

# Spline-Based Deforming Ellipsoids for Interactive 3D Bioimage Segmentation

Ricard Delgado-Gonzalo, Nicolas Chenouard, and Michael Unser

**Abstract**—We present a new fast active-contour model (a.k.a. snake) for image segmentation in 3D microscopy. We introduce a parametric design that relies on exponential B-spline bases and allows us to build snakes able to reproduce ellipsoids. We have designed our bases to have the shortest-possible support, subject to some constraints. Thus, computational efficiency is maximized. The proposed 3D snake can approximate blob-like objects with good accuracy and can perfectly reproduce spheres and ellipsoids, irrespective of their position and orientation. The optimization process is remarkably fast due to the use of Gauss’ theorem within our energy computation scheme. Our technique yields successful segmentation results, even for challenging data where object contours are not well defined. This is due to our parametric approach that allows one to favor prior shapes. Together with this work, we provide software that gives full control over the snakes via an intuitive manipulation of few control points.

**Index Terms**—Active contour, active surface, parametric snake, exponential B-spline, segmentation, parameterization, microscopy, 3D, sphere, ellipsoid.

## I. INTRODUCTION

ACTIVE surfaces are effective tools for 3D image segmentation. An active surface is a 3D generalization of a snake (also known as active contour). It evolves from an initial position, typically specified by the user, toward the boundary of an object in a 3D volumetric image. Its evolution is formulated as a minimization problem. The associated cost function is called *snake energy*. Snakes have become popular because it is possible for the user to interact with them, not only when specifying its initial position, but also during the segmentation process.

Many snake variants have been proposed utilizing different types of surface representation and various energy terms. Implicit methods based on a level-set formulation of the Chan-and-Vese problem have been investigated extensively over the past decade [1]. Some effort was invested to obtain semi-parametric approaches using simplex meshes [2] and, more recently, 3D triangular meshes [3]. A first approach to fully parametric snakes named active geometric functions (AGF) was proposed in [4] using the variational framework and the Mumford-Shah energy functional. Then, a refinement of the AGF method was presented using polynomial B-splines [5].

Copyright (c) 2013 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to [pubs-permissions@ieee.org](mailto:pubs-permissions@ieee.org).

R. Delgado-Gonzalo, Nicolas Chenouard, and M. Unser are with the Biomedical Imaging Group, École polytechnique fédérale de Lausanne (EPFL), Switzerland.

This work was funded by the Swiss SystemsX.ch initiative under Grant 2008/005 and the Swiss National Science Foundation under Grant 200020-121763.

Lastly, a spline-based parametric snake with a sphere-like topology was proposed in [6]. This last model incorporated special points that broke the smoothness and symmetry of the model.

In this paper, we propose a generic 3D spline-based parametric snake. To the best of our knowledge, it is the first of its kind where the topology can be constrained to segment ellipsoid-like objects. It is well adapted to handle the quasi-elliptical biological objects encountered in 3D microscopy. Moreover, our scheme is flexible and can also deal with more complex shapes.

Our snake surface is parameterized by few control points and uses a special kind of exponential B-splines as basis functions. Their most important feature is that they can be made to perfectly reproduce ellipsoids. Moreover, they have the shortest-possible support that is compatible with the reproduction of ellipsoids. Because they are also refinable, they provide a good approximation of any closed surface with a sphere-like topology in 3D.

Because speed is crucial for biological applications such as cell tracking in time-lapse sequences of 3D images, which produce tremendous amounts of data, we have designed edge and region energies that admit a fast implementation due to the use of pre-integrated images and Gauss’ theorem. We have analyzed extensive sets of real microscopic images and are reporting real-time performance. In addition, the spline-based parameterization and, more precisely, the use of our exponential B-splines, allow us to derive a fast algorithm for image segmentation. We also propose a simple method to detect self-intersection of the surface during the snake evolution.

Our parametric snakes proposed here can be made fully automatic. Meanwhile, they are also consistent with semi-automatic segmentation scheme that allows for user-friendly interaction. Because the snake is parameterized by only a few 3D control points, the user is able to easily guide it by interacting with anchors in dedicated 2D and 3D image views. These views feature a live display of the snake and provide feedback to the user. This ability is precious for crowded biological environments. The software implementing our techniques is given as an open-source library in an effort to provide useful tools for the bioimaging community.

This paper is organized as follows: In Section II, we review the general framework of parametric surface representations based on B-splines. Then, we formalize the mathematical conditions that basis functions must satisfy to make the parametric surfaces suitable for segmentation. Next, we specify a 3D snake model with a sphere-like topology. Implementation

details such as energy functionals and discretization issues are addressed in Section III. Finally, we illustrate the capabilities of our snake with synthetic and real data in Section IV.

## II. SPLINE SURFACES FOR OBJECT SEGMENTATION

In this section, we extend to 3D the representation of 2D curves originally developed in [7]. Topology and smoothness issues make the extension non-trivial.

### A. Parametric Representation of Surfaces

We consider a parametric representation of a surface  $\sigma(u, v)$  in 3D space that is described by a triplet of Cartesian coordinate functions  $x(u, v)$ ,  $y(u, v)$ , and  $z(u, v)$ , where  $u, v \in \mathbb{R}$  are continuous parameters. The two-dimensional functions  $x$ ,  $y$ , and  $z$  are represented by linear combinations of suitable basis functions. Among all possible bases, we focus on those derived from a compactly supported generator  $\Phi: \mathbb{R}^2 \mapsto \mathbb{R}$  and its multi-integer shifts  $\{\Phi(u - i, v - j)\}_{(i,j) \in \mathbb{Z}^2}$ . The parametric representation of the curve is then given by the vectorial equation

$$\sigma(u, v) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \mathbf{c}[i, j] \Phi\left(\frac{u}{T_1} - i, \frac{v}{T_2} - j\right), \quad (1)$$

where  $\{\mathbf{c}[i, j] \in \mathbb{R}^3\}_{(i,j) \in \mathbb{Z}^2}$  are the control points in 3D that define the shape of the surface and where  $T_1, T_2 \in (0, \infty)$  are the sampling steps for each parametric dimension. We denote by  $S \subset \mathbb{R}^3$  the set of points of the surface, and we are especially interested in the case when  $S$  is a closed surface.

In view of the nature of the domain set defined by  $u$  and  $v$ , a common strategy is to consider using tensor-products for the construction of the base function  $\Phi$ . Then, the generator can be written as

$$\Phi(u, v) = \phi_1(u) \phi_2(v). \quad (2)$$

This representation of closed surfaces has been studied by several authors [8], [9], [10], albeit not in the context of snakes. Various choices of  $\phi_1$  and  $\phi_2$  have been considered, such as polynomials, polynomial B-splines, and trigonometric B-splines. This tensor-product decomposition (2) allows one to take advantage of fast and stable interpolation algorithms [11], [12], [13].

We define the tangent space at any point  $\mathbf{p} = \sigma(u_0, v_0) \in S$  as the vector space generated by the tangent vectors to  $S$  at  $\mathbf{p}$

$$\mathbf{T}_1 = \frac{\partial \sigma}{\partial u}(u, v)|_{(u_0, v_0)} \quad (3)$$

$$\mathbf{T}_2 = \frac{\partial \sigma}{\partial v}(u, v)|_{(u_0, v_0)}. \quad (4)$$

The tangent bundle is defined as the disjoint union of all tangent spaces indexed by the points on the surface  $\mathbf{p} \in S$ . The tangent bundle is said to be well defined if all tangent spaces have their dimension equal to two—they must be planes. In this case, the surface  $S$  is said to be regular [14]. Requiring  $S$  to be regular implies that the surface should not self-intersect or have any border. Loosely speaking,  $S$  should locally *look*

*like a plane*. Under these conditions, the vector normal to  $S$  at  $\mathbf{p} = \sigma(u_0, v_0)$  can be computed by

$$\mathbf{n} = \mathbf{T}_1 \times \mathbf{T}_2, \quad (5)$$

where  $\times$  denotes the three-dimensional cross product.

