HAL
open science

Variational B-Spline Level-Set: A Linear Filtering Approach for Fast Deformable Model Evolution

Olivier Bernard, Denis Friboulet, Philippe Thevenaz, Michael Unser

To cite this version:

Olivier Bernard, Denis Friboulet, Philippe Thevenaz, Michael Unser. Variational B-Spline Level-Set: A Linear Filtering Approach for Fast Deformable Model Evolution. IEEE Transactions on Image Processing, 2009, 18 (6), pp.1179-1191. hal-00819449

HAL Id: hal-00819449

https://hal.science/hal-00819449v1

Submitted on 1 May 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Variational B-Spline Level-Set: A Linear Filtering Approach for Fast Deformable Model Evolution

Olivier Bernard, Denis Friboulet, *Member, IEEE*, Philippe Thévenaz, *Member, IEEE*, and Michael Unser, *Fellow, IEEE*

***Abstract*—In the field of image segmentation, most level-set-based active-contour approaches take advantage of a discrete representation of the associated implicit function. We present in this paper a different formulation where the implicit function is modeled as a continuous parametric function expressed on a B-spline basis. Starting from the active-contour energy functional, we show that this formulation allows us to compute the solution as a restriction of the variational problem on the space spanned by the B-splines. As a consequence, the minimization of the functional is directly obtained in terms of the B-spline coefficients. We also show that each step of this minimization may be expressed through a convolution operation. Because the B-spline functions are separable, this convolution may in turn be performed as a sequence of simple 1-D convolutions, which yields an efficient algorithm. As a further consequence, each step of the level-set evolution may be interpreted as a filtering operation with a B-spline kernel. Such filtering induces an intrinsic smoothing in the algorithm, which can be controlled explicitly via the degree and the scale of the chosen B-spline kernel. We illustrate the behavior of this approach on simulated as well as experimental images from various fields.**

***Index Terms*—Active contours, variational method, level-sets, B-spline, segmentation.**

I. Introduction

LEVEL-SET-BASED formulations have become a well-established tool in the field of image processing [1], [2], [3]. In image segmentation, level-set-based methods correspond to a class of deformable models where the shape to be recovered is captured by propagating an interface represented by the zero level-set of a smooth function which is usually called the level-set function. The evolution of the interface is generally derived through a variational formulation: the segmentation problem is expressed as the minimization of an energy functional that reflects the properties of the objects to be recovered. Formally, the minimization of this functional provides the evolution of the level-set function as a time-dependent partial differential equation (PDE) that is usually solved using finite-difference methods. These numerical schemes have been developed to obtain an accurate and unique solution; they involve upwind differencing, essentially non-oscillatory schemes borrowed from the numerical solutions of conservation laws and Hamilton-Jacobi equations [2].

O. Bernard and D. Friboulet are with CREATIS, INSA, UCB, CNRS UMR 5220, Inserm U630, Building Blaise Pascal, 7 avenue Jean Capelle, 69621 Villeurbanne Cedex, France (e-mail: bernard@creatis.insa-lyon.fr).

P. Thévenaz and M. Unser are with the Biomedical Imaging Group, École polytechnique fédérale de Lausanne, CH–1015 Lausanne VD, Switzerland (e-mail: michael.unser@epfl.ch)

As an alternative to this now well-known discrete scheme, we present in this paper a continuous approach where the level-set function is expressed as a continuous parametric function using B-splines. Let us note that B-splines have already been used in the context of variational active contours. In [4] the active contour is modelled as quadratic B-spline curves in order to introduce statistical shape knowledge into the Mumford-Shah segmentation functional. The philosophy of representing the active contour by a few control points has been pushed to the limit in [5], where a polygon is used to model the evolving curve and a polygon regularizer is introduced in order to avoid degeneracy in the course of the propagation. In another context [6], B-splines are used for shape gradient-based level-set evolution in order to constrain the domain transformations to a linear combination of a set of pre-defined transformations.

The idea of using a continuous representation of the level-set has recently been proposed in [7], [8], where the continuous representation is based on radial-basis functions (RBFs), which are then used to solve the level-set PDE. In contrast with these approaches, we start from the initial level-set energy functional and show that our formulation allows us to compute the solution as a restriction of the variational problem onto the space spanned by the B-splines—in other words, as a projection onto the space of B-splines. As a consequence, the minimization of the functional is directly obtained in terms of the B-spline coefficients.

By representing the level-set as a linear combination of continuous basis functions, we benefit from several interesting properties. This representation provides an overall control of the level-set (*i.e.*, over the whole computational domain of the level-set), in contrast to the narrow-band implementation usually associated to the finite-difference scheme. Moreover, the re-initialization step of the level-set used in most implementations may be avoided by constraining the propagation of the interface, via constraints on the coefficients of the linear expansion.

The use of B-splines as a basis for the level-set representation provides specific additional benefits. As shown in the sequel, the minimization of the level-set functional with respect to the B-spline coefficients may be expressed as a discrete convolution operation. Moreover, since the B-spline basis functions are separable, this convolution may be performed as a sequence of simple 1-D convolutions. This leads to a very efficient implementation and also yields clues for the following interesting interpretation of the optimization process: Each step of the level-set evolution can be seen as

a filtering operation with a B-spline kernel. Such filtering induces an intrinsic smoothing in the algorithm, which can be controlled explicitly via the degree and the scale of the chosen B-spline kernel. Given that the level-set implementation of active contours are sensitive to noise [9], [10], this provides an efficient way to deal with noisy feature images and yields a more robust segmentation.

The noise sensitivity of the level-set method has been addressed in several recent works using two different types of approaches. The first type of approaches modify the data term of the functional: a region approach is used to describe image information through an a priori statistical model, such as the exponential family of distribution [11], [12], [13] or the Generalized Gaussian [14], [15]. The second kind of approaches works on the energy functional and the associated PDE. By solving the weak form of the level-set PDE Liu [10] proposed to perform smoothing of the level-set function and associated flow through local integration of the level-set function and external image forces. Very recent studies [9], [16] allow tackling this issue by observing that the definition of the energy functional gradient directly depends on the choice of an inner product structure. This inner product is most often chosen as the canonical L_2 inner product, which does not favour smooth evolution flows. These studies have generalized the approach by showing that using other inner products, such as Sobolev-type inner products, allows obtaining smooth gradient flow. Our approach differs from the above mentioned studies in that we start from a level set formulation and obtain a smooth solution as the result of constraining the solution to be a projection the space spanned by the B-splines.

Finally, let us note that the above mentioned set of characteristics yields also interesting properties in terms of computational speed. As noted in the very recent paper by Shi *et. al.* [17], the speed of the usual finite difference narrow band implementations of the level set PDE is mainly linked to numerical stability, which implies small time steps [18], [19] and periodical re-initialization of the level-set function. Moreover, many applications require the calculation of level-set curvature which is costly. By contrast, due to the smooth, continuous B-spline representation of the level-set, we may use large steps (thus need much less iterations to reach convergence), re-initialization is replaced by an efficient re-normalization of the B-spline coefficients and the computation of curvature term may be skipped in most cases. These aspects yield a fast algorithm, as illustrated in the experimental section.

The paper is structured as follows. In Section II, we recall the general form of the level-set energy functional, describe the B-spline formulation of the problem, and derive the minimization of the functional in terms of the B-spline coefficients. In particular, we show how the gradient of the functional—hence the evolution of the level-set—may be expressed as a convolution. In Section III, we discuss the implementation issues of our method. We show in particular how the re-initialization of the level-set may be avoided by normalizing the B-spline coefficients. In Section IV, we evaluate the behavior of the method using simulated images as well as biological (fluorescence) and medical (CT) images. We give the main conclusions and perspectives of this work in Section V.

II. Problem Formulation

A. Energy Criterion

Let Ω be a bounded open subset of $\mathbb{R}^d$ and let $f : \Omega \mapsto \mathbb{R}$ be a given d-dimensional image. In the level-set formalism, the evolving interface $\Gamma \subset \mathbb{R}^d$ is represented as the zero level-set of a Lipschitz-continuous function ϕ of dimension $d+1$ that satisfies

$$\begin{cases} \phi(\mathbf{x}) > 0, & \forall \mathbf{x} \in \Omega_{\text{in}} \quad (1) \\ \phi(\mathbf{x}) < 0, & \forall \mathbf{x} \in \Omega_{\text{out}} \quad (2) \\ \phi(\mathbf{x}) = 0, & \forall \mathbf{x} \in \Gamma, \quad (3) \end{cases}$$

where Ω_{in} is a region in Ω bounded by $\Gamma = \partial\Omega_{\text{in}}$. The region Ω_{out} is defined as $\Omega \setminus \Omega_{\text{in}}$.

We now consider the classical problem of segmenting one object (possibly having several nonconnected components) from the background[1]. This problem is typically handled by the evolution of one level-set; its steady state partitions the image into two regions that delimit the boundaries of the object to be segmented. In this framework, a general expression of the energy functional that drives the level-set can be formulated as [20]

$$\begin{aligned} J(\phi) &= \nu_{\text{in}} \int_\Omega g_{\text{in}}(\mathbf{x}, \phi(\mathbf{x})) H(\phi(\mathbf{x})) \mathrm{d}x_1 \cdots \mathrm{d}x_d \\ &+ \nu_{\text{out}} \int_\Omega g_{\text{out}}(\mathbf{x}, \phi(\mathbf{x})) \left(1 - H(\phi(\mathbf{x}))\right) \mathrm{d}x_1 \cdots \mathrm{d}x_d \\ &+ \nu_{\text{c}} \int_\Omega g_{\text{c}}(\mathbf{x}, \phi(\mathbf{x})) \delta(\phi(\mathbf{x})) \left\| \boldsymbol{\nabla} \phi(\mathbf{x}) \right\| \mathrm{d}x_1 \cdots \mathrm{d}x_d, \quad (4) \end{aligned}$$

where the first and second terms (often referred to as region terms) are energy criteria attached to the inside and outside regions delimited by the interface Γ, respectively. The last term is an energy criterion attached to the interface Γ (often referred to as the contour term). The functions $g_{\text{in}}(\cdot)$ and $g_{\text{out}}(\cdot)$ describe the object region and the background region [21], [20], and $g_{\text{c}}(\cdot)$ is a function describing the contour [22]. The symbols H and δ correspond to the Heaviside and Dirac univariate functions, respectively. The variables ν_{in}, ν_{out}, and ν_{c} are some positive hyper-parameters.

B. B-Spline Level-Set Model

The model is obtained by expressing the level-set function $\phi(\mathbf{x})$ as the linear combination of B-spline basis functions [23]

$$\phi(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{Z}^d} c[\mathbf{k}] \, \beta^n(\frac{\mathbf{x}}{h} - \mathbf{k}). \quad (5)$$

In this expression, $\beta^n(\cdot)$ is the uniform symmetric d-dimensional B-spline of degree n. This function is separable and is built as the product of d one-dimensional B-splines, so that $\beta^n(\mathbf{x}) = \prod_{j=1}^d \beta^n(x_j)$. The knots of the B-splines are located on a grid spanning Ω, with a regular spacing given by h. The coefficients of the B-spline representation are gathered in $c[\mathbf{k}]$.

