Automatic Outer Lip Contour Extraction in Facial Images

Juan-Bernardo Gómez-Mendoza, Flavio Prieto
Universidad Nacional de Colombia
Sedes Manizales y Bogotá. Colombia.
e-mail: jbgomezm@unal.edu.co,
faprietoo@unal.edu.co

Juan-Bernardo Gómez-Mendoza, Tanneguy Redarce
Laboratoire Ampère (UMR 5005). INSA de Lyon.
69621 Villeurbanne, France.
e-mail: juan-bernardo.gomez-mendoza@insa-lyon.fr,
tanneguy.redarce@insa-lyon.fr

Abstract— In this paper a new method for outer lip contour approximation in facial images is presented. The method is inspired in the *Jumping Snakes* [1,2], improving the later in terms of snake final position refinement. This new method is aimed to generate an approximated outer lip contour that properly encloses the whole of the mouth region present in the image. The problem of automatic parameter selection and seed positioning is also tackled.

Keywords-active contours, lip contour extraction.

I. INTRODUCTION

In recent years lip contour extraction has recovered attention of research community, mainly due to its great potential in human machine interface and communication systems development. Most of the lip contour extraction techniques have been designed for audio-visual speech recognition (AVSR) [2,3,4,5], and in some cases extended to speaker identification [6]. In this task only a few landmarks suffice to estimate key features, as in the case of MPEG-4 facial animation parameters [7]. Moreover, mouth position does not vary considerably from the resting position while speaking, thus enabling strong geometrical constraints to be applied. Some of the aforementioned constraints restrict techniques' usage to images with quasi-symmetrical mouth positions presenting clear cupid arc definition.

Over-specialization of lip contour extraction methods for the task of AVSR makes them inappropriate in cases where mouth appearance may change subjected to pose changes and/or specific lip malformations. This also excludes them when an accurate description of the whole mouth contour is needed, as in general gesture detection or accurate mouth structures segmentation. As a given example, some typical mouth positions like wide open mouth fall outside the realm of normal speaking expected mouth positions.

Jumping Snakes technique is, however, a method that can be easily extended in order to cope with these limitations. It was introduced in [1] and later refined in [2], and its goal is to find a set of six feature points and six curves – four third order polynomials and two line segments – which represents lip outer contour. Those features are tracked in video sequences using a modified version of Lucas-Kanade optical flow estimation. Initial upper and lower lip contour approximation is performed by maximizing the gradient flow of a color transformation which is designed to enhance lips from skin and beard. The *Jumping Snake* algorithm has been used in AVSR applications like [3,4,5].

In order to extend *Jumping Snakes* aiming to obtain an accurate representation of the outer lip contour the curve fitting stage should be avoided. Instead, an incremental snake growing and refining methodology should be added. In this work, we introduce a modified version of the update iteration for upper lip snake, which uses the same principle of the original technique but also back-propagate its evolution in order to improve convergence speed and seed re-location. We also introduce a later refining stage that improves cupid's arc approximation and controls overall line segment length.

This document is presented as follows. Section II depicts the general flow of our method: in Section II-A the modified snake update technique for the algorithm presented in [1,2] is introduced. Section II-B shows how to refine the final position of the snake. Section II-C treats the automatic selection of the algorithm parameter set. Section III describes the test framework and presents the results. Finally, conclusions are drawn in Section IV.

II. AUTOMATIC OUTER LIP CONTOUR EXTRACTION

In this Section our methodology for automatic lip contour extraction is presented. In the following paragraphs the process to approximate both upper and lower lip contour is described. Sections II-A, II-B and II-C provide the insights of each step in the proposed methodology.

First, upper lip snake is initialized following the guidelines established in Section II-A. Then, points are added at both left and right side of the snake using the principle stated in [1,2], until reaching mouth Region of Interest (RoI) limits.