### B. Desirable Properties of Basis Functions for Segmentation

We now enumerate the formal conditions that our parametric surface model should satisfy for the purpose of 3D shape segmentation.

- 1) *Unique and Stable Representation.* We want our parametric functions  $x$ ,  $y$ , and  $z$  to be uniquely determined by a single sequence of coefficients  $\{\mathbf{c}[i, j]\}_{(i,j) \in \mathbb{Z}^2}$  for all  $u, v \in \mathbb{R}$ . Furthermore, for computational purposes, we ask this interpolation procedure to be numerically stable, which require that there exist two constants  $0 < A \leq B < \infty$  such that

$$A \|\mathbf{c}\|_{\ell_2} \leq \|\sigma\|_{L_2} \leq B \|\mathbf{c}\|_{\ell_2} \quad (6)$$

for all  $\mathbf{c} \in \ell_2$ , where  $\sigma$  takes the form (1).

A direct consequence of the lower inequality of (6), which is called the Riesz basis condition, is that the equality  $\sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \mathbf{c}[i, j] \Phi\left(\frac{u}{T_1} - i, \frac{v}{T_2} - j\right) = 0$  for all  $(u, v) \in \mathbb{R}^2$  implies that  $\mathbf{c}[i, j] = 0$  for all  $(i, j) \in \mathbb{Z}^2$ . Thus, the basis functions are linearly independent and every function is uniquely specified by its coefficients. The upper inequality ensures the stability of the interpolation process [13].

It has been shown in [15] that, due to the integer-shift-invariant structure of the representation, the Riesz basis condition has the following equivalent expression in the Fourier domain:

$$A \leq \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \left| \hat{\Phi}(\omega_1 + 2\pi i, \omega_2 + 2\pi j) \right|^2 \leq B \quad (7)$$

for all  $(\omega_1, \omega_2) \in \mathbb{R}^2$ , where  $\hat{\Phi}$  denotes the two-dimensional Fourier transform of  $\Phi$ . Once expressed in the Fourier domain, the Riesz condition provides a practical way to verify whether a given generating function  $\Phi$  satisfies (6) or not.

Given the fact that  $\Phi$  is built from the tensor product of the two one-dimensional functions  $\phi_1$  and  $\phi_2$ , a sufficient condition to satisfy (7) is to require  $\phi_1$  and  $\phi_2$  to be valid Riesz generators. In particular, they should satisfy

$$A_1 \leq \sum_{i=-\infty}^{\infty} \left| \hat{\phi}_1(\omega + 2\pi i) \right|^2 \leq B_1$$

$$A_2 \leq \sum_{j=-\infty}^{\infty} \left| \hat{\phi}_2(\omega + 2\pi j) \right|^2 \leq B_2,$$

where  $\hat{\phi}_1$  and  $\hat{\phi}_2$  are the one-dimensional Fourier transforms of  $\phi_1$  and  $\phi_2$ , respectively, and  $A_1, B_1, A_2$ , and  $B_2$  are the corresponding Riesz bounds.

- 2) *Affine Invariance.* Since we are interested in outlining shapes irrespective of their position and orientation,

we would like our model to be invariant to affine transformations. We formalize this by requiring that

$$\begin{aligned} \mathbf{A}\boldsymbol{\sigma}(u, v) + \mathbf{b} \\ = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} (\mathbf{A}\mathbf{c}[i, j] + \mathbf{b}) \Phi\left(\frac{u}{T_1} - i, \frac{v}{T_2} - j\right), \end{aligned} \quad (8)$$

where  $\mathbf{A}$  is a  $(3 \times 3)$  matrix and  $\mathbf{b} \in \mathbb{R}^3$ . From (8), it is easy to show that affine invariance is ensured if and only if

$$\sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \Phi\left(\frac{u}{T_1} - i, \frac{v}{T_2} - j\right) = 1 \quad (9)$$

for all  $(u, v) \in \mathbb{R}^2$ . In the literature, this constraint is often named the *partition-of-unity* condition [16]. Since  $\Phi$  is built from the tensor product of two one-dimensional functions, (9) holds if and only if

$$\begin{aligned} \forall u \in \mathbb{R} \quad &: \sum_{i=-\infty}^{\infty} \phi_1\left(\frac{u}{T_1} - i\right) = 1 \\ \forall v \in \mathbb{R} \quad &: \sum_{j=-\infty}^{\infty} \phi_2\left(\frac{v}{T_2} - j\right) = 1. \end{aligned}$$

Thus, both  $\phi_1$  and  $\phi_2$  are required to satisfy the one-dimensional partition-of-unity condition.

- 3) *Well-Defined Gaussian Curvature*. The Gaussian curvature of a parametric surface at a point  $\mathbf{p} = \boldsymbol{\sigma}(u_0, v_0)$  can be expressed as the ratio of the determinants of the second and first fundamental forms

$$K = \frac{\det \mathbf{II}}{\det \mathbf{I}}.$$

The first fundamental form is the inner product on the tangent space of a surface in three-dimensional Euclidean space which is induced canonically from the dot product of  $\mathbb{R}^3$ . It can be expressed as the symmetric matrix

$$\mathbf{I} = \begin{pmatrix} \mathbf{T}_1 \cdot \mathbf{T}_1 & \mathbf{T}_1 \cdot \mathbf{T}_2 \\ \mathbf{T}_1 \cdot \mathbf{T}_2 & \mathbf{T}_2 \cdot \mathbf{T}_2 \end{pmatrix},$$

where  $\mathbf{T}_1$  and  $\mathbf{T}_2$  are the tangent vectors defined in (3) and (4), respectively.

The second fundamental form is a quadratic form on the tangent plane in the three-dimensional Euclidean space,

$$\mathbf{II} = \begin{pmatrix} \frac{\partial^2 \boldsymbol{\sigma}}{\partial u^2} \cdot \hat{\mathbf{n}} & \frac{\partial^2 \boldsymbol{\sigma}}{\partial u \partial v} \cdot \hat{\mathbf{n}} \\ \frac{\partial^2 \boldsymbol{\sigma}}{\partial u \partial v} \cdot \hat{\mathbf{n}} & \frac{\partial^2 \boldsymbol{\sigma}}{\partial v^2} \cdot \hat{\mathbf{n}} \end{pmatrix},$$

where  $\hat{\mathbf{n}} = \frac{\mathbf{n}}{\|\mathbf{n}\|}$  denotes the normal unit vector that can be computed using (5). Together with  $\mathbf{I}$ ,  $\mathbf{II}$  serves to define extrinsic invariants of the surface.

We would like to be able to compute the Gaussian curvature  $K$  for every point on the surface. To do so, each coordinate function (or, equivalently, the functions  $\phi_1$  and  $\phi_2$ ) must be at least twice differentiable, and its second derivative must be bounded.

- 4) *Reproduction of Particular Shapes*. There are several shapes that appear repeatedly in segmentation problems in 3D microscopy (e.g., ellipsoids). For that reason,

it is important that our surface model (1) perfectly reproduces ellipsoids. Formally, we want a parametric description of all ellipsoids to lie within the span of (1). In other words, the basis  $\{\Phi(u - i, v - j)\}_{(i,j) \in \mathbb{Z}^2}$  should reproduce the family of functions that describe any ellipsoid. Since our model is vectorial, we can impose this condition component-wise.

In mathematical terms, we say that a generating function  $\Phi$  reproduces a function  $f : \mathbb{R} \mapsto \mathbb{R}$  if and only if there exists a sequence of coefficients  $\{c[i, j] \in \mathbb{R}\}_{(i,j) \in \mathbb{Z}^2}$  such that

$$f(u, v) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} c[i, j] \Phi(u - i, v - j)$$

holds almost everywhere. The analysis is further simplified if the function  $f$  is separable in  $u$  and  $v$ , so that there exists a decomposition such that  $f(u, v) = f_1(u) f_2(v)$ . In this situation, it is sufficient to approach the problem separately for each variable. Then, we say that  $\Phi$  reproduces the function  $f$  if there exists sequences  $\{c_1[i]\}_{i \in \mathbb{Z}}$  and  $\{c_2[j]\}_{j \in \mathbb{Z}}$  such that

$$\begin{aligned} f_1(u) &= \sum_{i=-\infty}^{\infty} c_1[i] \phi_1(u - i) \\ f_2(v) &= \sum_{j=-\infty}^{\infty} c_2[j] \phi_2(v - j) \end{aligned}$$

holds almost everywhere. In addition, we say that  $\phi_1$  reproduces  $f_1$  and that  $\phi_2$  reproduces  $f_2$ .

