contours are usually applied on the image intensity gradient in order to extract edge points on face features. However active contours are known to be complex to tune and sensitive to initialization. In this paper, active contours are run on the segmented label fields instead of being applied on intensity images as usual. This technique avoids manual parameter tuning and bad convergence: working on binary masks is a convenient way to define a stable external energy, always confined in the same range. In that case, the active contour parameters only need to be evaluated once beforehand and stay adequate to future segmented fields. No user initialization is required thanks to the rectangular ROI. Such active contours applied on MRF label fields are therefore reliable in face analysis applications.

VI. Experimental Results

This section presents specific implementations of the hierarchical segmentation scheme for face, lip and eye tracking, respectively.

Each application depends on a specific choice of the ROI and of the number N of segmentation passes. The scheme below details the application dependent implementation stages. The

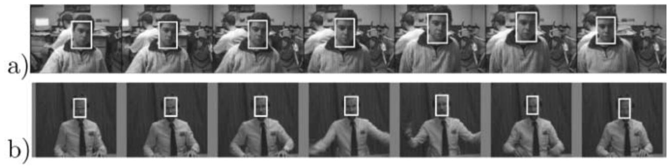

Fig. 8. Face tracking with rectangular shape ROI (the ROI detected is marked in white): (a) office sequence with mobile cam tracking the ROI. (b) Trevor sequence.

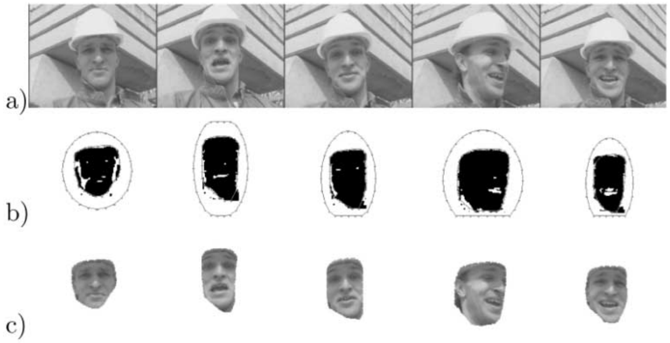

Fig. 9. Face tracking with arbitrary face orientation: (a) sample frames of the Foreman video sequence. (b) Masks obtained after 1-pass segmentation and initial positionning of active contours. (c) Face extraction with final active contours.

final active contour stage may provide geometrical parameters (width, height) of the facial features.

1) Set an initial ROI (automatic or manual).
2) Cluster this region up to pass N.
3) Compute the ROI associated to a selected hue (i.e., feature).
4) Apply active contours on binary masks.

A. Face Tracking

Face tracking is the first task in face analysis. A 1-pass segmentation is considered $(N = 1)$. This comes out to estimate only the preponderant red hue mode in the color image. Therefore, while looking for the head, any other object or surface (like the hands) should represent a smaller red hue area.

For face tracking, the initial ROI is simply the outer border of the first image. The 1-pass algorithm segments the face and extracts the border of the binary masks. The temporal tracking is obtained by taking the final ROI at time t to initiate the next one at time $t + 1$. Fig. 8 shows the robustness of the proposed face tracking algorithm in complex environment (busy office with two persons) or when the face represents a small area in front of a brown (i.e., close to red locus) background.

Note that the same method simply fails when combined with *HSI* or *YCrCb* transforms (cf. Fig. 1(c)–(d)).

Fig. 9 shows binary masks after 1-pass segmentation and final results of active contours on test sequence Foreman. In this well-known sequence, the camera moves while the speaker's head changes in orientation. Though this sequence presents numerous motion artefacts and moving shadows, the algorithm tracks the head with only 4 false detections among 300 images. The four misdetections correspond to images with superimposition of the hand over the head. Indeed, no distinction is actually made in the modeling between a talking head and a moving hand.

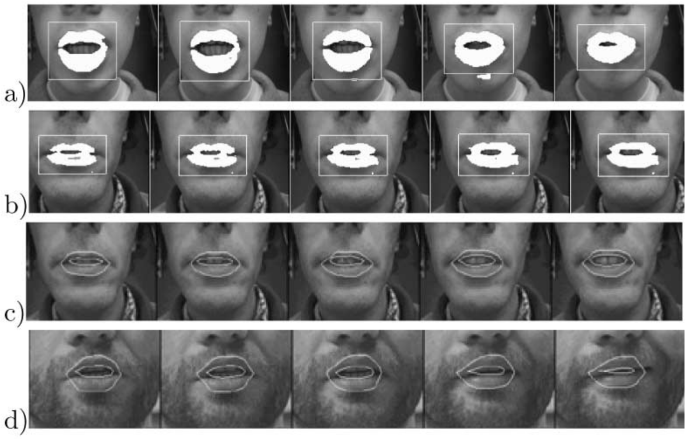

Fig. 10. (a) and (b) Two lip sequences with detected ROI and lip masks superimposed in white. (c) and (d) Two lip sequences with final active contours superimposed in white.

