Automatic Lip Tracking: Bayesian Segmentation and Active Contours in a Cooperative Scheme

M.Liévin, P.Delmas, P.Y. Coulon, F. Luthon and V. Fristot

Signal and Image Laboratory, Grenoble National Polytechnic Institute,
LIS, INPG, 46 av. Félix-Viallet, 38031 Grenoble Cedex, France
email : lievin,delmas@lis-viallet.inpg.fr fax : +33 (0)4 76 57 47 90

Abstract

An algorithm for speaker's lip contour extraction is presented in this paper. A color video sequence of speaker's face is acquired, under natural lighting conditions and without any particular make-up. First, a logarithmic color transform is performed from RGB to HI (hue, intensity) color space. A bayesian approach segments the mouth area using Markov random field modelling. Motion is combined with red hue lip information into a spatiotemporal neighbourhood. Simultaneously, a Region Of Interest and relevant boundaries points are automatically extracted. Next, an active contour using spatially varying coefficients is initialised with the results of the preprocessing stage. Finally, an accurate lip shape with inner and outer borders is obtained with good quality results in this challenging situation.

1. Introduction

It is commonly observed that visual information provides a precious help to the listener under degraded acoustical conditions [1]. The motivation of the present work is to extract visual information for automatic speech recognition (ASR), videoconferencing and speaker's face synthesis under natural lighting conditions with few assumptions. Some approaches proposed in this area are based on grey level analysis (*e.g.* Luettin in [7]). Others use color analysis but need to determine optimal values of some parameters (*e.g.* Coianiz in [7]). A wide range of papers describe the applications of active contours for lip boundary detection but often focus on inner (*e.g.* Petajan in [7]) or outer lip contours only, rarely both.

Here, an algorithm is proposed for inner and outer lip contour tracking under natural conditions, the requirement being that a micro–camera is mounted on a light helmet worn by the speaker so that it is fixed w.r.t. the head. The RGB video sequence (8 bits/color/pixel) contains the region of the face spanning from chin to nostrils. The purpose of the process is to obtain accurate inner and outer lip borders even if the mouth is closed. A Bayesian segmentation [6] is used as an initialisation step for a snake convergence.

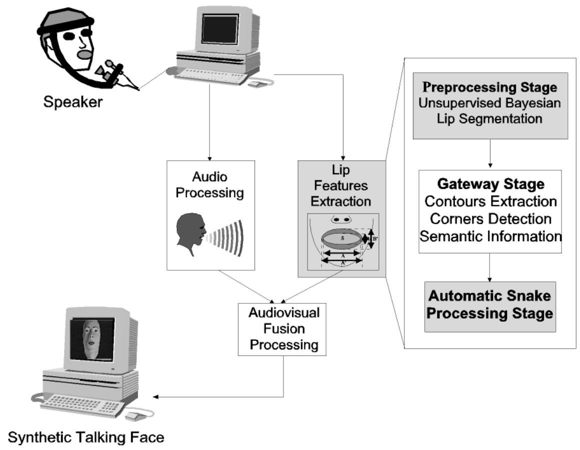

Figure 1: Context of lipreading: from sequence of speaker images, lip tracking provide parameters for talking face synthesis.

The cooperative scheme is divided into three stages:

- *Preprocessing stage*:
  Logarithmic color-space transform, *RGB* to *HI*. Unsupervised spatiotemporal segmentation of lip area and mouth location estimation.
- *Gateway*: Mouth characteristics extraction:
  Boundaries and semantic information from the segmented lips. Mouth corners detection.
- *Automatic Snake processing*:
  Snake initialization from preprocessing results. Convergence of automatic snake (outer) and balloon snake (inner).

This work is part of the Labiophone project (ELESA Federation n. 8 (CNRS-INPG)), an advanced audio-visual communication tool (Fig. 1). This project aims at providing a very low bit rate coding communication system, integrating both audio and visual features.