### C. 3D Snake Model

We define our 3D snake model as a closed parametric surface  $\boldsymbol{\sigma}$  following the parametric vectorial equation (1). Since the surface is closed, it is not necessary to consider that the parameters  $(u, v)$  take all possible values in  $\mathbb{R}^2$ . It is enough to consider a domain that is a compact set  $\Omega \subset \mathbb{R}^2$ . By convention, we normalize the range of the parameters  $u, v$  to lie within  $[0, 1]$ , setting the domain to be  $\Omega = [0, 1]^2$ . Then, the parametric representation of the surface is given by

$$\boldsymbol{\sigma}(u, v) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} \mathbf{c}[i, j] \phi_1(M_1 u - i) \phi_2(M_2 v - j), \quad (10)$$

where we have substituted the sampling steps  $T_1$  and  $T_2$  by the positive integers  $M_1 = \frac{1}{T_1}$  and  $M_2 = \frac{1}{T_2}$ . Together with the compactness of  $\phi_1$  and  $\phi_2$ , the fact that we impose  $M_1$  and  $M_2$  to be positive integers guarantees that the functions  $x, y$ , and  $z$  are represented by an integer number of basis functions within their domain  $\Omega = [0, 1]^2$ . The larger these values are, the more basis functions come into the domain  $\Omega$ . In this situation,  $\boldsymbol{\sigma}$  has more degrees of freedom and can represent a larger variety of shapes.

The particular choice of  $\phi_1$  and  $\phi_2$  determines the properties of the surface generated by (10), such as smoothness, computational load of the resulting model, or reproduction of particular shapes. With the appropriate choice of  $\phi_1$  and  $\phi_2$ , we can force the surface generated by (10) to take the topology of

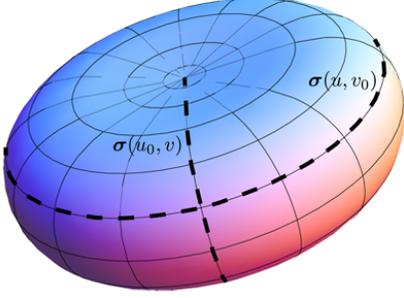


Fig. 1. Rendering of a snake surface taking the form of an oblate spheroid flattened at the poles. The mesh has been obtained by sampling (11). The circles of latitude are obtained when  $v$  is fixed to  $v_0$  in (11), and the meridians are obtained by fixing  $u$  to  $u_0$  in (11).

an ellipsoid by imposing boundary conditions on the sequence of control points  $\{\mathbf{c}[i, j]\}_{(i, j) \in \mathbb{Z}^2}$ .

1) *Topology*: To describe the parametrization, we shall adopt an earth-like cartographic terminology referring to meridians and circles of latitude. Then, the curves that are obtained when fixing the second parameter in (10) correspond to circles of latitude, as shown in Figure 1. It follows that, the curves obtained when  $v$  is constant in  $\sigma$  must be closed. This is achieved by letting the functions of each component of  $\sigma$  in  $u$  to be 1-periodic when  $v$  is constant. Thus, it is necessary to apply periodic boundary conditions along the first index of the sequence of control points. The sequence of coefficients becomes  $M_1$ -periodic and satisfies  $\mathbf{c}[i, j] = \mathbf{c}[i + M_1, j]$ . Under these conditions, we can reorganize the first infinite summation in (10) to a finite one involving periodized basis functions. Then, the parametric representation of the surface is expressed as

$$\sigma(u, v) = \sum_{i=0}^{M_1-1} \sum_{j=-\infty}^{\infty} \mathbf{c}[i, j] \phi_{1, \text{per}}(M_1 u - i) \phi_2(M_2 v - j), \quad (11)$$

where  $\phi_{1, \text{per}}(u) = \sum_{n=-\infty}^{\infty} \phi_1(u - M_1 n)$  for all  $u \in \mathbb{R}$ . Moreover, continuing with our earth simile, the curves that are obtained when fixing  $u$  in (10) correspond to meridians, which are open curves starting at the north pole  $\mathbf{c}_N$  and ending at the south pole  $\mathbf{c}_S$ .

2) *Reproduction of Ellipsoids*: We want to let our snakes reproduce ellipsoids perfectly, irrespective of their size, position, and orientation. Then, since every ellipsoid can be obtained by an affine transformation of a sphere of unit radius, we focus on the reproduction of this simpler shape and take advantage of (8).

The classical parameterization of a unit-radius sphere is

$$\begin{aligned} x(u, v) &= \cos 2\pi u \sin \pi v \\ y(u, v) &= \sin 2\pi u \sin \pi v \\ z(u, v) &= \cos \pi v, \end{aligned} \quad (12)$$

where  $u, v \in [0, 1]$ . These equations are separable in  $u$  and  $v$ , and each part can be efficiently taken care of with our separable model (11). For  $\Phi$  to be able to reproduce (12) and satisfy the condition of Section II-B,  $\phi_1$  must reproduce constants and

sinusoids of unit period, while  $\phi_2$  must reproduce constants and sinusoids of period equal to 2.

The optimal choice of  $\phi_1$  and  $\phi_2$  is dictated by a theorem in spline theory about the reproduction of exponential polynomials [17], that we include here for the sake of completeness.

*Theorem 1*: The centered generating function with minimal support and maximal smoothness that satisfies the Riesz-basis condition, the partition-of-unity condition, and that reproduces sinusoids of unit period with  $M$  coefficients is

$$\varphi_M(\cdot) = \sum_{k=0}^3 (-1)^k h_M[k] \varsigma_M(\cdot + \frac{3}{2} - k), \quad (13)$$

where

$$\begin{cases} \varsigma_M(\cdot) &= \frac{1}{4} \operatorname{sgn}(\cdot) \frac{\sin^2(\frac{\pi}{M} \cdot)}{\sin^2 \frac{\pi}{M}} \\ h_M &= [1, 1 + 2 \cos \frac{2\pi}{M}, 1 + 2 \cos \frac{2\pi}{M}, 1]. \end{cases}$$

Therefore, we take  $\phi_1(u) = \varphi_{M_1}(u)$  and  $\phi_2(v) = \varphi_{2M_2}(v)$ . If we expand (13) we obtain

$$\phi_1(u) = \begin{cases} \frac{\cos \frac{2\pi|u|}{M_1} \cos \frac{\pi}{M_1} - \cos \frac{2\pi}{M_1}}{1 - \cos \frac{2\pi}{M_1}} & 0 \leq |u| < \frac{1}{2} \\ \frac{1 - \cos \frac{2\pi(3/2 - |u|)}{M_1}}{2(1 - \cos \frac{2\pi}{M_1})} & \frac{1}{2} \leq |u| < \frac{3}{2} \\ 0 & \frac{3}{2} \leq |u| \end{cases}$$

for all  $u \in \mathbb{R}$ , and

$$\phi_2(v) = \begin{cases} \frac{\cos \frac{\pi|v|}{M_2} \cos \frac{\pi}{2M_2} - \cos \frac{\pi}{2M_2}}{1 - \cos \frac{\pi}{2M_2}} & 0 \leq |v| < \frac{1}{2} \\ \frac{1 - \cos \frac{\pi(3/2 - |v|)}{M_2}}{2(1 - \cos \frac{\pi}{2M_2})} & \frac{1}{2} \leq |v| < \frac{3}{2} \\ 0 & \frac{3}{2} \leq |v| \end{cases}$$

for all  $v \in \mathbb{R}$ . Note that  $\phi_1$  and  $\phi_2$  are equal if and only if  $M_1 = 2M_2$ . We show in Figure 2 some members of the family of functions  $\varphi_M$  indexed by  $M$ . They are continuous, have a finite support of length  $W = 3$ , and tend to be bump-like. Moreover, when  $M \rightarrow \infty$ , they converge to the quadratic B-spline. These functions allow the snake to perfectly replicate constants and sinusoids of the appropriate frequency at each component. This means that they can perfectly reproduce the parametric equations (12) with the appropriate configuration of control points. The approximation properties of our basis functions [17] guarantee that any coordinate function can be approximated as finely as desired by decreasing the sampling steps  $T_1$  and  $T_2$  (or, equivalently, by increasing  $M_1$  and  $M_2$ ). Moreover, the approximation error decreases quadratically with  $M_1$  and  $M_2$ . The most remarkable feature, though, is that the size of the support of these functions is the shortest possible to satisfy the conditions from Section II-B. This results in a maximally efficient scheme due to the fact that the computation of each point on the surface of the snake depends on the minimum-possible number of basis functions. It also ensures a local control of the surface by modifying single control points.