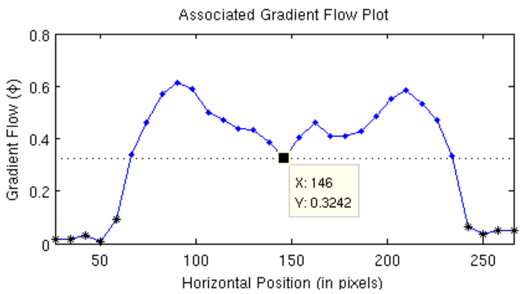
Associated gradient flow (φ) value is used in order to select those points that should be trimmed from the

This work was partially supported by the ECOS-Nord/Colciencias/ICFES/ICETEX Franco-Colombian Program; and by the "Identificación de posturas labiales en pacientes con labio y/o paladar hendido corregido" research project, granted by Colciencias Program No. 405 of 2007.

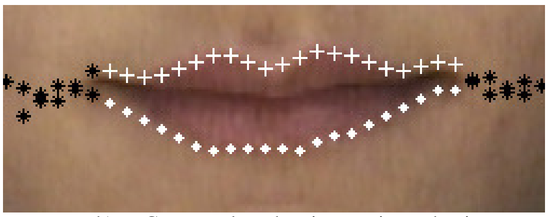
snake. Associated gradient flow for each upper lip snake point ($\mathbf{p}_i$) is computed as

$$\varphi_i = \begin{cases} \displaystyle\int_{\mathbf{p}_i\mathbf{p}_{i+1}} \frac{[\nabla(ph-L)]\cdot \mathbf{dn}_l}{|\mathbf{p}_i\mathbf{p}_{i+1}|}, \mathrm{i}\in[1,N] \\ \displaystyle\int_{\mathbf{p}_{i-1}\mathbf{p}_i} \frac{[\nabla(ph-L)]\cdot \mathbf{dn}_r}{|\mathbf{p}_{i-1}\mathbf{p}_i|}, \mathrm{i}\in[N+2,2N+1] \end{cases} \qquad (1)$$

where $2N+1$ is the total number of points in the snake, $N+1$ standing for the seed point index in the snake points set; *ph* represent the Pseudo-hue component values of the pixels in the line segment; *L* the corresponding Luminance value; and $\mathbf{dn}_l$ and $\mathbf{dn}_r$ are normalized vectors which lie perpendicular to line segments conformed by the points located at left or right side of the seed with the seed itself. Gradient flow usually increases in value when approaching mouth corners, and then decreases rapidly when points are added outside mouth contour. Points located outside the region enclosed by the two maximums in associated gradient flow for whose associated gradient values fall behind the minimum inside that region should be trimmed. This can be easily seen in Figure 1. Upper lip snake can be optionally refined following the procedure in Section II-B.



a) Associated gradient flow plot.



b) Contour plots showing untrimmed points.

Figure 1. Associated gradient flow use for point trimming.

Initial lower lip snake is approximated using the same methodology as in [2]. As for the upper snake case, points are added until reaching RoI limits. Finally, points outside horizontal leftmost and rightmost limits of the upper snake are trimmed from the snake.

A. *The Backward Jumping Snake*

The *Jumping Snake* proposed by Eveno *et al.* [1,2] is a simplified form of active contour that properly approximates the outer lip contour in color images. Pseudo-Hue (*ph*) and luminance (*L*) color components are used in order to compute the gradient flow that controls the snake evolution. *N* Points are added at both left and right side of the seed incrementally, preserving a horizontal distance (Δ) and aiming to maximize the normalized gradient flow of *ph-L* passing through each generated line segment. At the end of the iteration, vertical seed position is re-computed as the mean vertical position of the *N* added points that led the creation of the line segments with highest gradient flows (for further details please refer to [2]). In next iterations only seed position is retained, while the rest of the point set is computed once again using the same principle.

Since points are added incrementally last added points lie closer to the actual outer lip contour than the rest. However, their positions are discarded in next algorithm's iterations – only a vague reflection of their position is kept through seed re-compute. In order to benefit from this fact we propose to back-propagate the algorithm prior to seed recalculation. This propagation follows the same update principle stated in [2], but in this case starting from both left and right snake limit points towards the seed.

