Liptracking and MPEG4 Animation with Feedback Control

Brice Beaumesnil, Franck Luthon, Marc Chaumont

LIPTRACKING AND MPEG4 ANIMATION WITH FEEDBACK CONTROL

B. Beaumesnil, F. Luthon

LiUPPA
IUT Informatique de Bayonne
Château Neuf, Place Paul Bert 64100 Bayonne, France

M. Chaumont

LIRMM
Computer Science Department
161 rue Ada 34392 Montpellier, France

ABSTRACT

This article deals with facial segmentation and liptracking with feedback control by face model synthesis. On this topic, the search community is divided into two parts : analysis and synthesis. We want to use all the knowledge to create a global analysis/synthesis chain where the image analysis needs the 3D synthesis and conversely. As it happens, applications like face tracking or augmented reality need a rapid, robust and descriptive-enough solution.

Our solution is based on a two step approach : the first step is a real-time facial segmentation with active contour models and the second step recovers a 3D-face model in order to extract more precise parameters to adjust the first step. The contribution of this paper is to couple two research fields for creating a real time application. The results obtained show rapid and robust performances which could be exploited in a more global real-time face tracking application.

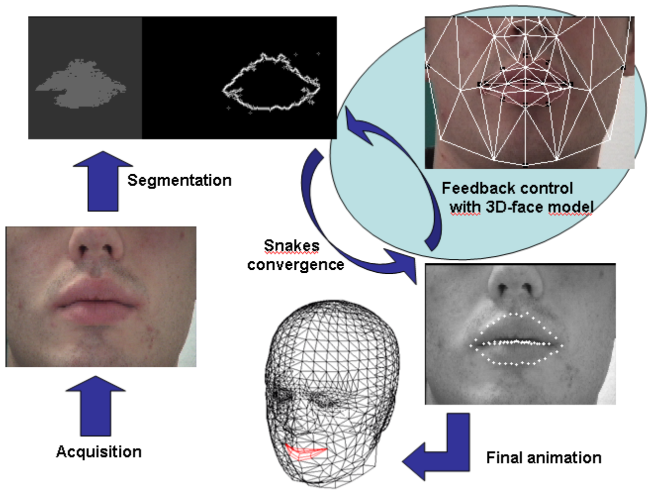

Fig. 1. Complete real-time analysis/synthesis framework.

1. INTRODUCTION

We present a complete real-time analysis/synthesis framework allowing lip tracking for animation of a clone with a single camera in unconstrained environment (typ. webcam in the office). The approach is based on lip segmentation from a hue component computed within a non-linear color space that is robust to luminosity variations. Internal and external active contours are extracted, and then interpreted to make real-time realistic animation of a clone's mouth.

To interpret them, we use a feedback loop with a 3D-face model. This model is shaped on the face and its animation parameters give a approximation of the mouth shape. Then we comparte it with the position of snake points to correct them.

Finally, robust snake points are sent to animate a 3D-face model (typ. a MPEG4 avatar) (see Fig.1).

Major methods don't implement any feedback, and many of them give us a average lip shape. They are often based on high level methods (AAM, ASM)[1], that require large sized video databases for off-line learning or use pattern matching algorithm to track manual points of initialisation[2]. All of them return a mouth shape estimated by their model.

Our approach departs from the above mentioned methods in the sense that we want to use low-level algorithms that can adapt to any mouth shape but without a priori knowledge about the face. The mouth shape is more precise because we use lip gradients directly on all images. The results obtained show real-time and robust performances without any learning stage, huge databases, or manual initialisation.

2. ANALYSIS : REAL TIME SEGMENTATION

Our work assumes that we have at our disposal an automatic tool for face detection and tracking, so that an optimal framing of the speaker's face is at our disposal (which is required in the case of videoconferencing for example). If this is not the case, the only constraint is that the speaker should stay in front of the camera with little motion (normal behavior in front of a webcam). In this paper, we simply ask the speaker to seat directly in front of the camera so that his face covers the major part of the image. Moreover, we ask him to have the mouth closed (neutral position) on the first frame of the sequence, in order to initialise the 3D-face model.