The control points that make our snake take the shape of a perfect unit sphere are a direct consequence of Theorem 1.

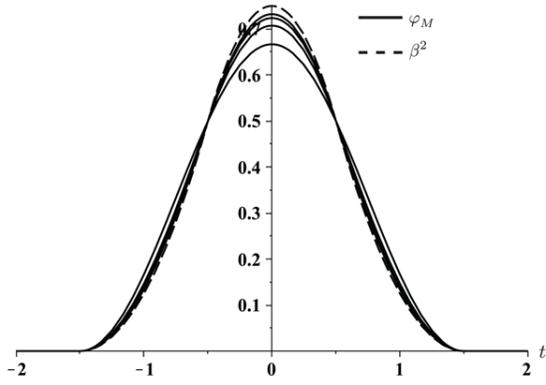


Fig. 2. Plot of a quadratic B-spline  $\beta^2$  and of the generating functions in (13) for  $M = 3, 4, 5$ , and  $6$ . The function with the lowest peak at  $t = 0$  corresponds to  $M = 3$ . As  $M$  increases, the central peak increases as well.

Their explicit expression is

$$\mathbf{c}[i, j] = \begin{pmatrix} c_{M_1}[i] s_{2 M_2}[j] \\ s_{M_1}[i] s_{2 M_2}[j] \\ c_{2 M_2}[j] \end{pmatrix},$$

where

$$c_M[k] = \frac{2(1 - \cos \frac{2\pi}{M})}{\cos \frac{\pi}{M} - \cos \frac{3\pi}{M}} \cos \frac{2\pi k}{M}$$

$$s_M[k] = \frac{2(1 - \cos \frac{2\pi}{M})}{\cos \frac{\pi}{M} - \cos \frac{3\pi}{M}} \sin \frac{2\pi k}{M}.$$

The derivation of the above result can be found in [17]. Note that the set of control points that generate the unit sphere is not unique, since we can arbitrarily choose the origin of the parameterization.

3) *Smoothness*: The chosen basis functions are twice differentiable and their second derivative is bounded. However, the parametric model (11) has two singular points where the continuity and smoothness are not guaranteed and need special attention: the poles  $\mathbf{c}_N$  and  $\mathbf{c}_S$ .

In order for the surface to be well-defined and continuous at the poles, we require the function  $\sigma(u, v)$  to be independent of  $u$  for  $v = 0$  and  $v = 1$ . Moreover, to ensure that the tangent plane varies continuously, we need to make some assumptions about the partial derivatives of  $\sigma$ . It was shown in [8] that a sufficient condition for continuity of the tangent plane is that the partial derivatives satisfy

$$\frac{\partial \sigma}{\partial v}(u, v)|_{v=0} = \mathbf{T}_{1,N} \cos 2\pi u + \mathbf{T}_{2,N} \sin 2\pi u \quad (14)$$

$$\frac{\partial \sigma}{\partial v}(u, v)|_{v=1} = \mathbf{T}_{1,S} \cos 2\pi u + \mathbf{T}_{2,S} \sin 2\pi u, \quad (15)$$

where  $\mathbf{T}_{1,N}, \mathbf{T}_{2,N}, \mathbf{T}_{1,S}, \mathbf{T}_{2,S} \in \mathbb{R}^3$  are constant vectors.

Tensor-product polynomial splines on the sphere have already been considered in [18] in the context of estimation techniques for fitting data on the sphere. However, no attempt was made to deal with the *pole problem* or to take full advantage of B-splines. Then, the sufficient conditions to obtain a continuously varying tangent plane or, equivalently,  $\mathcal{C}^1$  surface continuity, were first formulated in [8]. However, the proposed

scheme could only fulfill the conditions approximately, the main reason being that (14) and (15) can only be satisfied if the basis function associated to  $u$  is capable of reproducing sinusoids of unit period. An extension of this work was presented in [10], where the basis function associated to  $u$  was substituted by periodic trigonometric splines, being able to satisfy (14) and (15). Here, we use B-splines instead, which leads to the perfect reproduction of sinusoids. To the best of our knowledge, this has not been done previously.

We would like our parameterization to include implicitly the conditions exposed in Section II. We translate them as conditions over the control points. We categorize the required conditions in two types.

1) *Interpolation Conditions at the Poles*: All meridians originate at the north pole if and only if  $\sigma(u, 0) = \mathbf{c}_N$  for all  $u \in [0, 1]$ . Likewise, all meridians terminate at the south pole if and only if  $\sigma(u, 1) = \mathbf{c}_S$  for all  $u \in [0, 1]$ . The condition concerning the north pole can be rewritten in terms of the control points by evaluating (11) at  $v = 0$ . This yields

$$\mathbf{c}_N = \mathbf{c}[i, 1] \phi_2(-1) + \mathbf{c}[i, 0] \phi_2(0) + \mathbf{c}[i, -1] \phi_2(1),$$

for all  $i \in [0 \dots M_1 - 1]$ , where we have used the fact that  $\phi_2$  satisfies the partition-of-unity condition and that its support is limited to the interval  $[-\frac{3}{2}, \frac{3}{2}]$ . Likewise, the condition concerning the south pole can be rewritten in terms of the control points by evaluating (11) at  $v = 1$ . This yields

$$\mathbf{c}_S = \mathbf{c}[i, M_2 + 1] \phi_2(-1) + \mathbf{c}[i, M_2] \phi_2(0) + \mathbf{c}[i, M_2 - 2] \phi_2(1),$$

for all  $i \in [0 \dots M_1 - 1]$ .

2) *Smoothness Conditions at the Poles*: The sufficient conditions (14) and (15) were stated for any generic parameterization. For our case, this translates into

$$\begin{aligned} & \mathbf{T}_{1,N} \cos 2\pi u + \mathbf{T}_{2,N} \sin 2\pi u \\ &= M_2 \sum_{i=0}^{M_1-1} \sum_{j=-\infty}^{\infty} \mathbf{c}[i, j] \phi_{1,\text{per}}(M_1 u - i) \phi_2'(-j) \end{aligned}$$

and

$$\begin{aligned} & \mathbf{T}_{1,S} \cos 2\pi u + \mathbf{T}_{2,S} \sin 2\pi u \\ &= M_2 \sum_{i=0}^{M_1-1} \sum_{j=-\infty}^{\infty} \mathbf{c}[i, j] \phi_{1,\text{per}}(M_1 u - i) \phi_2'(M_2 - j) \end{aligned}$$

by evaluating the left-hand-side of (14) and (15). Here,  $\mathbf{T}_{1,N}, \mathbf{T}_{2,N} \in \mathbb{R}^3$  represent two free vectors that determine the tangent plane at the north pole, and  $\mathbf{T}_{1,S}, \mathbf{T}_{2,S} \in \mathbb{R}^3$  are two free vectors that determine the tangent plane at the south pole. The proposed smoothness conditions can be satisfied if and only if the model can reproduce sinusoids of unit period over  $u$  with a specified sampling rate. Since  $\phi_1$  was designed to reproduce sinusoids of unit period over  $u$  with a sampling rate of  $M_1$ , we are guaranteed that the smoothness conditions can be satisfied.