In order to update seed position we propose to change to a bilateral update approach. In this case, the goal is to maximize the gradient flow passing through both seed's associated left and right line segments, ensuring that overall line segment size is kept bounded. The goal can be translated into maximizing the objective function

$$\varphi_{N+1} = \int_{\mathbf{p}_N\mathbf{p}_{N+1}} \frac{[\nabla(ph-L)]\mathbf{dn}_-}{|\mathbf{p}_N\mathbf{p}_{N+1}|} + \int_{\mathbf{p}_{N+1}\mathbf{p}_{N+2}} \frac{[\nabla(ph-L)]\mathbf{dn}_+}{|\mathbf{p}_{N+1}\mathbf{p}_{N+2}|} \qquad (2)$$

subject to

$$|\mathbf{p}_N\mathbf{p}_{N+1}|<\gamma \qquad (3)$$

$$|\mathbf{p}_{N+1}\mathbf{p}_{N+2}|<\gamma \qquad (4)$$

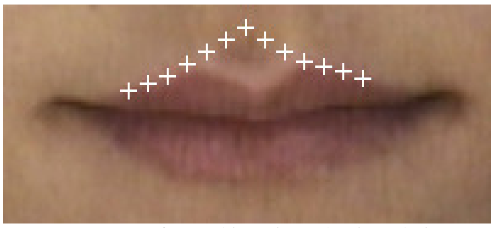
Where $\mathbf{p}_{N+1}$ is the seed, φ_{N+1} is the gradient flow associated to point $\mathbf{p}_{N+1}$, $\mathbf{dn}_-$ and $\mathbf{dn}_+$ are the normalized gradient vectors of left and right line segments, and γ is a parameter that controls the average line segment length. The value of γ should lie inside the range (Δ, 2Δ-5]. Smaller γ values will lead to smoother lip contour, but it increases the fitting error around corners in cupid's arc.

This method also ensures that seed final position lies on the actual lip contour, in contrast to [2]. Therefore, the seed position can be used as any other snake point for later procedures.

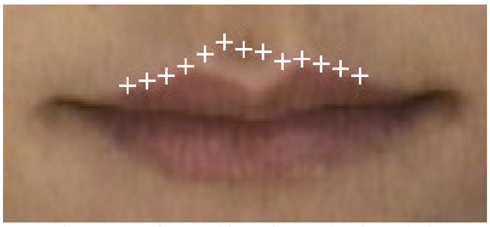
Some result of comparing two consecutive traditional *Jumping Snake* iterations against one forward-backward *Jumping Snake* iteration can be seen in Figure 2. For the image in the example, four forward iterations were needed to achieve full convergence, while only one forward-backward iteration was needed for the same purpose. The introduction of the backward iteration improves dramatically the snake convergence speed to outer upper lip contour.

B. *Upper Lip Snake Position Refinement*

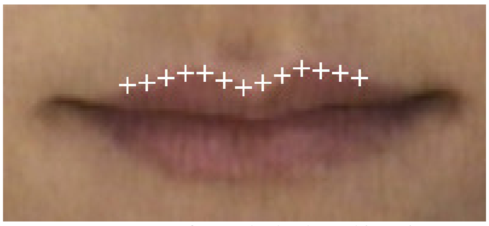
In order to refine contour approximation and snake positioning when looking for upper lip contour we propose to apply same methodology as used for seed re-calculation in Section II-A. The general form of the flow function in (2) is



a) One forward iteration (classic technique).



b) Two forward iterations (classic technique).



c) One forward – backward iteration.