The segmentation algorithm is divided into four parts :

2.1. Color Conversion

For face and lip segmentation, we use the LUX color space defined in [3]. This color space is non linear with respect to the RGB color components and is little sensitive to lighting variations. It exhibits very distinctly skin and lip hue areas. In this color space, most of the color information relative to a human face is coded by the U component (red chromaticity) in the particular case when $R > L$ (L being the luminance Eq.1) (see Fig.2-a).

$$L = (R+1)^{0.3}(G+1)^{0.6}(B+1)^{0.1} - 1 \tag{1}$$

$$U = \begin{cases} 256\frac{L+1}{R+1} & \text{if } R > L, \\ 255 & \text{otherwise or if } L < \delta. \end{cases} \tag{2}$$

The threshold δ is introduced in order to detect very dark areas like the inner side of the mouth or nostrils.

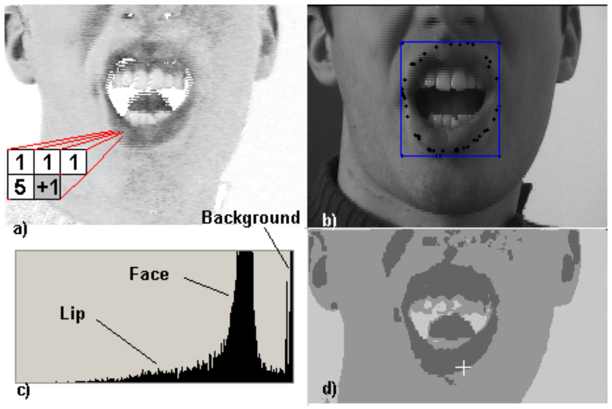

Fig. 2. a) Hue U with nonlinear filter mask ; b) Original color image, with bounding-box and outer lip contour points ; c) Hue Histogram ; d) 3-class map given by the k-means algorithm, with pixel P in white

2.2. Clustering

Since the hue difference between face and lips is more contrasted in this space, we can easily classify pixels as lips or face (see Fig.2-c). For that purpose we use the k-means classification algorithm. It works with three classes : lip, face and background ; it exploits two types of low-level information : the mean value of hue in a given neighborhood, and the maximum deviation from this mean value for any pixel in the neighborhood. Various empirical tests lead to set ad hoc parameter values that only depend on the mean-value of speaker's face hue in the whole image. These parameters are used to initialise the centers of the three classes (see Fig.2-d).

After processing, this technique gives us new precise centers of the three classes (because they are specific of the image).

2.3. Region Of Interest

In order to initialise active contour models, we need to know where the mouth is in the current image.

For that, we use a lip tracker based on Lucas-Kanade algorithm applied on a few relevant points detected on the outer lip contour (see Fig.2-b). We can get mouth optimal bounding box by a horizontal minimal enclosing area of all lip points.

But, for the first image of the sequence, we cannot use it. Thus we've implemented a non linear filter that finds, in a single pass, the better candidate point to be on the lip (named P), usually at the lower left edge of the lips (see Fig.2-d). The filter is generated to detect the bigger horizontal reddish form on the image (Eq.3). A mouth optimal bounding box (named BB) can be obtained by a simple scan of the connected component lip-hue area that includes P (point giving the maximum value of M).

$$M(i) = \begin{cases} 1 + \sum_{j \in \nu(i)} a(j)M(j) & \text{if } U(i) \in \text{“}lips\text{''} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

2.4. Active Contour Model

For estimating the lips outer border, one active contour [4] is initialised with the BB detected as explained above.

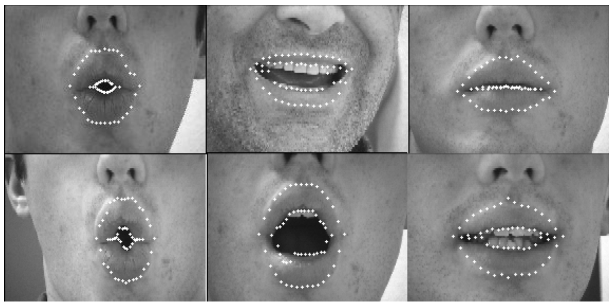

Fig. 3. Typical snake convergence results for different visemes corresponding to French phonemes : top) vowels [ø], [e] plus neutral position ; bottom) vowels [o], [a] and [i]

This snake is made of a finite number of control points that are forced to undergo only vertical displacements during iterations. The points are initialised on cubic curves computed from the BB and from the lip map (to position the lip corners).

