

# A NEW APPROACH TO TRACKING WITH ACTIVE CONTOURS

*Montse Pardàs 1,2 and Elisa Sayrol 2*

1 Bell Labs, Lucent Technologies, Murray Hill, NJ  
2 Universitat Politècnica de Catalunya, Barcelona, Spain

## ABSTRACT

This paper addresses the application of active contours or snakes for robust tracking of contours. Conventional snake approaches to tracking initialize the current frame snake with the snake obtained in the previous frame and then optimize this result considering only the current frame information. In our approach motion estimation is embedded in the energy minimization process of the snake. This is possible using a dynamic programming approach for this minimization and introducing a new term in the energy of the snake. In this way, larger displacements of the contour are allowed, and the tracking is more robust. This technique has been applied for tracking facial features such as the contour of the face, the mouth, the nostrils and the eyebrows.

## 1. INTRODUCTION

Tracking of contours has many applications such as human motion modeling, biomedical image analysis, surveillance or autonomous vehicle navigation. Great attention in the past years has been devoted to contour tracking by means of snake models. Snakes were first introduced by Kass et al. [6]. They proposed energy minimization as a framework where low-level information (such as image gradient or image intensity) can be combined with higher-level information (such as shape, continuity of the contour or user interactivity). In their original work the energy minimization problem was solved using a variational technique. In [1] Amini et al. proposed Dynamic Programming (DP) as a different solution to the minimization problem in order to overcome issues like optimality, numerical stability due to the usage of high order derivatives of the discrete data, convergence and enforcement of hard constraints. Other approaches to the minimization through DP are found in [4] and [5].

These approaches have been applied to the problem of contour tracking. In the original work by Kass et al. [6], active snakes were already used to track contours through image sequences. The initialization problem was solved by

using as initial position for the current frame, the optimal snake position found in the preceding frame. The same approach is followed in [7]. In both works it is claimed that the contour can be tracked along the sequence if there are no large deformations of this contour and if the frame rate is high enough. In [4] a similar approach was employed for tracking within the DP implementation.

Another method was proposed in [3]. In this case two new terms were introduced in the Energy functional to account for inter-frame constraints, one for the internal forces and one for the external ones. The first one introduced continuity of the snake in the time dimension while the second one tried to maintain unchanged the intensity value of every snaxel. The minimization was done with a multiscale DP algorithm. The fact of making it multiscale means that the solution is reached faster, although the optimum might not be reached. Besides, the intensity value of the particular snaxels does not preserve enough information to track a specific contour. Moreover, the time continuity restriction is not compatible with large deformation of the contours. Other modifications that adapt the basic algorithm to the tracking of contours can be found in [2] and [9].

All the previous approaches fail when the motion between frames is large or when there are deformations of the contour which lead the minimization algorithm towards other local minima. The method we propose tries to solve these problems by using a motion estimation procedure which results are introduced in the search for the minimum of the energy. For this aim, a DP approach similar to the one presented in [4] and [5] is used. However, only those pixels in the current frame that correspond to feasible displacements of snaxels in the previous frame are considered as snaxels candidates in the Dynamic Programming algorithm. Besides, a new energy term in the minimization procedure is introduced, based on the Motion Compensation error.

Section 2 covers the basic concepts of the snakes that will be used in our approach. Section 3 describes the implementation of the snakes by DP. In Section 4 we will describe how we introduce the motion estimation into the snakes algorithm. In Section 5 we will apply this algorithm

to the tracking of facial features. Finally, Section 6 will outline the advantages of this method.

## 2. SNAKES

In the discrete formulation of active contour models the contour to be tracked is represented as a set of snaxels  $v_i=(x_i, y_i)$  for  $i=0, \dots, N-1$ , where  $x_i$  and  $y_i$  are the  $x$  and  $y$  coordinates of the snaxel  $i$ , and its energy, which is going to be minimized, is defined by:

$$E_{snake}(v) = \sum_{i=0}^{N-1} (E_{int}(v_i) + E_{ext}(v_i)) \quad (1)$$

We use a discrete approximation of the second derivative to compute  $E_{int}$ .

$$E_{int} = |v_{ss}(s)| \approx |v_{i-1} - 2v_i + v_{i+1}| \quad (2)$$

This is an approximation to the curvature of the contour at snaxel  $i$ , if the snaxels are equidistant. We will force the snaxels to be equidistant when the snaxel is initialized in the first image.

Minimizing this energy will produce smooth curves. This is only appropriate for snaxels that are not corners of the contour. In the case of corners the energy has to be low when this second derivative is high. We use:

$$E_{int}(v_i) = [\beta |v_{i-1} - 2v_i + v_{i+1}| + (1-\beta)(B - |v_{i-1} - 2v_i + v_{i+1}|)] \quad (3)$$

where  $\beta$  is set to 1 if  $v_i$  is not a corner point and to 0 if it is.  $B$  represents the maximum value that the approximation to the second derivative can take.