2. Lip segmentation

2.1. Logarithmic color transform

Face features detection is often illuminance dependent. To gain independence from lighting conditions, we compute here a color transform for illuminant–invariant recognition. Angular transforms give poor results in noisy conditions (mono-CCD camera). Therefore, a logarithmic hue transform is defined using G and B channels from the RGB color space.

We compute the hue in a mathematical framework based on a logarithmic image processing model [4]. The intensity I of an image is represented by its associated gray tone function $\boldsymbol{i} = M(1 - \frac{I}{I_0})$. This model satisfies the saturation characteristics of the human visual system and is justified from a physical point of view. Specific algebraic and functional operations are redefined in a vectorial structure. The difference between logarithmic tone of the channels G and R corresponds to the logarithmic hue tone h. With few assumptions (I_0 close to the maximum value of white M), the logarithmic difference becomes a ratio between G and R components. Finally, from the RGB color space, a HI logarithmic color space is defined (Eq. 1) (Fig. 2).

$$H = 256 \times \frac{G}{R} \quad \text{and} \quad I = \frac{R+G+B}{3} \qquad (1)$$

Figure 2: *Top*: 5 images of a typical luminance sequence; *Bottom*: the corresponding hue sequence.

2.2. Lip Hue Segmentation

2.2.1. Observations

To detect lip regions, motion information is combined with red hue. From the HI color space, two kinds of observations o are derived (Eq. 2): $h(s)$ consists in filtering the hue value $H(s)$ at pixel s with a parabola centred on the mean value of lip hue H_{lip} with a standard deviation of the hue value Δ_H; $fd(s)$ is defined as the unsigned difference between the luminance of two consecutive images. $I(s)$ represents the intensity (or luminance) at pixel s.

$$\begin{aligned} h(s) &= \left[256 - \left(\frac{H(s) - H_{lip}}{\Delta_H}\right)^2\right] \times 1_{\frac{|H(s)-H_{lip}|}{\Delta_H} \leq 16} \\ fd(s) &= |I_t(s) - I_{t-1}(s)| \end{aligned} \qquad (2)$$

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

2.2.2. Hue and motion estimation

Hue and motion parameters are estimated automatically. In the previous work [6], these thresholds were determined before segmentation by hand. The hue observation needs three parameters to be estimated: $H_{lip}, \Delta_H, \theta_h$. For that purpose, the hue histogram is a useful representation of the hue distribution over the image. We can detect two main modes: the first for the skin-lip face, the second for the background. In natural conditions (no make-up), the lip mode and the skin mode overlap (Fig. 3).

In order to estimate H_{lip} accurately, the processing respects the following steps:

- Estimate H_{skin} from the hue distribution computed over the whole image (*Left* of the Fig. 3), the only assumption being that the main mode corresponds to the hue skin.
- Cluster all pixels respecting the condition given in Eq. 2 with H_{skin} instead of H_{lip} (*Middle* of the Fig. 3).
- Evaluate H_{lip} from the hue distribution after discarding all pixels belonging to skin mode. The remaining are the lip mode (*black* of the Fig. 3) and the background mode.

The threshold hue field is then defined by $h > \theta_h$.

Figure 3: *From left to right*: histogram of hue image (*In black*: overlap between lip and skin distribution); the corresponding segmentation of skin hue (*In black*); histogram of hue image when the skin mode is discarded (*In black*: the lip mode).

The algorithm requires an appropriate threshold θ_{fd} to suppress the camera noise without cutting significant temporal changes. We compute here the entropy $E_{fd}(S)$ over an image. The threshold motion field is then defined by $fd > \theta_{fd}$ with $\theta_{fd}(S) = 2^{E_{fd}(S)}$.

The thresholded fields appear non homogeneous and noisy. Therefore, we need a statistical relaxation to segment more accurately the lips.

2.3. The Segmentation Algorithm

2.3.1. Observations and Labels in an MRF Framework

From these two thresholded observations, four initial labels (a_0,a_1,b_0,b_1) are derived for coding four pixel classes: pixels with ($_1$) (resp. without ($_0$)) motion, belonging (a) (resp. not belonging (b)) to red hue areas. This label field is supposed to follow the main MRF (Markov Random Field) property related to a *spatiotemporal neighborhood* structure (Fig. 4), *i.e.* the label l_s of the current pixell s depends only on the labels of its spatiotemporal neighbors n.

