International Journal of Computer Vision 65(1/2), 29–42, 2005

DOI: 10.1007/s11263-005-3847-y

Detecting Codimension—Two Objects in an Image with Ginzburg-Landau Models

GILLES AUBERT
Laboratoire J.A. Dieudonné, UMR CNRS 6621, Université de Nice Sophia-Antipolis, Parc,Valrose 06108 Nice Cedex 2, France
gaubert@math.unice.fr

JEAN-FRANÇOIS AUJOL*
Laboratoire J.A. Dieudonné, UMR CNRS 6621, Université de Nice Sophia-Antipolis, Parc,Valrose 06108 Nice Cedex 2, France; ARIANA, projet commun CNRS/INRIA/UNSA, INRIA Sophia Antipolis 2004, route des Lucioles, BP93, 06902 Sophia Antipolis Cedex, France
aujol@cmla.ens-cachan.fr

LAURE BLANC-FÉRAUD
ARIANA, projet commun CNRS/INRIA/UNSA, INRIA Sophia Antipolis 2004, route des Lucioles, BP93, 06902 Sophia Antipolis Cedex, France
Laure.Blanc_Feraud@sophia.inria.fr

Received July 13, 2004; Revised March 16, 2005; Accepted May 9, 2005

First online version published in January, 2006

Abstract. In this paper, we propose a new mathematical model for detecting in an image singularities of codimension greater than or equal to two. This means we want to detect isolated points in a 2-D image or points and curves in a 3-D image. We drew one's inspiration from Ginzburg-Landau (G-L) models which have proved their efficiency for modeling many phenomena in physics. We introduce the model, state its mathematical properties and give some experimental results demonstrating its capability in image processing.

Keywords: Ginzburg-Landau model, points detection, segmentation, PDE, biological images, SAR images

1. Introduction

The goal of this paper is to propose a new mathematical model for detecting in an image singularities of codimension greater than or equal to two. This means we want to detect isolated points in a 2-D image or points and curves in a 3-D image. To the best of our knowledge there exist in the literature few works tackling this problem. Most of existing models are devoted to the detection of singularities of codimension-one, e.g. curves in $\mathbb{R}^2$ or surfaces in $\mathbb{R}^3$. Recently Lorigo et al. (1999) have developed a codimension-two geodesic active contour scheme for the segmentation of thin structures. Their algorithm is based on work in differential geometry (Ambrosio and Soner, 1996) concerning the evolution of arbitrary dimensional manifolds in arbitrary dimensional space. See also Ruuth et al. (1998) for a diffusion-generated motion scheme for codimension-curves. Lorigo et al. have applied

*Author is now with CMLA (CNRS UMR 8536), ENS Cachan, France.

their algorithm for automatically segmenting blood vessels in volumetric resonance angiography images.

Here our approach is quite different. We drew one's inspiration from Ginzburg-Landau (G-L) models which have proved their efficiency for modeling many phenomena in physics and in particular in the theory of superconductors. G-L models are well-adapted to detect singularities in signals which modelize material conductivity. In this paper, our objective is to apply this type of model to capture singularities in images. There exists a general theory of G-L models involving functions u from $\mathbb{R}^{n+k}$ into $\mathbb{R}^k$ for the study of singularities of codimension-k in an ambient space of dimension $n+k$ (see Alberti et al., 2003). Our future objective is to process 3D images and to detect curves in $\mathbb{R}^3$. As a preliminary work, we focus in this paper on 2D-images. In this context, we introduce the simplest G-L model, state its mathematical properties and give some experimental results demonstrating its capability in image processing. More precisely, we examine the case $k = 2$ and $n = 0$, which results in detecting isolated points in 2D images. Though simple detectors (for example based on the Laplacien of the 2D image) could be used for this task, we think that the study of the G-L 2-D model is a necessary first step to show the ability of the proposed model to detect singularities of codimension 2 and to understand more complicated topological situations in $\mathbb{R}^3$.

We will also show some experiments in the case $k = 1$ and $n = 1$. In this case we will see that our algorithm is able to detect curves which are not necessarily closed and that we can also capture certain quadruple junctions.

The plan of the paper is organized as follows. In Section 2 we introduce the G-L model and give its main physical and mathematical properties. Then in Section 3 we show how such a model can be adapted to the detection of points in 2-D images. In Section 4 we display some numerical results demonstrating that our algorithm also applies for the detection of curves in 2-D images and in particular its capability to process the detection of nonclosed curves. We also display results showing the robustness of our approach against noise.

2. The Ginzburg-Landau Model

In this section we introduce the Ginzburg-Landau model. We first present the origin of the model, then we give its main mathematical properties and finally we show how this model can be used for detecting in an image singularities of codimension-two.

The Ginzburg-Landau model was designed in the fifties by Ginzburg and Landau (1950) to modelize phenomenological patterns in superconductor material near their critical temperature. Semiconductors have the particularity that, when they are cooled down below a critical temperature, they become "superconducting" which means that there can be permanent currents without dissipation.

There exists an important literature concerning G-L models. It is not the place here to review all these results. We will only give those which are the most linked to our purpose. Most of them rely on the simplified energy:

$$E_\varepsilon(u) = \frac{1}{2}\int_\Omega \left(|\nabla u|^2 + \frac{1}{2\varepsilon^2}(1-|u|^2)^2 \right) dx \qquad (1)$$

or on the associated flow governed by the evolution equation:

$$\frac{\partial u}{\partial t} = \Delta u + \frac{1}{\varepsilon^2} u\,(1-|u|^2) \qquad (2)$$

where $u : \Omega \subset \mathbb{R}^2 \to \mathbb{R}^2$. The parameter ε is a small positive constant which has the dimension of a length and depends on the material and its temperature. Of course G-L models are interesting and useful if we associate to (1) or (2) some singular data (for example Dirichlet data $u(x) = u_0(x)$ on the boundary $\partial\Omega$). That also allows to avoid trivial solution. The function u is a complex-valued function which indicates the local state of the material (in the superconductor theory $|u|$ is proportional to the density of superconducting electrons). Usually a solution $|u_\varepsilon|$ takes values close to 1 almost everywhere for ε small enough, and takes values close to 0 on small regions of characteristic size ε which are singularity regions for the physical model under consideration (for example, loss of superconductivity). Therefore if we want to detect these singularities we only have to display the set where $|u_\varepsilon|$ takes values close to 0. Let us remark that points can be considered as the intersection of two curves $u_1(x) = 0$ and $u_2(x) = 0$. So we stress that u needs to be a vector and not a scalar if we want to detect singularities of codimension 2 in $\mathbb{R}^2$ (see Bethuel et al., 1994) for a theoretical explanation). Let us note incidentally that there exist other works in image processing using complex-values functions (see for example Gilboa et al., 2001; 2004).

As Dirichlet data are not quite natural in image analysis, we thus incorporate in (1), a data term of the type $\int_\Omega |u - u_0|^2 dx$ and we consider Neumann boundary conditions. For this type of model as pointed out by Chen et al. (1998), it may happen that singularities of initial data can eventually disappear from the domain (for example a singularity can merge with other singularities). Hence to stabilize each singularity we need to create an energy barrier around the singularities by incorporating an appropriate diffusion coefficient $a(x)$ in the first diffusion term of (1). Before writing down our final model (see Section 3) we quote from Chen et al. (1998) a result which was the starting point of our study. Let us consider the following Ginzburg-Landau equation with a variable diffusion coefficient and Neumann boundary condition:

$$\mathrm{div}(a(x)\nabla\Phi) + (1 - |\Phi|^2)\Phi \text{ in } \Omega, \ \frac{\partial\Phi}{\partial\vec{n}} = 0 \text{ on } \partial\Omega \tag{3}$$

where $a(x)$ is a positive smooth function, $\vec{n}$ is the unit outward vector on $\partial\Omega$, and the unknown Φ is a complex-valued function.

