Towards Autonomy in Active Contour Models

Marie-Odile BERGER
INRIA Lorraine / CRIN
B.P. 239 54506 Vandœuvre CEDEX , France
email: berger@loria.crin.fr

Roger MOHR
LIFIA
46, av Felix Viallet 38031 Grenoble Cedex, France
mohr@lifia.imag.fr

Abstract

The Active Contour Models provide a unified treatment involving the minimization of energy functionals for numerous computer vision problems. Nevertheless, numerical problems arise during minimization of the functional and the results depend heavily on the initialization...
The strengths and also the drawbacks of this kind of method are described in this paper and we show the absolute necessity to have at disposal an assessment criterion of the solutions. This leads us to propose a method called "snake growing", based on successive lengthenings of the snake. The strength of this approach is that at each stage, we are in good convergence conditions and that it allows us to get rid of initialization problems.

1 Introduction

One of the main problems with standard edge detectors is their inability to find the most salient edges. Too many or too few contours are in fact detected. Recently, new algorithms based on Active Contour Models have been proposed [Kas 88], [Zuc 88]. An active contour is described as an elastic line which is slithering under forces created by some kind of energy. Local minima of this energy constitute the searched image features.

Methods using minimization principles have been described in [Mon 71] and particularly by Kass, Witkin and Terzopoulos [Kas 88]. An active Contour is called a snake. The energy functional always consists in the sum of two terms:

The **Internal energy** E_{int}, which describes the properties of the elastic line $C = (x(t), y(t))$. In the Snake system, the curve model is a controlled continuity spline [Ter 86]. E_{int} is written

$$E_{int} = \int_C (\alpha|v'|^2 + \beta|v''|^2) \tag{1}$$

(Parameters α and β influence the elasticity or stiffness of the curve)

The **External energy** E_{ext} depends on the features which are searched for in the image (dark lines, white lines, edges, termination of line segments ...). For instance, in edge detection, E_{ext} is defined as

$$E_{ext} = \int_C (-|\nabla I(v(t))|)dt \tag{2}$$

where $\nabla I(v(t))$ is the gradient of the intensity image. Each of the terms can be weighted. We then have to minimize the global energy $E = \lambda E_{int} + E_{ext}$;

Such a method is a way to regularize the edge detection problem, which is ill posed [Pog 85]. The curves for which $|\nabla I(v(t))|$ is maximal are searched for in the restricted class of splines under tension.

This class of edge detector is of great interest in some situations: accurate localization of edges near a given initialization, detection of subjective contours, motion tracking....Nevertheless, the minimization of the energy functional E is still problematic. For instance :

- How can the numerous parameters be calculated ? (the weight of various energies, the regularization parameter, the parameter of the numerical minimization method used to minimize E).
- Numerical problems: high degree derivatives appear while solving Euler equations to minimize E and the algorithm is not numerically stable.
- A judicious initialization is required.

Part of those instability problems are actually due to the fact that several minima of functional E can be found in the searching area. Section 2 comes back to numerical problems and section 3 deals with the estimation of the quality of the numerical solution. Finally, we propose in the last two sections a method whose main idea is to be always in good convergence conditions and which frees us from initialization drawbacks.

2 Study of the method

The resolution method proposed in [Kas 88] is studied here. Other methods have been proposed [Ami 88] using dynamic programming, but we don't deal with them.

2.1 An ill posed problem

A problem is well posed in the sense of Hadamard when its solution is unique and depends continuously on the initial data. More precisely, in accordance with Tikhonov [Tik 76], the problem of finding a solution z from initial data u is well posed if

1. For all u, a unique solution $z = R(u)$ exists.

2. The problem is stable :

   Given any $\epsilon > 0$, we can find δ so that $d(u_1, u_2) < \delta$ in such a way that $d(z_1, z_2) < \epsilon$.

   This means that a small perturbation of the initial data induce a small variation of the solution.

The ill posed problem notion is fundamental because it deals with the ability of using numerical methods from initial data which are generally roughly known. What is the state of the present problem ?

Contours in the image are local minima of the energy functional E. Those minima are determined with variational calculus using the Euler equations :

$$
\begin{aligned}
-\alpha x'' + \beta x^{iv} - \frac{\partial |\nabla(x(t), y(t))|}{\partial x} &= 0 \\
-\alpha y'' + \beta y^{iv} - \frac{\partial |\nabla(x(t), y(t))|}{\partial y} &= 0
\end{aligned} \tag{3}
$$

But we must keep in mind that solutions of (3) are only stationary points and not always minima. An assessment criterion of numerical solutions is therefore required.