The purpose of the term  $E_{ext}$  is to attract the snake to desired feature points or contours in the image. In this work we have used the mean value of the image ( $I(x, y)$ ) or its gradient, depending on the feature that is being tracked, along the contour from  $v_i$  to  $v_{i+j}$ . Thus, the  $E_{ext}$  at snaxel  $v_i$  will depend only on the position of the snaxels  $v_i$  and  $v_{i+1}$ . That is,

$$E_{ext}(v_i) = E_{contv_i, v_{i+1}} = f(I, v_i, v_{i+1}) \quad (4)$$

## 3. DYNAMIC PROGRAMMING

We will use the DP approach to minimize the energy in Eq. (1). Let us express the Energy of the snake remarking the dependencies of its terms:

$$E_{snake}(v) = \sum_{i=0}^{N-1} E_{int}(v_{i-1}, v_i, v_{i+1}) + E_{ext}(v_i, v_{i+1}) = \sum_{i=0}^{N-1} E_i(v_{i-1}, v_i, v_{i+1}) \quad (5)$$

Although snakes can be open or closed, the DP approach can be applied directly only to open snakes. In this case the limits of Eq. 5 are adjusted to 1 and  $N-2$  respectively.

Now, as described in [1], this energy can be minimized via discrete DP defining a two-element vector of state variables in the  $i$ th decision stage:  $(v_{i+1}, v_i)$ . The optimal value function is a function of two adjacent points on the

contour  $S_i(v_{i+1}, v_i)$ , and can be calculated, for every couple of possible positions for snaxels  $v_{i+1}$  and  $v_i$ , as:

$$S_i(v_{i+1}, v_i) = \min_{v_{i-1}} [S_{i-1}(v_i, v_{i-1}) + E_i(v_{i-1}, v_i, v_{i+1})] \quad (6)$$

$S_0(v_1, v_0)$  is initialized to  $E_{ext}(v_0, v_1)$  for every possible candidate pair  $(v_0, v_1)$  and from this,  $S_i$  can be computed iteratively from  $i=1$  up to  $i=N-2$  for every candidate position for  $v_i$ . The total energy of the snake will be

$$E_{snake}(v) = \min_{v_{N-1}} S_{N-2}(v_{N-1}, v_{N-2}) \quad (7)$$

Besides, we have to store at every step  $i$  a matrix which stores the position of  $v_{i-1}$  that minimizes Eq. (6), that is,

$$M_i(v_{i+1}, v_i) = v_{i-1} \quad \text{such that } v_{i-1} \text{ minimizes (6).}$$

By backtracking from the final energy of the snake and using matrix  $M_i$ , the optimal position for every snaxel can be found.

In the case of a closed contour the solution proposed in [4] is to impose the first and last snaxels to be the same, and fix it to a given candidate for this position. The application of the DP algorithm will produce the best result under this restriction. Then this initial and final snaxel is successively changed to all the possible candidates, and the one that produces a smaller energy will be selected. We use an approximation proposed in [5] that requires only two open contour optimization steps. They first find an open contour solution without any restriction at the end. Then the two points at the mid of this contour are taken as start and end points for the closed contour, and another optimization is run with these two points fixed. We will call this method the two-steps algorithm, as it implies two optimization steps.

## 4. INTRODUCING MOTION ESTIMATION

In the previous Section we have assumed that for every snaxel  $v_i$  there are a finite (and hopefully small) number of candidates, but we have omitted how to select these candidates. The computational complexity of each optimization step is  $O(nm^3)$ , where  $n$  is the number of snaxels and  $m$  the number of candidates for every snaxel. Thus, it is very important to maintain  $m$  low.

In [1] only a small neighborhood around the previous position of the snaxel was considered. However, the algorithm was iteratively applied starting from the obtained solution until there was no change in the total energy of the snake. This method has several disadvantages. First, like in the approaches which use variational techniques for the minimization, the snake can fall into a local minimum. Second, the computational time can be very high if the initialization is far from the minimum.

In [4] and [5] a different set of candidates is considered for every snaxel. In particular, [4] establishes uncertainty lists for the high curvature points and defines a search space between these uncertainty lists. In [5] the search

zone is defined with two initial concentric contours. Each contour point is constrained to lie on a line joining these two initial contours. This approach gives very good results if the two concentric contours that contain the expected contour are available and the contour being tracked is the absolute minima in this area. However, these concentric contours are not always available for tracking. In tracking applications the motion and deformations of the contour can be very large from one image to the successive image. Thus, using a strip around the contour obtained in the previous image does not produce good results. Besides, we might be interested in tracking a local minimum.