Theorem 2.1 (Chen et al., 1998). Let Ω be a bounded domain in $\mathbb{R}^2$ with regular boundary. Given arbitrarily a finite number of distinct points $\{a_j\} \subset \Omega, j = 1, \ldots, N$ and let $\rho < \rho_0$ with $\rho_0 = min\{min\frac{1}{2}|a_j - a_k|, 1 \leq j < k \leq N; min\, dist(a_j, \partial\Omega), 1 \leq j \leq N\}$, then there exists a regular function $a(x) > 0$ such that equation (3) with Neumann boundary conditions has a stable solution $\Phi(x)$ whose zero set $Z[\Phi] = \{x \in \overline{\Omega}/\Phi(x) = 0\}$ is ρ-close to the prescribed configuration in the sense that

$$Z[\Phi] \subset \bigcup_{j=1}^{N} B_\rho(a_j), \ \ Z[\Phi] \cap B_\rho(a_j) \neq 0$$

where $B_\rho(a_j)$ is a disc of center a_j and radius ρ.

Typically the function $a(x)$ is close to zero only in $B_\rho(a_j)$. In the spirit of the above theorem, our objective in what follows is to propose some functions $a(x)$ of the initial image which allow us to detect singularities of codimension 2 and 1 in a 2D-image.

3. Detection of Singularities of Codimension-2 in 2-D Images

Let us now begin with the problem of detecting singularities of codimension-two in a 2-D image. From the previous results, this means that we need to consider complex functions defined on Ω. First, we have from an initial 2-D image $f(x)$ to construct a complex-valued image u_0 (we only consider gray-level images). There are many ways for doing it. We choose the one's proposed by Grossauer and Scherzer (2003). We first rescale the intensity image $f(x)$ to the interval $[-1, 1]$, then $f(x)$ is identified with the real part of a complex valued function u_0: $\Omega \to \mathbb{C}$ by defining $\mathrm{Im}(u_0) = \sqrt{1 - f(x)^2}$, so that $|u_0| = 1$. For detecting singularities of codimension-two (points) we propose to search for minimizers $u_\varepsilon \in H^1(\Omega; \mathbb{C})$ of the following G-L functional:

$$F_\epsilon(u) = \mu \int_\Omega a(x)|\nabla u|^2 + \frac{1}{\epsilon^2}\int_\Omega \left(1 - |u|^2\right)^2 + \frac{\lambda}{2}\int_\Omega |u - u_0|^2 \tag{4}$$

where $a(x)$ is a diffusion coefficient and u is a complex valued function as u_0. The first term of F_ϵ is a regularization term, the second one forces a solution u to be such that $|u| = 1$ almost everywhere, and the third one is a data term. The choice of the data term is a current subject of investigation.

If we denote by u (for the sake of clarity, we omit the ε dependence) a minimizer of $F_\varepsilon(u)$ then it satisfies the Euler-Lagrange system:

$$-\mu\mathrm{div}(a(x)\nabla u) - \frac{1}{\epsilon^2}u(1 - |u|^2) + \lambda(u - u_0) = 0 \text{ in } \Omega \tag{5}$$

and $\frac{\partial u}{\partial \vec{n}} = 0$ on $\partial\Omega$ (where $\vec{n}$ is the outward unit normal to $\partial\Omega$).

Now, since we want to detect singularities created by discontinuity points (or Dirac singularities whose support are points), we choose $a(x)$ as follows:

$$a(x) = W(\Delta f) \tag{6}$$

where f is the initial gray level image. The operator Δf stands for the Laplacien and W is a nonincreasing function with $W(0) = 1$ and $W(+\infty) = 0$. Let us remark that isolated points can be detected as maximum of the absolute value of the Laplacien. Typically, we choose

$$W(t) = \frac{1}{1 + (t/\alpha)^2} \tag{7}$$

where α is a parameter modelling the size of the discontinuity step. Moreover, since f may not be twice differentiable (in fact f is not continuous at points we want to detect), we first smooth it by convolution with a Gaussian kernel before computing W. We will see in Section 3.3 that this model behaves as the one given in Theorem 2.1: the set $\{x \in \Omega / |u_\epsilon| \simeq 0\}$ is the set of the singularities (discontinuity points) of u if ε small enough, and thus also the discontinuity points of the image f.

3.1. Evolution Equation

As it is often done, to solve equation (5), we embed it into a dynamical scheme:

$$\frac{\partial u}{\partial t} = \mu \mathrm{div}(a(x)\nabla u) + \frac{1}{\epsilon^2} u \left(1 - |u|^2\right) - \lambda \left(u - u_0\right) \tag{8}$$

with Neumann boundary conditions and initial condition $u(t = 0, x) = u_0(x)$. We write $u = (u_1, u_2)$, so that we can rewrite (5) as:

$$\begin{cases} \dfrac{\partial u_1}{\partial t} = \mu \mathrm{div}(a(x)\nabla u_1) + \dfrac{1}{\epsilon^2} u_1 \left(1 - \left(u_1^2 + u_2^2\right)\right) \\ \qquad -\lambda \left(u_1 - (u_0)_1\right) \\ \dfrac{\partial u_2}{\partial t} = \mu \mathrm{div}(a(x)\nabla u_2) + \dfrac{1}{\epsilon^2} u_2 \left(1 - \left(u_1^2 + u_2^2\right)\right) \\ \qquad -\lambda \left(u_2 - (u_0)_2\right) \end{cases} \tag{9}$$

$(u_0)_1$ is the original image, after it has been rescaled between -1 and 1. We take $(u_0)_2 = \sqrt{1 - (u_0)_1^2}$ (so that $(u_0)_1^2 + (u_0)_2^2 = 1$).

3.2. Discretization of the Model

The image is a two dimension vector of size $N \times N$. We denote by X the Euclidean space $\mathbb{R}^{N \times N}$ and $Y = X \times X$. The space X will be endowed with the scalar product $(f, g)_X = \sum_{1 \le i, j \le N} f_{i,j} g_{i,j}$ and the norm $\|f\|_X = \sqrt{(f, f)_X}$. We introduce a discrete version of the gradient operator. If $f \in X$, the gradient ∇f is a vector in Y given by:

$$(\nabla f)_{i,j} = \left((\nabla f)^1_{i,j}, (\nabla f)^2_{i,j}\right)$$

with

$$(\nabla f)^1_{i,j} = \begin{cases} f_{i+1,j} - f_{i,j} & \text{if } i < N \\ 0 & \text{if } i = N \end{cases}$$

and

$$(\nabla f)^2_{i,j} = \begin{cases} f_{i,j+1} - f_{i,j} & \text{if } j < N \\ 0 & \text{if } j = N \end{cases}$$

We also introduce a discrete version of the divergence operator. We define it by analogy with the continuous setting by div $= -\nabla^*$ where ∇^* is the adjoint of ∇: that is, for every $p \in Y$ and $f \in X$, $(-\mathrm{div} p, f)_X = (p, \nabla f)_Y$. It is easy to check that:

$$(\mathrm{div}(p))_{i,j} = \begin{cases} p^1_{i,j} - p^1_{i-1,j} & \text{if } 1 < i < N \\ p^1_{i,j} & \text{if } i = 1 \\ -p^1_{i-1,j} & \text{if } i = N \end{cases} + \begin{cases} p^2_{i,j} - p^2_{i,j-1} & \text{if } 1 < j < N \\ p^2_{i,j} & \text{if } j = 1 \\ -p^2_{i,j-1} & \text{if } j = N \end{cases} \tag{10}$$

Finally, we define a discrete version of the Laplacian operator by setting $\Delta f = \mathrm{div}(\nabla f)$ if $f \in X$.