In practice, we don't want to determine all solutions, but only edges in the vicinity of the initial curve S_0 given by the user. The snake S_0 deforms itself into conformity with the nearest salient contour.

But this problem is ill posed. When several contours exist in the vicinity of S_0, the solution is not unique (two different curves can yield the same energy E). Furthermore, the solution does not always depend continuously on the initial data. For instance, let us consider the case of two rectilinear contours $[A, B]$ and $[A, C]$, and let D be the bisector of (B, A, C). An initialization by a segment containing A and making an angle $\theta > 0$ with D will be attracted by $[A, C]$ but will be attracted by $[A, B]$ if θ is negative, whatever angle we take.

When S_0 cuts several edges or lies in the neighborhood of a junction, the active contour is attracted by several contours at the same time. The numerical method is then instable or stabilizes numerically towards a position which does not fit any contour.

Those remarks bring an obvious fact to light: we must **estimate the quality** of the numerical solution. Often, only part of the resulting curve C is truly a contour. This case particularly arises when open contours are searched for; numerical calculus is then interrupted (because the solution doesn't move any more) whereas the resulting solution is not a local minima. This case also arises when a curve stabilizes on several edges at the same time. Some curves $C_1, ..C_n$ which are parts of the contour can then be extracted from C. We will see in section 4 how those extracted curves can be used.

2.2 Numerical study

The Euler equations (3) are solved iteratively in the following way:

The problem is discretized and the curve C is represented as the set of equidistant points $(v_i = (x_i, y_i))_{0 \leq i < N})$. Approximating the derivatives with finite differences gives rise to the equation

$$
\begin{aligned}
&\alpha_i(v_i - v_{i-1}) - \alpha_{i-1}(v_{i+1} - v_i) \\
&+[\beta_{i-1}[v_{i-2} - 2v_{i-1} + v_i] - 2\beta_i[v_{i-1} - 2v_i + v_{i+1}] \\
&+\beta_{i+1}[v_i - 2v_{i+1} + v_{i+2}]] + (f_x(i), f_y(i)) = 0
\end{aligned} \tag{4}
$$

This can be written in matrix form as

$$
\begin{aligned}
AX + f_x(x, y) &= 0 \\
AY + f_y(x, y) &= 0
\end{aligned} \tag{5}
$$

where A is a pentadiagonal matrix whose band is $[\beta_{i-1}; -\alpha_i - 2\beta i - 1 - 2\beta_i; (\alpha_i + \alpha_{i+1}) + \beta_{i-1} + 4\beta_i + \beta_{i+1}; -\alpha_{i+1} - 2\beta_i - 2\beta_{i+1}; \beta_{i+1}]$

This system is solved iteratively using

$$
\begin{aligned}
X^t &= (A + \gamma I)^{-1}(\gamma X^{t-1} - f_x(X^{t-1}, Y^{t-1})) \\
Y^t &= (A + \gamma I)^{-1}(\gamma X^{t-1} - f_y(X^{t-1}, Y^{t-1}))
\end{aligned} \tag{6}
$$

The parameter γ determines in fact the rate of convergence of the process. When γ is small, the snake moves quickly whereas the process converges more slowly when γ is great. A good determination of γ is essential in order to avoid oscillations of the sequence (X^t, Y^t).

Intuitively, it is advised to choose a little value of γ when the process starts (because the curve is far from the solution) and a great value when the snake is close to the contour.

In [Fua 89], P. Fua and Y. Leclerc proposed an adaptive procedure for computing γ. γ is chosen so that the average displacement of the points (X_i^t, Y_i^t) remains equal to a given value Δ. We then obtain

$$
\gamma = \frac{1}{\sqrt{n}\Delta} |\frac{\partial E}{\partial X}| \tag{7}
$$

But this value is global. It is well adapted when all the curve points have a similar behavior but not in the case where parts of the curve are close to a contour whereas the remainder is very far, as in figure 1.

It's the reason why we propose to split up the curves into several parts on which points have a similar behavior. We can then use adaptive determination of γ on each part.

For instance, in figure 1, we cut C in two parts on which different evolutions of γ are used.

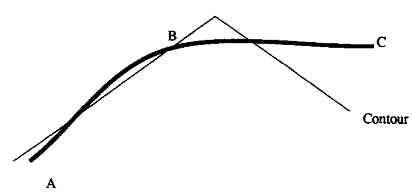

Figure 1: A non homogeneous behavior.

