

# Genetic Snakes: Application on Lipreading

Renaud Séguier, Nicolas Cladel<sup>1</sup>

## Abstract

Contrary to Genetics Snakes, the current methods of mouth modeling are very sensitive to initialization (position of a snake or a deformable contour before convergence) and fall easily into local minima. We propose in this article to make converge two snakes in parallel via a genetic algorithm. The coding of the chromosome takes into account at the same time gradients and region type information contained in the image. In addition we introduce the use of STM (Sparse Template Matching) into the field of lipreading. Thanks to a temporal filter, word signatures (stored in Sparse Templates) make it possible to recognize various words pronounced several times at one week interval.

## 1 Introduction

Genetic Algorithms have been used for a many years in computer vision for their optimisation quality [1] and their aptitude to avoid local minima [2]. In particular, Genetics Snakes [3] [4] [5] make it possible to overcome the problem of snakes initialization.

This problem of initialization is crucial in lipreading. The potential applications (speech recognition, lip-synchronization in audio-video flows, synthesis of talking heads) require a robust algorithm of mouth modeling. This robustness is not achieved in current modeling techniques (shape model [6], dynamic contours [7], deformable models [8], cumulative histograms analysis [9]). For that reason certain teams turned to a low level mouth modeling (DCT Discrete Cosinus Transform) which requires very big learning bases [10]. These last techniques are restrictive: they do not analyze explicitly the shape of the mouth, and cannot thus be used in talking head type applications. We propose in the section 2 an implementation of Genetic Snakes (GS) to model the mouth in a robust way.

In the field of embedded system we need to implement simple and light memory classifiers. The section 3 introduce the use of such a classifier in lipreading: the STM (Sparse Template Matching) [11]. We will see that with the help of a temporal low-pass filter, the sparse matrices become robust with the possible temporal variation which appear when a person pronounces a word more or less quickly.

## 2 Genetic Snakes

First of all we will present the preprocessing which enable us to generate images of differently filtered mouths. We will introduce afterwards the coding which bind the GS to the images, and then the energy that we will maximize using the genetic algorithm.

### 2.1 Image preprocessing

A face detector [12] locates the face in the image. We use then the same process as [13] to locate the mouth. In the V values (from YUV color coordinate system), the lips has a strong level of intensity while the teeth and the dark interior of the mouth are confused and rather dark (Fig. 1b). It is enough then to binarize the image to obtain a zone characterizing the interior of the mouth after a rapid morphological treatment (erosion and area growing starting from the center of the image, see Fig. 1c)). The same morphological treatment (erosion and area growing starting from points belonging to the upper and lower lips) is applied on light pixels to obtain the image d). We carry out finally the fusion of the images c) and d) to obtain the area of external lips contours (Fig. 1e)). The gravity center of the interior of the lips  $C_G$  is then evaluated starting from the image c). We apply then a Sobel filter to extract edges from these images and store the information of the direction of contour (see Fig. 2).

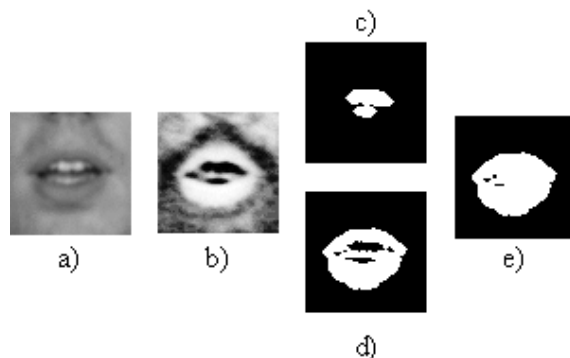


Fig. 1. Mouth preprocessing

<sup>1</sup>Supélec - Team ETSN, Avenue de la Boulaie - BP28 35511 Cesson Sévigné, France. E-mail: Renaud.Segulier@supelec.fr, Nicolas.Cladel@supelec.fr

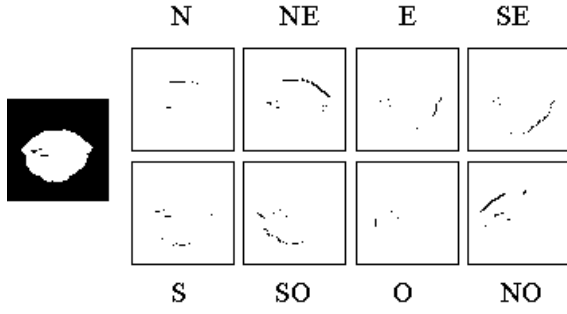


Fig. 2. Edges extraction

## 2.2 Chromosome coding

Our aim is to find a first snake on the external lips contour and a second one on the interior contour. Each one of these snake is defined on eight nodes. To accelerate optimization, we make evolve the snakes only on contours points. Thus the node C (Fig. 4) will be defined only on North, North-East and North-West directed points in the area RC (see Fig. 3 and 3). This area is defined starting from the gravity centre  $C_G$  of the mouth previously given.

