Image segmentation using snake-splines and anisotropic diffusion operators

G.-H. Cottet [1] , J. Demongeot [2], M. El Ayyadi [1] and F. Leitner [2]
July 1995

Abstract

The objective of image segmentation is to give a practical representation of object shapes. Filtering methods usually obtain nice results but do not extract explicitely objects of interest from images. Meanwhile active contour algorithms directly deal with objects but are far less efficient than local methods. So, a grey-level approach of both contrast enhancement and segmentation is presented using B-spline active contours and anisotropic diffusion equations. The goal of this combination is the use of a pre-processing method based on a selective filter that exhausts crest lines on a grey-level image. Then an active contours method permits the grouping of contour's indices in a parametric representation. The interest of such a method resides in the complementary performances of these two tools. Experimental results are shown on both synthetic and real images.

Keywords

Segmentation, contrast enhancement, B-splines, active contours, anisotropic diffusion, continuous modeling.

Animus fert dicere formas mutatas in corpora nova
(Ovide, Metamorphoses, 1st Book)

I. Introduction

THE study of grey-level images generally involves pattern segmentation. For several years such a procedure has been discussed through a large variety of operator [13]. However end user needs are not satisfied, because used operators often propose non truely modifiable results (according to some parameters).
Active contours and relative methods seem to present the best result form, because it can be easily displayed through a graphical user interface and manually modified. The main advantage of active contours resides in the knowledge to be gained about the targeted object edges. We know that these edges are closed and can be described with a parametric representation. Of course, the segmentation problem can not be solved in this way since parameter optimisation requires information from the image such as gradient values, which are often noisy. Furthermore, whatever deformation an active contour may undergo, it retains at least its own geometric properties as originally formulated in a mathematical framework.
The method [15], [19], [17], [2], [5], [24] identifies a function which describes the boundary of the object to be segmented. A classical way for finding the right border (the curve

[1] Laboratoire LMC - Institut IMAG, Université Joseph Fourier, B.P. 53X. 38041 Grenoble, France
[2] Laboratoire TIMC - Institut IMAG, Faculté de Médecine, Université Joseph Fourier, Institut Albert Bonniot, Domaine de la Merci, 38706 La Tronche Cedex, France

C) is to solve the variational problem where the function for finding $C(u) = (x(u), y(u))$ minimizes the energy:

$$\int_0^1 \Big[\underbrace{\alpha G(C(u))}_{geometrical\ constraint} + \underbrace{\gamma E_{f(C(u))}}_{image\ constraint} \Big] du \tag{1}$$

The snake equation 1 contains an *image constraint* induced by the image to be processed, and a *geometrical constraint* corresponding to the properties we expect for the curve. In the geometrical term, a function G models tension energy and/or smoothing energy using different derivatives of the curve C. f represents the initial image, and E is some sort of energy adapted to image segmentation (for instance $E_f = \Leftrightarrow ||\nabla(f)||^2$). α, and γ are parameters that can be optimized to find a compromise between intrinsic and extrinsic terms.

On the opposite, filtering methods or contrast enhancement algorithms now offer a higher performance in image treatment. The method discussed here is based on nonlinear anisotropic diffusion. The property sought for these operators is to reduce noise in the original image without altering its significant features. If too much noise persists, the contours used in the segmentation step will be drastically slowed, whereas too much smoothing will prevent accurate edge detection by the operators.

A solution for a really useful segmentation operator may use an efficient filtering operator coupled with a snake method. In the present work, quality of filtering was optimized in order to obtain highly constraining hypothesis for the active contour method. Such a characteristic permits to only deal with "interesting" contours, without special cases for impulsional noise for instance. In the following, both aspects are detailed. In Section 2 snake strategy is explained while Section 3 details diffusion operator. Then Section 4 shows results of method combination.

II. The active contour method

A. *Sketch of the method*

A.1 The snake model

As presented above, snake searches for a compromise between curve internal constraints and image constraints. Snake parametrisation is usually a polygonal approximation, and in the present case we have chosen to use B-spline functions to model the active contour or *snake*. We called this method *snake-splines* [19]. In the intrinsic term, a *smoothing term* controls the tension of the curve, and a *perimeter term* controls the inner size. The smoothing term is already satisfied by the spline functions [18]. So when B-splines are used, only a perimeter constraint adapted to the image constraint needs to be managed. We can specify these constraints as :

- Intrinsic energy : a perimeter tension, and a local surface minimization (Section II-C.3).
- Extrinsic energy : a growth procedure (Section II-B.2) which maximizes the area inside the curve. The domain within which the curve can move remains in the object's interior.

Therefore, although the snake-splines solution starts with uncertain initial edge indices, by using a growth rule these indices will be interpolated to the contour targeted.

A.2 Target points

During curve growth information contained by the image must be respected. It is well known that in an image the object's boundaries are closed to the maximum of the gradient norm applied on the grey level function. Two different methods can be used to extract these borders:

- Canny-Deriche filtering [3], [10]. Points to reach are locally maximum along a direction.
- Use of a first order operator. Image is binarized using the following rule: if the value of a point is greater than the average value of its neighborhood (a parameter b defines its size), result value will be 1, else 0. Points to reach are the frontiers between 0 and 1 areas.