[1]The interested reader will find in [3] approaches that extend this method to several regions.

C. Functional Minimization

In the classical variational settings, the energy criterion is minimized with respect to the level-set function using either Euler-Lagrange equations [24] or the Fréchet/Gâteaux derivatives [20], [25]. In contrast with these approaches, we use the B-spline formulation (5) and perform the minimization with respect to the B-spline coefficients $c[\mathbf{k}]$. Such minimization implies computing the derivatives of (4) with respect to each B-spline coefficient $c[\mathbf{k_0}]$. We show in Appendix A that these derivatives may be expressed as

$$\frac{\partial J}{\partial c[\mathbf{k_0}]} = \int_\Omega w(\mathbf{x})\, \beta^n(\frac{\mathbf{x}}{h} - \mathbf{k}_0)\, \mathrm{d}x_1 \cdots \mathrm{d}x_d, \tag{6}$$

where

$$\begin{aligned} w(\mathbf{x}) \;=\; & \nu_{\mathrm{in}}\, (\frac{\partial g_{\mathrm{in}}(\mathbf{x}, \phi(\mathbf{x}))}{\partial \phi(\mathbf{x})}\, H(\phi(\mathbf{x})) \\ & + g_{\mathrm{in}}(\mathbf{x}, \phi(\mathbf{x}))\, \delta(\phi(\mathbf{x}))) \\ & + \nu_{\mathrm{out}}\, (\frac{\partial g_{\mathrm{out}}(\mathbf{x}, \phi(\mathbf{x}))}{\partial \phi(\mathbf{x})}\, (1 - H(\phi(\mathbf{x}))) \\ & - g_{\mathrm{out}}(\mathbf{x}, \phi(\mathbf{x}))\, \delta(\phi(\mathbf{x}))) \\ & + \nu_{\mathrm{c}}\, (\frac{\partial g_{\mathrm{c}}(\mathbf{x}, \phi(\mathbf{x}))}{\partial \phi(\mathbf{x})}\, \|\boldsymbol{\nabla}\phi(\mathbf{x})\| \\ & - \mathrm{div}(g_{\mathrm{c}}(\mathbf{x}, \phi(\mathbf{x}))\, \frac{\boldsymbol{\nabla}\phi(\mathbf{x})}{\|\boldsymbol{\nabla}\phi(\mathbf{x})\|}))\, \delta(\phi(\mathbf{x})). \end{aligned} \tag{7}$$

Here, $w(\mathbf{x})$ reflects the features of the object to be segmented and will be called the feature function in the sequel. It is worth noting that whenever the region and contour terms $g_{\mathrm{in}}(\cdot)$, $g_{\mathrm{out}}(\cdot)$, and $g_{\mathrm{c}}(\cdot)$ do not depend on the level set function, the derivatives of these functions with respect to ϕ in (7) vanish and $w(\mathbf{x})$ simplifies to

$$\begin{aligned} w(\mathbf{x}) \;=\; & (\nu_{\mathrm{in}}\, g_{\mathrm{in}}(\mathbf{x}) \,-\, \nu_{\mathrm{out}}\, g_{\mathrm{out}}(\mathbf{x}) \\ & -\, \nu_{\mathrm{c}}\, \mathrm{div}(g_{\mathrm{c}}(\mathbf{x})\, \frac{\boldsymbol{\nabla}\phi(\mathbf{x})}{\|\boldsymbol{\nabla}\phi(\mathbf{x})\|}))\, \delta(\phi(\mathbf{x})). \end{aligned} \tag{8}$$

The minimization of the energy criterion (4) with respect to the B-spline coefficients generally does not lead to a closed-form solution. In order to obtain a local minimum, we then perform a gradient-descent method which yields

$$\mathbf{c}^{(i+1)} = \mathbf{c}^{(i)} - \lambda\, \boldsymbol{\nabla}_{\mathrm{c}} J(\mathbf{c}^{(i)}), \tag{9}$$

where λ is the iteration step and $\boldsymbol{\nabla}_{\mathrm{c}}$ corresponds to the gradient of the energy relative to the B-spline coefficients, the components of which being given by (6) and (7).

Equation (6) yields an interesting interpretation of the minimization process. Let us define $\beta^n_h(x) = \beta^n(\frac{x}{h})$, which is the B-spline of degree n upscaled by a factor h. The expression of the energy gradient is then given by

$$\boldsymbol{\nabla}_{\mathrm{c}} J = \frac{\partial J}{\partial c[\mathbf{k}]} = \int_\Omega w(\mathbf{x})\, \beta^n_h(\mathbf{x} - h\,\mathbf{k})\, \mathrm{d}x_1 \cdots \mathrm{d}x_d. \tag{10}$$

This last equation shows that the computation of the gradient of J with respect to a set of B-spline coefficients may be interpreted as convolving the feature function $w(\mathbf{x})$ with the B-spline $\beta^n_h(\mathbf{x})$, and sampling the result with a period h.

D. Discretization: Gradient Calculation as a Discrete Separable Convolution

Since the image information is available only on a discrete grid, the practical computation of the level-set evolution implies the discretization of the gradient $\boldsymbol{\nabla}_{\mathrm{c}} J$ given in (10), *i.e.* the discretization of the feature function w and the B-spline basis.

1) Discretization of the feature function w: Discretization of w as given in (7) or (8) implies the approximation of the Heaviside function and the one-dimensional Dirac. We follow here a now classical framework [24], [25], [26], by considering H_ϵ and δ_ϵ, *i.e.* any C^1 approximations and regularizations of the Heaviside function H and Delta function δ, as $\epsilon \to 0$ and with $H'_\epsilon = \delta_\epsilon$. The detailed expressions of H_ϵ and δ_ϵ chosen for our implementation are given in the Experiments Section IV-A [2]. As noted in [27], this is equivalent to consider a regularized version of the functional J, *i.e.*

$$\begin{aligned} J_\epsilon(\phi) = & \nu_{\mathrm{in}} \int_\Omega g_{\mathrm{in}}(\mathbf{x}, \phi(\mathbf{x})) H_\epsilon(\phi(\mathbf{x}))\mathrm{d}x_1 \cdots \mathrm{d}x_d \\ & + \nu_{\mathrm{out}} \int_\Omega g_{\mathrm{out}}(\mathbf{x}, \phi(\mathbf{x}))\, (1 - H_\epsilon(\phi(\mathbf{x})))\, \mathrm{d}x_1 \cdots \mathrm{d}x_d \\ & + \nu_{\mathrm{c}} \int_\Omega g_{\mathrm{c}}(\mathbf{x}, \phi(\mathbf{x}))\delta_\epsilon(\phi(\mathbf{x}))\, \|\boldsymbol{\nabla}\phi(\mathbf{x})\|\, \mathrm{d}x_1 \cdots \mathrm{d}x_d, \end{aligned} \tag{11}$$

The associated feature function $w_\epsilon(\mathbf{x})$ is then simply obtained from (7) and (8) by replacing the Heaviside and the Dirac by their regularized counterparts H_ϵ and δ_ϵ. Let us call $w_\epsilon[\mathbf{u}]$ the corresponding discrete feature function, with $\mathbf{u} \in \mathbb{Z}^d$.

2) Gradient Calculation: $w_\epsilon[\mathbf{u}]$ being defined above, we immediately obtain the discrete version of the gradient (10) by applying the discrete B-spline formulation of [28]. The centered d-dimensional discrete B-spline of degree n is noted $b^n[\mathbf{u}]$; it is obtained by sampling its continuous version $\beta^n(\mathbf{x})$ at integer values. Similar to its continuous counterpart, the sequence $b^n[\cdot]$ is separable and is built as the product of d one-dimensional B-splines, so that $b^n[\mathbf{u}] = \prod_{j=1}^d b^n[u_j]$. By using an integer spacing of the knots (*i.e.*, $h \in \mathbb{N} \setminus \{0\}$), we may define the discrete B-spline of degree n scaled by a factor h as $b^n_h[\mathbf{u}] = \beta^n(\frac{1}{h}\,\mathbf{u})$.

The discrete version of the formulation is then obtained from (10) as

$$\langle \boldsymbol{\nabla}_{\mathrm{c}} J \rangle\, [\mathbf{k}] = \left\langle \frac{\partial J}{\partial c[\mathbf{k}]} \right\rangle = \sum_{\mathbf{u} \in \mathbb{Z}^d} w_\epsilon[\mathbf{u}]\, b^n_h[\mathbf{u} - h\,\mathbf{k}]. \tag{12}$$

The energy gradient thus corresponds to the convolution of the feature image and the B-spline, down-sampled by a factor h; put differently,

$$\langle \boldsymbol{\nabla}_{\mathrm{c}} J \rangle\, [\mathbf{k}] = (w_\epsilon * b^n_h)_{\downarrow h}\, [\mathbf{k}]. \tag{13}$$

This last expression provides an efficient way of calculating the gradient and thus the evolution of the level-set through (9). Since the B-spline kernel $b^n_h[\mathbf{u}]$ is separable, the gradient may indeed be computed as a simple series of d convolutions of the feature image with a one-dimensional B-spline kernel.

[2] Note that shape gradient-based approaches [20], [27] allow expressing the level-set evolution directly as a function of the level-set gradient, avoiding thereby such regularization

In effect, each step of the level-set evolution corresponds to filtering the feature image with a B-spline kernel—that is, performing a low-pass filtering operation. This induces an intrinsic smoothing in the algorithm. For a fixed B-spline degree n, the amount of smoothing can be controlled by choosing the scale of the filter $b_h^n[\mathbf{u}]$ (*i.e.*, by choosing the knot spacing h). This provides a way to efficiently deal with noisy images.

III. Implementation Issues

A. Bounded Level-Set

In the course of propagation, the level-set function may develop steep or flat gradients, which, in turn, yield inaccuracies in the numerical approximation [3]. This is usually taken into account in classical implementations by reshaping the level-set through periodical re-initialization of the level-set function as the distance function to the zero level. This scheme has two main drawbacks:

- It increases the computational cost of the method;
- As noted in [3], it reduces the topological flexibility of the method since it prevents the level-set from creating new contours (*i.e.* , new zero-level components) far away from the initial interface.

As shown below, bounding the level-set function allows preventing steep gradients and avoids the re-initialization step. Moreover, due to the linearity of the expansion, bounding the level-set may be easily performed through a simple re-normalization of the expansion coefficients. This feature was used in [8] in the case of Radial Basis Function (RBF) by using the ℓ_1-norm of the expansion coefficients. Here, we propose instead to take advantage of the B-spline formulation and re-normalize the level-set by constraining the ℓ_∞-norm of the expansion. Such a procedure has the following advantages:

- It has a modest computational cost;
- It does not prevent the creation of new zero level components, thus making the solution topologically more flexible.