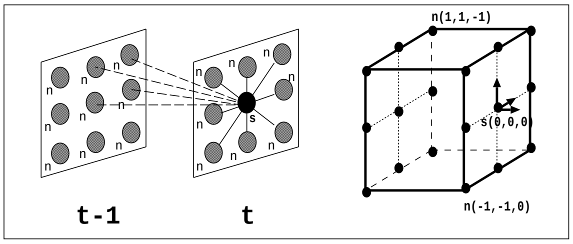

Figure 4: *Left*: Spatiotemporal neighbourhood structure η with binary cliques $c = (s,n)$. s is the current pixel (in black), n is any spatiotemporal neighbour of s (in gray); *Right*: corresponding elementary cube C_{xyt}

Maximizing the A Posteriori probability (MAP criterion) of the label field is equivalent to minimizing a global energy function [3]:

$$W(S) = \sum_{o \in \{fd,h\}} U_o(S) + \alpha . U_m(S) \tag{3}$$

where U_o and U_m represent respectively the *attachment energies* (expressing the link between labels and observations, Eq. 4) and the *model energy* (corresponding to spatial and temporal a priori constraints) (Eq. 5) over the image S, α is a weighting coefficient between the two energies.

$$U_o(S) = \sum_{s \in S} \left[\frac{[o_s - \psi_o(l_s)]^2}{2\sigma_o^2} \right] \tag{4}$$

where ψ_o is an attachment function, mean value of the observation o over S and σ_o^2 is the corresponding variance. Both are estimated on line.

The *a priori* model energy is defined as a sum of interaction potential functions over the neighborhood:

$$U_m(S) = \sum_{s \in S} \Big[\sum_{n \in \eta(s)} V_{st}(l_n, l_s) \Big] \tag{5}$$

The spatiotemporal potential function V_{st} is defined as the inverse of the Euclidian distance between two neighbors. The distance integrates two elementary potentials β_s and β_t as scale factors (Eq. 6).

$$V_{st}(l_n, l_s) = \frac{\beta_s(l_n,l_s)\beta_t(l_n,l_s)}{\sqrt{\beta_t(l_n,l_s)^2\left(\delta_x^2 + 4\delta_y^2\right) + \beta_s(l_n,l_s)^2\delta_t^2}} \tag{6}$$

where $\overrightarrow{(s,n)} = (\delta_x, \delta_y, \delta_t)$ and $\delta \in \{-1;0;1\}$

The elementary potentials β_s and β_t are defined to constrain the model respectively to spatial homogeneity of labels and temporal homogeneity of hue when no motion is detected.

2.3.2. Results and ROI estimation

An iterative deterministic algorithm (ICM : Iterated Conditional Modes) is implemented to compute the minimum energy at each site, starting from the initial label configuration L_t^0.

From lip red hue relevant labels, the ROI (Region Of Interest) is evaluated *on line* by maximizing a cost function $\Gamma(S)$ on each image (details in [6]) after each step of the relaxation. The ROI estimation reduces the relaxation time by surrounding the mouth precisely. Moreover, it increases the accuracy of parameter's estimation.

After a few iterations (typ. 10) on the label field, convergence is achieved. One obtains homogeneous red hue and motion lip fields.

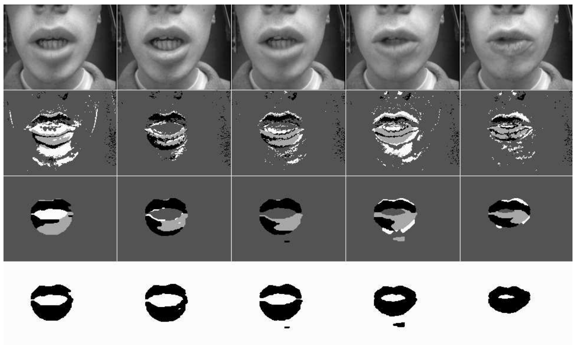

Figure 5: *From top to bottom*: sequence of luminance images; initial labels; label fields after relaxation: *the 4 labels are shown in gray levels (from white to black:* b_1, a_1, b_0, a_0*)*; sequence of hue relevant label images (a_0 and a_1).