Time Discretization. We use an explicit Euler scheme with respect to the time variable t, that is we approximate $\frac{\partial u}{\partial t}$ by $\frac{u^{n+1}_{i,j} - u^n_{i,j}}{\delta t}$ (where n stands for the iteration time). To solve (8), we use an explicit scheme:

$$u^{n+1}_{i,j} = u^n_{i,j} + \delta t \Big(\mu (\mathrm{div}(a_{i,j} u^n_{i,j}) + \frac{1}{\epsilon^2} u^n_{i,j} \left(1 - \left|u^n_{i,j}\right|^2\right) - \lambda \left(u^n_{i,j} - u^0_{i,j}\right)\Big) \tag{11}$$

with $u^0_{i,j} = (u_0)_{i,j}\ \forall (i, j)$.

We thus get the following system (we omit subindices i, j refering to the pixel location):

$$\begin{cases} u_1^{n+1} = & u_1^n + \delta t \left(\mu(\mathrm{div}\left(a u_1^n\right)\right. \\ & + \frac{1}{\epsilon^2} u_1^n \left(1 - \left(\left(u_1^n\right)^2 + \left(u_2^n\right)^2\right)\right) \\ & -\lambda \left(u_1^n - (u_0)_1\right)) \\ u_2^{n+1} = & u_2^n + \delta t \left(\mu(\mathrm{div}\left(a u_2^n\right)\right. \\ & + \frac{1}{\epsilon^2} u_2^n \left(1 - \left(\left(u_1^{n+1}\right)^2 + \left(u_2^n\right)^2\right)\right) \\ & - \lambda \left(u_2 - (u_0)_2\right)) \end{cases} \tag{12}$$

3.3. Numerical Results

3.3.1. Parameters. We need to fix several parameters before running our algorithm. Fortunately, they have an intuitive explanation which makes them easy to fix. We first need to fix the parameters λ, μ and ε used in system (9).

1. ε is to be small. We use values ranging from 0.1 to 1.0 (we have mainly used $\epsilon = 0.1$ and $\epsilon = 0.5$). It controls the critical size of the points our algorithm detects, i.e. the resolution of the segmented image. The smaller it is, the finer the resolution is. Nevertheless, one must not set it too small, because the spatial discretization of the image is fixed. Moreover, the smaller ε is, the smaller the time discretisation step δt has to be fixed (otherwise, the numerical algorithm does not converge).
2. λ is the fidelity parameter to the initial data. Since we initialize u to u_0, we do not need to use a large value. In our numerical experiments, we have almost always used $\lambda = 0.1$.
3. μ is the regularization parameter. We use values ranging from 0.1 to 50.0 (but we have mainly used $\mu = 1$). It mainly depends on how noisy the initial image is. The larger the noise is, the larger μ should be.

We also need to fix the parameter α in equation (7). It represents the critical size of the step of the discontinuity that our algorithm detects. The smaller it is, the smaller the detected steps are.

And as we have said, before computing Δf in the diffusion term (6), we regularize f by convolution with a Gaussian kernel of standard deviation σ. We use values ranging from 3 to 7. The larger the noise is, the larger we set σ (in the case when the original image has not been degraded by some noise, we sometimes do not regularize f).

Computation Time. Since we minimize the functional by solving the associated Euler-Lagrange system with a gradient descent method, we have to choose the time discretization step δt as well as a small stopping criterion TOL. Concerning δt, it needs to be chosen quite small so that the scheme be stable. To check if the scheme is stable, one just needs to look at the differences $|u_1^{n+1} - u_1^n|$. If δt is correct, then these differences decrease. Otherwise, they explode within a few iterations. We want to choose δt as large as possible, so that the algorithm be fast. But the smaller ε is, the smaller δt needs to be. In practice, we have used values of δt ranging from 0.00005 to 0.01. Concerning TOL, we have used the value 0.0001. We iterate algorithm (9) until $\max\left\{\frac{|u_1^{n+1} - u_1^n|}{|u_1^n|}, \frac{|u_2^{n+1} - u_2^n|}{|u_2^n|}\right\} \leq TOL$.

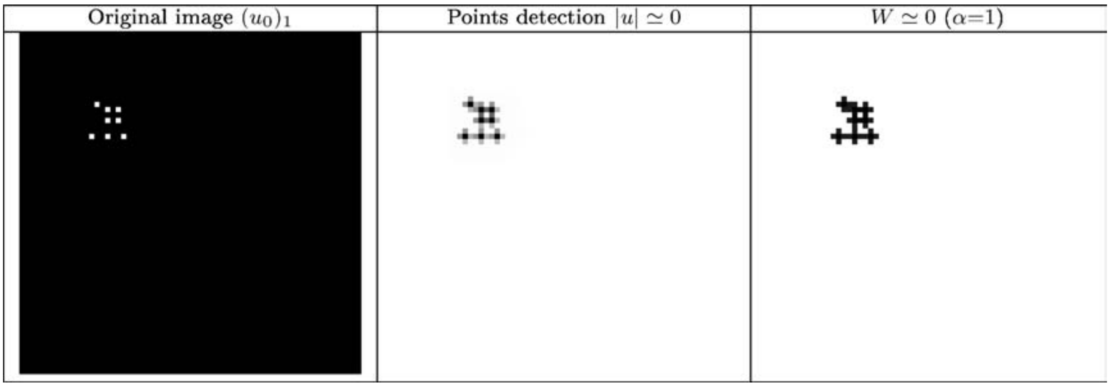
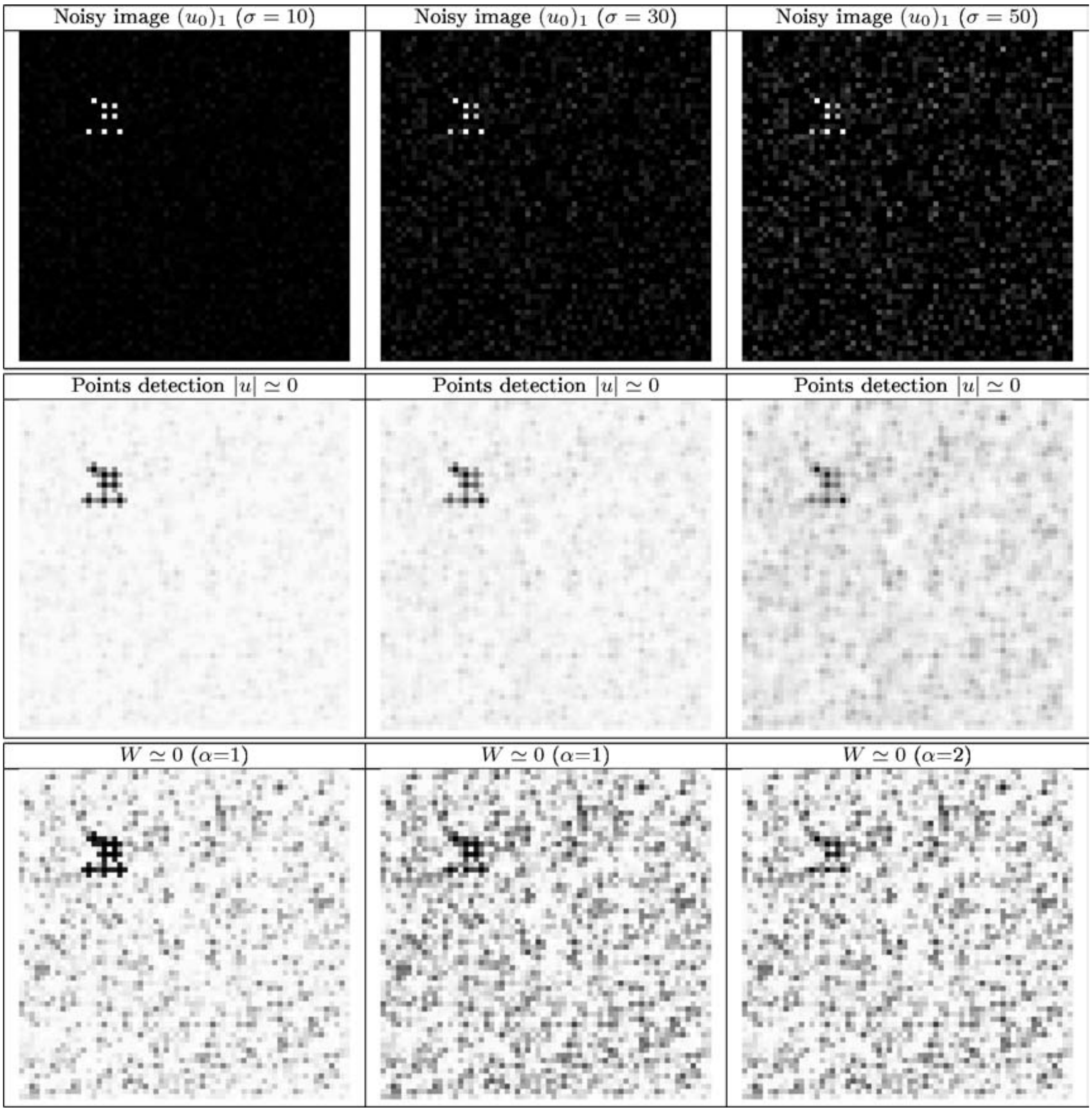