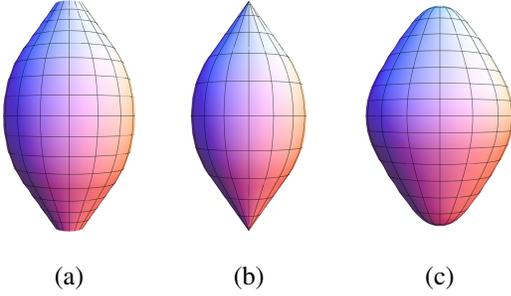


Fig. 3. Surfaces generated by the snake model (11). (a) Interpolation conditions are not satisfied. (b) Smoothness conditions are not satisfied. (c) Interpolation and smoothness conditions are satisfied.

In Figure 3, we show some surfaces generated by (11) that fail to satisfy some of the conditions on the poles. If the interpolation conditions are not satisfied at the poles, the surface may not be completely closed. If the smoothness condition is not satisfied either, the surface may have kinks at the poles, leaving an ill-defined tangent plane at these points.

The final step is to incorporate the exposed conditions into the parameterization and to obtain explicit formulas.

*Theorem 2:* A parametric spline-based surface with a sphere-like topology,  $C^1$  continuity, and the capability of reproducing ellipsoids irrespective of their position and orientation can be expressed as

$$\sigma(u, v) = \sum_{i=0}^{M_1-1} \sum_{j=-1}^{M_2+1} \mathbf{c}[i, j] \phi_{1,\text{per}}(M_1 u - i) \phi_2(M_2 v - j), \quad (16)$$

restricted to

$$\mathbf{c}[i, -1] = \mathbf{c}[i, 1] + \frac{\mathbf{T}_{1,N} c_{M_1}[i] + \mathbf{T}_{2,N} s_{M_1}[i]}{M_2 \phi_2'(1)} \quad (17)$$

$$\mathbf{c}[i, 0] = \frac{\mathbf{c}_N}{\phi_2(0)} + \frac{\phi_2(1) (\mathbf{c}[i, -1] + \mathbf{c}[i, 1])}{\phi_2(0)} \quad (18)$$

$$\mathbf{c}[i, M_2] = \frac{\mathbf{c}_S}{\phi_2(0)} - \frac{\phi_2(1) (\mathbf{c}[i, M_2 - 1] + \mathbf{c}[i, M_2 + 1])}{\phi_2(0)} \quad (19)$$

$$\mathbf{c}[i, M_2 + 1] = \mathbf{c}[i, M_2 - 1] - \frac{\mathbf{T}_{1,S} c_{M_1}[i] + \mathbf{T}_{2,S} s_{M_1}[i]}{M_2 \phi_2'(1)}, \quad (20)$$

where  $\{\mathbf{c}[i, j]\}_{i \in [0 \dots M_1 - 1], j \in [1 \dots M_2 - 1]}$ ,  $\mathbf{c}_N$ ,  $\mathbf{c}_S$ ,  $\mathbf{T}_{1,N}$ ,  $\mathbf{T}_{2,N}$ ,  $\mathbf{T}_{1,S}$ , and  $\mathbf{T}_{2,S}$  are free parameters. This adds up to a total number of  $(M_1(M_2 - 1) + 4)$  free control points.

*Proof:* First, since the support of  $\phi_2$  is limited to the interval  $[-\frac{3}{2}, \frac{3}{2}]$  and  $v$  lies within the interval  $[0, 1]$ , the second summation in (11) can be restricted to the indices  $j \in [-1 \dots 1]$ .

The restrictions (18) and (19) are obtained directly from the interpolation conditions at the poles. We rewrite the condition at the north pole as

$$\mathbf{c}[i, 0] = \frac{\mathbf{c}_N + \phi_2(1) (\mathbf{c}[i, -1] + \mathbf{c}[i, 1])}{\phi_2(0)}$$

for all  $i \in [0 \dots M_1 - 1]$ , where we have used the fact that  $\phi_2(1) = \phi_2(-1)$  since  $\phi_2$  is symmetric, giving us (18). Likewise, the interpolation condition at the south pole is rearranged as

$$\mathbf{c}[i, M_2] = \frac{\mathbf{c}_S - \phi_2(1) (\mathbf{c}[i, M_2 - 1] + \mathbf{c}[i, M_2 + 1])}{\phi_2(0)}$$

for all  $i \in [0 \dots M_1 - 1]$ , giving us (19).

Now, we simplify the right-hand side of the smoothness condition at the north pole to

$$\sum_{i=0}^{M_1-1} M_2 (\mathbf{c}[i, -1] \phi_2'(1) - \mathbf{c}[i, 1] \phi_2'(1)) \phi_{1,\text{per}}(M_1 u - i),$$

where we have used the fact that  $\phi_2'$  is an antisymmetric function that satisfies  $\phi_2'(\cdot) = -\phi_2'(-\cdot)$ . We expand the sinusoids of the left-hand side of the smoothness condition at the north pole using  $\phi_{1,\text{per}}$ . It results in

$$\begin{aligned} & \mathbf{T}_{1,N} \cos 2\pi u + \mathbf{T}_{2,N} \sin 2\pi u \\ &= \mathbf{T}_{1,N} \sum_{i=0}^{M_1-1} c_{M_1}[i] \phi_{1,\text{per}}(M_1 u - i) \\ & \quad + \mathbf{T}_{2,N} \sum_{i=0}^{M_1-1} s_{M_1}[i] \phi_{1,\text{per}}(M_1 u - i) \\ &= \sum_{i=0}^{M_1-1} (\mathbf{T}_{1,N} c_{M_1}[i] + \mathbf{T}_{2,N} s_{M_1}[i]) \phi_{1,\text{per}}(M_1 u - i). \end{aligned}$$

By identification of the coefficients, we obtain

$$\mathbf{c}[i, -1] = \mathbf{c}[i, 1] + \frac{\mathbf{T}_{1,N} c_{M_1}[i] + \mathbf{T}_{2,N} s_{M_1}[i]}{M_2 \phi_2'(1)}$$

for all  $i \in [0 \dots M_1 - 1]$ , which gives (17). The expression for (20) is obtained analogously using the smoothness condition at the south pole. ■

Theorem 2 is crucial when specifying the snake, since it ensures that any point on the resulting snake surface has at least  $C^1$  regularity, and eliminates the singularities that would otherwise exist at the poles.

### III. ENERGIES AND IMPLEMENTATION

#### A. Snake Energies

We follow the standard paradigm and formulate the surface evolution as an energy-minimization problem. The snake energy is typically the linear combination of three terms [19]:

- the image energy, which is responsible for guiding the snake toward the boundary of interest;
- the internal energy, which ensures that the segmented region has smooth boundaries;
- the constraint energy, which provides a means for the user to interact with the snake.

The total energy of the snake is written as

$$E_{\text{snake}}(\mathbf{c}) = E_{\text{image}}(\mathbf{c}) + E_{\text{int}}(\mathbf{c}) + E_c(\mathbf{c}).$$

The optimal spline parameters are obtained as

$$\mathbf{c}^* = \arg \min_{\mathbf{c}} E_{\text{snake}}(\mathbf{c}).$$

The quality of segmentation depends on the choice of the energy terms. The energy-minimization process is then essentially an optimization procedure, where we iteratively update the snake coefficients to approach the minimum of the energy function from a starting position. In our model, we obviated the constraint energy by accommodating the user interaction as a hard constraint and leaving the coefficients be specified by the user of the optimization routine.