2.3.3. Lip red hue labels and ROI

From the final label fields, one can extract lip red hue relevant label (a_0 and a_1) (Fig. 6). Those results are shown with a ROI evaluated on line. Several typical sequences have been tested, some with a soft natural red make-up, others with very poor lighting conditions without any make-up. It shows the robustness of the algorithm to the variability of the lighting conditions.

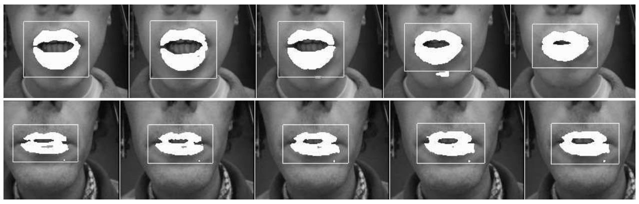

Figure 6: Two sequences of final lip hue fields with ROI superposed on the corresponding luminance.

2.3.4. Preprocessing lip contours extraction

As a preprocessing stage, the lip segmentation offers robust information to lighting conditions:

- automatic mouth location (ROI) (Fig. 6)
- unsupervised segmentation of the mouth shape
- semantic information: open/close detection

But, the borders seem to appear irregular when tongue or gum are segmented, the segmentation is elusive when close to the mouth corners, the inner contour is not segmented when the mouth is closed. We need then a higher level mouth detection algorithm: active contours. A gateway has been defined to initialise the next stage with relevant data from the preprocessing.

3. Intermediate processing: a gateway towards active contours

3.1. Mouth corners detection

ROI information is used to locate mouth corners and vertical limits of the lips. Areas of darkness occur at the inner border of lips on horizontal mouth transitions (*e.g.*: upper lips and tooth, tooth and mouth interior, tooth and lower lips). Indeed, the vertical minima of image locates lip frontiers and corners with accuracy.

The mouth corners are estimated with the following steps:

- Find the grey level minima pixel over image columns; compute the distribution (with ζ as core (Eq. 7)).

$$\zeta_j = 4 \times \frac{j * (N_{col} - j)}{(N_{col})^2} \tag{7}$$

  where N_{col} is the number of columns of the image, j the current column, the weighting coefficient ζ_j varies from 0 to 1, from the border to the center of the image.

- Detect the highest peak and deduce the horizontal symmetry axis of the mouth.
- Extract lip corners following the line of minima, from the center of the image to the left (respectively to the right).
- Estimate the width and orientation angle of the mouth.

Finally, a good estimate of the width and the orientation of the mouth can be used (Fig. 7).

Figure 7: *From left to right*: vertical minima (in white) on a grey level image; the corresponding distribution (in white); mouth corners position for an open mouth.

3.2. Lip shape extraction

Starting from the ROI coordinates, a reduced number of edges (*e.g.* 30) are detected within the inner and the outer borders of the lip shape. These edges are linked together with the mouth corners we found previously to provide an excellent snake initialization (Fig 8).

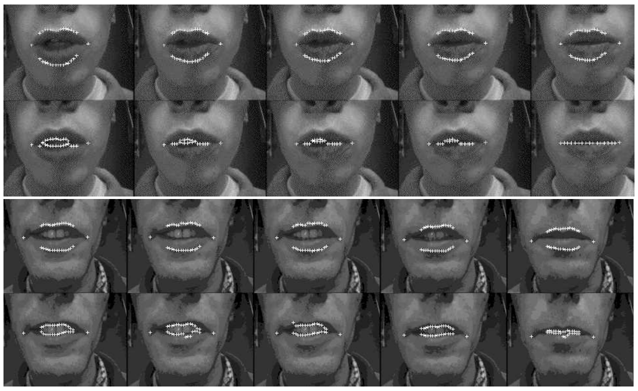

Figure 8: Inner and outer borders detection from the segmented lips and the corresponding mouth corner positions.