The forces used for snake convergence are the following :

– Internal forces emanate from the shape of the snake (elasticity and stiffness)
– External forces come from higher level image understanding processes (gradients)
– A contraint force that is specific of the problem at hand (the snake is forced to converge towards the gravity center of the BB)

$$\begin{pmatrix} s.u'_i \\ s.v'_i \\ s \end{pmatrix} = \underbrace{\begin{pmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{intrinsic\ parameters} . \underbrace{\begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{extrinsic\ parameters} . \begin{pmatrix} X'_i \\ Y'_i \\ Z'_i \\ 1 \end{pmatrix} = T.M'_i,\ with \begin{cases} \alpha_u = -k_u.f \\ \alpha_v = k_v.f \end{cases} . \quad (4)$$

After convergence of the outer snake, another snake (inner one) is initialised on the outer one, then shrinked by a non isotropic scaling w.r.t. the mouth center and taking into account the actual thickness of lips.

3. USED OF AN A PRIORI INFORMATION : A 3D-FACE MODEL

A contrario to the first step of lip segmentation (2), this step is based on a priori knowledge : a face own a specific 3D structure. In this paper we use the 3D-face model named CANDIDE-3 [5] in order to model this specific 3D-face structure. The 3D-face model is extracted and then help us to limit the face deformations degrees and the mouth deformation degrees. Thus, the 3D-face model allows to constrain the previous segmentation solution (obtained thanks to active contours) and then to keep the mouth position in an acceptable solution space.

The 2D features points stemming from the lip segmentation may be noisy (2D positions are un-precise) and their number is small. Our solution to extract a 3D-face model (for more details see [6]), with the knowledge of 2D features points, takes care of those difficult constraints and moreover is well-suited for real-time applications.

The solution is divided in two steps. The first step recovers an approximation of the 3D-pose (section 3.1), the second step recovers an approximation of the 3D-face model (section 3.2).

3.1. Pose approximation

To extract the 3D-face model and the 3D-pose, we minimize the distance error E (see equation 5) between the observed set of 2D image points $\{(u_i, v_i)^t\}$ and the projected set of points $\{(u'_i, v'_i)^t\}$. The projected set of points $\{(u'_i, v'_i)^t\}$ are obtained by projecting all the corresponding 3D-face model vertex using the T projection (see equation 4).

$$E = \sum_i (u_i - u'_i)^2 + (v_i - v'_i)^2. \quad (5)$$

The projection of a set of points $\{(u'_i, v'_i)^t\}$ belonging to the 3D model is the result of a shape displacement ($S_i.\sigma$), an animation displacement ($A_i.\alpha$) and a projection (T) of an CANDIDE-3 average 3D-face model as expressed in the following equation :

$$\begin{pmatrix} s.u'_i \\ s.v'_i \\ s \end{pmatrix} = T.[\underbrace{M_i + S_i\sigma + A_i.\alpha}_{M'_i}]. \quad (6)$$

S_i and and A_i are respectively the shape unit and the animation unit matrix, expressing the possible displacement of a vertex i. The displacement intensity is expressed by the weighting vectors σ and α. More details are given in Ahlberg's report [5].

The computation of projection T (given in equation 5) is not an easy task ; projection T should then be simplified. This simplification consists in supposing that all the 3D vertex are in a same 3D plan. This is a realist hypothesis when there are small depth differences between 3D points in comparison to the distance between the camera and the face.

By canceling, from equation 5, each E's partial derivative in function of T's parameters, we obtain 2 linear systems (σ and α are set to zero). Those two systems are solved by using classical linear algebra tools. Note that the matrix involved in that system is very small (matrix size=4×4) ; its computation and its inversion is very rapid.

3.2. Shape approximation

Once $T_{2\times4}$ projection is computed, shape adaptation is processed. The minimization problem of equation 5, is solved by fixing $T_{2\times4}$ projection. Equation 5 is re-written such that :

$$\begin{aligned} E &= \sum_i [U_i - N.S_i.\sigma]^t.[U_i - N.S_i.\sigma], \quad (7) \\ & \quad with\ U_i = \begin{pmatrix} u'_i \\ v'_i \end{pmatrix} - T. \begin{pmatrix} M_i \\ 1 \end{pmatrix}, \\ & \quad and \quad N = T_{2\times3}. \end{aligned}$$

We obtain a linear system (equation 8) by canceling the partial derivative $\frac{\partial E}{\partial \sigma}$:

$$\underbrace{(\sum_i S_i^t.N^t.N.S_i)}_{A} .\sigma = \underbrace{\sum_i S_i^t.N^t.U_i}_{B} . \quad (8)$$

Solution is such that $\sigma = (A^tA)^{-1}A^t.B$. The matrix involved in the system is small and sparse ; its computation and its inversion are very rapid. The same reasoning may be done for α computation.