First, let us note that the multiplication of an implicit function by a non-null coefficient does not change its associated interface. Since ϕ is represented through the set c[k] of the B-spline expansion, multiplying ϕ by a coefficient ξ simply corresponds to multiplying $\{c[\mathbf{k}]\}$ by ξ. Furthermore, appendix B shows that the ℓ_∞-norm , $\|c\|_\infty$, of the B-spline coefficients provides us with a bound on the level-set function as:

$$|\phi(\mathbf{x})| < \|c\|_\infty \tag{14}$$

Hence, we can re-normalize the level-set function to the range $[-1, 1]$ by the following simple modification of the initial algorithm provided by (9):

$$\mathbf{c}^{(i+1)} \leftarrow \frac{\mathbf{c}^{(i+1)}}{\left\|\mathbf{c}^{(i+1)}\right\|_\infty}. \tag{15}$$

In Appendix C, we show that re-normalization also imposes a bound on the gradient norm of the level-set. This bound is

$$\| \nabla\phi(\mathbf{x}) \|_2 < (n+1)^d \; K_n \sqrt{d} \tag{16}$$

where $K_n = \max_x \left(\left| \frac{\partial \beta^n(\mathbf{x})}{\partial \mathbf{x}} \right| \right)$. K_n may easily computed for any degree n. For B-splines of degrees $n = 1$, 2 and 3, we have $K_1 = 1$, $K_2 = 1$ and $K_3 = 2/3$.

B. Gradient-Descent Algorithm

We minimize the energy given in (4) thanks to a gradient-descent algorithm with feedback step adjustment [29]. At each step (i), we use (9), (15), and the current estimate $\mathbf{c}^{(i)}$ to compute a candidate update $\mathbf{c}^{(i+1)}$ and its associated energy. If this update decreases J, then the step is considered successful, the corresponding B-spline coefficients $\mathbf{c}^{(i+1)}$ are accepted, and the step size λ is multiplied by a factor $\alpha_f \geq 1$. Else, we perform a more conservative update by dividing the step size by $\alpha_f^{'} \geq 1$, and we repeat the test.

As described in Section III-A, the range of ϕ is controlled during minimization thanks to the normalization procedure, which is applied every time the updated $\mathbf{c}^{(i)}$ is accepted. This strategy makes the traditional level-set re-initialization unnecessary.

Let us note here that other strategies than gradient descents have been proposed in order to obtain level set evolution. The very recent contribution of Grady [30] describes an approach based on combinatorial optimization instead of gradient descent in order to minimize the influence of undesirable local minima and obtain overall control of the level set evolution through non-local movement.

C. Computational Cost

The complexity of our algorithm depends on the support of the discrete B-spline $b_h^n[\mathbf{u}]$; more specifically, it essentially depends on its chosen degree n. The support of a 1-D discrete B-spline of degree n expanded by a factor h is given by $(n+1)\,h$. Calling d the number of dimensions and N^d the number of pixels of the image, the number of spline knots is then $(N/h)^d$. The cost of computing one update (9) of the B-spline coefficients $\mathbf{c}^{(i+1)}$ is dominated by the convolution (13). Because of separability, the downsampling can be applied on-the-fly, and the overall cost for updating all coefficients in (9) is $K_h = (n+1)\,N^d\,\frac{1-h^{-d}}{1-h^{-1}}$. For a sufficiently large h, it thus corresponds to about $K_\infty = (n+1)\,N^d$ operations for a pool of $(N/h)^d$ coefficients. Note that the total cost K does depend on $h \gg 1$, but only weakly so; at worst, it can reach $(K_\infty\, d)$ for $h = 1$.

Since the gradient-descent algorithm described above requires the computation of the energy through the sampled version of (4), it requires the evaluation of the level-set itself. By a similar analysis, the total cost of evaluating the level-set function expressed in (5) corresponds again to $(n+1)\,N^d$ operations. Note that the computational cost of the energy terms involved in the computation of the feature image $w(\mathbf{x})$ and of the energy J is application-dependent because of the terms $g_{\text{in}}(\mathbf{x}, \phi(\mathbf{x}))$, $g_{\text{out}}(\mathbf{x}, \phi(\mathbf{x}))$, and $g_{\text{c}}(\mathbf{x}, \phi(\mathbf{x}))$; it is therefore not expressed here.

IV. Experiments

In this section, we evaluate the proposed approach using a series of simulated images and biomedical images. The

interested reader will find numerous other examples and may test the segmentation algorithm in real time at the following website address: http://bigwww.epfl.ch/demo/levelset-spline/. The method has been developped using a Java platform.

A. Segmentation Functional

Our segmentation experiments are based on the Chan-Vese functional [24], which aims at partitioning the image into regions with piecewise-constant intensity. This approach corresponds to a particular case of the Mumford-Shah functional [31], known as the minimal-partitioning problem. The corresponding functional is given as

$$\begin{aligned} J_\epsilon(\phi, \mu_{\rm in}, \mu_{\rm out}) &= \int_\Omega \left(f(\mathbf{x}) - \mu_{\rm in}\right)^2 H_\epsilon(\phi(\mathbf{x})) {\rm d}x_1 \cdots {\rm d}x_d \\ &+ \int_\Omega \left(f(\mathbf{x}) - \mu_{\rm out}\right)^2 \left(1 - H_\epsilon(\phi(\mathbf{x}))\right) {\rm d}x_1 \cdots {\rm d}x_d \\ &+ \nu \int_\Omega \|\boldsymbol{\nabla}\phi(\mathbf{x})\| \, \delta_\epsilon(\phi(\mathbf{x})) {\rm d}x_1 \cdots {\rm d}x_d, \end{aligned} \tag{17}$$

where $(\mu_{\rm in}, \mu_{\rm out})$ are the two parameters of the energy function, and where ν is a hyper-parameter that balances the influence of the regions terms (first and second integrals) and of the contour term (third integral). The functions $H_\epsilon(\cdot)$ and $\delta_\epsilon(\cdot)$ are C^∞ regularized versions of the Heaviside and of the Dirac functions[3], respectively. The corresponding expression is

$$\begin{cases} H_\epsilon(x) = \dfrac{1}{2} + \dfrac{1}{\pi} \arctan(\dfrac{x}{\epsilon}) \\ \delta_\epsilon(x) = \dfrac{\rm d}{\rm dx} H_\epsilon(x), \end{cases} \tag{18}$$

where ϵ is a real positive constant that controls the scale of the regularized Dirac.

From Appendix A, the derivation of this criterion leads to the feature function

$$\begin{aligned} w_\epsilon(\mathbf{x}) \;=\; & \left(\left(f(\mathbf{x}) - \mu_{\rm in}\right)^2 - \left(f(\mathbf{x}) - \mu_{\rm out}\right)^2 \right. \\ & \left. \nu \, {\rm div}(\frac{\boldsymbol{\nabla}\phi(\mathbf{x})}{\|\boldsymbol{\nabla}\phi(\mathbf{x})\|})\right) \delta_\epsilon(\phi(\mathbf{x})). \end{aligned} \tag{19}$$

From (18) and (19), the evolution of the level-set is given by (9). In our implementation, the one-dimensional convolution filter was applied using mirror boundary conditions.

We estimate the parameters $\mu_{\rm in}$ and $\mu_{\rm out}$ thanks to a conventional Expectation Maximization (EM) technique which keeps ϕ (*i.e.*, $c[\mathbf{k}]$) fixed and minimizes the energy criterion with respect to $\mu_{\rm in}$ and $\mu_{\rm out}$. These parameters are then given as

$$\begin{cases} \mu_{\rm in} = \dfrac{\int_\Omega f(\mathbf{x}) \, H_\epsilon(\phi(\mathbf{x})) \, {\rm d}x_1 \cdots {\rm d}x_d}{\int_\Omega H_\epsilon(\phi(\mathbf{x})) \, {\rm d}x_1 \cdots {\rm d}x_d} \\ \mu_{\rm out} = \dfrac{\int_\Omega f(\mathbf{x}) \, \left(1 - H_\epsilon(\phi(\mathbf{x}))\right) \, {\rm d}x_1 \cdots {\rm d}x_d}{\int_\Omega \left(1 - H_\epsilon(\phi(\mathbf{x}))\right) \, {\rm d}x_1 \cdots {\rm d}x_d}. \end{cases} \tag{20}$$

[3]The interest of $\delta_\epsilon(\cdot)$ relies on its infinite support. As noted in [24], this makes the evolution term (13) act on all level curves at once, which reduces the sensitivity of the algorithm to the initialization.

B. Segmentation Parameters

The following settings are applied to all experiments:

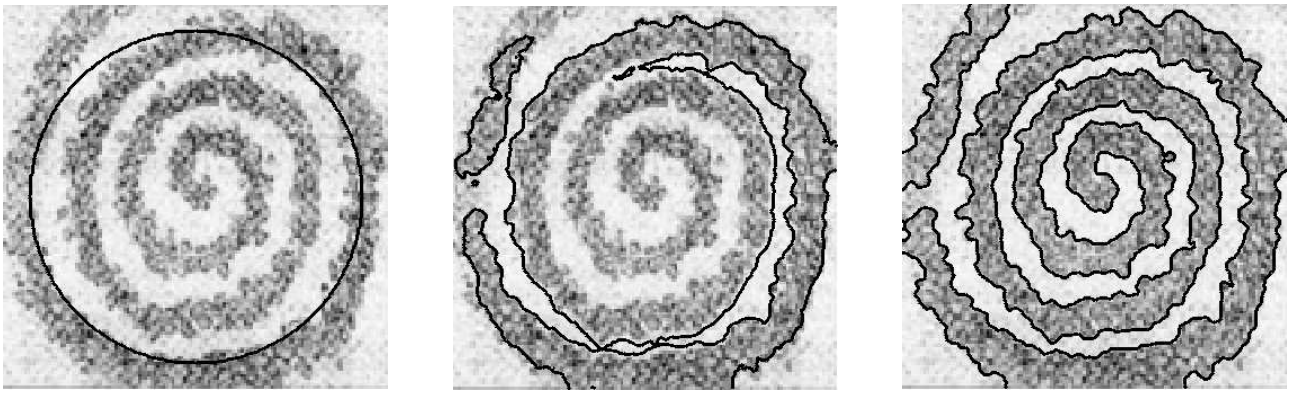
- All images have an eight-bit dynamics and their dimension is 256^d, where d is the number of dimensions;
- We use a cubic B-spline function as basis for the level-set representation. This function provides a good tradeoff between smoothing properties and computational cost. The length of the support of this function is equal to $4\,h$;
- The scale of the regularized Dirac defined in (18) has to be large enough, so that the evolution equation acts on all level curves and yields a global minimizer [24]. Because the level-set is bounded in the interval $[-1, 1]$, we simply set $\epsilon = 1$;
- The parameters that adjust the steps of the gradient-descent are fixed as $\alpha_f = 1$ and $\alpha_f^{'} = 1.5$;
- Unless otherwise mentioned, we initialize the level-set with the implicit function $\phi_0(\mathbf{x}) = 110 - \|\mathbf{x} - (128, 128)\|$.