B. Lip Tracking

Lip contour extraction is achieved by applying the proposed scheme with a 2-pass segmentation $(N = 2)$. The first pass corresponds to the face tracking implementation. The second pass segments the lip hue mode. The search for the second mode is constrained by the condition: $U_2 < U_1$, where U_1 and U_2 are the mean values of red chroma for face and lips respectively. This condition drives the histogram estimate toward the appropriate mode, whenever the background is non uniform or noisy. In undefined viewing conditions, where these modes are generally mixed, the hierarchical segmentation succeeds in labeling the lip areas from the face. Fig. 10(a)–(b) shows lip masks obtained on two sequences acquired with a micro-camera.

With an approach similar to face tracking, two active contours are initiated and computed on the binary masks. One contour corresponds to the outer border of the mouth whereas the other contour extracts the inner border of the mouth. Lip corners need to be fixed during active contour convergence. Their position is located at the middle of the estimated lip ROI. Fig. 10(c)–(d) shows two sequences with superimposed final active contours. Fig. 10(c) corresponds to a common case in videophone application. An ambient light induces shadows on lip boundaries while the speaker is talking. Fig. 10(d) shows the robustness of the processing even when the speaker wears a beard.

C. Eye Tracking

Eyes are an essential feature of face expression. In the rest of this paragraph, only the right eye tracking is detailed without loss of generality. Considering the ROI obtained from the previous face tracking process, this area is partitionned into smaller bounding boxes corresponding to the various face feature locations. For instance, the right eye is located a priori at the upper left side of the image. A more accurate solution may consist in using anthropometric parameters taken from a speaker database built in a previous learning phase. Here, the sub-area is predefined as the upper left quadrant of the image. This assumption is sufficient as an initial guess for the ROI. The segmentation scheme is then run with two passes $(N = 2)$, followed by an active contour stage. The approach is the same as for lip contour

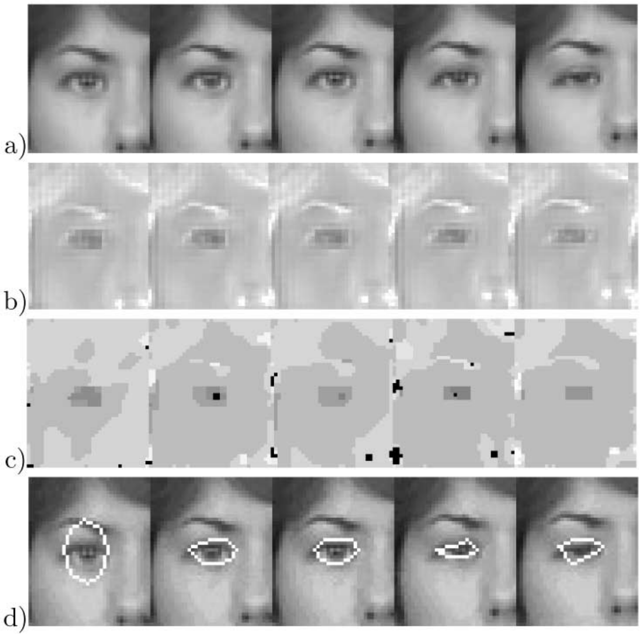

Fig. 11. Four stages of the right eye contour extraction. (a) Bounding box positioning; (b) *LUX* transform; (c) label field after hierarchical segmentation; (d) eye contour extraction by active contours (initial position is shown on the first image).

extraction: one active contour is initialized on the outer border with two centered corners, corresponding to the eye corners.

Following is an overview of the eye tracking stages:

1) face tracking;
2) locating anthropomorphic areas: mouth, left eye, right eye;
3) computing chroma U from the *LUX* transform;
4) segmenting the eye area;
5) extracting the eye contour.

Fig. 11 presents the four last stages when applied to Claire sequence. Even after zooming on the eye area, the algorithm tracks correctly the outer border of the eye.