3 Estimation of the result

We have seen previously that numerical solutions may be a stationary point of E or even a point where the sequence (X^t, Y^t) doesn't move any more. In the first case, (X, Y) is a solution of the Euler equations but is not a minimum of E whereas, in the second case, the numerical algorithm is stopped while the limit is not reached yet. In this case, only parts of the curve can be considered as contours. The quality of the numerical solution must then be estimated.

We can use the general definition of an edge [Fua 89]: an edge is a curve C whose points have a gradient magnitude that is maximal in the direction normal to the curve. Points along the curve then satisfy

$$\frac{\partial|\nabla I(v(s))|}{\partial n(v(s))} = 0 \tag{8}$$

where $n(v(s))$ is the normal vector at the point having the arc length s. We can then compute a set of curves C_i extracted from C which can be considered as contours within some tolerance. Only curves whose length is great enough are retained.

Such a criterion is not easy to calculate because we only know the gradient at points whose coordinates are integer. Moreover, it is not really necessary to know if C is exactly a contour but only to determine if C is close enough to a contour. Actually, when using an adaptive value of γ, we use as initialization the curve which had been determined at the precedent stage. The contour position becomes finer when γ increases. Therefore, we only have to know if C lies in the vicinity of a contour in order to adapt α and we can use the following criterion: compare the average M_{gra} of the gradient on the curve C with the average M_{neigh} of gradient in a neighborhood of C. If $M_{gra} > M_{neigh}$ within some tolerance, a contour certainly lies in the considered neighborhood. The snake assessment gives rise to a sequence of contours $Cont_1, ..., Cont_n$ extracted from C whose lengths are sometimes small. We will see in the next section a method to infer longer contours from $Cont_1, ..., Cont_n$.

4 Snake growing

We propose in this section a method to get rid of initialization and convergence problems based on a **local** study. We first propose a method called "snake growing" which permits to build a snake incrementally. We then suggest some improvements to take into account contours which present great curvature variations.

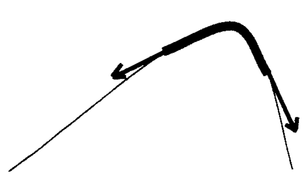

Figure 2: Lengthening in tangent direction.

4.1 Initialization problems

Given numerical and convergence problems described previously, it seems important to dispose of a correct initialization S_0. The following conditions must generally be satisfied :

- S_0 must cut the contour in at least one point, because the convergence process starts with the sections of initialization and contour.
- S_0 must have a direction close enough to the contour which will be detected.
- S_0 must lie in the influence area of at most one contour.

In the particular case of closed snakes, the elastic line tends to retract itself. Thus, an exterior initialization permits to detect contours even when the initialization is far from the contour.

4.2 Snake growing

The "Snake growing" method is based on the following remark: when S_0 lies in the vicinity of the contour, the iterative method will converge quickly, especially if S_0 is short. We therefore build a sequence $S_1, .., S_n$ of snakes which will yield contours $Cont_1, .., Cont_n$ whose lengths are increasing. Since only smoothed edges are searched, it seems natural to take as S_{i+1} the curve inferred from $Cont_i$ by lengthening its extremities in the direction of the tangents as shown in (Fig.2).

We can then solve iteratively the Euler equations with S_{i+1} which is a good initialization by construction. We then have to estimate the quality of the resulting curve C_{i+1}. Nothing can ensure that the entire curve C_{i+1} is a contour, particularly when a great curvature variation occurs at the extremities. The estimation procedure, described below, allows us to determine $Cont_{i+1}$ from C_{i+1}.

The growing algorithm can be written as:

- *start from a curve S_0 close to a contour,*
- *run the snake algorithm with S_0 as initialization, which yields C_0.*
- *While lengthening is possible, do:*

(build a sequence C_i of contours whose lengths increase)

- *lengthen C_i in the tangent direction to have the initialization curve S_i.*
- *run the traditional algorithm which converges towards Cl_{i+1}*
- *assessment of the curve Cl_{i+1} which yields C_{i+1} and the result quality.*

The growth of the length can be a given value l or can be an adaptive value inferred from previous results. For instance, when the convergence is fast at a stage, we can suppose that the tangent direction is close to the contour direction; l is then increased.

4.3 Critical study

4.3.1 Advantages

The major advantage of snake growing is that at each step, we are in conditions such that the iterative method converges in a satisfying manner. More precisely:

- The only time when initialization is far from the contour is at the beginning of the process. But there are only few discretization points (S_0 is small) and the cost is very weak.