Figure 1. Synthetic image: in this very simple example, the points are well detected. W is the thresholding function (7) based on the Laplacien.



Noisy image $(u_0)_1$ $(\sigma = 10)$

Noisy image $(u_0)_1$ $(\sigma = 30)$

Noisy image $(u_0)_1$ $(\sigma = 50)$

Points detection $|u| \simeq 0$

Points detection $|u| \simeq 0$

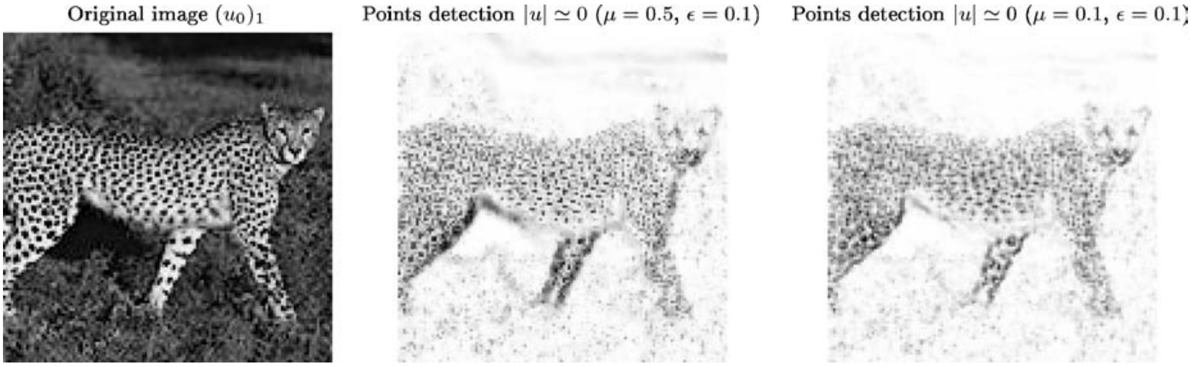
Points detection $|u| \simeq 0$

$W \simeq 0$ $(\alpha{=}1)$

$W \simeq 0$ $(\alpha{=}1)$

$W \simeq 0$ $(\alpha{=}2)$

Figure 2. Noisy synthetic images: as the standard deviation of the noise increases, then the detection of the points gets less accurate.



Original image $(u_0)_1$

Points detection $|u| \simeq 0$ $(\mu = 0.5, \epsilon = 0.1)$

Points detection $|u| \simeq 0$ $(\mu = 0.1, \epsilon = 0.1)$

Figure 3. Detecting points: as expected, the dashes of the leopard are detected as singularities.

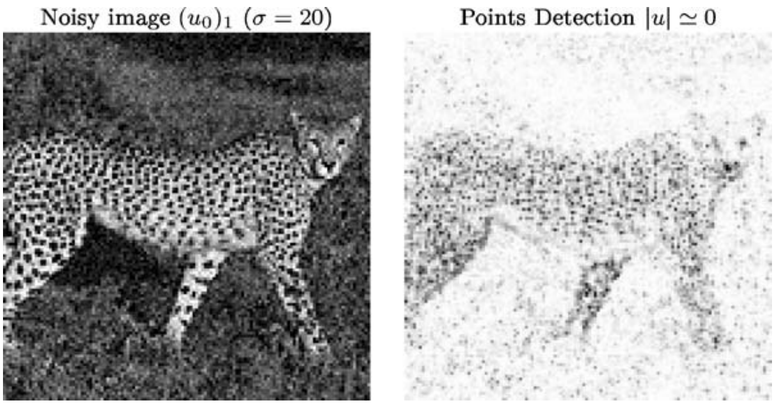

Noisy image $(u_0)_1$ $(\sigma = 20)$
Points Detection $|u| \simeq 0$

Figure 4. Detecting points in a noisy image ($\sigma = 20$).

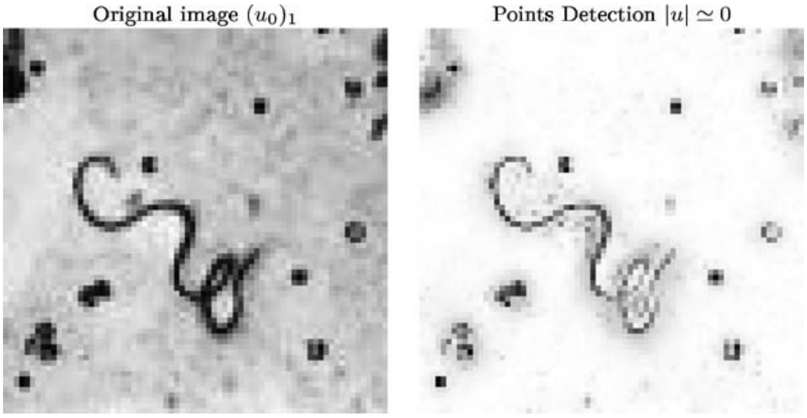


Figure 5. Biological points: the structures are detected as sequences of points.

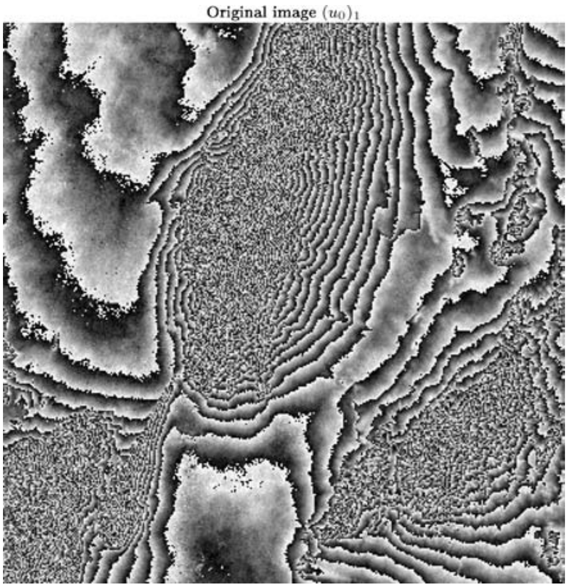


Figure 6. Original SAR image.