In the experiments, we shall focus on the influence of the two main parameters of the method: the regularization factor ν, which weights the curvature term in the functional (see (19)), and the scale h, which controls the smoothing linked to the B-spline formulation (see (13)). Moreover, we give for each experiment the cpu time of the computation, performed on a $1.4\,$GHz Intel with $1\,$GB of RAM.

C. Segmentation Results

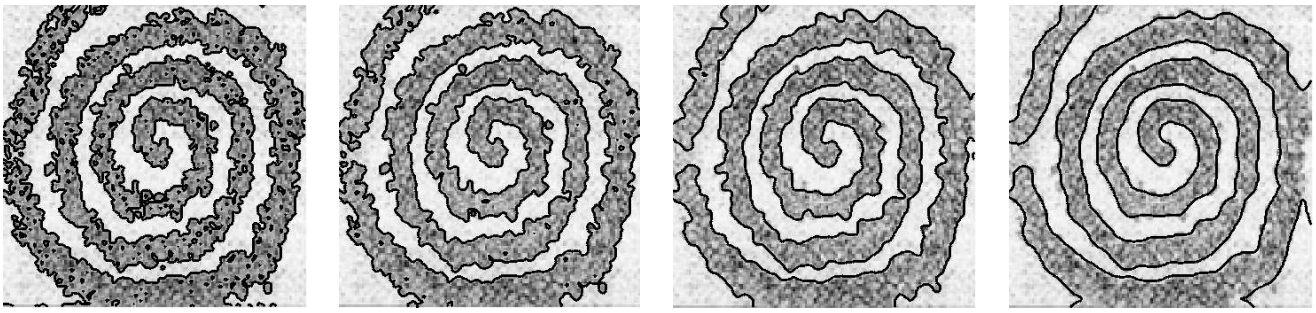
We show in Figure 1 the segmentation results obtained on an image containing a textured spiral, borrowed from [24], and corresponding to an art picture from the Los Angeles Times by Brian Forrest. From the initial solution, the zero-level interface propagates into the spiral and converges to the final curve after 46 iterations. In this case, we have applied the B-spline representation at full scale ($h = 1$), which corresponds to a knot placed at every pixel of the image of size 256^2. In this example, we have set the curvature regularization as $\nu = \frac{2}{25}\,255^2$. Thanks to the separable nature of the convolution formulation of the level-set evolution, we obtain the final result in only $20\,$s cpu time.

We illustrate in Figures 2(a)–(d) the influence of the scale of the level-set B-spline representation. The computations are performed for increasing scale and without curvature regularization (*i.e.*, $\nu = 0$ in (19)). The results demonstrate a corresponding increase in the degree of smoothness of the segmentation. Comparing these results to the previous ones (see Figure 1(d) and Figure 2(c)), it appears that the scale of the B-splines or the curvature term have a similar influence on the smoothness of the final shape. It is however interesting to note that it is much faster to segment without curvature regularization: The cpu time is equal to either $2\,$s or $3\,$s for scales $h \in [2 \ldots 8]$, whereas the cpu time is $20\,$s for the result shown in Figure 1(d). This difference is due to the fact that we spare the computation of the term corresponding to the curvature in the feature image (see (19)). Moreover, relaxing the curvature term makes the active contour less rigid; thus, the algorithm reaches the solution within fewer iterations.



(a) initialization (b) 11 iterations (c) final result, 46 iterations

Fig. 1. Segmentation of a textured spiral image. Full-scale model ($h = 1$) with curvature weight $\nu = \frac{2}{25}\,255^2$. (a) Initialization; (b) Level-set after 11 iterations; (c) Segmentation result obtained after 46 iterations, cpu = 20 s.



(a) Final result at scale 1 (50 iterations, cpu = 11 s) (b) Final result at scale 2 (8 iterations, cpu = 3 s) (c) Final result at scale 4 (6 iterations, cpu = 2 s) (d) Final result at scale 8 (11 iterations, cpu = 3 s)

Fig. 2. Segmentation of a textured spiral image with different scales and no curvature constraint ($\nu = 0$). (a) Segmentation result for $h = 1$; (b) Segmentation result for $h = 2$; (c) Segmentation result for $h = 4$; (d) Segmentation result for $h = 8$.

We illustrate in Figure 3 the behavior of the approach in the presence of various amounts of additive Gaussian noise and show the respective influence of the scale and curvature regularization. In Figures 3(d)–(f) (*i.e.*, second row), we show the results obtained at full scale ($h = 1$) using only the curvature term. Because this term corresponds to a purely geometric smoothing, it cannot provide satisfying segmentation when the amount of noise is important. Indeed, increasing the constraint (*i.e.*, increasing ν) beyond the values corresponding to those of Figures 3(d)–(f) cannot improve the results, since doing this would yield a quasi-rigid active contour which would not evolve from the initial solution anymore.

By contrast, Figures 3(g)–(l) (third and bottom row) hint that an increase of the scale of the B-spline is more successful at coping with an increase in the amount of noise. This illustrates the double influence of the scale parameter of the B-spline level-set:

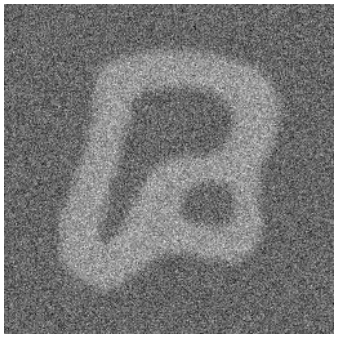
- A scale increase yields smoother shapes since this is equivalent to represent the implicit function of the level-set with fewer B-splines that have a larger support. Thus, this is comparable to the geometric smoothing provided by the curvature regularization mentioned above;
- The evolution of the B-spline level-set is formally equivalent to smoothing the feature image $w(\mathbf{x})$ with a B-spline kernel, as shown in (13). Consequently, increasing the scale (*i.e.*, the support of the kernel) leads to stronger smoothing, which in turn reduces the effect of the noise. Such a smoothing cannot be provided by the curvature term.

As previously noted, the results of Figure 3 show that the segmentation performed without the curvature term (see Figures3(d)–(f)) results in a faster computation.

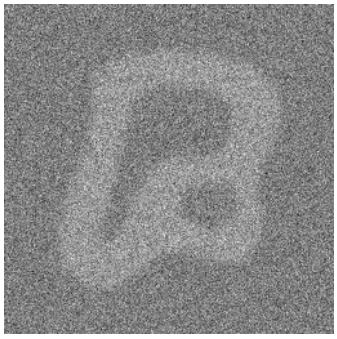
These above observations are quantitatively evaluated by comparing the obtained segmentation and the reference shape using the popular Dice coefficient [17], [32]. Noting S_S and S_R the segmented and the reference regions, the Dice coefficient DC is given as:

$$DC(S_S, S_R) = \frac{2\,Area(S_S \bigcap S_R)}{Area(S_S) + Area(S_R)} \qquad (21)$$

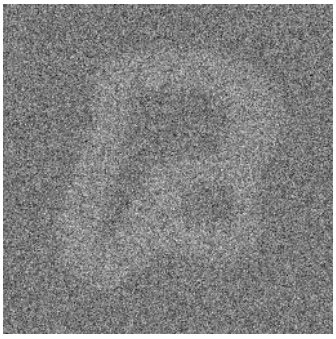
The Dice coefficient varies from 0 to 1: it is 1 when the two regions are identical and 0 when they are completely different. The corresponding results are given in Figure 4, where we



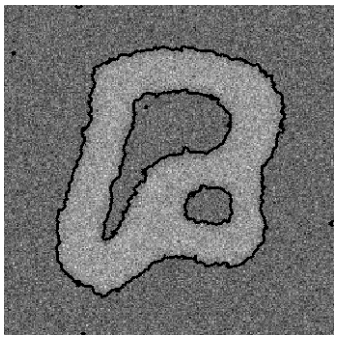
(a) Image at 25 dB



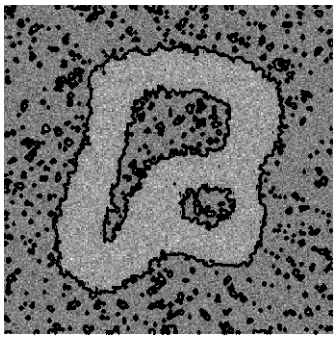
(b) Image at 20 dB



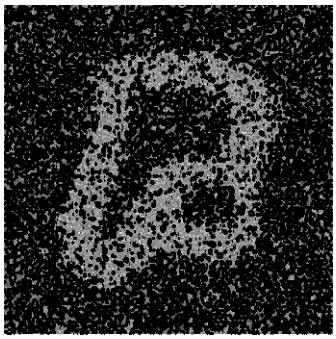
(c) Image at 15 dB



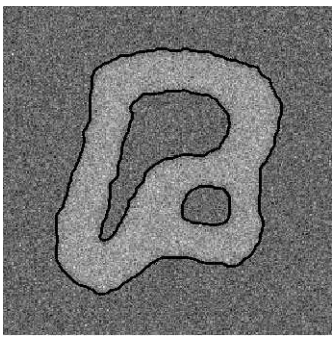
(d) $h = 1$, $\nu = \frac{3}{200}\,255^2$, cpu 14 s



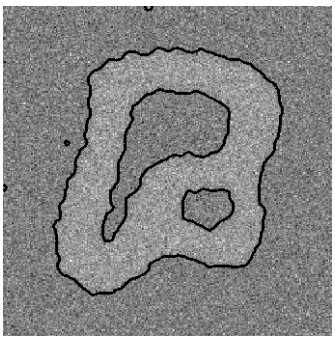
(e) $h = 1$, $\nu = \frac{3}{500}\,255^2$, cpu 18 s



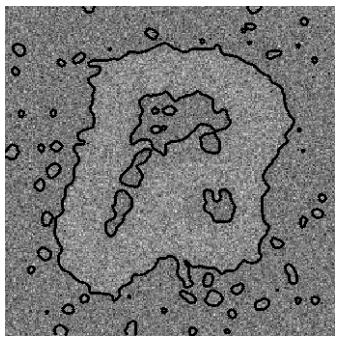
(f) $h = 1$, $\nu = \frac{1}{1000}\,255^2$, cpu 24 s