The set of points extracted by these methods defines the targeted edges. They are called edge indices. As demonstrated earlier, the active contour starts inside the object to be modeled and will grow until it reaches the edge indices.

A.3 Finding the edge parametrisation

Let us define the active contour C with n parameters. These parameters are the control points of the control polygon and correspond to the expression of the B-spline curve. Details about the B-spline expression itself will be explained in Section II-C because its properties are mainly useful for the refinement method. Two problems are encountered in defining C:
1. finding the expression of the curve itself, or in other words, finding n parameters;
2. finding the value of the parameters (position of control points).

Both the number and the values of the parameters have to be optimized. So a loop is constructed to optimize the position of a reduced set of points (Section II-B). Then this set (Section II-C) is gradually increased. For a given set of control points, we use a growth procedure (Section II-B.2) in order to locate the points. A local adaptive procedure is then used to make a perimeter length adjustement (Section II-C.3) in order to preserve the necessary properties during the growth of the curve.

B. Positioning control points

B.1 Definitions

To clarify discussion some definitions need to be given. Algorithms rely on the notion of *control points* of the spline curve. This term is used so often here that we simply call them *points*. When we refer to the real points along the curve, we explicitly say *points of the curve*. We are working with closed curves, so an interior and an exterior can be defined. Because of the growth procedure (Section II-B.2) used, the vertices of a closed polygon are classified using two types : e-points and i-points. The criterion which distinguishes the two types of points is that *an external point (e-point) forms a bump on the polygon while an internal point (i-point) creates a hollow.* This is closely related to the orientation of the curve, and we define an e-point (resp. an i-point) as : Let P_i be a vertex of a polygon (P_n), and $\vec{k}$ the normal to the plane where the polygon is defined. We assume the polygon has a trigonometric orientation, then :

$$\vec{k} \cdot (\overrightarrow{P_{i-1}P_i} \wedge \overrightarrow{P_{i-1}P_{i+1}}) \geq 0 \text{ (resp. } < 0) \Leftrightarrow P_i \text{ is an e-point (resp. an i-point)} \quad (2)$$

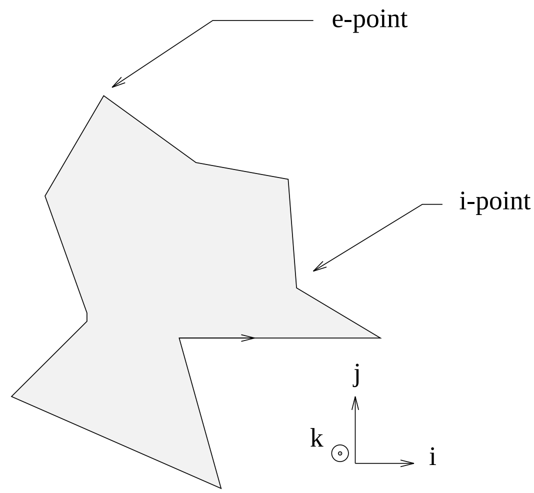

Fig. 1. *Naming rule for points, assuming curve has a trigonometric orientation.*

B.2 The growth procedure

The growth procedure consists in moving each e-point in order to increase the area inside the closed curve. The curve is defined at each iteration by its control polygon (P_n). Depending of spline order o in use, a part of the curve is defined by π_{o+1}, which is a subset of the main polygon. Such a part of the curve defines a set of points into the filtered image, which are used as the start of a filling algorithm. Using a bowl of parametrized size, the algorithm connects all points of same value under the part of the curve. Getting this new area frontier will give the new shape of the curve (Fig.6) :

1. starting as first position with the old part of the curve, new frontier is found using an edge following algorithm. New control points interpolate such an edge and replace π_{o+1}.
2. some structures can exist inside this new area. They are detected by comparing area surface with number of pixels filled. In the case a new curve is born. If it has a sufficient surface (surface thresholding parameter sp), it belongs to the solution. Parametrization of such an inner border is made in the same manner, with detection of a first point, and then border following.
3. if the new area border reaches a point of another part of the old curve position, the curve will be split, and inner part will be kept if it has a sufficient surface (same surface thresholding parameter sp).

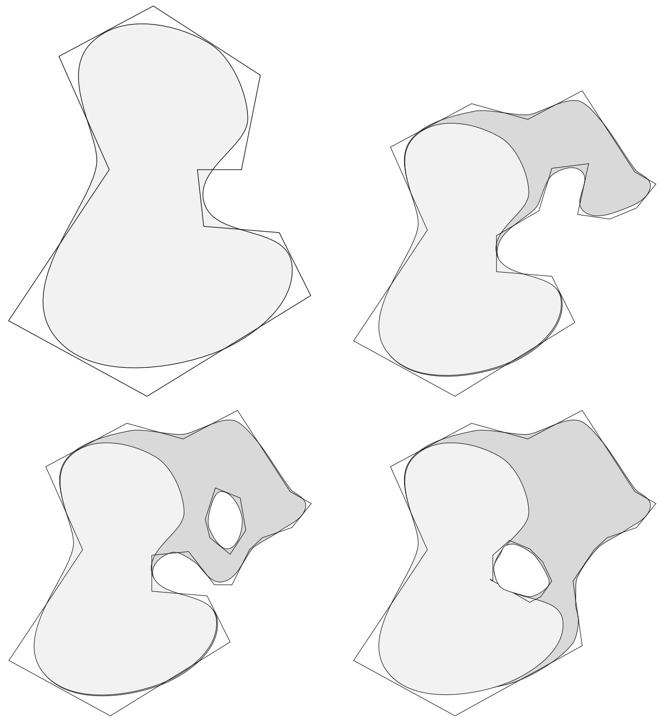

Fig. 2. *Growing principle. Starting from the same shape (top-left), three different results can be obtained as explained above.*