4. Active contours

Three major problems are classically encountered while using snakes: initialization, parameter estimation and convergence of the algorithm. Initialization is commonly done by hand, close to the object to provide good convergence.

Snake evolution is sensitive to parameter values which are usually evaluated manually after several tests. Snake convergence needs a good fitting between its energy and the desired image features.

Xu [8] solves most of the initialization and convergence problems for concave areas by creating a new external force called gradient vector flow. But the diffusion process required too much time and therefore has no quasi real time applications. L.D Cohen [2] introduced balloon forces to provide inflation of snakes or push them far away from their initial guess. This approach helps to ensure the snake convergence or gives snake dilatation properties.

The lip tracking algorithm uses non-convex energy minimization to succeed in detecting mouth boundaries. The first energy minimum reached is likely to be a local one. Therefore, a good mouth parameter estimation is an essential step for good results.

4.1. Energy minimizing curves

Introduced by Kass and al [5] active contours were designed for interactive interpretation in which the user guides (by external forces modification) the snake near the desired solution. A snake is a parameterised curve v defined (Eq. 8) by its Cartesian coordinates x and y along the curvilinear abscissa s which evolves through the minimization of its functional Φ (Eq. 9).

$$v(s) = [x(s), y(s)], \quad s\epsilon[0,1] \tag{8}$$

$$\Phi : v(s) \longrightarrow \int_0^1 (E_{int}(s) + E_{ext}(s))\, ds \tag{9}$$

The internal energy (Eq. 10) is a second order regularization term derived from Tikhonov ill–posed problems theory. It controls the curve smoothness via weighting parameters α and β. α controls the snake tension and β its curvature. External energy (Eq. 11) represents the fitting of image data to current vector. We focus on lip boundaries. We then decided to use the image gradient to extract edge points. To do so we use a classical gradient filter (such as Sobel or Canny-Deriche).

$$E_{int}(s) = \alpha|v'(s)|^2 + \beta|v''(s)|^2 \tag{10}$$

$$E_{ext}(s) = -\left|\nabla\left(G_\sigma \otimes I\right)(v(s))\right|^2 \tag{11}$$

∇ represents the gradient operator, G_σ the 2D Gaussian kernel and I the current image. This leads us to the classical dynamic scheme (Eq. 12) where I_d is the identity matrix, A the Toeplitz snake matrix, V the snake control points vector and $\frac{1}{\gamma}$ the time step coefficient.

$$V(t) = (A + \gamma I_d)^{-1}\left(\gamma V(t-1) - F(V(t-1))\right) \tag{12}$$

F represents forces derived from external energy. Higher level information forces such as Distance map or Balloon forces [2] are added there.

4.2. Adapted snakes

Our automatic snakes integrate α and β as non spatially constants values. The Toeplitz matrix obtained is not detailed here. Forces calculation is done by bilinear interpolation to reduce numerical instabilities which occur through snake energy minimization. Sampling the snake curve by spline functions during the process enforces a constant distance between snake points. Moreover, it helps moving points trapped by spurious edges. Finally we impose all parameters constant through different images for a given number of snake points.

As mouth corners are finely detected, the snake gets its extremities fixed. That kind of active contours is less unstable than the traditional ones. Convergence is tested after each resampling (classically every 10 iterations). This solves the final oscillation convergence problema of snakes. We authorize a maximum quadratic error (Eq. 13) calculated between two successive sampled snakes.

$$\epsilon = \sum_{i\in[0..N-1]} |V_i(t+t_s) - V_i(t)| \tag{13}$$

where t_s represents the sampling time step chosen and N the size of the snake control points vector.

4.3. Shape constraint

Our aim is to maintain a mouth shaped snake even without external constraints. Thus, we test non spatially constant snake parameters derived from physical considerations based on mouth geometry. For example, the lower lip contour usually has a curvature that is minimal at the middle and maximal at the corners. We choose a higher β coefficient at the middle and a lower one around corners. The same kind of adjustment is applied to the upper lip.