With such parameters, it takes less than 30 seconds to process a $256 * 256$ image.

3.3.2. Commentaries. In all our numerical examples, we also show the result we get with a Laplacien based detector. We just use the function W proposed in (7) (we mention the value of α we use in the Figures). This detector is based on the values of the Laplacien of the image (after it has been smoothed by a Gaussian kernel). Large values of the Laplacien correspond to a singularity in the image. This means that W is close to 0 on a singularity, and close to 1 in flat regions. We could also have used a classical detector such as the Harris detector (Harris and Stephens, 1988). But this detector was originally designed to capture the corners of an image, and we have checked numerically that it performs not as well as the W-detector on pointwise singularities.

On Fig. 1, we show an example on a synthetic image: our algorithm catches the points very well. We also

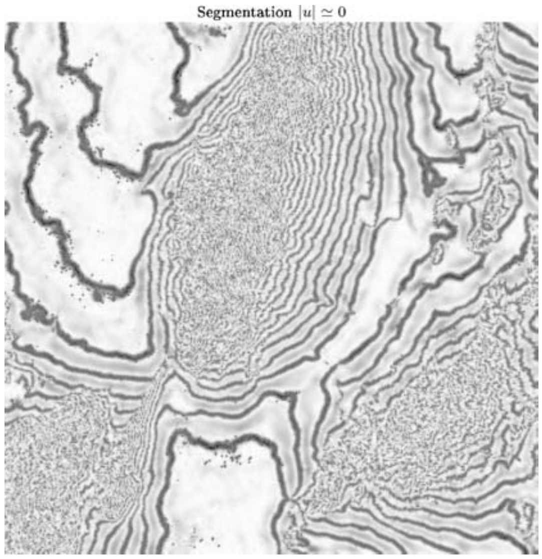


Figure 7. Segmentation of the SAR image of Figure 15: the proposed model merges the detected points into lines.

display the result we get with the W-detector (7). Our model clearly compares well. On Fig. 2, we have added a Gaussian noise (respectively with standard deviation $\sigma = 10$, 30 and 50). The algorithm still performs well. In practice, to create the noisy images, we have first added the Gaussian noise to f, and then rescaled it to the interval $[-1, 1]$. As the level of the noise increases, our model seems to compare well with the W-detector.

On Fig. 3, we show what happens on a real image: we catch the dashes of the leopard. Figure 4 is the same example, but with an additive Gaussian noise with standard deviation $\sigma = 20$. On both examples, the shape of the leopard is given by its dashes. Figure 3 also clearly illustrates the role of the μ parameter: the larger μ is, the more regularized the solution is.

Figure 5 shows an example of a biologic image. Although we use the model (4) which is designed to catch points, we nevertheless detect lines as sequence of points.

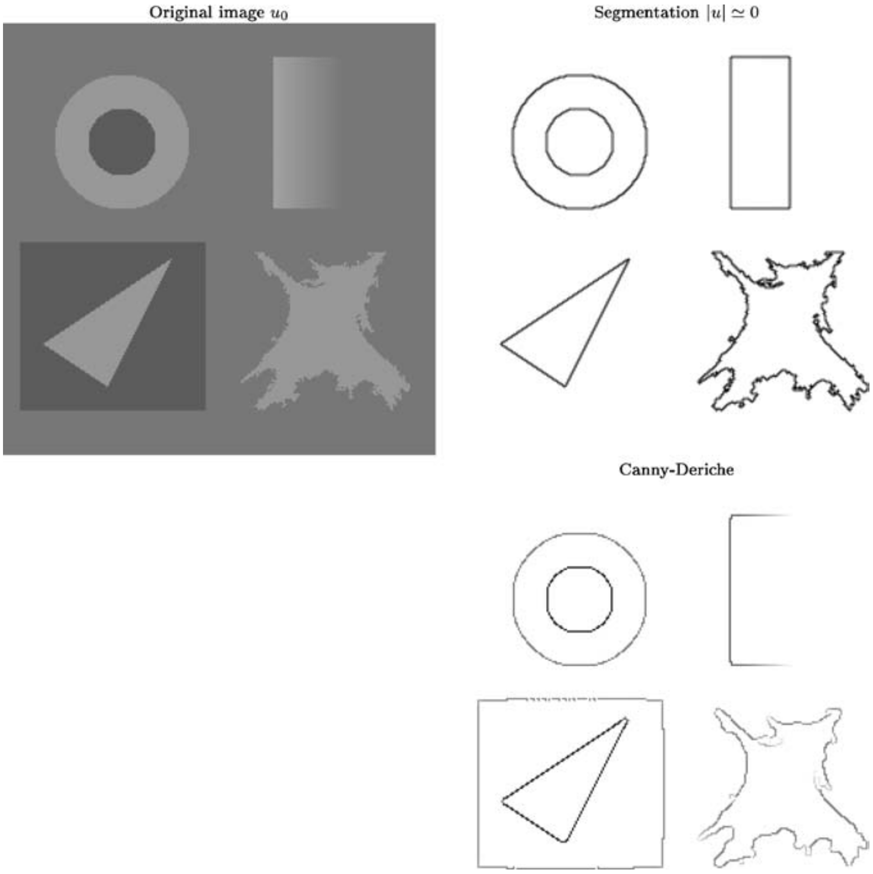


Figure 8. A synthetic image: with the proposed model, the edges are well-detected except for the square.

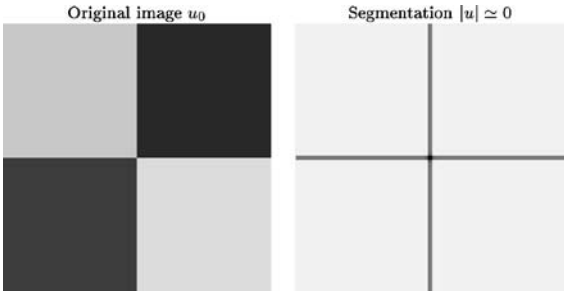


Figure 9. Quadruple junction (good case): in this situation, the proposed model can deal with a quadruple junction.

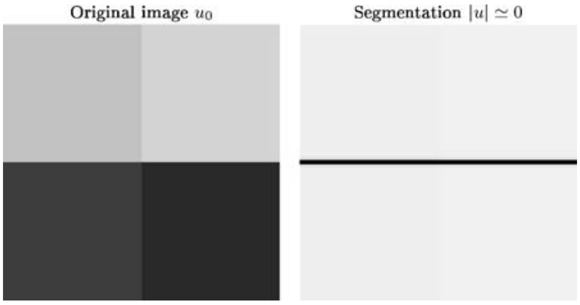


Figure 10. Quadruple junction (wrong case): in this situation, the proposed model cannot deal with a quadruple junction. This is the reason why the square is not segmented in Fig. 8.

Finally, Figs. 6 and 7 shows an application to SAR interferometry (Lewis, 1998). In this case too, the lines are formed by sequences of points. We will come back to this application in the next section. It should be noticed that with our model, the detected points seem to be merged into lines in this case.

4. Detection of Codimension-1 Structures in 2D-Images

As we mentionned in the introduction, in this case, we do not work with complex-values function but with scalar-values functions and we search for a minimizer of problem (4).

In the existing literature, there are many approaches to detect lines in an image. For instance, based on the gradient, there is the classical Canny-Deriche approach (Aubert and Kornprobst, 2002). There has also been a lot of approaches using snakes and active contours (Aubert and Blanc-Feraud, 1999; Aubert and Kornprobst, 2002; Caselles et al., 1993, 1997; Malladi et al., 1994; Osher and Fedkiw, 2001; Sussman et al., 1994).