C. Increasing the number of control points

C.1 Spline curves subdivision algorithm

The polynomial property of the snake and the proposed spline evaluation algorithm can be explained in detail.
The active contour C has a classical B-spline [16], [9], [28], [17] expression :

$$\forall x \in [0,1], C(x) = \sum_{i=0}^{i=n-1} P_i \alpha_i(x) \tag{3}$$

where $P_i \in \Re^2$, $0 \leq i < n \Leftrightarrow 1$ are the control points, and α_i is the basis of the spline functions. The number n of coefficients is of course adapted to the order o of the spline basis. The method (Section II-A), requires the use of a spline function that is easy to evaluate and that can be refined using more coefficients. So a subdivision algorithm [20] [25] is used which is a variation of the Oslo algorithm to quickly give a discretisation of the curve. This is useful for testing the position of the curve in the image. However, starting with (P_n) polygon this algorithm yields to double the number of control points (Q_{2n}) at each iteration, until this new set of points converges to the set of points of the curve itself. This property will permit us to continuously increase the number of parameters of the curve. Figure 3 shows an example of this subdivision algorithm. Starting with the $(P_0, \ldots, P_5)$ polygon, at the first step we obtain the $(Q_0, \ldots, Q_{11})$ subdivided polygon. A second order spline has been used to provide this example. The sequence of polygons converges to the B-spline curve when the number of iterations tends to infinity.

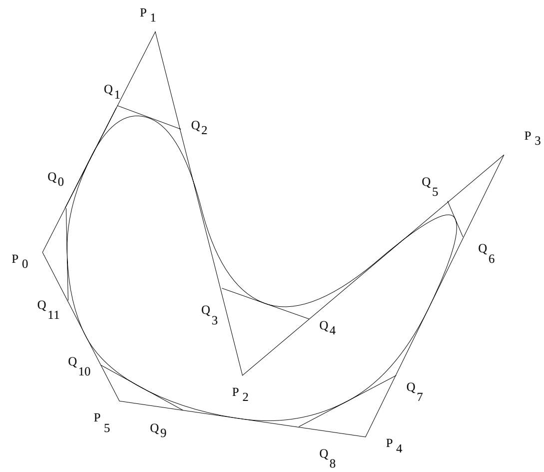

Fig. 3. *subdivision algorithm :* (P_6) *polygon subdivided into* (Q_{12}) *polygon.*

C.2 Refinement of a spline curve

The refinement method is clearly explained in Figure 4 on which we can see that the point Q_3 of the Figure 3 can be moved in order to locally modify the curve. The technique consists in using some of the Q_j control points in order to locally refine the curve expression [11]. In fact, (P_n) and (Q_{2n}) define the same curve but if we want to locally adapt it, the curve must be defined hierarchically with (P_n) and some of Q_i. So Figure 4 shows how to locally refine the $(P_0, \ldots, P_5)$ polygon when Q_3 is moved. The refined curve can be defined by the $(Q_0, \ldots, Q_{11})$ polygon or by the $(P_0, P_1, P_2, P_3, P_4, P_5, Q_3)$ polygon. This technique manages only the significant parameters of the curve's representation.

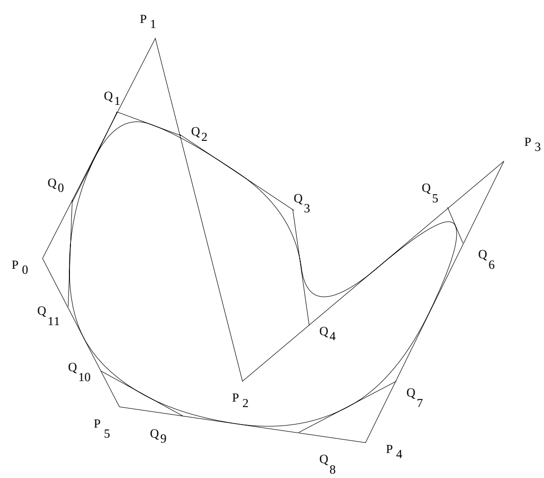

Fig. 4. *refinement method :* (P_6) *polygon refined in* $(P_0, P_1, P_2, P_3, P_4, P_5, Q_3)$.

C.3 Local adaptation

This paragraph describes how parameters can be modified in order to perform intrinsic constraints. Snake has to keep a certain rigidity, defined by user. After any new point generation, an adjustment is performed in order to optimize the curve's shape. Concerned points are e-points, because a smoothing on i-points would permit to the curve to go outside the object. Both following methods can be used according to weighting parameters:

1. Decrease the perimeter length by moving Q_i point to the center of its neighbours Q_{i-1} and Q_{i+1} (Figure 5). This method takes advantage of the homogeneous repartition of points.