- At the following steps, the snake is quite near a unique contour, except at the extremities. The numerical solution is then stable.

We can also notice that this method allows to detect easily rectilinear edges and contours which present small curvature variations, as lengthening is quite close to the contour.

4.3.2 Drawbacks

One of the drawbacks of the method is the cost of the algorithm. In the initial method, we have to inverse an $N * N$ matrix where N is the number of discretization points. Here, we have to inverse a new matrix at each lengthening of the snake. However, as the algorithm is more stable, the number of iterations needed is lower. Nevertheless, we cannot get theoretical estimation of the cost because the number of iterations depends on the initialization curve and on the image.

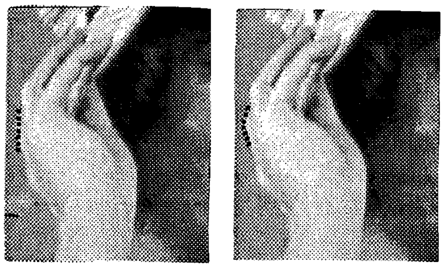

Figure 3: First step of growing.

As lengthening is made in the direction of the tangent, it is clear that the method is hardly adapted when heavy curvature variations occur at the extremities (figure 8). In the following section, we propose an improvement of the method to get rid of this drawback.

4.3.3 Results

To illustrate our method, consider the image shown in (Fig.3). We want to detect the woman's hand using snake growing. Figure 3) shows the initialization curve and the resulting snake.

Different steps of growing and the associated edges are shown in (Fig.4). In this example, only 10 iterations at each step are sufficient to ensure convergence.

Consider now the image of the lamp in figure 5. We voluntarily took a bad initialization. We can see the numerical solution which is almost stabilized. We then use the evaluation criterion to determine a curve C which can be considered as an edge. The result is shown in (Fig.6) and the snake is lengthened as in the previous case (Fig.7).

5 Improvement

5.1 Search for the best prolongation

When the growing algorithm fails, one of the following cases occurs

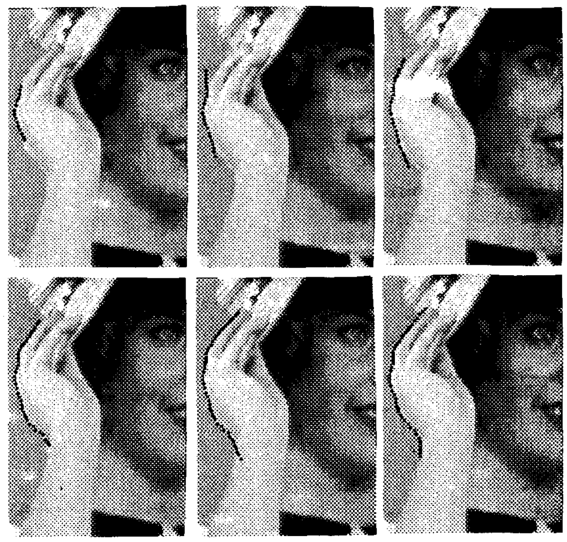

Figure 4: Following steps of growing.

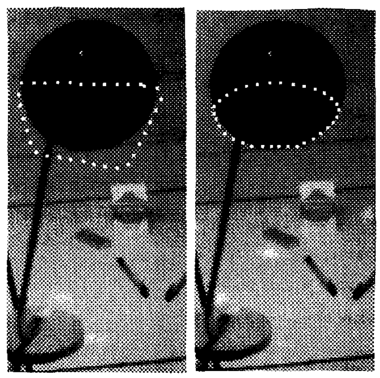

Figure 5: Traditional use of the snake.

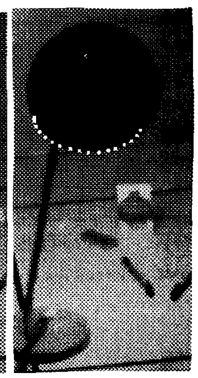

Figure 6: evaluation

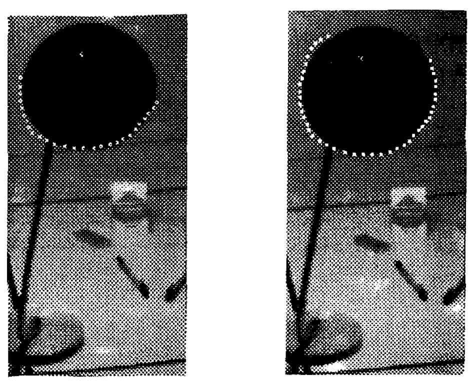

Figure 7: Snake growing.