The solution we propose uses motion estimation in order to select the search space for every snaxel.

A small region around every snaxel is selected as basis for the motion estimation. The shape of this region is rectangular and its size is application dependent. However, the region should be small enough so that its motion can be approximated by a translational motion. The compensation error for all the possible displacements  $(dx, dy)$  of the block in a given range is computed as:

$$MCE_{v_{i0}}(dx, dy) = \sum_{j=-Ry}^{j=Ry} \sum_{i=-Rx}^{i=Rx} |I(x_0 - i, y_0 - j) - I(x_0 - i - dx, y_0 - j - dy)|^2 \quad (8)$$

Being  $(x_0, y_0)$  the  $x$  and  $y$  coordinates of the snaxel  $v_i$  in the previous frame, which we have called  $v_{i0}$ . The region under consideration is centered at the snaxel and with size  $2Rx$  in the horizontal dimension and  $2Ry$  in the vertical dimension.

The range for  $(dx, dy)$  determines the maximum displacement that a snaxel can suffer. The matrix  $MCE_{v_{i0}}(dx, dy)$  is stored for every snaxel, and the  $M$  best results are selected as possible new locations for snaxel  $v_i$ .

The DP algorithm described in Section 3 is now applied considering as candidates for every snaxel those locations that have been selected by the motion estimation algorithm.

Besides, we have introduced a new term in the external Energy function in Eq. (4) that improves the tracking capabilities of the algorithm. It is a memory term which corresponds to the compensation error obtained in the motion estimation. In this way preference is given to those positions with the smaller compensation error. That is, the energy will be lower in those positions which texture is most similar to the texture around the position of the corresponding snaxel in the previous frame. This introduces the possibility to track smooth contours placed near strong contours. For instance, let us consider the tracking of the lips. In general, the outer contour of the lips is not a very clear contour. When the mouth is slightly open, the inner contour is much stronger and if a global search was made, the snake would change its position to occupy this other contour. However, by restricting the search to those positions with a similar texture to the position of the snaxel in the previous image and by

introducing this new term in the energy we are able to track the outer contour even with strong motion of the lips. Thus, the external energy will be composed of two terms, the one which makes the snake be attracted by contours of the image, and this new one. Therefore, the expression for the external energy will be:

$$E_{ext}(v_i) = \gamma E_{cont_{v_{i-1}-v_i}} + (1 - \gamma) MCE_{v_{i0}}(v_i) \quad (9)$$

The constant  $\gamma$  can be set depending on the strength of the contour that is being tracked. If it is a strong contour  $\gamma$  is chosen close to 1. Otherwise, more importance is given to the Motion Compensation Error term.

## 5. TRACKING FACIAL FEATURES

We have applied the previously described algorithm to facial feature tracking. The features of the face that are going to be followed are the contour of the face, the nostrils, the mouth and the eyebrows. Slightly adapted snakes are used in each case, that is, closed snakes without corner points for the contour of the face, closed snakes with two corner points for the lips and open snake for the line that connects the nostrils.

## 6. CONCLUSIONS

We have presented a contribution to contour tracking using snakes. The objective is to take advantage of motion estimation in order to select a set of candidate positions for a given snaxel in the current frame, according to the information that we have for this snaxel in the previous frame. This selection of candidate positions is compatible with a Dynamic Programming implementation of the active contours algorithm. Furthermore, we have introduced a new term in the computation of the Energy of the snake, in order to introduce motion information. This approach reduces the computational cost with respect to other DP implementations, as only a small number of pixels have to be considered as candidates for every snaxel. It allows to track contours which are not the global minimum, as the snake is actually tracking the texture around the snaxel. It is also much more robust to deformations and large motion of the contours than the classical approaches. Furthermore, all the advantages of the DP approach are preserved, such as the possibility to enforce hard restrictions in the contour.

## 7. REFERENCES

- [1] A. Amini, T. Weymouth and R. Jain, "Using Dynamic Programming for Solving Variational Problems in Vision", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 9, September 1990.

[2] P. Delagnes, J. Benois, D. Barba, "Active contours approach to object tracking in image sequences with complex background", *Pattern Recognition Letters*, Vol. 16, pp. 171-178, February 1995.

[3] K. Fujimura, N. Yokoya and K. Yamamoto, "Motion tracking of deformable objects by active contour models using multiscale dynamic programming", in *Proceedings of the 11th ICPR International Conference on Pattern Recognition*, Vol. 1, pp. 83-86, The Netherlands, 1992.

[4] D. Geiger, A. Gupta, L. Costa and J. Vlontzos, "Dynamic Programming for Detection, Tracking, and Matching Deformable Contours", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 3, March 1995.