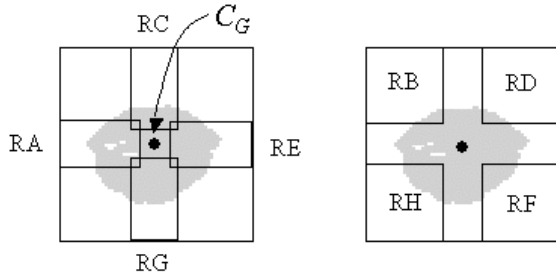


Fig. 3. Mouth regions

All the other nodes of the first snake are defined in the same manner by taking into account the area in which they must evolve and the contours orientation which characterize them. With regard to interior contour, insofar as the mouth is sometimes closed, it is difficult to define in a robust way the areas in which the nodes of the second snake must evolve. This is why we take into account only the contours orientation of the image d) of the figure 1 knowing that the nodes of the second snake can belong to any area of the image.

The position of each node of both snakes is coded on a chromosome gene as the figure 4 indicates it. Thus, the tenth gene codes the value  $x + yL$  if  $L$  is the width of the image and  $(x, y)$  the coordinates of the node  $J$  belonging to the second snake.

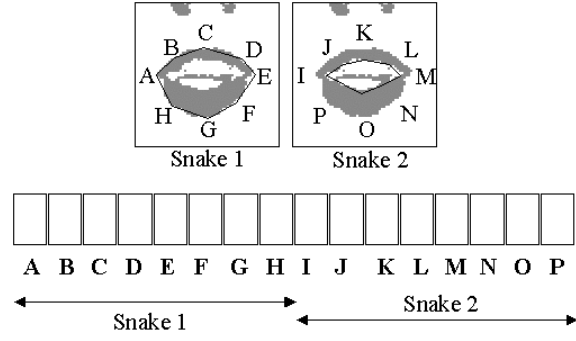


Fig. 4. Chromosome coding

## 2.3 Energie evaluation

The energy we wish to minimize is constituted by three terms controlled by three coefficients  $\alpha_1, \alpha_2$  et  $\alpha_3$ .

$$E = \alpha_1.E_{Bend} + \alpha_2.E_{Snake2} + \alpha_3.E_{InterSnake} \quad (1)$$

The first term makes it possible to control the rigidity of the curve. This constraint is evaluated on the whole set of the nodes of both snakes except for those corresponding to mouth corners ( $A, E, I, \text{ and } M$ , see Fig 4).

$$E_{Bend} = \sum_{i=2, i \neq 1+N/2}^N \|V I_{i-1} - 2V I_i + V I_{i+1}\|^2 + \sum_{i=2, i \neq 1+N/2}^N \|V 2_{i-1} - 2V 2_i + V 2_{i+1}\|^2 \quad (2)$$

where  $i$  is the number of the node on the curve knowing that a snake contains  $N$  nodes. The first node is on the left corner of the mouth ( $A$  or  $I$  in the figure 4), the node  $1 + N/2$  is on the right corner of the mouth ( $E$  or  $M$ ).  $V_j$  is  $i$ -th node of the snake  $j$ .

The second term  $E_{Snake2}$  takes into account the quantity of pixels  $NbPix_{S2}$  belonging to the interior mouth (white pixels of the figure 1c) belonging to the area R1 of figure 5).

$$E_{Snake2} = \frac{1}{NbPix_{S2}} \quad (3)$$

The last term  $E_{InterSnake}$  makes it possible both snakes to surround the lips. It takes into account at the same time the sum of the pixels levels belonging to the lips ( $V$  values, Fig 1b) in the area R2 of the figure 5):

$$Pix_{Lips} = \sum_{R2} V_{value} \quad (4)$$

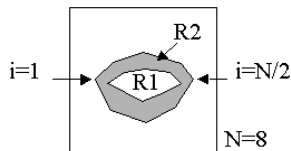
and its complement evaluated as:

$$Pix_{nonLips} = NbPix_{Inter} - Pix_{Lips} \quad (5)$$

Where  $NbPix_{Inter}$  is the amount of pixels between the two snakes (area R2 of the figure 5). Considering at the same time these two terms makes it possible to take into account the most clear pixels in the area R2 while minimizing the dark number of pixels in this same area.