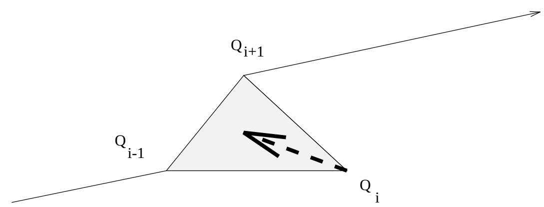

Fig. 5. *minimizing perimeter length : moving* Q_i *to* $\frac{Q_{i-1}+Q_{i+1}}{2}$*, or delete* Q_i *if it controls a too little area.*

2. Decrease the inner surface by computing local controlled surface. Such a surface is defined by the vector product $\overrightarrow{Q_{i-1}Q_i} \wedge \overrightarrow{Q_iQ_{i+1}}$. This area is compared with all other corresponding areas of the curve, and if it is the smallest, the point Q_i will be removed.

C.4 Global smoothing

In order to avoid a large set of control points, a smoothing procedure was designed. Some points have to be removed and a useful criterion must take into account the influence of the removed point on the curve shape. This influence is due to the angle between the two vertices of which the point belongs and to their length. The removing criterion consists in determining the controlled area within the control polygon. As shown on Figure 5, such an area is defined by $||\overrightarrow{Q_{i-1}Q_i} \wedge \overrightarrow{Q_iQ_{i+1}}||$.
Then an area threshold called sm can be determined in order to smooth the curve.

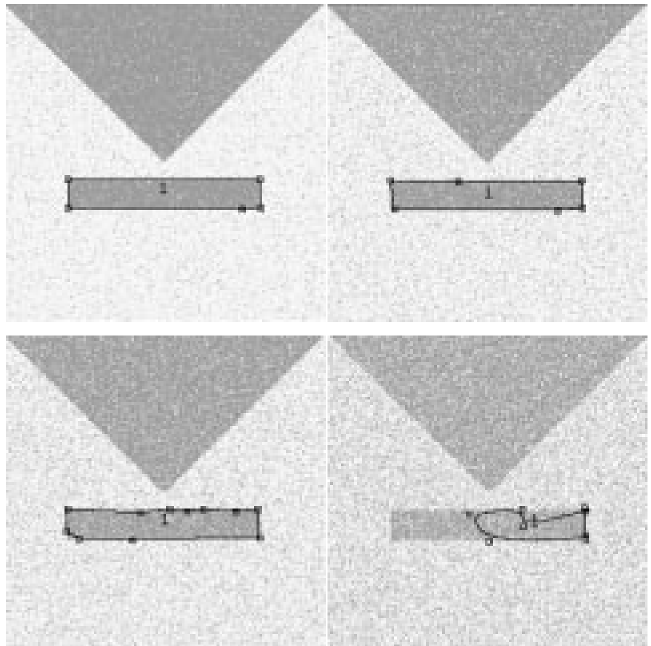

Fig. 6. *Results on synthetic images using increasing noise (clockwise). Starting from center point of the rectangle, the curve grows. Noise on image is 10%, 20%, 30% and 50%. On the two first images, the edges are correctly detected, but on the third a corner is missed. Then on last image noise is too important to determine anything.*

III. Low-level operator : Anisotropic diffusion

Many methods for contrast enhancement have been used in the past to reinforce the grey level gradient on the boundary of objects of interest. This step obviously precedes the steps of segmentation and contouring which give the final shape of these objects and allow them to be viewed as targets of a robotized action. Classical Gaussian filters [8] and the Gaussian scale-space [30] are unfortunately inadequate as an effective compromise solution for these two goals which in the absolute condition are mutually exclusive in the present state of the art. More sophisticated diffusion operators [23], [4], [1], [7], [22] have recently been designed where diffusion is somewhat inhibited across significant edges of the signal. However these operators, which must be understood in the framework of multi-scale analysis, require the definition of a minimum scale to be preserved. This minimum scale must then be translated into an integration time after which the filter has to be extinguished. The notion of minimum scale is not always clear on a given image. This is particularly true when very thin objects are studied. In this case, a relevant orientation for a scale would be along the longest uni-directional series of points of an object, but not across them. A multi-scale approach does not adequately distinguish between both. The class of operators presented below have the striking feature that they preserve the 1-D objects as long as their contours are smooth enough [7]. Moreover, they exhibit strong stability properties which permit acquisition of the desired filtered image on the asymptotic (in time) state of the system. In other terms, their use does not require any *a priori* knowledge about the desired image.