[5] S. Gunn and M. Nixon, "Global and Local Active Contours for Head Boundary Extraction", *International Journal of Computer Vision* 30(1), pp. 43-54, 1998.

[6] M. Kass, A. Witkin and D. Terzopoulos, "Snakes: Active Contour Models", *International Journal of Computer Vision*, Vol. 1, No. 4, pp. 321-331, 1988.

[7] F. Leymarie and M. D. Levine, "Tracking Deformable Objects in the Plane Using an Active Contour Model", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 6, June 1993.

[7] M2VTS <http://www.tele.ucl.ac.be/M2VTS>

[8] M. Pardas, "Automatic Face Analysis for Model Calibration", in *Proceedings of the II IWSNHC*, Santorini, Greece, September 1999.

[9] C. Vieren, F. Cabestaing, J. Postaire, "Catching moving objects with snakes for motion tracking", *Pattern Recognition Letters*, Vol. 16, pp. 679-685, 1995.

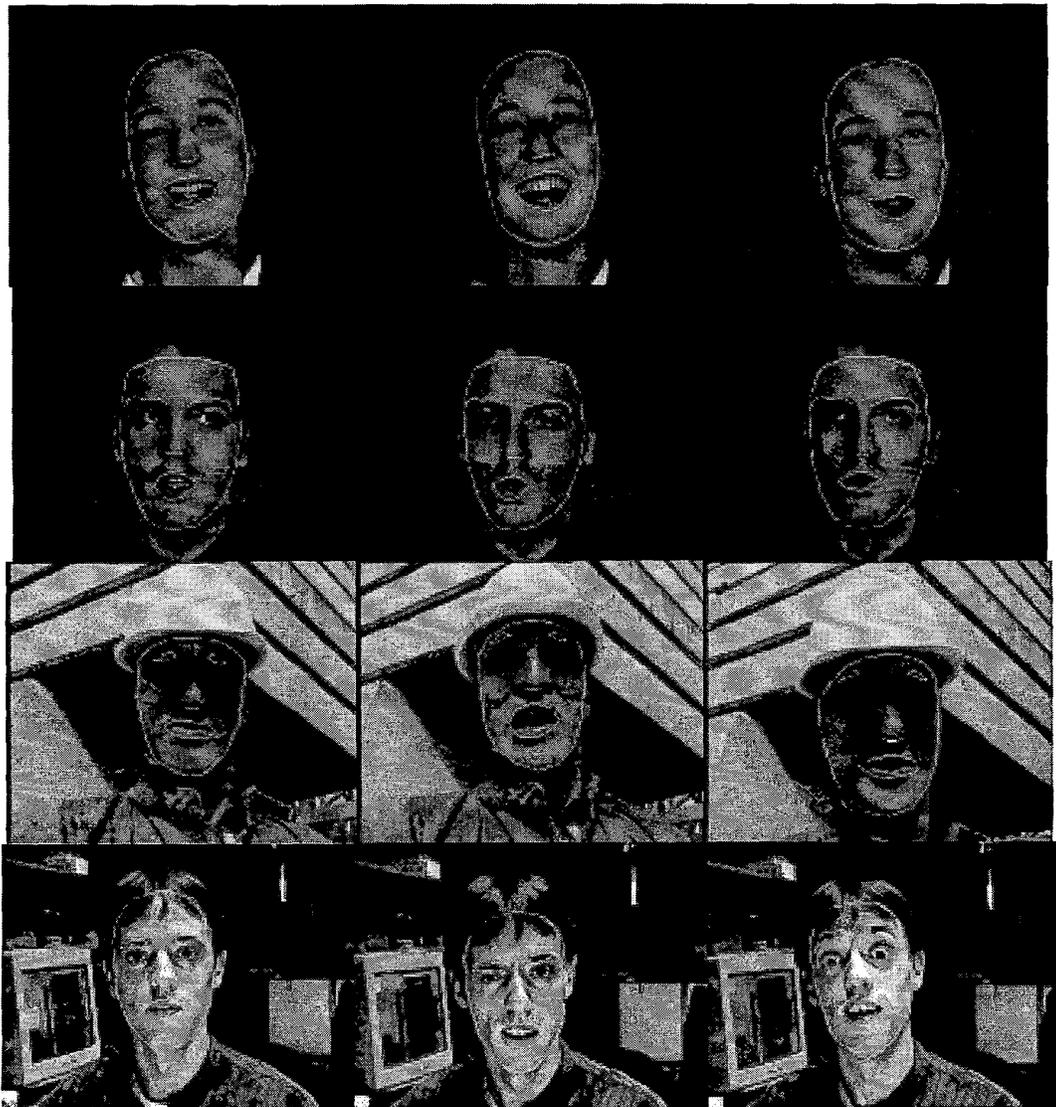


Figure 1. Examples of the tracking of facial features.