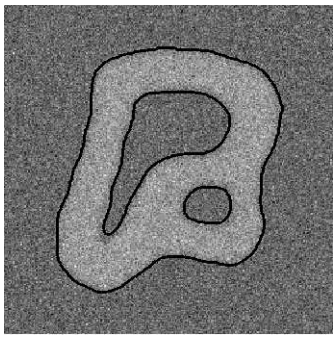
(g) $h = 4$, $\nu = 0$, cpu 3 s



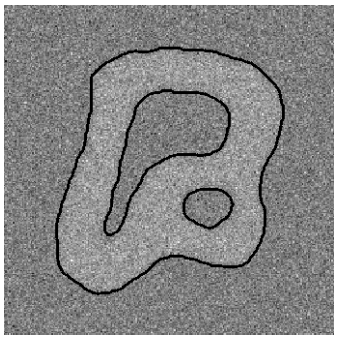
(h) $h = 4$, $\nu = 0$, cpu 3 s



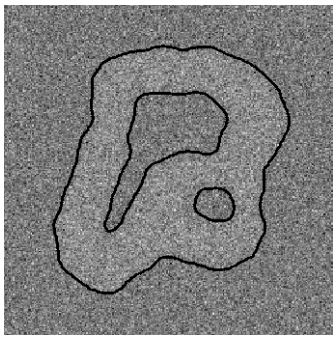
(i) $h = 4$, $\nu = 0$, cpu 3 s



(j) $h = 8$, $\nu = 0$, cpu 4 s

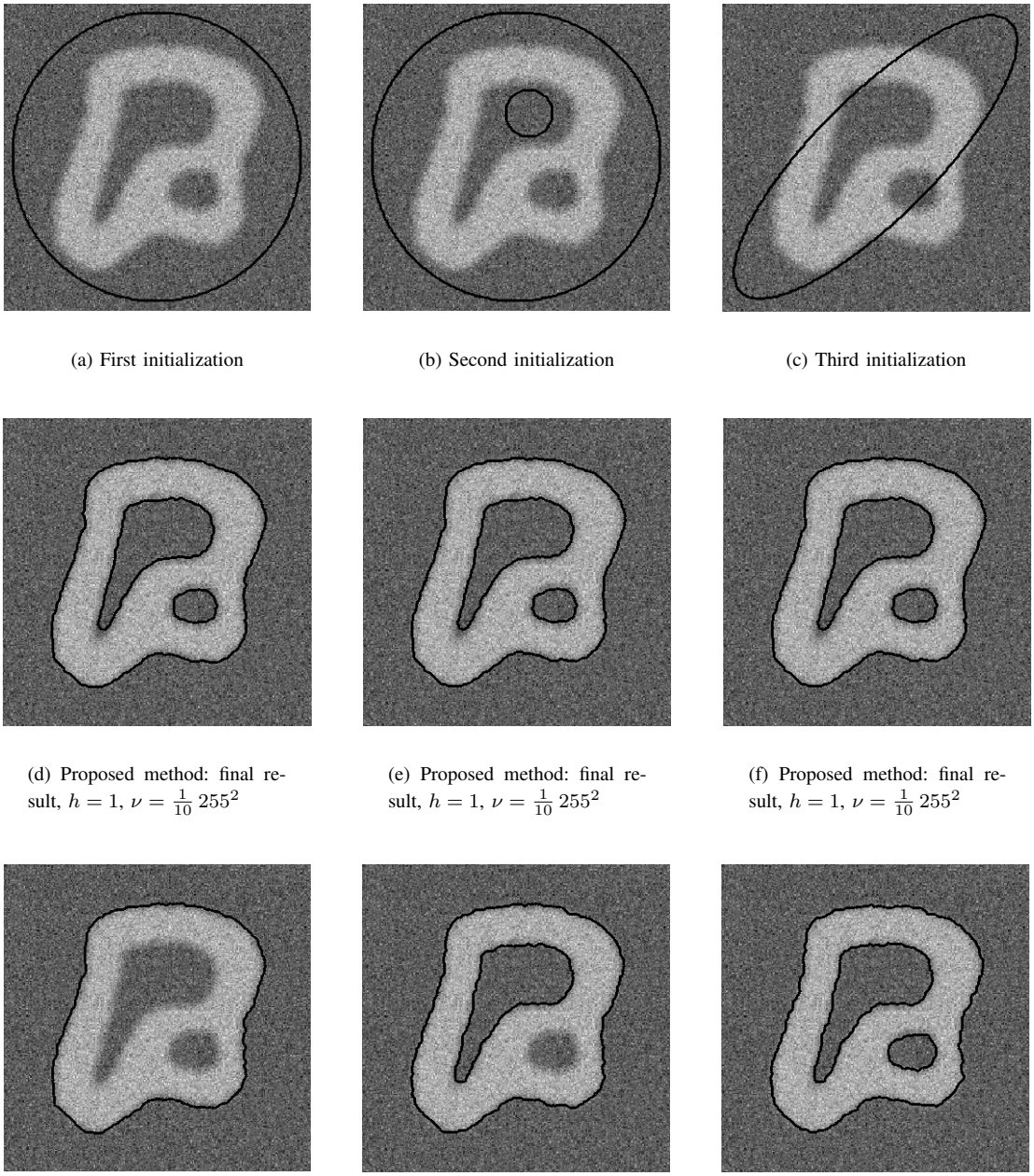


(k) $h = 8$, $\nu = 0$, cpu 3 s



(l) $h = 8$, $\nu = 0$, cpu 3 s

Fig. 3. Segmentation of images with different SNR values. Each column corresponds to a given noise level (left: 25 dB, middle: 20 dB, right: 15 dB); each row corresponds to a given scale. Top row: Original images. Second row: full scale ($h = 1$) with curvature term. Third row: $h = 4$, no curvature term. Bottom row: $h = 8$, no curvature term.



(a) First initialization

(b) Second initialization

(c) Third initialization

(d) Proposed method: final result, $h = 1$, $\nu = \frac{1}{10}\,255^2$

(e) Proposed method: final result, $h = 1$, $\nu = \frac{1}{10}\,255^2$

(f) Proposed method: final result, $h = 1$, $\nu = \frac{1}{10}\,255^2$

(g) Classical level-set method: final result, $\nu = \frac{1}{10}\,255^2$

(h) Classical level-set method: final result, $\nu = \frac{1}{10}\,255^2$

(i) Classical level-set method: final result, $\nu = \frac{1}{10}\,255^2$

Fig. 5. Segmentation of an image at 30 dB with different initializations. Each column corresponds to a given initialization. Top row: original images with the level-set initialization. Second row: segmentation results using the proposed approach. Bottom row: segmentation results using a classical level-set method [33].

have plotted the Dice coefficient, for various amounts of additive Gaussian noise as a function of the scale parameter h. Each Dice coefficient has been estimated by averaging 20 segmentation results obtained on different noise realizations. From Figure 4 the following can be observed. When the noise level is very low (30 dB), only weak smoothing is needed and low values for h provide the best results. Increasing h yields oversmoothing of the shape and thus a decrease of the Dice coefficient. At higher noise levels, the amount of smoothing has to be increased, thus the best results corresponds to higher values of h (*i.e.* 2 for 25 dB, 4 for 20 dB and 8 for 15 dB). Increasing h beyond these values yields again oversmoothing. Note also that even for these "ideal" values the Dice coefficient decreases with the noise level: this is consistent with the fact that the details of the object cannot be recovered at such noise levels.

We next illustrate the interest of bounding the level-set using the normalization procedure described in Section III-A. We show in Figures 5(d)–(f) the segmentation results obtained on a shape having multiple components using different ini-

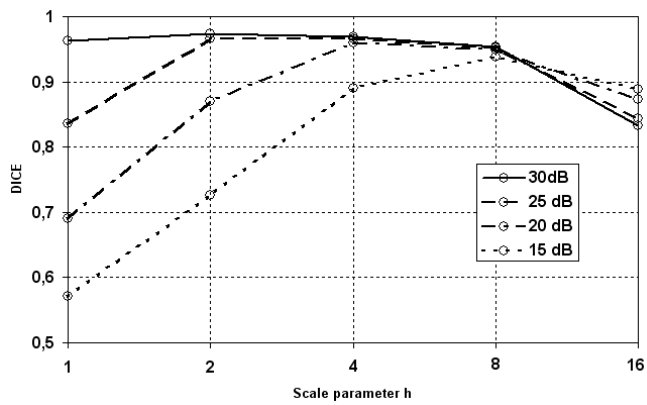


Fig. 4. Dice coefficient in function of the scale parameter for segmentation of images having various amounts of additive Gaussian noise. Each dice value has been estimated on 20 realizations.

tializations (see Figures 5(a)–(c)). By bounding the level-set, we avoid the periodic re-initialization of the implicit function and make the level-set topologically more flexible: Whatever the initialisation, the level-set develops the necessary new contours (*i.e.*, new zero-level components) and provides the proper segmentation of the object. In Figures 5(g)–(i), we show the segmentation results obtained using a classical level-set algorithm [33] where the implicit function is represented through a signed distance function defined on a narrow band around the zero level. At each iteration, an optimized re-initialization strategy is used to keep the signed-distance properties over time. The results clearly show that such algorithms have difficulties to develop new components and thus strongly depend on the initialization.

We give in Figure 6 an example of applying our approach on a fluorescence micrograph that contains labeled yeast cells. As the shapes to be detected are smooth, we set the scale of the B-spline level-set to $h = 2$, and we do not use the curvature term by setting $\nu = 0$. We show the final segmentation in Figure 6(c), where multiple components are detected thanks to the topological flexibility of the level-set.

We consider in Figure (7) a satellite image of European night lights. Its proper segmentation is a challenging problem because we would like to retain the small details of the shape of the coast borders, while the regions corresponding to solid ground are inhomogeneous. In this case, we have segmented the image by using a moderate scale ($h = 2$), with the assistance of a non-vanishing curvature term. We have shown the evolution of the zero level in Figures 7(a) to 7(c), and we have given the associated inside and outside regions in Figures 7(d) to 7(f).

As shown in Figure 8, we have also applied the proposed segmentation approach to a 3-D image of a calcaneus bone, acquired on a micro-CT with a voxel size of $80\,\mu\mathrm{m}^3$. The segmentation was obtained at full scale without curvature term to preserve the details structure of the calcaneus bone structure. We provide in Figure 8 a 3-D visualization of the resulting segmentation, as well as three image slices. These results show the ability of the model to handle complex topology.