A. The model

The starting point in the derivation of efficient filters is always to inhibit diffusion across edges. If u is the grey level representation of an image (u maps $[0,1]^2$ onto $[\Leftrightarrow 1,+1]$), the normal to these edges coincides rather well with the vector $\vec{\nabla} u_\varepsilon$, where u_ε denotes a regularized version of u. A natural idea is thus to allow diffusion only along the direction orthogonal to these vectors. If we denote the orthogonal projection by $P_{\vec{\nabla} u_\varepsilon^\perp}$ on the direction $\vec{\nabla} u_\varepsilon^\perp$, this leads to the following filter equation:

$$\frac{\partial u}{\partial t} \Leftrightarrow \mathrm{div}(P_{\vec{\nabla} u_\varepsilon^\perp} \vec{\nabla} u) = f(u). \tag{4}$$

The parameter ε above is a scale parameter, which in practice is a few pixels wide, and u_ε is typically the average of u over the nearest neighbors of a given pixel, and the reaction term $f(u)$ is used for contrast enhancement. Model (4) is decribed in [7]. In this reference it is proved that the diffusion is strongly inhibited at the edges of coherent signals, no matter how thin these signals are. Unlike an isotropic scale parameter, the parameter ε turns out to be the scale on which the contours defined by the edges must be smooth. On the other hand, where the signal is noisy, no coherence between $\vec{\nabla} u$ and $\vec{\nabla} u_\varepsilon$ can be detected, and the diffusion operator acts as a purely isotropic diffusion operator, which is highly efficient for noise reduction.

We now come to a variant of this method which will be implemented in the next section. The underlying idea is still to prevent diffusion across the edges. However the edge directions are now determined using time references averages of the image instead of spatial references averages. More precisely the model is based on the following system:

$$\frac{\partial u}{\partial t} \Leftrightarrow \mathrm{div}(L[\vec{\nabla} u]) = 0 \tag{5}$$

$$\frac{dL}{dt} + \frac{1}{\beta} L = \frac{1}{\beta} F[\vec{\nabla} u] \tag{6}$$

where

$$F[\vec{\nabla} u] = \begin{cases} s^2\, P_{\vec{\nabla} u^\perp} & \text{if} |\vec{\nabla} u| > s \\ |\vec{\nabla} u|^2 P_{\vec{\nabla} u^\perp} + \frac{3}{2}(s^2 \Leftrightarrow |\vec{\nabla} u|^2)\mathrm{Id} & \text{if} |\vec{\nabla} u| \leq s. \end{cases}$$

The notation P stands for the orthogonal projection, in 2D:

$$P_{\vec{\nabla} u^\perp} = \frac{1}{|\vec{\nabla} u|^2} \begin{pmatrix} u_y^2 & \Leftrightarrow u_x u_y \\ \Leftrightarrow u_y u_x & u_x^2 \end{pmatrix},$$

and L denotes a 2×2 matrix. This system has to be supplemented with initial conditions: $u(\cdot, 0) = u_0$ which is the grey level of the original image, while the initial value of L is the identity, denoted by Id. This corresponds to the isotropic Gaussian filter. Unlike other diffusion models, the idea is that the image to process is a perturbation of an exact one that we wish to obtain on the asymptotic states of (5)-(6).

Let us now comment on the form of the right hand side of (6). A direct integration of (6) in zones where the gradient is larger than our parameter s yields

$$L(t) = \mathrm{Id}\exp\left(\Leftrightarrow t/\beta\right) + \frac{s^2}{\beta}\int_0^t P_{\vec{\nabla}u(\cdot,\tau)^\perp}\exp(\Leftrightarrow\frac{t\Leftrightarrow\tau}{\beta})\,d\tau$$

This is why we presented our method as a counterpart of (5)-(6) where space averages are somehow, although not exactly, replaced by time averages (averages do not commute with the operator P).
The (small) parameter β is the time scale on which averaging is essentially performed. The effect of the threshold s in (6) is to select stationary stable states. If it was not present, all couples of the form $(u, P_{\vec{\nabla}_u{}^\perp})$ would be stationary solutions to (5)-(6), and thus candidates for asymptotic states. Because of s, only solutions which have gradients that are either vanishing or stronger than s correspond to steady-states.
These solutions correspond to patterns separated by layers stiffer than s. The parameter s is thus a contrast parameter whose effect allows diffusion to homogenize coherent structures. Let us observe that these patterns include those with edges having corners.
The equation (6) can then be interpreted as a Hebbian dynamic learning rule [14] for the synaptic weights as follows. Starting from a translation invariant synaptic matrix associated to isotropic diffusion, that is $L \equiv \mathrm{Id}$, equation (6) allows the system to recognize the significant edge and to make diffusion vanish across these edges on a time scale $\simeq \beta$. At the discrete level this means that the synaptic weights linking neurons on opposite sides of the edges are strongly inhibited. In this analogy, the parameter s is directly related to the threshold above which variations of activity must exist between two neurons in order to tend to render inhibitory the connection between these 2 neurons.
For a mathematical analysis of (5)-(6) refer to [6].
Numerical experiments suggest that even though no reaction terms compensate for diffusion, this model has strong stability properties around steady states consisting of piecewise constant signals. Another convenient feature is that it requires the choice of only two parameters which are β (the learning relaxation time) and the contrast parameter s.
In practice, β is of the order of a few time steps. A general rule is that the more noisy the original image is, the bigger β must be chosen to allow isotropic diffusion to remove noise. However, the numerical results that we present below seem to indicate that the final results are not very sensitive to the choice of β. As for s, the smaller it is, the finer will be the structures extracted from the picture.