VII. Discussion

While real-time, accuracy and robustness are still bottlenecks for user-friendly multimedia applications, the low-level tools presented here are an attempt to address these key issues in order to improve forthcoming facial communication platforms.

The three cues of our pixel-based approach for robust face feature segmentation are the following.

- A new nonlinear color transform is derived from the association of LIP model with *YCrCb* space. Note that the *LUX* color transform may also be applied to other multimedia applications, like video coding, video indexing or scene description. We are currently investigating its use not only for color segmentation but also for compression within the JPEG2000 standard. Our first results (to be published) confirm that it yields visual improvement compared to *YCrCb* when dealing with very high compression ratios (typ. 1:100).
- A hierarchical segmentation algorithm allows robust tracking of face and facial features under unsupervised conditions, thanks to the robustness of the spatiotemporal MRF model that jointly handles color and motion.
- Active contours applied on label fields (instead of images) make it possible to extract accurately and automatically all feature edges.

Whereas simplifications were done for real-time implementation, this approach proved to be efficient not only for classical test sequences like Trevor, Foreman or Claire, but also for sequences acquired with various cameras (micro-cam, mobile cam, desktop webcam). Moreover, the quality of results is actually little affected by MPEG compression since the proposed logarithmic space is derived from the $YCrCb$ coding. Face tracking was implemented at a rate of 12 full-color video fps on a 1.4 GHz processor, the complete face feature segmentation requiring less than 1 second per frame. The whole algorithm may run in real-time at 30 fps on a specialized DSP board.

The proposed method works automatically for front-view face sequences with complex background. Its capacity in extracting object contours enables its integration in MPEG7 bit-stream description. Its robustness to lighting variations makes it also suitable for outdoor applications. Of course, one limitation is when dealing with profile images: in that case, the method should be partially supervised (for instance to set the initial ROI for tracking the visible eye).

A forthcoming development of such a framework is the synthesis of 3-D realistic animated faces, fed with geometrical parameters measured on the face features extracted by the algorithm.

References

[1] M. H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, pp. 34–58, Jan. 2002.

[2] E. Hjelmås and B. K. Low, "Face detection: A survey," *Comput. Vis. Image Understand.*, vol. 83, no. 3, pp. 236–274, Sept. 2001.

[3] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 23–38, Jan. 1998.

[4] S. Ben-Yacoub, B. Fasel, and J. Luettin, "Fast face detection using MLP and FFT," in *Proc. Int. Conf. on Audio and Video-Based Biometric Person Authentication*, Washington, DC, 1999.

[5] S. J. McKenna, S. Gong, and Y. Raja, "Modeling facial color and identity with Gaussian mixtures," *Pattern Recognit.*, vol. 12, pp. 1883–1892, Dec. 1998.

[6] A. Colmenarez *et al.*, "Detection and tracking of faces and facial features," in *Proc. IEEE Int. Conf. Image Processing*, Kobe, Japan, Oct. 1999.

[7] M. S. Lew and N. Huijsmans, "Information theory and face detection," in *Proc. Int. Conf. Pattern Recognition*, Vienna, Austria, Aug. 1996, pp. 601–605.

[8] C. H. Lin and J. L. Wu, "Automatic facial feature extraction by genetic algorithm," *IEEE Trans. Image Processing*, vol. 8, pp. 834–845, June 1999.

[9] A. L. Yuille, P. W. Hallinan, and D. S. Cohen, "Feature extraction from faces using deformable templates," *Int. J. Comput. Vis.*, vol. 2, no. 8, pp. 99–111, 1992.

[10] T. Chen, "Audiovisual speech processing. Lip reading and lip synchronization," *IEEE Signal Processing Mag.*, vol. 18, pp. 9–21, Jan. 2001.

[11] D. DeCarlo and D. Metaxas, "Combining information using hard constraints," in *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Ft. Collins, CO, 1999, pp. 132–138.

[12] J. Ström, T. Jebara, S. Basu, and A. Pentland, "Real time tracking and modeling of faces: An EKF-based analysis by synthesis approach," in *Proc. IEEE Int. Conf. on Computer Vision*, Zurich, Switzerland, 1999.

[13] R. Ishiyama and S. Sakamoto, “Geodesic illumination basis: Compensating for illumination variations in any pose for face recognition,” in *Proc. Int. Conf. on Pattern Recognition*, Aug. 2002.

[14] I. A. Essa and A. P. Pentland, “Coding, analysis, interpretation, and recognition of facial expressions,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 7, pp. 757–763, 1997.