D. Computation Cost Results

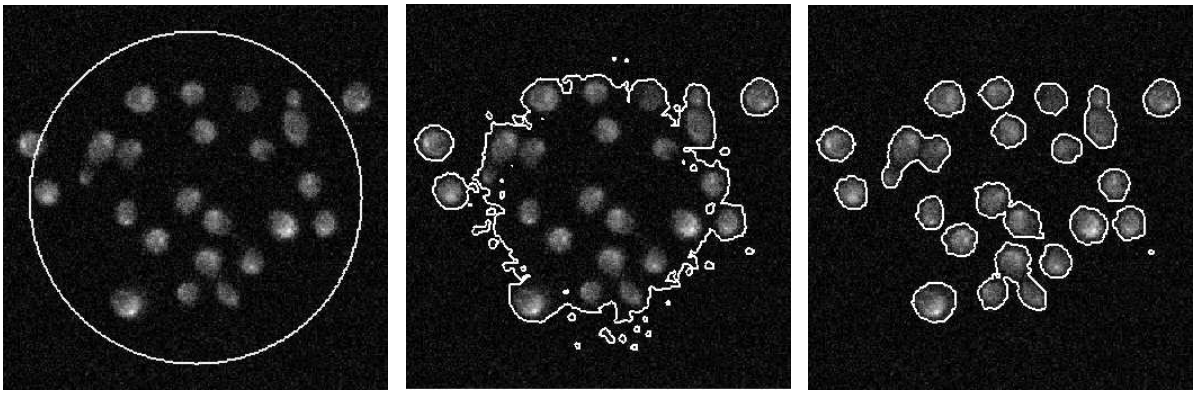
The interest of the proposed approach in terms of computational cost relies on the following:

- Because we use a continuous, smooth representation of the level-set through B-splines, we may use large steps and thus need much less iterations (on the order of several ten at worst), as compared to finite difference implementation which demands very small time steps to maintain numerical stability, yielding a large number of iterations (generally on the order of several hundred).
- As detailed in section III-A, we do not re-initialize the level-set, but use re-normalization of the coefficients.
- The smoothness of the solution is implicitly enforced, through the intrinsic smoothness of the B-spline representation, which may be adjusted through parameter h. As a consequence, the use of the costly curvature term may be avoided in most cases.

We propose to further quantify the above observations by comparing our method with the speed-optimized fast two-cycle (FTC) algorithm recently described in [17]. This algorithm is composed of a data-dependent term and a curve smoothness regularization term. The level-set evolution process is separated into two different cycles: one cycle for the data-dependent term and a second cycle for the smoothness regularization. The speed of the FTC algorithm then results from two main features: it uses a discrete approximation of narrow band level-set which yields evolution without the need of solving partial differential equations (PDE*s*) and the smoothing curvature term is approximated by a Gaussian filtering process. The FTC algorithm is thus controlled via four parameters: the number iterations of the data-dependent cycle N_a, the number of iterations of the smoothing cycle N_s, the variance of the Gaussian filter σ^2, and the number of points used to numerically approximate the filter N_g. As a consequence, the smoothness of the final segmentation increases with N_s, σ^2 and N_g.

The algorithms have been applied to the image shown on Figure 9, which was constructed by adding a small amount of Gaussian noise (SNR of 30 dB) to a binary 256^2 image of a leaf. The data-dependent term used in the tests is the Chan-Vese region terms, as given in (17) and used the previous results section. Note that this image has been selected such that the 2 methods may yields comparable results in terms of segmentation (*i.e.* there are no topological difficulties for the FTC narrow band method). The same initialization (a centred circle with radius 110 pixels) has been used. The parameters N_a, and N_s, of the FTC algorithm are set to 30 and 3, as in [17]. The remaining parameters of the 2 approaches have then been selected to yield similar segmentations.

The segmentations are quantitatively compared through the Dice coefficient [17], [32], using the leaf binary image as a reference. This allows controlling that the segmentation results are close enough for the running times comparison to be meaningful. The obtained segmentation are given in Figure 9 and Table I provides the corresponding results in terms of total running time, number of iterations, time spent to estimate the parameters μ_{in} and μ_{out} and Dice coefficient.



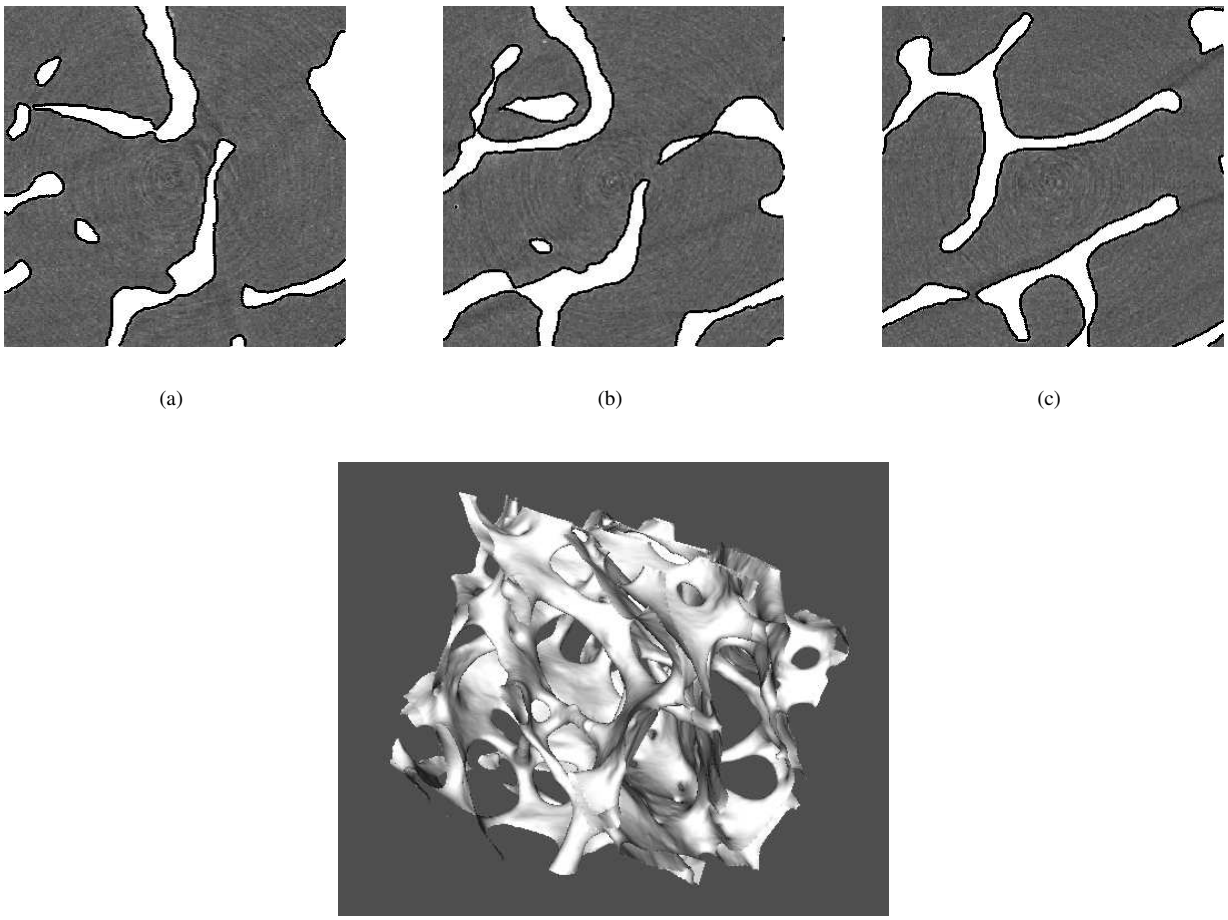
(a) Initialization (b) Three iterations (c) Final result, 29 iterations

Fig. 6. Segmentation of a fluorescence micrograph of yeast cells. The scale of the level-set model is $h = 2$, without any regularization ($\nu = 0$). (a) Initialization; (b) Level-set after 3 iterations; (c) Segmentation result obtained after 29 iterations. The cpu time is 5 s.

(a) Initialization (b) Three iterations (c) Final result, 15 iterations

(d) Inside and outside regions delimited by Γ at initialization (e) Inside and outside regions delimited by Γ at iteration 2 (f) Inside and outside regions delimited by Γ at iteration 10

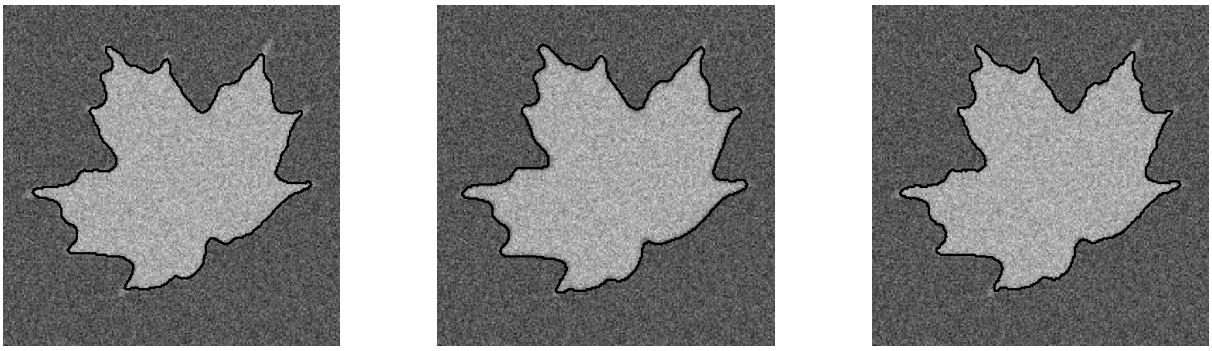
Fig. 7. European night lights. Level-set model at scale $h = 2$ with a curvature term ($\nu = \frac{1}{10} \cdot 255^2$). (a) and (d) Initialization; (b) and (e) Level-set after 3 iterations; (c) and (f) Segmentation result obtained after 15 iterations. The cpu time is 7 s.



(a) (b) (c)

(d)

Fig. 8. Segmentation of 3-D micro-CT images of a calcaneus bone. Level-set model at full-scale $h = 1$, without any curvature term ($\nu = 0$). (a)–(c) Three slices through the original data volume, along with the obtained contours; (d) 3-D rendering of the resulting segmentation. The cpu time is 540 s.



(a) FTC ($N_g = 5$, $\sigma^2 = 2$)

(b) B-spline level-set ($h = 4$, $\nu = 0$)

(c) B-spline level-set ($h = 1$, $\nu = \frac{2}{25}\,255^2$)

Fig. 9. Segmentation of the leaf image using the FTC and B-spline level-set algorithms

	Total running time(s)	Number of iterations	Param. estimation(s)	Dice coefficient
FTC ($N_g = 5$, $\sigma^2 = 2$)	1.68	82	1.33	0.985
B-spline level-set ($h = 4$, $\nu = 0$)	2.03	5	0.13	0.980
B-spline level-set ($h = 1$, $\nu = \frac{2}{25}\,255^2$)	15.48	33	4.19	0.985

TABLE I
COMPUTATION TIMES AND DICE COEFFICIENT CORRESPONDING TO THE SEGMENTATION OF THE LEAF IMAGE

	Total running time(s)	Number of iterations	Param. estimation(s)	Dice coefficient
FTC ($N_g = 9$, $\sigma^2 = 3$)	44.10	1573	33.96	
B-spline level-set ($h = 4$, $\nu = 0$)	1.86	5	1.16	0.926 (Compared to FTC)
B-spline level-set ($h = 1$, $\nu = \frac{2}{25}\,255^2$)	28.99	80	6.39	0.913 (Compared to FTC)

TABLE II
COMPUTATION TIMES AND DICE COEFFICIENT CORRESPONDING TO THE SEGMENTATION OF THE SPIRAL IMAGE. SINCE THERE IS NO REFERENCE FOR THIS IMAGE, THE DICE COEFFICIENT IS USED HERE TO COMPARE THE SEGMENTATION PROVIDED BY FTC WITH THE TWO SEGMENTATIONS PROVIDED BY THE B-SPLINE LEVEL SET

Figure 9 qualitatively show the closeness of the segmentation results, which is confirmed by the corresponding Dice coefficients displayed in the fourth column of Table I. Let us note that the running time corresponding to FTC (1.68s) is much larger than the time given in the original paper [17]. This is however quite natural, since the evolution requires the numerical evaluation of integrals over the image through (20) to estimate the parameters $\mu_{\rm in}$ and $\mu_{\rm out}$. On the opposite, in the paper [17] a simple threshold-based data-dependent term is used, which does not requires intermediate parameter estimation. This is confirmed in Table I which shows that the algorithm spends a large amount of time (1.33s) in estimating parameters $\mu_{\rm in}$ and $\mu_{\rm out}$.