B. Numerical experiments

B.1 Discretisation

First let us give some notations. We denote by u_{ij}^n an approximation of $u(ih, jh, n\Delta t)$ where $h = \frac{1}{N}$, $0 \le i, j \le N$, by $\beta = k\Delta t$ the relaxation time and by

$$L_{ij}^n = \begin{pmatrix} L_{(ij)xx}^n & L_{(ij)xy}^n \\ L_{(ij)yx}^n & L_{(ij)yy}^n \end{pmatrix}$$

an approximation of L. The divergence and the gradient are approximated by classical one-sided finite differences as follows:

$$\begin{aligned} \mathrm{div}_h(u_1, u_2) &= h^{-1}(\Delta_+^x u_1 + \Delta_+^y u_2) \\ \vec{\nabla}_h u &= h^{-1} \begin{pmatrix} \Delta_-^x u \\ \\ \Delta_-^y u \end{pmatrix} \end{aligned}$$

where

$$\Delta_+^x u_{i,j} = u_{i+1,j} \Leftrightarrow u_{i,j} \;\;,\;\; \Delta_-^x u_{i,j} = u_{i,j} \Leftrightarrow u_{i-1,j},$$

and similarly for Δ^y by exchanging the roles of i and j. These are the formulas which lead to the classical five points scheme for the Laplacian operator.
Then we discretise $L(\vec{\nabla} u)$ by:

$$h^{-1} \begin{pmatrix} L^n_{(ij)xx} \Delta_-^x u^n_{ij} + L^n_{(ij)xy} \Delta_-^y u^n_{ij} \\ \\ L^n_{(ij)yx} \Delta_-^x u^n_{ij} + L^n_{(ij)yy} \Delta_-^y u^n_{ij} \end{pmatrix} \tag{7}$$

and $\mathrm{div}(L(\vec{\nabla} u))$ by:

$$h^{-2} \begin{bmatrix} \Delta_+^x \Big(L^n_{(ij)xx} \Delta_-^x u^n_{ij} + L^n_{(ij)xy} \Delta_-^y u^n_{ij} \Big) + \\ \\ \Delta_+^y \Big(L^n_{(ij)yx} \Delta_-^x u^n_{ij} + L^n_{(ij)yy} \Delta_-^y u^n_{ij} \Big) \end{bmatrix} \tag{8}$$

We use an explicit time discretisation for equation (5), and an implicit one for equation (6). The resulting algorithm is given by :

$$\begin{cases} u^{n+1}_{ij} = u^n_{ij} + \\ \\ \dfrac{\Delta t}{h^2} \begin{bmatrix} \Delta_+^x \Big(L^n_{(ij)xx} \Delta_-^x u^n_{ij} + L^n_{(ij)xy} \Delta_-^y u^n_{ij} \Big) \\ \\ + \Delta_+^y \Big(L^n_{(ij)yx} \Delta_-^x u^n_{ij} + L^n_{(ij)yy} \Delta_-^y u^n_{ij} \Big) \end{bmatrix} \\ \\ \qquad L^{n+1}_{ij} = (\frac{k}{1+k})(\frac{1}{k} P^n_{ij} + L^n_{ij}) \\ \\ \qquad u^0_{ij} = u_0(ih, jh) \qquad 0 \leq i, j \leq N \end{cases} \tag{9}$$

where P^n_{ij} is the evaluation of the right-hand side of (6) at time t_n.
To ensure stability in (9), we have to choose a time step satisfying

$$\Delta t \leq \frac{h^2}{6} \Big| \max_{i,j} L_{ij} \Big|. \tag{10}$$

IV. Experimental results

The snake-spline tool is now ready to be used in the segmentation process. The method is the following:
After applying the anisotropic diffusion algorithm (9) with parameters (threshold s and memory coefficient β) the snake parameters are defined (scale space sp, radius b, and smooth sm).

• In the first experiment, the classical benchmark image of Lena is shown. Since the original image (top left on Figure 7) is very textured in the searched area, a large number of iterations and a large memory coefficient ($\beta = 20\Delta t$, and s=5) is needed for restoration. Snake is manually initialized on the diffused image with four control points (Figure 3). Then snake (sp=3, b=1,sm=0) reaches the boundary of the searched area.
• In the second experiment, especially hard to analyze images were chosen (heart ultrasound images). Such a noisy image requires an appropriate filtering, and a large number of iterations is needed too (β =6.E-5, and s=10). Nevertheless searched areas (ventricles) are smooth and snake parameters are (sp=8, b=3, and sm=14). Two snakes were used to obtain both right and left ventricles. Gradient image (Canny-Deriche [3], [10] with α = 0.35) (top right of Figure 8) shows dashed gradient crests which are sufficient to correctly reconstruct the borders.

Fig. 7. Lena image. From top to bottom and left to right: the original image (512×512), the Canny-Deriche gradient image ($\alpha = .35$), the snake-spline intialization on the diffused image (100 iterations, β=6 E-05, s=5 and time to calculate equals 245 secondes), and the snake spline result with (sp=3, b=1, sm=0).