1) *Image Energy*: There are many construction strategies which can be categorized in two main families: 1) edge-based methods, which use gradient information to detect contours [19], [20], [21] and 2) region-based methods, which use statistical information to distinguish different homogeneous regions [22], [23]. Both have advantages and disadvantages. On one hand, the edge-based energy can give a good localization of the contour near the boundaries. Unfortunately, it is very sensitive to noise and has a small basin of attraction, thus requiring a good initialization. On the other hand, the region-based energy has a large basin of attraction and can converge even if explicit edges are not present. However, it does not give a localization that is as good as the edge-based energy at the image boundaries. In order to benefit from the advantages of both strategies, a unified energy was proposed in [24]. In our case, we are going to follow a similar approach by using the convex combination of gradient and region energies

$$E_{\text{image}} = \alpha E_{\text{edge}} + (1 - \alpha) E_{\text{region}},$$

where  $\alpha \in [0, 1]$ . The tradeoff parameter  $\alpha$  balances the energy contributions. Its value depends on the characteristics of each application.

For the gradient-based (or edge) energy, traditional snakes rely on the guidance provided by edge maps. The most popular approach is based on the magnitude  $\|\nabla f\|$  of the gradient [19]. Mathematically, it is desired minimize

$$E_{\text{mag}} = - \oint_S \|\nabla f\| dS,$$

where the integration is performed on the closed surface  $S$ , and  $dS$  represents the area increment.

The major drawback of this approach is that the snake gets distracted by edges of nearby targets, since it does not distinguish between the gradients generated by different objects in the image. We solve this issue by considering the direction of the gradient as well and by imposing that the direction of the gradient and the normals of the surface be aligned. The parameterization (10) offers us the opportunity to choose the orientation of the normal vectors at initialization time. From now on, and without loss of generality, we assume it to be outwards. If we want to segment a bright object surrounded by a darker region, we expect the directions of the image gradients to be the same as the directions of the normal vectors when the energy reaches its minimum. We formalize this concept with the following image energy:

$$E_{\text{grad}} = - \oint_S \nabla f \cdot d\mathbf{S} = - \oint_S \left( \nabla f \cdot \frac{\mathbf{n}}{\|\mathbf{n}\|} \right) dS, \quad (21)$$

where  $d\mathbf{S}$  represents the vector differential of area. Using Gauss' (or the divergence) theorem, our edge energy can also

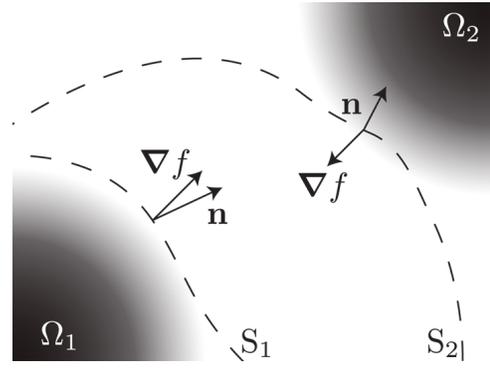


Fig. 4. 2D schematic representation of two parametric snakes  $S_1$  and  $S_2$  (dashed lines), of their interaction with the objects  $\Omega_1$  and  $\Omega_2$ , of normal vectors  $\mathbf{n}$ , and of the gradient vectors  $\nabla f$  of the image.

be expressed as the volume integral

$$E_{\text{grad}} = - \iiint_V \text{div}(\nabla f) dV = \iiint_V \underbrace{-\Delta f}_{f_{\text{edge}}} dV, \quad (22)$$

where  $V$  is the volume enclosed by  $S$ ,  $dV$  represents the volume increment, and  $f_{\text{edge}}$  can be precomputed. This allows for easy combination of region energies with edge-map-based energies.

The advantage of (22) is that it uses the direction of the gradient to discriminate between edges of the same target and between different targets. Its minimization makes the surface of the snake stick to edges where the image gradient has similar direction as the normal vector  $\mathbf{n}$  and to be repelled from edges with different orientations.

In Figure 4, we present a 2D schematic with the configuration of the various quantities involved in (21). A first snake is represented by the dashed line  $S_1$ . It is used to segment the area labeled as  $\Omega_1$ . The normal to the surface  $\mathbf{n}$  and the image gradient  $\nabla f$  are pointing in the same direction, which will add a strong negative contribution to  $E_{\text{grad}}$ . A second snake is represented by the dashed line  $S_2$ . It is used to segment the same area, but this time it is closer to the region labeled as  $\Omega_2$ . In this case, the normal to the surface  $\mathbf{n}$  and the image gradient  $\nabla f$  are pointing in opposite directions, which adds a strong positive contribution to  $E_{\text{grad}}$  because of the minus sign in (21). Thus, the first snake is more optimal than the second one, as far as  $E_{\text{grad}}$  is concerned.

For the region-based energy, we adopt a strategy similar to [25]. We first build an enclosing shell  $S_{\text{sh}}$  around the snake. Then, our region-based energy discriminates an object from its background by maximizing the signed contrast between the intensity of the data averaged within the volume  $V$  enclosed by the snake, and the intensity of the data, averaged within the volume  $V_{\text{sh}}$  enclosed by the shell. When  $V \subset V_{\text{sh}}$ , the region energy is expressed as

$$E_{\text{region}} = \frac{1}{|V|} \left( \int_V f dV - \int_{V_{\text{sh}} \setminus V} f dV \right), \quad (23)$$

where  $|V|$  is the volume of the snake.

To enforce that the criterion remains neutral when  $f$  takes a constant value  $f_0$ , we build the enclosing shell  $\sigma_{\text{sh}}$  such

that  $|V_{\text{sh}}| = 2|V|$ . Under these conditions,  $E_{\text{region}}|_{f=f_0} = 0$  depends neither on the snake nor even on  $f_0$ . We take full advantage of the affine-invariance property of our snake model and we build the parameterization of the shell  $\sigma_{\text{sh}}$  as the affine transformation of the snake surface (16) with the same center of gravity and with volume  $|V_{\text{sh}}|$ . The explicit parametric expression is

$$\sigma_{\text{sh}}(u, v) = \sqrt[3]{2} \sigma(u, v) + \left(1 - \sqrt[3]{2}\right) \sigma_{\text{g}},$$

where

$$\sigma_{\text{g}} = \sum_{i=0}^{M_1-1} \sum_{j=-1}^{M_2+1} \mathbf{c}[i, j]$$

is the center of gravity of the snake. Since the shell is an affine transformation of the original snake, it is possible to express  $\sigma_{\text{sh}}$  with the same basis functions.

2) *Internal Energy*: The internal energy is responsible for ensuring the smoothness of the snake. It was first proposed in [19] in the context of 2D active contours as a linear combination of the length of the contour and the integral of the square of the curvature along the contour. This smoothness term is used most often in most widely-used in applications. Its direct extension to active surfaces gives

$$E_{\text{int}} = \lambda_1 |S| + \lambda_2 \iint_S |K|^2 dS, \quad (24)$$

where  $K$  is the Gaussian curvature of the surface.

This internal energy can also be expressed specifically in terms of the tangent vectors and the fundamental forms as

$$\begin{aligned} E_{\text{int}} &= \lambda_1 \int_0^1 \int_0^1 \|\mathbf{T}_1 \times \mathbf{T}_2\| du dv \\ &+ \lambda_2 \int_0^1 \int_0^1 \left| \frac{\det \mathbf{II}}{\det \mathbf{I}} \right|^2 \|\mathbf{T}_1 \times \mathbf{T}_2\| du dv, \end{aligned}$$

where the first term makes the snake contract and the second favors smooth solutions. In the framework of active contours, most parametric schemes rely on the smoothness of the representation, thus eliminating the need for an explicit internal energy term. However, these approaches can ensure a low value of the curvature only when the curves are parameterized at constant speed (proportional to arc-length). For example, a spline curve may be rough if some of the spline knots accumulate locally. Similar problems exist with Fourier and other parametric representations. A practical workaround is to reparameterize the curve to constant arc-length after each step of the optimization algorithm, which is quite expensive [26]. Another approach is to substitute the curvature term in (24) by an energy term that penalizes the curve for not being in the curvilinear abscissa [24]. This energy is called *curvilinear reparameterization energy*. Minimizing this energy causes the curve knots to move tangentially to the snake, thus bringing it to curvilinear abscissa. The use of this energy is equivalent to reparameterizing the snake at each step, but with a much lower computational load. We adopt a similar approach and modify the internal energy (24) to

$$E_{\text{int}} = \lambda_1 |S| + \lambda_2 \int_0^1 \int_0^1 (|\det \mathbf{I}| - |S|)^2 du dv. \quad (25)$$

Letting the curve be driven by (25) will cause the control points to distribute uniformly over the snake surface and avoid accumulation. Then, the smoothness is implicitly enforced by the underlying parametric representation.