We propose here a powerfull algorithm to catch curves in a 2-D image. Comparing with active contour methods, our new algorithm can catch nonclosed curves, and the initilization is completely automatic.

4.1. Evolution Equation

As before, we embed PDE (5) into a dynamical scheme:

$$\begin{aligned}\frac{\partial u}{\partial t} = \mu \operatorname{div}(a(x)u) + \frac{1}{\epsilon^2}u(1-|u|^2)&\\ -\lambda(u-u_0) \quad \text{in}\,\Omega&\end{aligned} \qquad (13)$$

with $\frac{\partial u}{\partial \vec{n}} = 0$ on $\partial\Omega$. Moreover, we impose $u(t=0,x) = u_0(x)$. λ and μ are positive weighting parameters. We then discretize the PDE (13) with finite differences.

In Section 3 we had chosen $a(x) = W(\Delta f)$ (see Eq. (6)). As the singularities we seek are no longer points but lines, we now choose:

$$a(x) = W(\nabla f) \qquad (14)$$

where f is the initial gray level image and W is the same function as in Section 3 (see Eq. (7)). Let us remark that lines can be detected as maximum of the absolute value of the gradient.

We use the same numerical scheme as in Section 3 (see Eq. (11)), but now the unknown u is a scalar function.

$$\begin{aligned}u^{n+1}_{i,j} = u^n_{i,j} + \delta t\Bigg(\mu\Delta u^n_{i,j} + \frac{1}{\epsilon^2}u^n_{i,j}\left(1-|u^n_{i,j}|^2\right)&\\ -\lambda\left(u^n_{i,j} - u^0_{i,j}\right)\Bigg)&\end{aligned} \qquad (15)$$

The initialization u_0 is the original image which has been rescaled between -1 and 1.

4.2. Numerical results

We set the parameters in the same way as in the preceeding section. We have decided to compare our model with the classical Canny-Deriche algorithm. We have used the implemantation in Megawave2 (http://www.cmla.ens-cachan.fr/Cmla/Megawave/).

Figure 8 is an example of segmentation of an image without any noise. One can see that it gives very good edges (comparing with the Canny-Deriche edge detector). The only problem is that our algorithm does

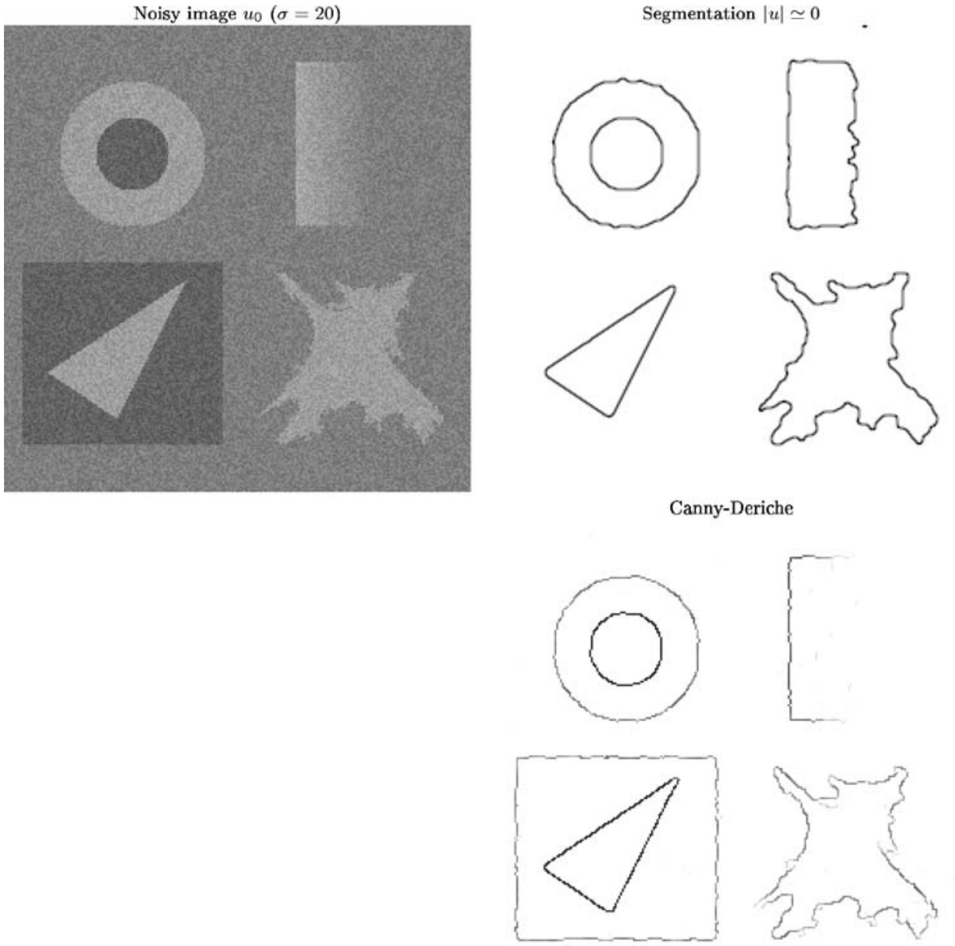


Figure 11. A noisy image: the result is to be compared with the one of Fig. 8 (which corresponds to the noise free case). Because of the noise, the fine details are not segmented as well as in Fig. 8, but the result is still good.

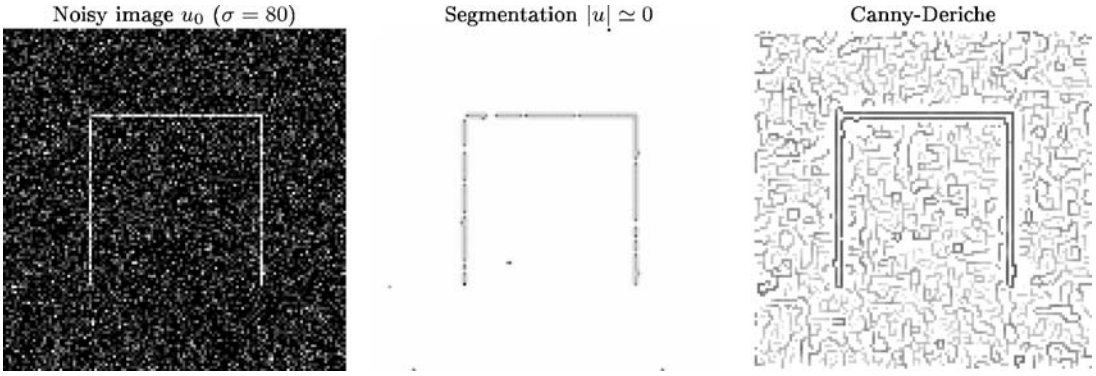


Figure 12. A nonclosed curve: even though the noise standard deviation is quite high, the proposed model is able to find the curve.