Figure 2. Snake evolution using Jumping Snakes with forward and forward-backward iterations, using the same parameter set in both cases.

$$\varphi_i = \int_{\mathbf{p}_{i-1}\mathbf{p}_i} \frac{[\nabla(ph-L)]\mathbf{dn}_-}{|\mathbf{p}_{i-1}\mathbf{p}_i|} + \int_{\mathbf{p}_i\mathbf{p}_{i+1}} \frac{[\nabla(ph-L)]\mathbf{dn}_+}{|\mathbf{p}_i\mathbf{p}_{i+1}|} \qquad (5)$$

subject to

$$|\mathbf{p}_{i-1}\mathbf{p}_i| < \gamma \qquad (6)$$

$$|\mathbf{p}_i\mathbf{p}_{i+1}| < \gamma \qquad (7)$$

The procedure is performed in a backward-forward manner, starting from the snake leftmost and rightmost pints towards the seed, and then all the way back to the extremes. Actual snake endpoints are not re-computed since they only have one associated line segment each.

C. Automatic Parameter Selection

In order to avoid manual seed selection, a RoI based approach is proposed. Mouth RoI can be automatically detected by using extensive search [8,9], or by template matching using, e.g., Haar transform [10], Active Appearance Models [11] or AdaBoost-based techniques [12].

Mouth RoI provides an approximate bounding box of the mouth, which can be used to compute the parameter Δ, and the location of upper and lower seeds. Upper seed position is chosen as the middle point of the upper row of the RoI. Also, even though method in [2] gives a first approach on how to choose lower lip contour seed, it can be alternatively chosen as the midpoint of the lower row of the RoI. This proved to be helpful for images where mouth is open.

An acceptable value for Δ can be chosen by dividing the RoI width by $4N$, whenever that operation leads to a value bigger than five pixels. Otherwise, decreasing the value of N is recommended. When is chosen Δ to be smaller than five pixels the flow through each line segment is highly unstable thus introducing undesired local minima in (1) and (2).

Pseudo-hue component usually presents noticeable low-scale noise that interferes with the snake evolution. A good strategy to avoid falling in early local minimums is to smooth the response of the *ph-L* component before computing the gradient. The application of a smoothing filter spreads the gradient flow through unreliable gradient areas, but it can also make the snake jump inside the lips. One way to overcome both problems is by starting with a relatively big smoothing factor and decreasing it along iterations. Smoothing filter size, as well as other algorithm parameters discussed before, should be carefully chosen depending on the mouth width and the noise level in the image.

III. Tests and Results

For test purposes, a set of 20 images taken from five different subjects were used. The image set primarily contains mouth rest positions, along with some closed mouth gestures. For every image, algorithm initialization was carried out by running one *Jumping Snake* forward – backward iteration as explained in Section II-A. Approximate mouth RoIs were manually selected, and the parameter set was estimated following the guidelines in Section II-C.

For eighteen of the twenty images outer lip contour was in appearance well fitted by the algorithm. Lip contour approximation through line segments proved to enclose more than 98% of pixels labeled as belonging to mouth region, while retaining less than 1% of non-mouth pixels. Error was mainly concentrated around lower lip contour and near the mouth corners.

In the remaining two cases lower lip contour was not properly fitted, mainly due to the presence of shadows below the lip. Even when *ph* is somewhat immune to illumination artifacts, low dynamic response of the sensor corresponding to dark areas in the scene makes noise effect noticeable in gradient calculation.

Figure 3 shows an example of applying the proposed methodology. Notice that final snake position encloses the whole mouth region in the image. Also, line segments preserve an approximate length all over the resulting snake. Snake final position is well adjusted to cupid's arc, while some problems still arise when approaching mouth corners.

IV. Conclusion

In this paper a novel strategy for automatic outer lip contour extraction in images is presented. The main objective of the technique is to enclose the majority of mouth information present in the image, discarding at same time non-mouth regions. The method proved to be in most cases accurate, enclosing more than 98% of mouth region within the resulting snakes.

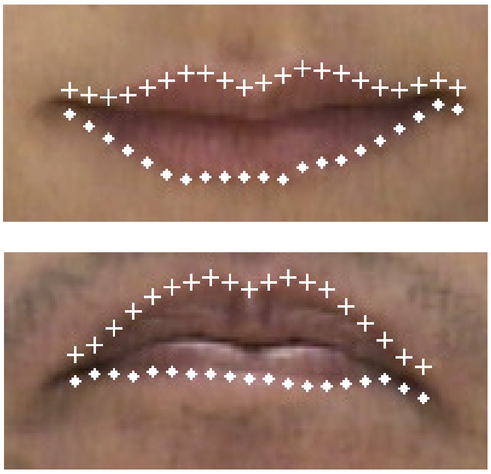


Figure 3. Examples of outer lip contour approximation.