We have thus:

$$E_{InterSnake} = Pix_{nonLips} + \frac{1}{Pix_{Lips}} \quad (6)$$



**Fig. 5.** Areas defined by both snakes

Après minimisation de l'énergie  $E$ , la snake décrite par le chromosome d'énergie minimum évalue le contours des levres. Notons que nous ne mettons pas en oeuvre un tracking de cette forme [14] étant donné la très grande variabilité des lèvres entre deux images (acquises à 25Hz dans notre système).

After minimization of energy  $E$ , the snake described by the chromosome of minimal energy evaluates contours of the lips. Because of the great variability of the lips between two images (acquired at 25Hz in our system), we do not implement a shape tracking like [14].

### 3 Lipreading System

The lipreading system includes a preprocessing and a classification module.

#### 3.1 Preprocessing

Our team considers halfway parameters in term of level of analysis between DCT and high-level models.

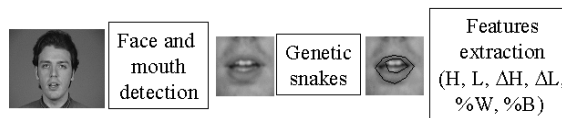
As figure 6 indicates it, six parameters only are sufficient to realize the lipreading: the height and the width of the mouth ( $H, L$ ), their temporal derivative ( $\delta H, \delta L$ ) and the percentage of light  $\%W$  and dark pixels  $\%B$  contained in the mouth [13]. The height and the width of the mouth are evaluated starting from the exterior snake:

$$H = G_y - C_y \quad (7)$$

Where  $P_y$  represent the Y-coordinate of the point  $P$  in the figure 4. With regard to the width, the first node of the interior snake always does not correspond to node I since we did not restrict the area in which the nodes of the second snake evolve contrary to those of the first one (see section 2.2). Thus we evaluate the X-coordinate of  $I$  and  $M$  by detecting the X-coordinates min and max values of the points  $I$  to  $P$ :

$$L = \max V_{2ix} - \min V_{2ix}, \text{ with } i = 1..N \quad (8)$$

where  $V_{2ix}$  is the X-coordinate of the second snake point  $i$ .



**Fig. 6.** Preprocessing

In a small window of 10 pixels height centered on the mouth, we calculate an eight-bit grey level histogram of the pixels. The ratio between the number of pixels whose values are lower than 50 and the total pixel number gives us the percentage of dark pixels, that between the number of pixels whose values are superior to 150 and the total pixel number gives us the percentage of light pixels. We add to these four parameters (height, width, dark percentage, light percentage), the temporal derivate of the height and the width.

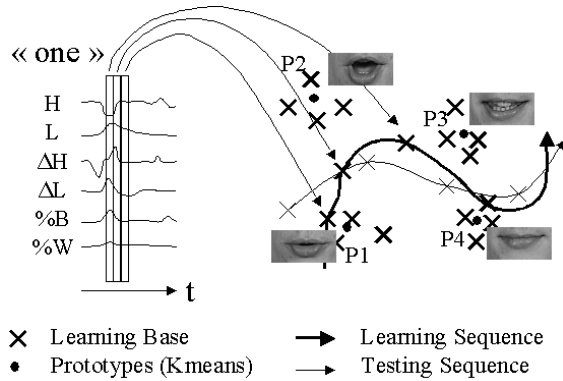
Those parameters are normalised taking into account their observed values all along the sequence.

#### 3.2 Classification

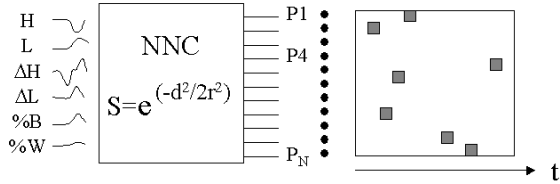
The STM (Sparse Template Matching) [11] have a very weak complexity and require not much memory, reasons why they are well adapted to embedded applications. To be able to exploit them, we use the same generic preprocessing as in [13] which makes it possible to convert continuous signals to a representation in the form of a sparse matrix. This generic preprocessing is simple. During the phase of training a K-means identifies  $N$  prototypes or vector codes on parameters vectors ( $H, L, \Delta H, \Delta L, \%W, \%B$ ) taken at every instant. As shows it figure 7 the pronunciation of a word can then be characterized by a trajectory in the space of these vectors, trajectory itself characterized by a sequence of prototypes (learning sequence of figure 7:  $P_1 \rightarrow P_1 \rightarrow P_3 \rightarrow P_4$ ).