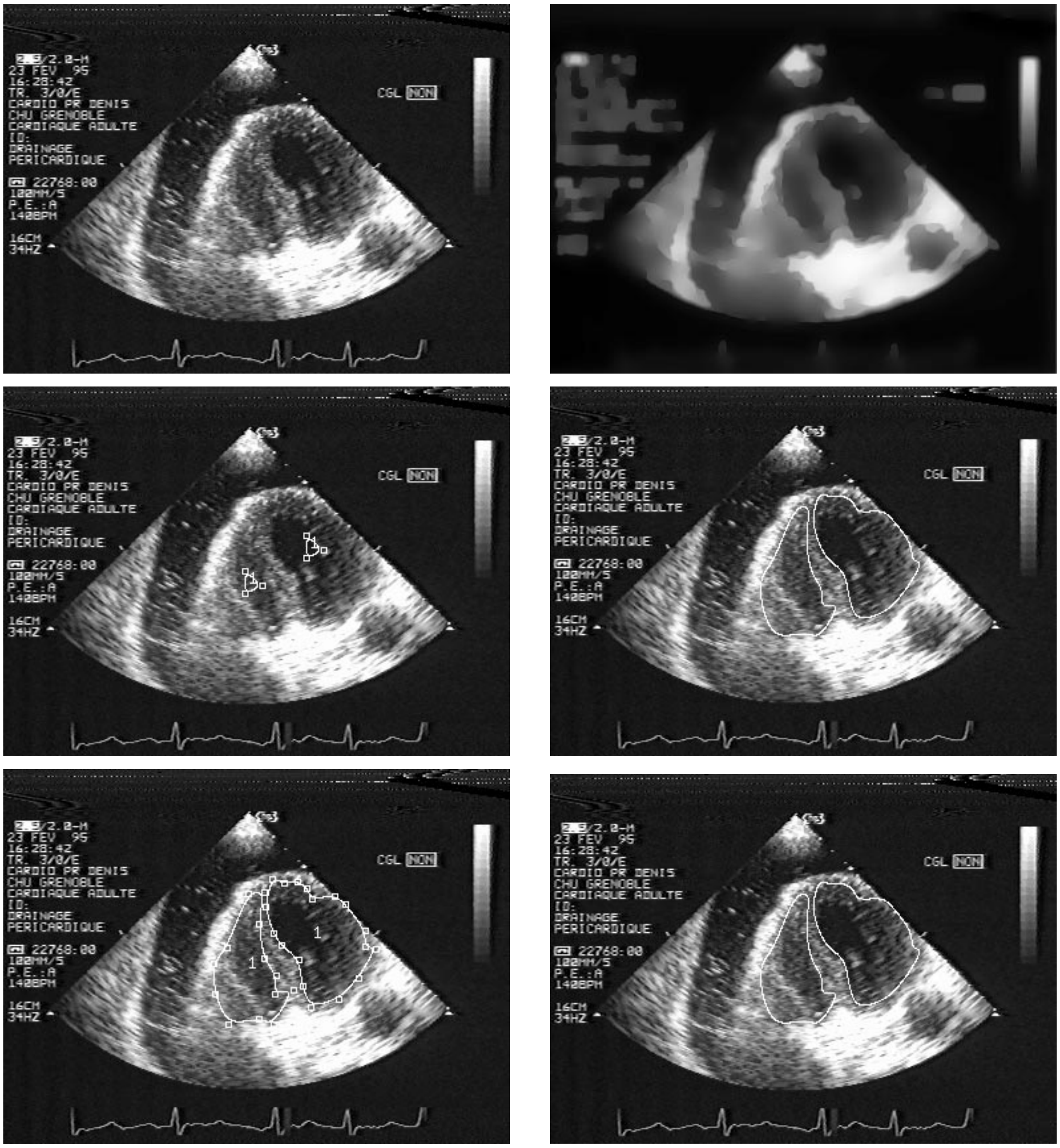

Fig. 8. Ultrasound image processing. From top to bottom and left to right: the original image (286×384), the Canny-Deriche gradient image (α=1.5), the diffused image (100 iterations, β=6.-05, s=5 and time to calculate equals 102 secondes), the snake-spline intialization on it, the snake spline processing with (sp=8, b=3, sm=25 and 19.), and the snake -spline result.

V. Discussion and Conclusion

We have presented a tool for image restoration which combines nonlinear anisotropic diffusion and snake-splines. A didactic way was chosen here in presenting 2-D version of algorithms, but a 3-D generalization is already available since both processes were parallelisable. We showed examples on very noisy and very textured images. Parameters correspond to specific geometrical features and their choice is closely related to edges. This permits to obtain a first result with reasonable accuracy, and its parametrical struc-

ture is fully available for any post-treatment. For example it allows a graphical interface to display or it can also be directly used in modeling algorithms such as shape tracking along image sequences.
Certain classical techniques already exist based on linear approximations of the trajectories [27]. They are Gaussian filtering of the velocity vector field [26], and Markov random field estimation. Real-time optical flow calculation using neural networks and parallel implementation techniques are now beginning to appear in the literature [29], [21]. Real-time contrast enhancement followed by real-time segmentation and real-time optical flow calculation is now possible for 2-D imaging. Fast 3-D acquisition like CT-morphometer at the macroscopic level and confocal microscopy even permit prediction of technical improvements for moving 3-D structures [12]. These techniques will soon be avalaible at the cellular level in order to follow the motion of micro-organisms and scar formation. These examples involve a final modeling step in order to explain which parameters and observables are critical for these dynamic processes. Such models already exist for Dyctiostelium motion, cardiac motion, cicatrisation, and tumor growth. Improvements in the first steps of data processing like those shown here will certainly push forward production of such explanatory models. An interesting example is fitting simple discrete models of insect motion with real data obtained through the procedures described here. Identifying parameters using these new methods represents a real progress, since motion can be quantified in reference to qualitative dynamic modeling approaches, possibly in real-time studies within a near future.