4.4. Results

Using the gateway lip shape information to initialize snakes gives us the opportunity to be closer to the desired object. Inner positioning is done in the mouth. Therefore, we should inflate the inner snake to reach inner lips boundaries. Cohen Balloon forces reduce the natural ability of snakes to shrink even without external forces. When the mouth is closed, we simply sample the inner snake along the straight line between lip corners.

We hold all the coefficients constant throughout our tests for both inner and outer snakes. Images from our image database including open and closed mouths from different faces are tested. Outer and inner snakes always reach good boundaries after few iterations, usually less than 100.

The top of Fig. 9 shows five successive frames of a closing mouth. Outer and inner lips are perfectly tracked with a few number of iterations (about 60). The inner snake is capable of detecting small and asymmetric mouth apertures (frames 2 to 4).

The bottom of Fig. 9 points out specific problems of lip tracking. Lips vary in shape from one speaker to another. Benny has thin upper lips. His mouth is longer than the previous speaker. His upper lip is very elusive but our algorithm succeeds in detecting both inner and outer lip frontiers.

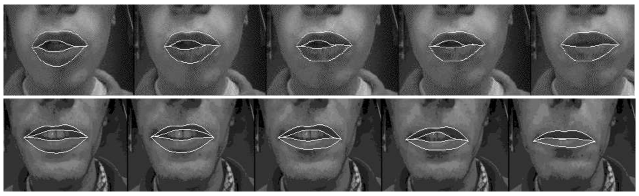

Figure 9: Convergence snakes results on two sequences (*Top*: Nico; *Bottom*: Benny).

5. Conclusion

An automatic lip contours extraction has been successfully applied to several sequences in natural conditions (natural images of speaker's face without any particular make-up or lighting).

First, the logarithmic transform and the following spatiotemporal Bayesian segmentation dealing with hue and motion information locate and segment the mouth with very few error rate. Then, a preprocessing gateway, combining low level information and mouth corners location, initializes inner and outer active contours. Thanks to the mouth corners location algorithm, the two snakes get their extremities fixed. Whereas snakes were described by their inventors as a *semi-automatic* process, these preprocessing steps prove that automatic snakes are viable. Finally, with no assumption about the lighting conditions, the lip contours (inner and outer) are extracted with accuracy, even when the mouth is closed or asymmetric.

Some problems still occur for a fine detection when tongue or gum are detected. We need also to deal with more difficult cases like colored people or faces with beard (Fig. 10). The proposed algorithm requires less than 3 to 4 seconds per image on a standard 150MHz workstation. Therefore, hardware implementation for both stages, spatiotemporal segmentation and active contours, are currently under study.

Figure 10: First results on beard faces sequences: the outer contour.

6. References

[1] C. Benoît, M.T. Lallouache, and T. Mohamadi. A set of french visemes for visual speech synthesis. In *Talking Machines: Theories, Models and Designs*, pages 485–504. Elseviers Science Publishers, 1992.

[2] L.D. Cohen. On active contour models and balloons. *Computer Vision, Graphics and Image Processing*, 53(2):211–218, 1991.

[3] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Trans. Pattern Analysis Machine Intell.*, 6(6):721–741, November 1984.

[4] M. Jourlin and J-C. Pinoli. Image dynamic range enhancement and stabilization in the context of the logarithmic image processing model. *Signal Processing*, 41(2):225–237, January 1995.

[5] M. Kass, A. Witkins, and D. Terzopoulos. Snakes: Actives contours models. *International Journal of Computer Vision*, 1(4):321–331, 1987.

[6] M. Liévin and F. Luthon. Lip features automatic extraction. In *Proc. of the* 5^{th} *IEEE Int. Conf. on Image Processing*, volume 3, pages 168–172, Chicago, Illinois, October 1998.

[7] D. Stork and M. Hennecke. *Speechreading by Humans and Machines*, volume 150. Springer-Verlag, Berlin, 1996.

[8] C. Xu and J.L. Prince. Gradien vector flow: A new external force for snakes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 66–71, San Juan, Porto Rico, 1997.