This sequence which characterizes the pronounced word is identified using a Nearest Neighbour Classifier (NNC) algorithm (Fig. 8). At every instant, one compares the parameters vector with each prototype identified during the training phase, and one emits a unit value on the output associated with the nearest prototype (testing sequence of figure 7:  $P_1 \rightarrow P_1 \rightarrow P_4 \rightarrow P_4$ ). A word is thus characterized by a sequence of prototypes, itself store in a binary sparse matrix.

Nevertheless a person always does not pronounce the same words at the same speed. To make a more robust detection we thus apply a temporal low-pass filter on this

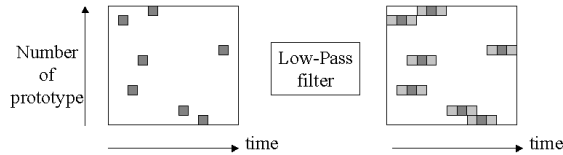


**Fig. 7.** Learning phase



**Fig. 8.** The Nearest Neighbour Classifier which produce a sparse matrix

matrix. Then, a sparse template is not represented anymore by a sparse binary matrix but by a sparse multivaluated one as indicates it the figure 9.

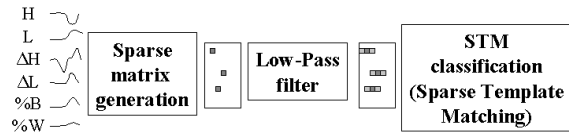


**Fig. 9.** Low-pass filtering along the time

Classification is extremely simple and takes as a starting point the techniques of tracking in image processing [15]. In the case of a one pass learning, the sparse matrix generated during the training represents the signature of the word which will have to be recognized. There is as many sparse template than words to be recognized. Classification is done simply by evaluating an Euclidean distance between the sparse matrix resulting from the new word presented and each sparse template. The smallest distance indicates the recognized word. If the training is done on several sequences, the sparse template characterizing a word is simply the average of the sparse matrices evaluated on each training sequence of the word to recognize. The figure 10 represents the whole classification treatment.

## 4 Results

Within the framework of separated word recognition, we tested our system on the first person of the Euro-



**Fig. 10.** Sparse Template Classification

pean Data Base M2VTS [16] (Multi Modal Checking for Teleservices and Security applications). This base is dedicated to Audio-visual recognition and identification. The person pronounces four times (at one week interval) the digits from 0 to 9. We chose this base because it characterizes well the conditions of use in which the real time implementation of our system will have to function. The images were acquired at 25Hz with a weak resolution (288x360pixels in 4:2:2).

The figure 11 makes it possible to visualize the difference between the real height of the mouth (dotted line) and that detected starting from the first snake (solid line). The variation between these two values is not significant. The important thing is that the detected height evolution follows well the real height one throughout sequence. Thus at instant 10, it seems that the snake is badly positioned contrary to the result provided at instant 114, whereas both exterior snakes are well positioned. On the other hand, at instant 71 it is clear that the first snake fell into a local minimum of the energy which induces a bad evaluation of the mouth height. On the four sequences, the mouth height is indeed badly estimated (as at instant 71, figure 11) in 1.2% of the cases only (in 8 images out of 669).

In the context of single-speaker lipreading, we tested two different configurations knowing that four sequences of ten digits were available: a one pass and a three sequences training. Ten prototypes were identified in training phase ( $N = 10$ , Fig. 8).