References

[1] L. Alvarez, P.L. Lions, and J.M. Morel. Image selective smoothing and edge detection by nonlinear diffusion.II. *SIAM J. Numer. Anal.*, 29:845–866, 1993.

[2] M.O. Berger. Les contours actifs : modélisation, comportement et convergence, february 1991. INPL thesis.

[3] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. & Machine Intell.*, 8:679–698, november 1986.

[4] F. Catté, J.-M. Morel, P.-L. Lions, and T. Coll. Image selectives smoothing and edge detection by nonlinear diffusion. *SIAM J. Numer. Anal.*, 29:182–193, 1992.

[5] I. Cohen. Modèles déformables 2D et 3D : application à la segmentation d'images médicales, 1992. Université Paris IX-Dauphine thesis.

[6] G.-H. Cottet and M. El Ayyadi. A Volterra type model for image processing. *In preparation.*

[7] G.H. Cottet and L. Germain. Image processing through reaction combined with nonlinear diff usion. *Math. Comp.*, 61:659–673, 1993.

[8] E. Hildreth D. Marr. Theory of edge detection. In *Proc. R. Soc. London*, volume B 207, pages 187–217, 1980.

[9] C. de Boor. *A practical guide to splines.* Springer Verlag, Berlin, 1978.

[10] R. Deriche. Using Canny's Criteria to derive a recursively implemented optimal edge detector. *Int. J. of Comp. Vision*, pages 167–187, 1987.

[11] D.R. Forsey and R.H. Bartels. hierarchical B-spline refinement. *Computer Graphics*, 22(4), 1988.

[12] L. Haibo, P. Roivainen, and R.Forchheimer. 3D motion estimation in model-based facial image coding. *IEEE Trans. Pattern Anal. & Machine Intell.*, 15:545–556, 1993.

[13] R.M. Haralick and S.G. Shapiro. Survey : image segmentation techniques. *Computer Vision and Image Processing*, 29:100–132, 1985.

[14] D.O. Hebb. *The Organization of Behavior: A Neuropsychological Theory.* John Wiley, New York, 1948.

[15] M. Kass, A. Witkin, and D. Terzopoulos. Snakes : active contour models. In *Proc. of ICCV*, pages 259–268, 1987.

[16] P.J. Laurent. *Approximation et optimisation.* Hermann, Paris, 1972.

[17] F. Leitner. Segmentation dynamique d'images tridimensionnelles, september 1993. INPG thesis.

[18] F. Leitner and P. Cinquin. From splines and snakes to snake-splines (medical images segmentation). In C. Laugier, editor, *Geometric Reasoning for Perception and Action*, pages 264–281. Springer-Verlag, 1991.

[19] F. Leitner, I. Marque, and P. Cinquin. Dynamic segmentation : finding the edge with snake-splines. In Academic Press, editor, *Curves and Surfaces*, pages 279–284, 1990.

[20] A. LeMehauté and P. Sablonnière. Courbes et surfaces Bézier/B-splines. In *A.T.P. du CNRS*, may 1987.

[21] J. Mattes, D. Trystram, and J. Demongeot. Parallel image processing: application to gradient enhancement of medical images. *Parallel Proc Letters*, 1995. (submitted).

[22] S. Osher and L. Rudin. Feature-oriented image enhancement using shock filters. *SIAM J. Numer. Anal.*, 27:919–940, 1990.

[23] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. & Machine Intell.*, 12(7):629–639, 1990.

[24] N. Rougon. Eléments pour la reconnaissance de formes tridimensionnelles déformables. Application à l'imagerie biomédicale, february 1993. ENST thesis.

[25] P. Sablonnière. Spline and Bézier polygons associated with a polynomial spline curve. *Computer Aided Design*, 10:257–261, 1978.

[26] M. Shah, K. Rangarajan, and P.S. Tsai. Motion trajectories. *IEEE Trans. Systems, Man & Cybern*, 23:1138–1151, 1993.

[27] B.E. Shi, T. Roska, and L.O. Chua. Design of linear neural networks for motion sensitive filtering. *IEEE Trans. Circuits & Systems II*, 40:320–332, 1993.

[28] L.L. Shumaker. *Spline fonctions : basic theory.* John Wiley & sons, 1981.

[29] S.T. Toborg and K. Huang. Cooperative vision integration trough data-parallel neural computations. *IEEE Trans on Computers*, 40:1368–1380, 1991.

[30] A. Witkin. Scale-space filtering. In *Proc of IJCAI*, pages 1019–1021, 1983.