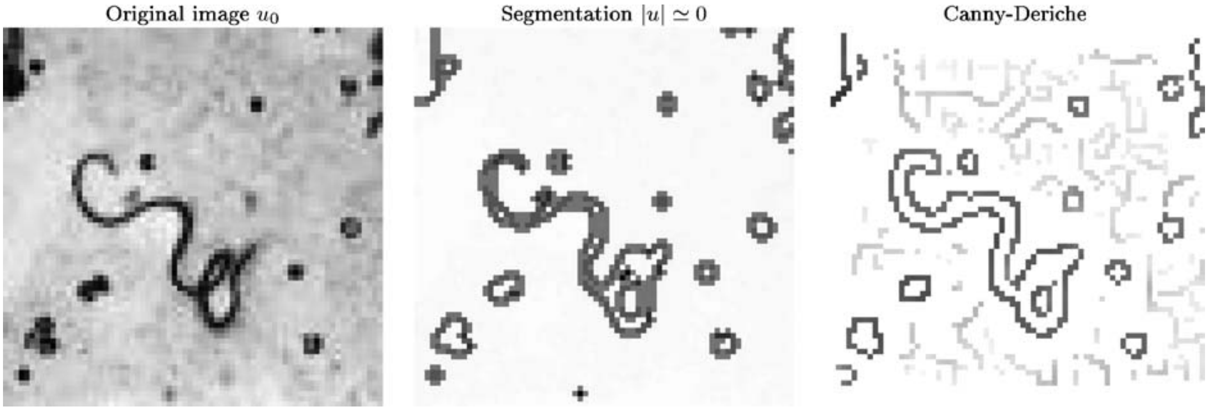


Figure 13. Biological lines: the original image is the same as the one of Fig. 5. This illustrates the difference between the proposed model to detect lines and the proposed one to detect points.

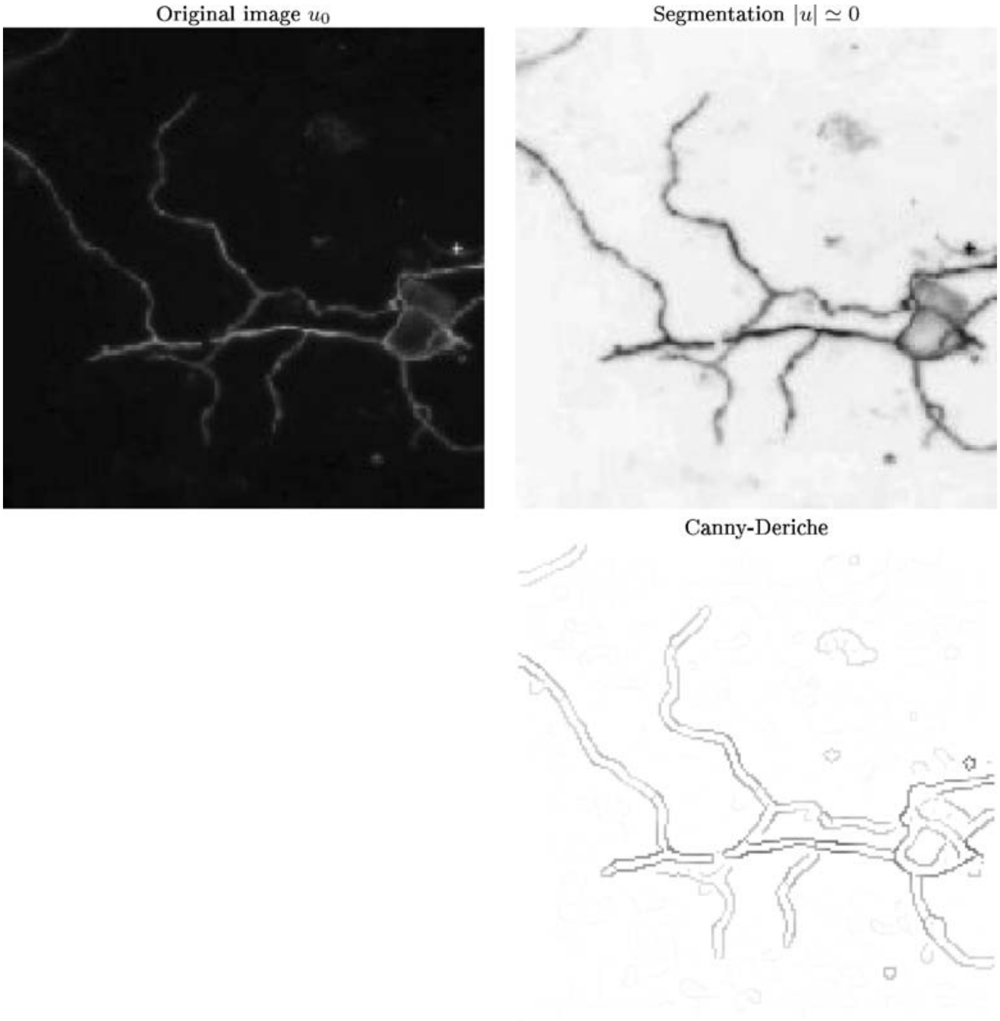


Figure 14. Biological image: the structures (which are lines in this image) are well segmented by the proposed algorithm.

not detect the square. We illustrate more precisely this problem on Figs. 9 and 10. Figure 9 gives an example of segmentation with a quadruple junction. In this case, the algorithm performs very well. But one can see on Fig. 10 that our model cannot handle any quadruple junctions. In fact, in many situations, one needs more than two phases. Here, for quadruple junctions we need four phases. In other words, the attracting term in the G-L functional must have four potential wells. Notice that for n-junctions, $n \geq 4$, we only need four potential wells. We are currently working on modifying our functional in this direction.

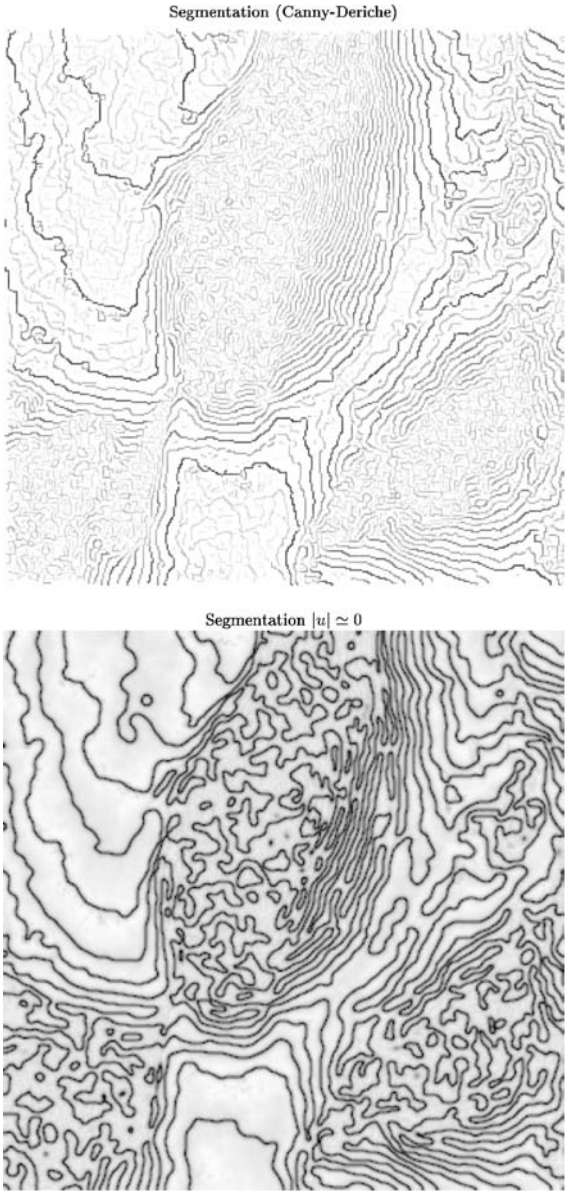
Figure 11 is an example of segmentation of the image of Fig. 8 with some Gaussian noise (with standard deviation $\sigma = 20$). The result is still a very good one even if the noise is quite strong.

On Fig. 12, we have tested our algorithm on a non-closed curve. We can see that we can detect it, even when there is a strong Gaussian noise (with standard deviation $\sigma = 80$).