### B. Fast Energy Computation

The computational cost of the evolution of the snake is dominated by that of the volume integrals in (22) and (23). An efficient way to implement these operations is the use of pre-integrated images. Let  $g$  be the function we are integrating ( $f$  or  $f_{\text{edge}}$ ), and let  $\Omega$  be the domain of integration ( $V$  or  $V_d$ ). Then, by Gauss' theorem, we rewrite the volume integral as the surface integral in different ways

$$\begin{aligned} \iiint_{\Omega} g dV &= \iint_{\partial\Omega} g^x dy \wedge dz \\ &= \iint_{\partial\Omega} g^y dz \wedge dx \\ &= \iint_{\partial\Omega} g^z dx \wedge dy, \end{aligned}$$

where  $\wedge$  is the wedge product,  $\partial V$  is the boundary of  $V$ , and

$$\begin{aligned} g^x(x, y, z) &= \int_{-\infty}^x g(\tau, y, z) d\tau \\ g^y(x, y, z) &= \int_{-\infty}^y g(x, \tau, z) d\tau \\ g^z(x, y, z) &= \int_{-\infty}^z g(x, y, \tau) d\tau. \end{aligned}$$

All three possibilities are equivalent and can be stored in lookup tables to speed up the access to the data. The translation of volume integrals into surface integrals reduces the computational load dramatically. This can only be achieved if the surface is defined continuously, as in (1).

Despite the fact that we are describing our surface continuously, in a real-world implementation we only have at our disposal a sampled version of the functions we want to pre-integrate. We therefore perform a bilinear interpolation of the sampled data and store the result in lookup tables. Then, the final estimation of the energies is obtained using quadrature rules with a fixed sampling step.

### C. Optimization Algorithm

The optimization of the snake is carried out efficiently by a Powell-like line-search method [27]. This method requires the derivatives of the energy function with respect to the parameters (*i.e.*, the knot coefficients) and converges quadratically to the solution.

The algorithm proceeds as follows: firstly, one direction within the parameter space is chosen depending on the partial derivatives of the energy. Secondly, a one-dimensional minimization is performed within the selected direction. The line optimization is performed by successive parabolic approximations until the minimum along the direction is achieved. Then, a new direction is chosen using the partial derivatives of the energy function once more, while enforcing conjugation properties. Among all possible implementations of Powell's

method, we used the version where the direction of maximum decrease is replaced with the average descent direction [27]. This scheme is repeated until convergence. This powerful optimization method is used here because the number of parameters that define the shape of the snake is very small.

#### D. Self-Intersection Detection

The optimization process can sometimes lead to self-intersecting surfaces. However, the probability of self-intersection is greatly reduced by the introduction of (25), which hinders the clustering of control points.

Despite this refinement, self-intersection may still arise occasionally when the image energy forces some control points to move faster than others. This compromises our approach since we use Gauss' theorem, which assumes non self-intersecting closed surfaces. An extensive body of research can be found on the intersection problem, with numerous articles presenting different approaches for the intersection of freeform curves and surfaces [28]. Unfortunately, these methods are too time-consuming for our purpose.

As an alternative, we devised a fast method for the detection of self-intersection that uses the Gauss-Bonnet formula. This formula states that the Euler characteristic  $\chi$  of a closed, non-intersecting surface  $S$  can be computed by integrating the Gaussian curvature, as in

$$\chi(S) = \frac{1}{2\pi} \iint_S |K|^2 dS. \quad (26)$$

The Euler characteristic is a number that describes the shape and structure of a topological space, regardless of the way it is bent or deformed, as long as it does not self-intersect. In the case of sphere-like topologies, we have  $\chi(S) = 2$ . Therefore, we know that the snake self-intersects if  $\chi(S) \neq 2$ .

Unfortunately, (26) can give a correct value even if the surface is self-intersecting. In principle, it is possible to detect these cases by splitting the integral (26) over a series of smaller intervals and checking if there is a self-intersection in each of the subintervals. However, such cases are unlikely to occur in practice, and it was not necessary to implement such a finer level of detection.

When a self-intersection is detected, there are several ways to proceed. Some authors split their shape descriptor in a way that new smaller snakes are born [3], [29]. Since we wanted to preserve the topology of our surface, we opted to stop the optimization routine and ask for user assistance.

#### E. User Interactions

Our 3D segmentation technique has been programmed as an open-source plugin for the multi-platform software Icy [30] and is available on the official plugin repository.<sup>1</sup>

By contrast with many other implicit and global parametric snakes, ours is fully parameterized by only a few control points, which eases the interactions with the user. This has encouraged us to develop a dedicated graphical user interface that lets the user initialize the 3D snake position and trim it

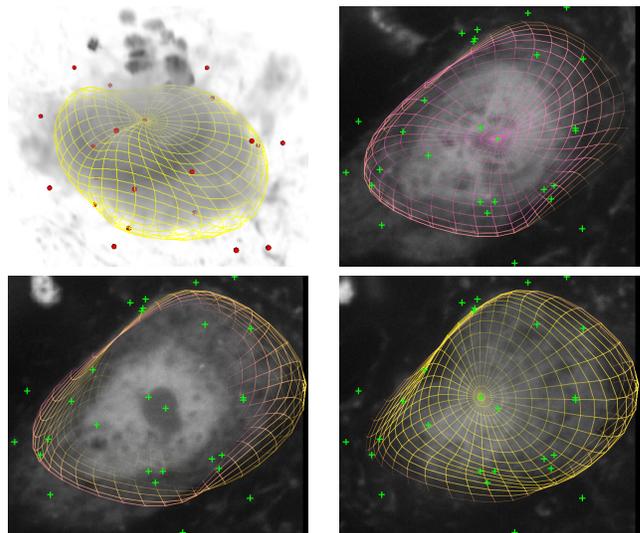


Fig. 5. Segmentation of a cell in a 3D confocal image of a murine brain (image courtesy of Sabine Scheibe and Sebastian Rhode at TILL Photonics). Top-left: 3D view of the segmented glomeruli. The solid spheres represent the control points of the model and their location define the effective contour of the snakes. From left to right and top to bottom: snake at different  $z$ -axis positions of the 3D volume, overlaid on the original 2D images. We color the mesh depending on the  $z$ -axis coordinate and set its transparency proportional to the distance of the slice displayed in the 2D viewer. The '+' elements are the projection to the XY plane of the control points of the model.

even after the optimization process. As illustrated in Figures 5, 10, and 11, the user can manipulate the position of any point by selecting it in either the 2D or 3D mesh representations of the snake. The 2D view also provides 3D cues, as we color the mesh depending on the  $z$ -axis coordinate and set its transparency proportional to the distance of the slice displayed in the 2D viewer. With this strategy, only parts of the snake close to the displayed image are shown. Editing the points can be performed with simple mouse actions. The live update of the 2D and 3D views of the snake is very fast. This is due to the fact that the change in one parameter affects the structure locally. Thus, just a limited region of the surface of the snake has to be recomputed. As a result, a fast, intuitive, and semi-automatic segmentation procedure that loops between snake initialization, optimization, and correction, is made possible.

#### F. Initialization

Our method can be operated in an automated fashion. For that, the algorithm only requires an initial position. Like in all segmentation problems, this initialization is strongly dependent on the imaging modality. For example, in fluorescence microscopy, a blurring or a DoG filter followed by the detection of local maxima/minima suffices to provide an initial position (see our previous work [7] for examples of automatic initializations of 2D parametric snakes). In medical imaging, refined methods based on atlases can also be used [31].