[15] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J. G. Taylor, “Emotion recognition in human-computer interaction,” *IEEE Signal Processing Mag.*, vol. 18, pp. 32–80, Jan. 2001.

[16] B. Menser and M. Wien, “Segmentation and tracking of facial regions in color image sequences,” *Proc. SPIE*, pp. 731–740, June 2000.

[17] D. Chai and K. N. Ngan, “Face segmentation using skin-color map in videophone applications,” *IEEE Trans. Circuits, Syst., Video Technol.*, vol. 9, pp. 551–564, June 1999.

[18] B. Martinkauppi, M. Soriano, and M. Laaksonen, “Behavior of skin color under varying illumination seen by different cameras in different color spaces,” *Proc. SPIE*, vol. 4301, pp. 102–103, Jan. 2001.

[19] C. Garcia and G. Tziritas, “Face detection using quantized skin color regions merging and wavelet packet analysis,” *IEEE Trans. Multimedia*, vol. 1, pp. 264–277, Sept. 1999.

[20] R.-L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, “Face detection in color images,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, pp. 696–706, May 2002.

[21] F. J. Huang and T. Chen, “Tracking of multiple faces for human-computer interfaces and virtual environnements,” in *Proc. IEEE Int. Conf. on Multimedia and Expo*, New York, July 2000.

[22] C. Wang and M. S. Brandstein, “A hybrid real-time face tracking system,” in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 1998.

[23] S. Shah and M. D. Levine, “Visual information processing in primate cone pathways—Part I: A model,” *IEEE Trans. Syst., Man, Cybern.*, vol. 26, pp. 259–289, Apr. 1996.

[24] G. Deng and J.-C. Pinoli, “Differentiation-based edge detection using the logarithmic image processing model,” *J. Math. Imag. Vis.*, vol. 8, pp. 161–180, 1998.

[25] M. Jourlin and J.-C. Pinoli, “Image dynamic range enhancement and stabilization in the context of the logarithmic image processing model,” *Signal Process.*, vol. 41, no. 2, pp. 225–237, Jan. 1995.

[26] M. M. Fleck, D. A. Forsyth, and C. Bregler, “Finding naked people,” in *Proc. Eur. Conf. on Computer Vision*, vol. 2, Cambridge, U.K., 1996, pp. 593–602.

[27] F. Luthon, A. Caplier, and M. Liévin, “Spatiotemporal MRF approach to video segmentation: Application to motion detection and lip segmentation,” *Signal Process.*, vol. 76, no. 1, pp. 253–260, 1999.

[28] Y. W. Lim and S. U. Lee, “On the color image segmentation algorithm based on the thresholding and the fuzzy C-means techniques,” *Pattern Recognit.*, vol. 23, no. 9, pp. 935–952, 1990.

[29] T. Coianiz, L. Torresani, and B. Caprile, “2D deformable models for visual speech analysis,” in *Proc. 29th Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 1995, pp. 391–398.

[30] F. Luthon and M. Liévin, “Entropic thresholding in image processing,” in *Proc. Eur. Signal Processing Conf.*, vol. 1, Toulouse, France, Sept. 2002, pp. 605–608.

[31] S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, pp. 721–741, Nov. 1984.

[32] P. Bouthémy and P. Lalande, “Recovery of moving object masks in an image sequence using local spatiotemporal contextual information,” *Opt. Eng.*, vol. 32, no. 6, pp. 1205–1212, June 1993.

[33] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *Int. J. Comput. Vis.*, pp. 321–331, 1988.

Marc Liévin received the M.S. degree in electrical engineering and the Ph.D. degree from the Polytechnic National Institute, Grenoble, France, in 1996 and 2000, respectively.

He was Research and Teaching Assistant at this Institute for four years. In October 2000, he joined the Research Center Caesar in Bonn, Germany, as Research Associate in medical imaging. His research interests include image and signal processing, real-time computer vision, and graphics in the scope of multimedia and medical applications.

Franck Luthon (M’01) received the electronics engineering degree and the Ph.D. degree from the Polytechnic National Institute, Grenoble, France, in 1985 and 1988, respectively.

He was an Assistant Professor at this Institute for ten years. Since September 2000, he has been a Professor at the University of Pau and Adour Province. His research interests are in the area of signal and image processing, with particular emphasis on spatiotemporal segmentation for video communication. He has been an Associate Editor for *Traitement du Signal* since 1995.