4. FEEDBACK CONTROL

3D-face model extraction advantage is the rigidity compared with snake convergence. Indeed the 3D-face model uses their animation units to find the mouth shape, thus the shape is more realistic than bad segmentation. But it could be a disadvantage compared with a good convergence, because the shape described by the model is an approximation. To use this information we have implemented a feedback control consisting in three steps :

- error measurement (between some snake's point and its corresponding 3D point model).
- backward transmission of some ROIs that need to be processed again (because of bad segmentation)
- re-run of the segmentation process on those ROIs

4.1. Error Measurement

To establish an error measurement we add two parameters to our algorithm :

- One to each snake point. This parameter can have two values : 0 et 1. If it is 0 the point can be moved to reduce snake energy, otherwise the point is fixed.
- One to each lip's 3D model point. It is used to stock error distance with its corresponding snake point.

To compute error measurement we must find the correspondance table between snake points and 3D model points (just 18 snake points can be used, because the model is an MPEG4 3D model). This table is easy to compute, we must find the snake point who are in the same row (or the nearest) that every lip's 3D model point. After that, we can compute error by equation 9 (where y is line component)

$$Error_i = 3Dmodel_i.y - Snake_{f(i)}.y \qquad (9)$$

The sign of error value indicates the position of the snake compared to the position of the 3D model.

4.2. Rescue treatment

If the distance between a snake control point and the 3D mesh corresponding position is clearly erroneous (see Fig.4-b) a rescue treatment is called to correct its position.

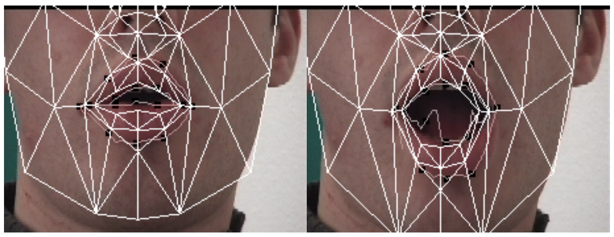

Fig. 4. Regularisation though the 3D model : a) Good match between segmentation and the model ; b) Bad match : a control point of the inner snake is clearly erroneous and is corrected by the 3D mesh corresponding position.

In this case, all good snake points are fixed and we reposition the remainder and choose a new method to treat them (typ. a new image gradient to converge snake). This local-recomputation can be made in a little area (a zoom-in in the neighborhood).

After convergence of all snake points (with a minimum error criterion), we can animate clone with our MPEG4 points. To have a realistic animation, we use snake corresponding points to shape the model.

5. CONCLUSION

This work demonstrates that one can build a complete analysis-synthesis chain that works in real-time for 3D head animation, without using sound information nor face databases for learning. The whole framework, implemented in non optimised C-code on an $i386$ processor at $1.4GHz$, works in real-time (i.e. processing rate better than $30Hz$).

Another direction of our current research is to segment not only the lips, but also other face features (namely nostrils, eyes, eyebrows and ears). This will help us to adjust 3D pose (using just the mouth points is a too poor input data for 3D pose extraction) and may enable a better understanding of some spoken phonemes.

Finally, as we want to be able to animate various clones and propose a generic solution, we are also working on the use of MPEG4-compliant 3D models (using FDP and FAP, facial animation parameters) already partly present in our framework.

6. REFERENCES

[1] I. Matthews and S. Baker, "Active appearance models revisited," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 135–164, November 2004.

[2] F. Dornaika and F. Davoine, "Head and facial animation tracking using appearance-adaptive models and particle filters," in *IEEE CVPR Workshop on Real-time Vision for Human-Computer Interaction*, Washington DC, 2004.

[3] Marc Lievin and Franck Luthon, "Nonlinear color space and spatiotemporal mrf for hierarchical segmentation of face features in video," *IEEE Trans. on Image Processing*, vol. 13, pp. 63–71, Jan. 2004.

[4] Andrew Witkin Micheal Kass and Demetri Terzopoulos, "Snake : Active contour models," *International Journal of Computer Vision*, vol. 1, pp. 321–331, 1987.

[5] J. Ahlberg, "CANDIDE-3 - un updated parameterised face," Tech. Rep., Department of Electrical Engineering, Linköping University, Jan. 2001.

[6] M. Chaumont and B. Beaumesnil, "Robust and real-time 3d-face model extraction," in *IEEE International Conference on Image Processing, ICIP'2005*, Sept. 2005, pp. 461–464.