The second line in Table I gives the results corresponding to the B-spline level-set when the curvature term is not used (*i.e.* $\nu = 0$), so the smoothing is performed using an appropriate scale ($h = 4$). The corresponding total computation time (2.03s) is slightly larger than FTC (1.68s). As previously mentioned, the number of iterations needed to segment the leaf is much smaller for the B-spline level-set (5 iterations) than for FTC (82 iterations). Thus, each iteration of the B-spline level-set algorithm is computationally intensive, but this is compensated for by the fact that fewer iterations are used.

The third line in Table I gives the results corresponding to the B-spline level-set when the curvature term is used (*i.e.* $\nu = \frac{2}{25}\,255^2$), and the scale is set to the minimum ($h = 1$). The computation time is much larger (15.48s) and the number of iteration increases accordingly. This confirms the observations made in the previous experimental section (see Figures 1 and 2): the curvature term makes the level-set more rigid and the algorithm takes more iterations to reach the solution.

It is to be noted that the performance of the algorithms clearly depends on the image. Because it works locally, a narrow band algorithm such as FTC will be penalized when the structure to be segmented requires propagation along a long path. This may be illustrated on a somewhat extreme case, corresponding to the spiral presented in the previous section on Figure 1. In this case, the initialization is a 225 pixels centred square. The segmentation results and associated running time are given in Figure 10 and Table II. In this particular case, the B-spline algorithm yields a much shorter computation time (1.86s) as compared to the FTC algorithm (44.10s). This is simply due to the fact that the proposed approach provides an overall control over the level set, whereas the local nature of narrow band FTC implies propagation along the internal path in the spiral, requiring a large number of iterations (1573). Here again, the use of the curvature constraint (third line of Table II) tends to rigidify the B-spline level-set, implies more iterations (80) and thus yields larger computation time (28.99s).

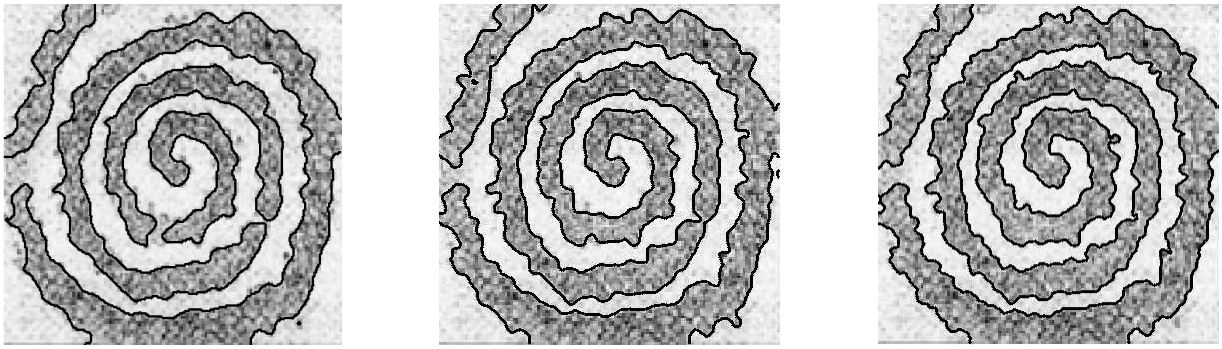
V. CONCLUSION

We propose in this paper a new formulation of active contours based on level-sets, where the implicit function is modeled as a continuous parametric function expressed on a B-spline basis. This representation provides an overall control of the level-set, and allows one to avoid the re-initialization step of the level-set via the normalization of the B-spline coefficients. Our proposed method has conceptual advantages such as:

- Simplification of the mathematical derivations. Our cost function depends on a finite number of parameters, and its gradient can be calculated exactly, without one having to recourse to approximate finite differences. There is no need for sophisticated calculus of variations, or, alternatively, Gâteaux derivatives, since the gradient of the cost function is obtained through elementary differentiation with respect to a finite number of parameters.
- Absence of arbitrary choices. The discretization scheme is natural since it is built in the problem from its inception. Thanks to the underlying B-spline model, all computations can be made exact, including derivatives.

Our B-spline formulation also offers specific features. The level-set evolution may be expressed as a sequence of 1-D convolutions, yielding an efficient algorithm. Moreover, this evolution corresponds to a smoothing-filter operation, the amount of smoothing being explicitly controlled via the scale and the degree of the selected B-spline basis.

The behavior of the proposed approach has been evaluated on various test images using the Chan-Vese functional. The segmentation results show the potential of our method in terms of computation time and flexibility. In particular, these results show that the intrinsic smoothing of the approach allows one to



(a) FTC ($N_g = 9\,,\,\sigma^2 = 3$)

(b) B-spline level-set ($h = 4\,,\,\nu = 0$)

(c) B-spline level-set ($h = 1\,,\,\nu = \frac{2}{25}\,255^2$)

Fig. 10. Segmentation of the spiral image using the FTC and B-spline level set algorithms

deal with additive noise, which is difficult to take into account with the conventional level-set implementations.

ACKNOWLEDGEMENT

Data from Synchrotron X-Ray CT of calcaneus bone are courtesy of Dr Françoise Peyrin from CREATIS-LRMN at Lyon and ESRF (European Synchrotron Radiation Facility) at Grenoble, France.

This work was partially supported by the Swiss Science Foundation under Grant 200020-109415.

APPENDIX A
ENERGY DERIVATION ACCORDING TO B-SPLINE COEFFICIENTS

We consider here the differentiation of the global energy criterion J (4) with respect to a given B-spline coefficient $c[\mathbf{k}_0]$. For brevity sake, let us skip the function arguments and note the 3 integral terms of J as

$$\begin{cases} J_{\rm in} \,=\, \displaystyle\int_\Omega g_{\rm in}\, H\, {\rm d}x_1 \cdots {\rm d}x_d \\ J_{\rm out} \,=\, \displaystyle\int_\Omega g_{\rm out}\, (1-H)\, {\rm d}x_1 \cdots {\rm d}x_d \\ J_{\rm c} \,=\, \displaystyle\int_\Omega g_{\rm c}\, \delta\, \|\boldsymbol{\nabla}\phi\|\, {\rm d}x_1 \cdots {\rm d}x_d. \end{cases}$$

Using differentiation with respect to parameter $c[\mathbf{k}_0]$, we have:

$$\begin{cases} \dfrac{d\,J_{\rm in}}{d\,c[\mathbf{k}_0]} \,=\, \displaystyle\int_\Omega \left[\frac{\partial\, g_{in}}{\partial\,\phi}\, H \,+\, g_{in}\, \frac{\partial\, H}{\partial\,\phi}\right] \cdot \frac{d\,\phi}{d\,c[\mathbf{k}_0]}\, {\rm d}x_1 \cdots {\rm d}x_d & (22) \\ \dfrac{d\,J_{\rm out}}{d\,c[\mathbf{k}_0]} \,=\, \displaystyle\int_\Omega \left[\frac{\partial\, g_{out}}{\partial\,\phi}\, (1-H) \,-\, g_{out}\, \frac{\partial\, H}{\partial\,\phi}\right] \cdot \frac{d\phi}{dc[\mathbf{k}_0]}\, {\rm d}x_1 \cdots {\rm d}x_d & (23) \\ \dfrac{dJ_{\rm c}}{dc[\mathbf{k}_0]} = \displaystyle\int_\Omega \left[\frac{dg_c}{d\phi}\delta|\nabla\phi| + \frac{d\delta}{d\phi}g_c|\nabla\phi| + g_c\delta\frac{d|\nabla\phi|}{d\phi}\right] \cdot \frac{d\,\phi}{d\,c[\mathbf{k}_0]}\, {\rm d}x_1 \cdots {\rm d}x_d. & (24) \end{cases}$$

In the last equation, using integration by part, it may be shown that

$$\frac{d\,\delta}{d\,\phi}\, g_c\, |\nabla\,\phi| \,+\, g_c\, \delta\, \frac{d\,|\nabla\,\phi|}{d\,\phi} \,=\, g_c\, \delta\, div\left(\frac{\nabla\phi}{|\nabla\phi|}\right) \qquad (25)$$

and the following expression follows:

$$\frac{d\,J_{\rm c}}{d\,c[\mathbf{k}_0]} \;=\; \int_\Omega \left[\frac{d\,g_c}{d\,\phi}\, |\nabla\,\phi| \,+\, g_c\, div\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right] \cdot\, \delta\, \frac{d\,\phi}{d\,c[\mathbf{k}_0]}\, {\rm d}x_1 \cdots {\rm d}x_d. \qquad (26)$$

Noting moreover that

$$\frac{d\,\phi}{d\,c[\mathbf{k}_0]} \,=\, \beta^n(\frac{\mathbf{x}}{h} - \mathbf{k}_0) \qquad (27)$$

yields equation (6) and (7).