Figure 13 is to be compared with Fig. 5. One clearly sees that both models do not perform the same way. The model of this section tries to find lines in an image, whereas the model of the previous section aims at finding points. Therefore, in this case, the points are represented by circles, and the lines by their edges (since they have a too large width with respect to the parameter ε which has been used).

Figure 14 shows a segmentation result on a biological image. Contrary to Fig. 13, the lines are represented by a single curve (as in Fig. 12). This comes from the width of the lines to be detected and the value of ε ($\epsilon = 0.5$ in this case). If we set ε smaller, then the lines are considered as objects with non-negligeable width (as in Fig. 13 or 8).

We come back to SAR image application (Lewis, 1998). We use our algorithm on the same interferometric image as in Fig. 7 (where the original image is displayed). On Fig. 15, we see that we get too many lines (in fact, we get twice as many line as we should want). One way to correct this problem is to use the result we get with the point version of the algorithm (Fig. 7). We just multiply both results (i.e. the gray level value of the final result in position (i, j) is the product of the gray-level value of the point detection in Fig. 7 in position (i, j) and the gray-level value of the segmentation in Fig. 15 in position (i, j)), so that we just keep the lines we want (see Fig. 16).


Segmentation (Canny-Deriche)

Segmentation $|u| \simeq 0$

Figure 15. Segmentation of the SAR image of Fig. 15. This result is to be compared with the one of Fig. 7. With our model to detect lines, we get twice as many lines as we should.

Computation Time. As we have explained before, we tune the time discretization step δt and the small stopping criterion *TOL* in the same way as in the preceeding section. With such parameters, it takes less than

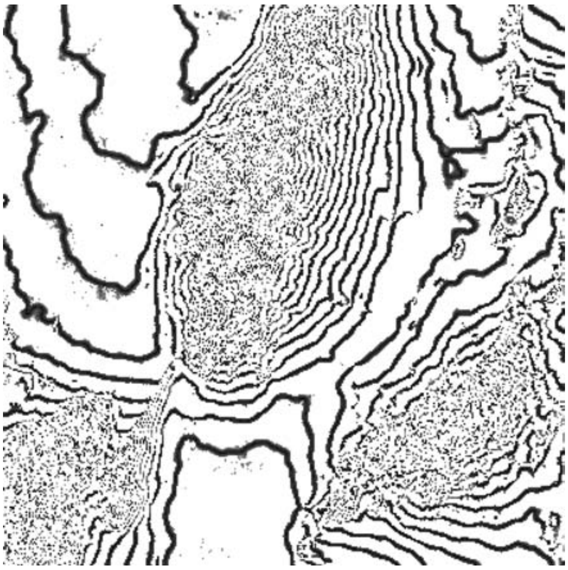


Figure 16. SAR image segmentation using both models: this result is obtained with a pixel by pixel multiplication of the results of Fig. 7 (with our model to detect points) and of Fig. 15 (with our model to detect lines).

15 seconds to process a $256 * 256$ image (in this case, the algorithm is faster since there is just one equation). This is of course longer than a line detector such as the Canny-Deriche one. But it compares pretty well to the methods based on snake approaches. For instance, in the case of geodesic active contours, the complexity is then higher since there is a periodic reinitialization procedure (Aubert and Kornprobst, 2002).

5. Conclusion and Future Prospects

In this paper, we have displayed some experimental results using Ginzburg-Landau functionals for the detection of objects of codimension 2 or 1 in a 2-D image. We got a new model to carry out such tasks. We have also stated some mathematical results about GL models. However these results are not directly applicable to our functional since we add a data term and we have to study theoretically the behaviour of u_ϵ as ϵ tends to 0. This will be made in a future work. Our numerical results confirm the interest in using such an approach. From a numerical point of view, we also have to go further into the tuning of the parameter ε. It is closely related to the mesh-size h. We conjecture that a relation of the type $h = O(\epsilon)$ must hold for ensuring the convergence of the discrete functional to the continuous one's. That is why we choose ε close to 1 in our experiments since classically in image processing $h = 1$. This type of results have been pointed out for similar problems in Aubert et al. (2004).

Acknowledgment

We thank Philippe Lecomte (INRA Sophia-Antipolis, France) for providing us with some biological images.

References

Alberti, G., Baldo, S., and Orlandi, G. 2003. Variational convergence for functionals of Ginzburg-Landau type. Preprint.

Ambrosio, L. and Soner, H.M. 1996. Level set approach to mean curvature flow in arbitrary dimension. *Journal of Differential Geometry*, 43.

Aubert, G. and Blanc-Feraud, L. 1999. Some remarks on the equivalence between 2D and 3D classical snakes and geodesic active Contours. IJCV, 34(1):19–28.

Aubert, G., Blanc-Féraud, L., and March, R. 2004. Γ-convergence of discrete functionals with non-convex perturbation for image classification, to appear in the SIAM. *Journal on Numerical Analysis*.

Aubert, G. and Kornprobst, P. 2002. Mathematical problems in image processing, vol. 147 of *Applied Mathematical Sciences*, Springer-Verlag.

Bethuel, F., Brezis, H., and Helein, F. 1994. Ginzburg-Landau vortices. Progress in Nonlinear Differential Equations and Their Applications. Birkhauser.

Caselles, V., Catte, F., Coll, T., and Dibos, F. 1993. A geometric model for active contours. *Numerische Mathematik* 66:1–31.

Caselles, V., Kimmel, R., and Sapiro, G. 1997. Geodesic active contours. *IJCV* 22(1):61–79.

Chen, X.Y., Jimbo, S., and Morita, Y. 1998. Stabilization of vortices in the Ginzburg-Landau equation with variable diffusion coefficients. *SIAM Journal Math. Anal.*

Gilboa, G., Zeevi, Y.Y., and Sochen, N. 2001. Complex diffusion processes for image filtering. In *Scale-Space '01*, vol. 2106 of *Lecture Notes in Computer Science*.

Gilboa, G., Sochen, N.A., and Zeevi, Y.Y. 2004. Image enhancement and denoising by complex diffusion processes. *IEEE Trans. Pattern Anal. Mach. Intell.* 26(8):1020–1036.

Ginzburg, V. and Landau, L. 1950. On the theory of superconductivity. *Zhekspe r. Teo. Fiz*, 20.

Grossauer, H. and Scherzer, O. 2003. Using the complex Ginzburg-Landau equation for digital inpainting in 2D and 3D. In *Scale-Space '03*, vol. 1682 of *Lecture Notes in Computer Science*.

C. Harris and M. Stephens 1988. A combined corner and edge detector. In *Proceedings of the Fourth alvey vision conference*. Manchester, pp. 147–151.

Henderson L. 1998. *Principle and Applications of Imaging Radar*, vol. 2 of 3rd edition. J. Wiley and Sons.

Lorigo, L., Faugeras, O., Grimson, W., Keriven, R., and Westin, C.F. 1999. Co-dimension-two geodesic active contours for the segmentation of tubular structures. In *Int. Conf. Information Processing in Medical Imaging*. Visegrad, Hungary.

Malladi, R., Sethian, J.A., and Vemuri, B.C. 1994. Evolutionary fronts for topology-independent shape modeling and recovery. In *ECCV 1994*, vol. 800 of *Lecture Notes in Computer Science*, pp. 3–13.

Osher, S. and Fedkiw, R.P. 2001. Level set methods: An overview and some recent results. *Journal of Computational Physics* 169:463–502.

Ruuth, S., Merriman, B., Xin, J., and Osher, S. 1998. Diffusion-generated motion by mean curvatures for filaments, Technical Report 98-47, UCLA Computational and Applied Mathematics.

Sussman, M., Smereka, P., and Osher, S. 1994. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics* 114:146–159.