- Edge termination or hole in the edge;
- High curvature variation in the contour (corner...) and the tangent direction is then far from the contour.

The zero crossings of the second derivative along the gradient direction are used to detect interruptions of contour in the flow direction [Kas 87]. But it does not permit to know if the contour carries on in another direction. We solve this problem in a **local** manner: to determine in which direction the contour lies, we study several lengthenings in different directions and we only retain the solutions giving rise to better results. In this way, we can find contours which present corners. Several solutions can be kept, so that junction points can be detected.

In our implementation, we lengthened the snake in eight directions and computed the resulting solution. We retain solutions which give rise to the best length increase.

An application of this method is shown in (Fig.8) to detect elevation curves in an image. Traditional methods fail because elevation curves are very close and present a high curvature variation in the middle of the back. The best prolongation method detects this variation.

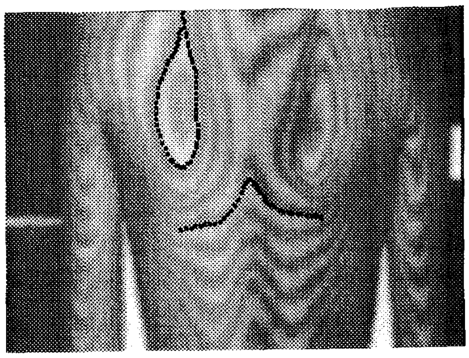

Figure 8: Using the best prolongation method to detect elevation curves.

5.2 Get rid of initialization

The method described in section 3 starts from a local initialization. But we can get rid of this drawback in the following way:

Given any initialization S_0, we can detect the intersection points (I_j) of S_0 with the contours using the gradient curve along S_0. We can then cut S_0 in curves S_j containing (I_j). They are the starting points of new snakes which can be lengthened as described before.

5.3 Discussion

The methods we proposed in this section are based on a local point of view, whereas the traditional snakes take into account a global point of view. This allows us to detect easily **local behaviors** in a contour like a corner or a high curvature variation. This approach can be very efficient if we have knowledge of the geometry of the contours and particularly on the presence of angular points on the contour. Other constraints can be used to control the edge detection. For instance, in [Fua 89], Fua uses parallelism constraints described in term of energy to detect roads in aerial images.

6 Conclusion

One of the strengths of the Active Contour Models is to provide a global point of view which permits to get rid of local anomalies or holes in the contour and to get regular and localized shapes.

One of the drawbacks of the proposed method (cutting and searching in several directions) is to lose this globality. Nevertheless, this is compensated by the ability to detect easily curves which are not C^2 (C^2 is the set of functions for which the second derivative is continuous) even in noisy surrounding.

This kind of method is only efficient if we have sufficient contour informations to have knowledge about privileged directions or if we integrate a geometric model of the contour.

Acknowledgment

This work was partially financed by GRECO PRC Communication homme-machine.

References

[Ami 88] A. Amini, S. Tehrani, and T. Weymouth. Using Dynamic Programming for Minimizing the Energy of Active Contours in the Presence of Hard Constraints. In *Second International conference on computer vision*, pages 95–99, Tampa Florida (USA), 1988.

[Fua 89] P. Fua and Y. Leclerc. Model Driven Edge Detection. *To appear in Journal of Machine Vision and Applications*, 1989.

[Kas 87] M. Kass and A. Witkin. Analyzing Orienting Patterns. *International Journal of Computer Vision*, 37:362–383, 1987.

[Kas 88] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, 321–331, september 1988.

[Mon 71] U. Montanari. On the Optimum Detection of Curves in Noisy Pictures. *CACM*, 8:335–345, 1971.

[Pog 85] T. Poggio, V. Torre, and C. Koch. Computational Vision and Regularization Theory. *Nature*, 314–319, 1985.

[Ter 86] D. Terzopoulos. Regularization of Inverse Visual Problems Involving Discontinuities. *IEEE Transactions on PAMI*, 8:413–424, 1986.

[Tik 76] A. Tikhonov and V. Arsenine. *Méthodes de résolution de problèmes mal posés*. Editions de Moscou, 1976.

[Zuc 88] S. Zucker, C. David, A. Dobbins, and L. Iverson. The Organization of Curve Detection: Coarse Tangent Fields and Fine Spline Coverings. In *Second International Conference on Computer vision*, pages 568–577, Tampa Florida (USA), 1988.