APPENDIX B
BOUNDING OF THE LEVEL-SET FUNCTION

Since the B-splines verify the conditions [23],

$$\begin{cases} \beta^n(\mathbf{x}) \geq 0, \qquad \forall \mathbf{x} \in \mathbb{R}^d \\ \displaystyle\sum_{\mathbf{k}\in\mathbb{Z}^d} \beta^n(\frac{\mathbf{x}}{h} - \mathbf{k}) = 1, \qquad \forall \mathbf{x} \in \mathbb{R}^d, \end{cases} \qquad (28)$$

then we have that

$$\begin{aligned} |\phi(\mathbf{x})| &= \left| \sum_{\mathbf{k}\in\mathbb{Z}^d} c[\mathbf{k}]\, \beta^n(\frac{\mathbf{x}}{h} - \mathbf{k}) \right| \\ &\leq \sum_{\mathbf{k}\in\mathbb{Z}^d} |c[\mathbf{k}]| \left| \beta^n(\frac{\mathbf{x}}{h} - \mathbf{k}) \right| \\ &\leq \sum_{\mathbf{k}\in\mathbb{Z}^d} \|c\|_\infty\, \beta^n(\frac{\mathbf{x}}{h} - \mathbf{k}) \\ &\leq \|c\|_\infty \sum_{\mathbf{k}\in\mathbb{Z}^d} \beta^n(\frac{\mathbf{x}}{h} - \mathbf{k}). \end{aligned} \tag{29}$$

This yields

$$|\phi(\mathbf{x})| \leq \|c\|_\infty . \tag{30}$$

Appendix C
Bounding of the level-set gradient

Let $\|\nabla\phi(\mathbf{x})\|_2$ be the norm of the gradient of the level-set. We have

$$\begin{aligned} \|\nabla\phi(\mathbf{x})\|_2 &= \left\| \sum_{\mathbf{k}\in\mathbb{Z}^d} c[\mathbf{k}]\, \nabla\beta^n(\frac{\mathbf{x}}{h} - \mathbf{k}) \right\|_2 \\ &\leq \sum_{\mathbf{k}\in\mathbb{Z}^d} |c[\mathbf{k}]| \left\| \nabla\beta^n(\frac{\mathbf{x}}{h} - \mathbf{k}) \right\|_2 \\ &\leq \sum_{\mathbf{k}\in\mathbb{Z}^d} \|c\|_\infty \left\| \nabla\beta^n(\frac{\mathbf{x}}{h} - \mathbf{k}) \right\|_2 \\ &\leq \|c\|_\infty \sum_{\mathbf{k}\in\mathbb{Z}^d} \left\| \nabla\beta^n(\frac{\mathbf{x}}{h} - \mathbf{k}) \right\|_2 . \end{aligned} \tag{31}$$

This yields

$$\|\nabla\phi(\mathbf{x})\|_2 \leq A\, \|c\|_\infty . \tag{32}$$

where $A = \max_x \left(\sum_{\mathbf{k}\in\mathbb{Z}^d} \left\| \nabla\beta^n(\frac{\mathbf{x}}{h} - \mathbf{k}) \right\|_2 \right)$

The determination of A may be then easily done using B-splines basic properties. The d-dimensional B-spline of degree n is given by

$$\beta^n(\mathbf{x}) = \prod_{j=1}^{d} \beta^n(x_j) \tag{33}$$

where $\beta^n(x_j)$ is the one-dimensional B-splines of degree n. The *i-th* component of the gradient $\nabla\beta(\mathbf{x})$ is thus given by:

$$\frac{\partial\, \beta^n(\mathbf{x})}{\partial\, x_i} = \frac{\partial\, \beta^n(x_i)}{\partial\, x_i} \prod_{j=1, j\neq i}^{d} \beta^n(x_j) \tag{34}$$

Since the B-splines verifies

$$0 \leq \beta^n(x_j) \leq 1 \qquad \forall\, x_j \in \mathbb{R} \tag{35}$$

We thus have

$$|\frac{\partial\, \beta^n(\mathbf{x})}{\partial\, x_i}| \leq |\frac{\partial\, \beta^n(x_i)}{\partial\, x_i}| \leq K_n \tag{36}$$

where $K_n = \max_x \left(|\frac{\partial\, \beta^n(\mathbf{x})}{\partial\, x}| \right)$. K_n may be easily computed for any degree n. For B-splines of degrees $n = 1$, 2 and 3, we have $K_1 = 1$, $K_2 = 1$ and $K_3 = 2/3$.

As a consequence, the norm of the gradient $\nabla\, \beta^n(\mathbf{x})$ is bounded as follows

$$\|\nabla\, \beta^n(\mathbf{x})\|_2 \leq \sqrt{\sum_{j=1}^{d} (K_n)^2} = K_n\sqrt{d} \tag{37}$$

Coming back to A and noting that only $(n+1)^d$ B-splines are non-zero for any $\mathbf{x}$, we thus have

$$A \leq (n+1)^d\, K_n\, \sqrt{d} \tag{38}$$

and finally

$$\|\nabla\, \phi(\mathbf{x})\|_2 \leq (n+1)^d\, K_n\, \sqrt{d}\, \|\mathbf{c}\|_\infty \tag{39}$$

Applying the re-normalization given in (15)

$$\mathbf{c}^{(i+1)} \leftarrow \frac{\mathbf{c}^{(i+1)}}{\left\|\mathbf{c}^{(i+1)}\right\|_\infty} \tag{40}$$

the new gradient norm is then bounded as

$$\|\nabla\, \phi(\mathbf{x})\|_2 \leq (n+1)^d\, K_n\, \sqrt{d} \tag{41}$$

References

[1] J. Suri, K. Liu, S. Singh, S. Laxminarayan, X. Zeng, and L. Reden, "Shape recovery algorithms using level sets in 2-D/3-D medical imagery: A state-of-the-art review," *IEEE Transactions on Information Technology In Biomedicine*, vol. 6, no. 1, pp. 8–28, 2002.

[2] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. New York: Springer-Verlag, 2002.

[3] R. Tsai and S. Osher, "Level set methods and their applications in image science," *Communications in Mathematical Sciences*, vol. 1, no. 4, pp. 1–20, 2003.

[4] D. Cremers, F. Tischhauser, J. Weickert, and C. Schnorr, "Diffusion snakes: Introducing statistical shape knowledge into the mumford-shah functional," *International Journal of Computer Vision*, vol. 50, no. 3, pp. 295–313, 2002.

[5] G. Unal, A. Yezzi, and H. Krim, "Information theoretic active polygons for unsupervised texture segmentation," *International Journal of Computer Vision*, vol. 62, no. 3, pp. 199–220, 2004.

[6] E. Debreuve, M. Gastaud, M. Barlaud, and G. Aubert, "Using the shape gradient for active contour segmentation: from the continuous to the discrete formulation," *Journal of Mathematical Imaging and Vision*, vol. 28, no. 1, pp. 47–66, 2007.

[7] B. Morse, W. Liu, T. Yoo, and K. Subramanian, "Active contours using a constraint-based implicit representation," in *SIGGRAPH '05: ACM SIGGRAPH 2005 Courses*. New York, USA: ACM Press, 2005, p. 252.

[8] A. Gelas, O. Bernard, D. Friboulet, and R. Prost, "Compactly supported radial basis functions based collocation method for level-set evolution in image segmentation," *IEEE Transactions on Image Processing*, vol. 16, pp. 1873–1887, 2007.

[9] G. Sundaramoorthi, A. Yezzi, and A. Mennucci, "Sobolev active contours," *International Journal of Computer Vision*, vol. 73, no. 3, pp. 345–366, 2007.

[10] H. Liu, Y. Chen, and W. Chen, "Neighborhood aided implicit active contours," in *CVPR*, vol. 1, no. 841–848, 2006.

[11] P. Martin, P. Refregier, F. Goudail, and F. Guerault, "Influence of the noise model on level set active contour segmentation," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 799–803, 2004.

[12] F. Lecellier, S. Jehan-Besson, J. Fadili, G. Aubert, and M. Revenu, "Statistical region-based active contours with exponential family observations," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'06)*, Toulouse, France, 2006, pp. 113–116.

[13] F. Lecellier, S. Jehan-Besson, J. Fadili, G. Aubert, M. Revenu, and E. Saloux, "Region-based active contour with noise and shape priors," in *IEEE International Conference on Image Processing (ICIP'06)*, Atlanta, Georgia, USA, 2006, pp. 1649–1652.

[14] O. Bernard, J. Dhooge, and D. Friboulet, "Segmentation of echocardiographic images based on statistical modelling of the radio-frequency signal," in *European Signal Processing Conference (Eusipco)*, Florence, Italy, 2006, pp. ID cr2205, 4 pages.

[15] O. Bernard, B. Touil, A. Gelas, R. Prost, and D. Friboulet, "A rbf-based multiphase level set method for segmentation in echocardiography using the statistics of the radiofrequency signal," in *IEEE International Conference on Image Processing (ICIP'07)*, San Antonio, Texas, USA, 2007, pp. 157–160.

[16] G. Charpiat, P. Maurel, J. Pons, R. Keriven, and O. Faugeras, "Generalized gradients: Priors on minimization flows," *International Journal of Computer Vision*, vol. 73, no. 3, pp. 325–344, 2007.

[17] Y. Shi and W. Karl, "A real-time algorithm for the approximation of level-set-based curve evolution," *IEEE Transactions on Image Processing*, vol. 17, no. 5, pp. 645–656, 2008.

[18] F. Gibou and R. Fedkiw, "A fast hybrid k-means level set algorithm for segmentation," in *4th Annual Hawaii International Conference on Statistics and Mathematics*, Hawaii, USA, 2005, pp. 281–291.

[19] G. Kuhne, J. Weickert, M. Beier, and W. Effelsberg, "Fast implicit active contour models," *In: Pattern Recognition, Lecture Notes in Computer Science*, vol. 2449, pp. 133–140, 2002.

[20] G. Aubert, M. Barlaud, O. Faugeras, and S. Jehan-Besson, "Image segmentation using active contours: Calculus of variations or shape gradients?" *SIAM Applied Mathematics*, vol. 63, no. 6, pp. 2128–2154, 2003.

[21] A. Tsai, A. Yezzi, W. Wells, C. Tempany, D. Tucker, A. Fan, W. Grimson, and A. Willsky, "A shape-based approach to the segmentation of medical imagery using level sets," *IEEE Transactions on Medical Imaging*, vol. 22, no. 2, pp. 137–154, 2003.

[22] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, pp. 61–79, 1997.

[23] M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Processing Magazine*, vol. 16, no. 6, pp. 22–38, November 1999.

[24] T. Chan and L. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, pp. 266–277, 2001.

[25] H. Zhao, T. Chan, B. Merriman, and S. Osher, "A variational level set approach to multiphase motion," *J. Comput. Phys.*, vol. 127, pp. 179–195, 1996.

[26] M. Sussman, E. Fatemi, P. Smereka, and S. Osher, "An improved level set method for incompressible twophase flows," *Journal of Computers and Fluids*, vol. 27, no. 5-6, pp. 663–680, 1998.

[27] S. Jehan-Besson, M. Barlaud, and G. Aubert, "Dream2s: Deformable regions driven by an eulerian accurate minimization method for image and video segmentation," *International Journal of Computer Vision*, vol. 53, no. 1, pp. 45–70, 2003.

[28] M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing: Part I—Theory," *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 821–833, February 1993.

[29] J. Kybic and M. Unser, "Fast parametric elastic image registration," *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1427–1442, November 2003.

[30] L. Grady and C. Alvino, "Reformulating and optimizing the mumford-shah functional on a graph - a faster, lower energy solution," in *European Conference on Computer Vision (ECCV'08)*, Marseille, France, 2008, pp. 248–261.

[31] D. Mumford and J. Shah, "Optimal approximation of piecewise smooth functions and associated variational problems," *Commun. Pure Appl. Math.*, vol. 42, pp. 577–685, 1989.

[32] L. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, pp. 297–302, 1945.

[33] R. Whitaker, "A level-set approach to 3D reconstruction from range data," *International Journal of Computer Vision*, vol. 29, pp. 203–231, 1998.