*One pass learning* The results are presented as an average performed on the four tests carried out starting from four different sequences of training; according to the number of the sequence used for the training, tests are made one the three other sequences. In this configuration, the system performed a correct classification in 68% of the cases. Let us note the temporal filter advantage (Fig. 9). If it is not implemented, the results are only of 58%,

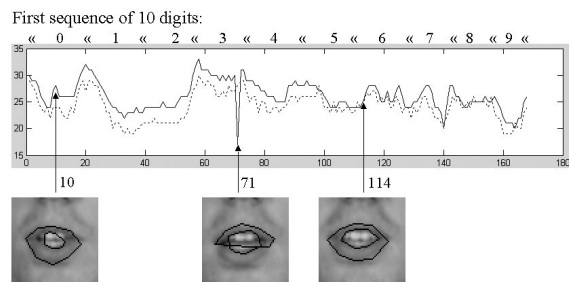
*Three sequences training* The average result evaluated on the four various sequences of tests were obviously better: the classification was correct in 88% of the cases. Without the temporal filter, the results were only of 83%.

## 5 Conclusion

We proposed in this article an implementation of Genetics Snakes in lipreading. Two snakes converge in

parallel towards lips contours. A very simple classifier (Sparse Template Matching) gives good results within the framework of this application and allows us to consider a real time implementation on an embeded system.

However the convergence of the GS is still too long (54s on P3-900Mz) even if the implementation was made in Matlab. Moreover, the coefficients  $\alpha_1$ ,  $\alpha_2$  and  $\alpha_3$  which control various energies to be optimized, were given at hand. It remains us to propose an automatic method who will allow us to identify the value of these coefficients on a person test and to check their robustness on the other people introduced in M2VTS.



**Fig. 11.** Dotted line : real mouth height along the first sequence, solid line : first genetic snake height

## 6 Acknowledgment

A part of this research was supported by Brittany Region ("Région Bretagne") in France.

## References

- [1] C. Bounsaythip and J.T. Alander. Genetic algorithms applied to image processing - a review. In *Proc. of the 3rd Nordic Workshop on Genetic Algorithms (3NWGA)*, 1997.
- [2] K. Sakaue, A. Amano, and N Yokoya. Optimization approaches in computer vision and image processing. *IEICE Trans. Inf. and Syst.*, 1999.
- [3] A. Cagnoni, A. Dobrzeniecki, R. Poli, and J. Yanch. Genetic algorithm-based interactive segmentation of 3d medical images. *Image and Vision Computing*, 17(12):881-895.
- [4] L. Ballerini. Genetic snakes for color images segmentation. *Lecture Notes in computer sciences 2037*, 2001.
- [5] N. Covavisaruch and T. Tanatipanond. Deformable contour for brain mr images by genetic algorithm: From rigid tottraining approaches. In *Proceedings, Image and Vision Computing New Zealand (IVCNZ '99)*, 1999.
- [6] Michael T. Chan, You Zhang, and Thomas S. Huang. Real-time lip tracking and bimodal continuous speech recognition. In *Workshop on Multimedia Signal Processing*, 1998.
- [7] P. Delmas, P.Y. Coulon, and V. Fristot. Automatic snakes for robust lip boudaries extraction. In *International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 1999.
- [8] Y. Tian, T. Kanade, and J. F. Cohn. Recognizing action units for facial expression analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2001.
- [9] R. Séguier, N. Cladel, C. Foucher, and D. Mercier. Lipreading with spiking neurons: One pass learning. In *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, Feb 2002.
- [10] G. Potamianos, J. Luettin, and C Neti. Hierarchical discriminant features for audio-visual lvcsv. In *International Conference on Acoustics Speech and Signal Processing (ICASSP)*, May 2001.
- [11] G. Sullivan, K. Baker, A. Worrall, C. Attwood, and P. Remagnino. Model-based vehicle detection and classification using orthographic approximations. In *Proc of 7th British Machine Vision Conference*, 1996.
- [12] R. Séguier, A. Le Glaunec, and B. Loriferne. Human faces detection and tracking in video sequence. In *Proc. 7th Portuguese Conf. on Pattern Recognition*, 1995.
- [13] R. Séguier and D. Mercier. Audio-visual speech recognition: one pass learning with spiking neurons. In *International Conference on Artificial Neural Networks (ICANN)*, 2002.
- [14] Michael T. Chan. Hmm-based audio-visual speech recognition integrating geometric and appearance-based visual features. In *Workshop on Multimedia Signal Processing*, 2001.
- [15] Per-Erik Forssén. Window matching using sparse templates. In <http://www.isy.liu.se/cvl/ScOut/TechRep/>, 2001.
- [16] S. Pigeon. M2vts. In [www.tele.ucl.ac.be/PROJECTS/M2VTS/m2fdb.html](http://www.tele.ucl.ac.be/PROJECTS/M2VTS/m2fdb.html), 1996.