The strategy is inspired in the *Jumping Snakes* presented in [1,2], including some improvements over the later in terms of contour accuracy and convergence speed. Both features are tackled by the introduction of a backward phase in traditional snake update iteration (see Section II-A), along with a final position refining stage (Section II-B).

In some cases, cupid's arc fitting is slightly improved when applying the position refining process described in Section II-B. However, since forward-backward iterating the Jumping Snake algorithm is accurate enough to approximate lip contour, it is up to user to decide whether or not to perform snake point refining in a specific application.

The method's low computational complexity makes it usable for developing real-time video applications. Most of the reckoning lies in smoothing *ph* and *ph-L* color components prior to gradient flow calculation. Those smoothing operations can nowadays be implemented at GPU level.

REFERENCES

[1] N. Eveno, A. Caplier, and P.Y. Coulon. "A new color transformation for lip segmentation," In. Proc. of IEEE Fourth Workshop on Multimedia Signal Processing, october 3-5, 2001, pp. 3-8.

[2] N. Eveno, A. Caplier, and P. Coulon. "Accurate and quasi-automatic lip tracking," IEEE Trans. on Circuits and Systems For Video Technology, vol. 14, no. 5, pp. 706-715, 2004.

[3] H.E. Çetingül, Y. Yemez, and A.M. Tekalp. "Robust lip-motion features for speaker identification," In. Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, march 18-23, 2005, pp 509-512.

[4] H.E. Çetingül, Y. Yemez, E. Erzin, and A.M. Telkap. "Discriminative analysis of lip motion features for speaker identification and speech-reading," IEEE Trans. on Image Processing, vol. 15, no. 10, pp. 2879-2891.

[5] S. Alizadeh, R. Boostani, and V. Asadpour. "Lip feature extraction and reduction for HMM-based visual speech recognition systems," In. Proc. of International Conference on Signal Processing, october 26-29, 2008, pp. 561-564.

[6] B.J. Borgström, and A. Alwan. "A low-complexity parabolic lip contour model with speaker normalization for high-level feature extraction in noise-robust audiovisual speech recognition," IEEE Trans. on Systems, Man and Cybernetics, vol. 38, no. 6, pp. 1273-1280, 2008.

[7] A. Ceballos, J.B. Gómez, F. Prieto, and T. Redarce. "Robot command interface using an audio-visual speech recognition system," In. Proc. of Congreso Iberoamericano de Reconocimiento de Patrones, november 15-18,2009, pp. 869-876.

[8] J.B. Gómez, F. Prieto, and T. Redarce. "Lips movement segmentation and features extraction in real-time," In. Proc. of International Conference on Industrial Electronics, Technology & Automation, december 4-14, 2006, pp. 205-210.

[9] J.B. Gómez, J.E. Hernández, F. Prieto, and T. Redarce. "Real-time robot manipulation using mouth gestures in facial video sequences," In Proc. of 2nd. International Conference on Brain, Vision and Artificial Intelligence, october 10-12, 2007, pp. 224-233.

[10] T. Wang, and P. Shi. "Facial components detection with boosting and geometric constraints," In Proc. of 18th International Conference on Pattern Recongnition, august 20-24, 2006, pp. 446-449.

[11] I.Matthews, T.F. Cootes, J.A. Bangham, S. Cox, and R. Harvey. "Extraction of visual features for lipreading," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 24, no. 2, pp. 198-213, 2002.

[12] M. Saradadevi, and P. Bajaj. "Driver fatigue detection using mouth and yawning analysis," Int. Journal of Computer Science and Network Security, vol. 8, no. 6, pp. 183-188, 2008.