When choosing the initial position of the snake, one must ensure that the object of interest is within the basin of attraction of the chosen image energy. The conditions for the convergence of our image energies are the following:

- $E_{\text{grad}}$  is only responsive to image gradients. Therefore,

<sup>1</sup>[http://icy.bioimageanalysis.org/plugin/Active\\_Cells\\_3D/](http://icy.bioimageanalysis.org/plugin/Active_Cells_3D/)

the snake surface must be initialized close to the boundary of the object. This basin of attraction is usually extended by preprocessing the input image with a smoothing or a distance transform [7].

- $E_{\text{region}}$  is responsive to image contrasts between the snake core and the snake shell. Therefore, the snake surface must be initialized such that the core intersects the object and the shell intersects the background.

#### IV. EXPERIMENTS AND SIMULATIONS

We present in this section five experiments. In the first one, we deform our snake to show how the self-intersection is detected by monitoring the Euler characteristic. In the second experiment, we investigate the sensitivity of our snake to the presence of noise. In the third setup, we perform a quantitative evaluation of our algorithm when segmenting neighboring targets; there the preservation of topology plays a crucial role. We move away from simulated data in the fourth experiment where we investigate the approximation properties of our snake with medical data of a spleen from a CT-scan. In the last experiment, we illustrate the application to real microscopic data where the ground truth is not available.

The calculations were performed on a MacPro 3.1 with two Quad-Core Intel Xeon processors, 16GB of RAM memory, and an NVIDIA GeForce 8800 GT with 512 MB running Mac OS X 10.8.2.

##### A. Twisting the Snake

We deform the snake away from a perfect sphere by rotating the central layer of control points by an angle  $\theta$ , thereby creating a twist in the center of the structure. In particular, we set the snake control points to

$$\mathbf{c}[i, j] = \begin{pmatrix} c_{M_1}[i] s_{2 M_2}[j] \\ s_{M_1}[i] s_{2 M_2}[j] \\ c_{2 M_2}[j] \end{pmatrix},$$

for  $i \in [0 \dots M_1 - 1]$  and for all  $j \neq \lfloor \frac{M_2}{2} \rfloor$ , and we set

$$\mathbf{c}[i, j] = \begin{pmatrix} \cos(\theta) c_{M_1}[i] s_{2 M_2}[j] - \sin(\theta) s_{M_1}[i] s_{2 M_2}[j] \\ \sin(\theta) c_{M_1}[i] s_{2 M_2}[j] + \cos(\theta) s_{M_1}[i] s_{2 M_2}[j] \\ c_{2 M_2}[j] \end{pmatrix},$$

for  $i \in [0 \dots M_1 - 1]$  and for  $j = \lfloor \frac{M_2}{2} \rfloor$ .

We show in Figure 6 different surface configurations as a function of the rotation angle  $\theta$ . We see that, for small angles, the surface is slightly twisted but not self-intersecting. However, as we approach  $\theta = \pi$ , the central part of the structure collapses and leads to two self-intersections. We show in Figure 6 how the Euler characteristic, computed according to (26), varies in terms of  $\theta$ . The predicted value is correct for  $\theta \leq \frac{7\pi}{9}$  and starts increasing when the snake starts self-intersecting. Hence, the conditions of the Gauss-Bonnet theorem are violated.

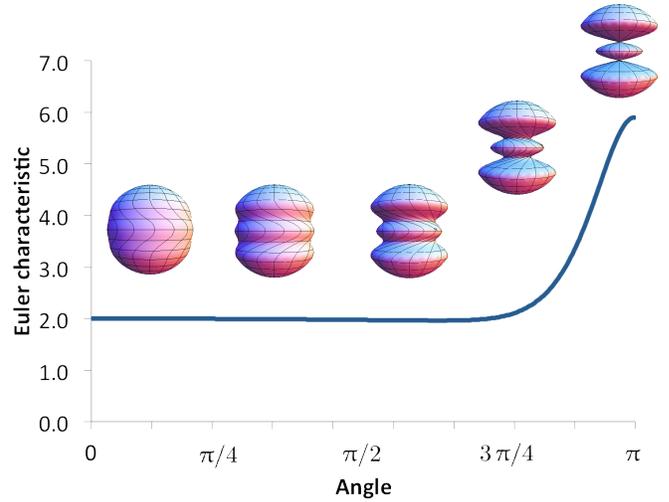


Fig. 6. Value of the Euler characteristic  $\chi(S)$  of the snake surface, computed using the Gauss-Bonnet formula when applying a rotation on the central layer of control points. As the layer rotates, the structure deforms and loses its sphere-like topology.

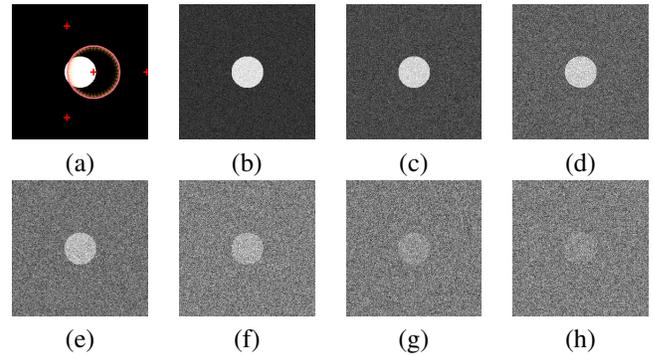


Fig. 7. Cross-sections of the test images used in the assessment of robustness. Only one realization is shown for each PSNR value. (a) Initialization. (b) PSNR = 20dB. (c) PSNR = 15dB. (d) PSNR = 10dB. (e) PSNR = 5dB. (f) PSNR = 0dB. (g) PSNR = -5dB. (h) PSNR = -10dB.

##### B. Robustness to Noise

Next, we investigate the sensitivity of our method to the presence of noise. We generated 100 realizations of a noisy sphere for each one of seven different peak-signal-to-noise ratios (PSNRs). Our test images were obtained by on-voxel-wise sampling a sphere of radius of 30 voxel units on a regular grid of  $(256 \times 256 \times 256)$  voxels. We show in Figure 7 one cross-section of the noisy volumes for every PSNR value.

The initial shape is a sphere with a fixed radius of 50 voxel centered at a distance of 25 voxel from the real center of the object to segment (see Figure 7). We constrained the number of control points to its minimum ( $M_1 = M_2 = 3$ ). Then, we ran the optimization process until convergence using exclusively the region-based energy (*i.e.*, we set the tradeoff parameter  $\alpha = 0$ ).

We used the Jaccard distance  $J = 1 - |\Theta \cap \Omega| / |\Theta \cup \Omega|$  to quantify, as a percentage, the quality of the segmentation. This distance provides a measure of dissimilarity between two binary objects where a low value reflects an accurate

PSNR [dB]	$J$ [%]	$\sigma_J$ [%]
20	0.20	0.16
15	0.44	0.26
10	0.61	0.35
5	2.16	2.68
0	12.16	10.05
-5	22.74	11.29
-10	34.69	9.82

TABLE I  
JACCARD DISTANCE WHEN SEGMENTING NOISY SPHERES.

segmentation. In the definition of the Jaccard distance,  $\Theta$  corresponds to the ground-truth region and  $\Omega$  corresponds to the region enclosed by the snake. We computed  $J$  following a on-voxel discretization of the data.

We show in Table I the value of  $J$  and its standard deviation  $\sigma_J$  across all noisy realizations. We observe from the results that our snake is robust against noise since it is capable of giving a proper segmentation even for low PSNRs. The quality of the segmentation deteriorates quickly when  $\text{PSNR} \geq 0$  due to the presence of too much noise.

### C. Segmentation of Overlapping Objects

In this section, we compare our snake against other segmentation methods in terms of accuracy and speed at the task of delineating different configurations of overlapping objects. The goal is to illustrate that the strong topological constraints imposed in the parametric model are advantageous when dealing with cluttered environments, without the introduction of any extra term in the energy functional. An application with real data of such advantage is presented in Section IV-E2.

We generated 4 volumetric images ( $256 \times 256 \times 256$ ) by voxel-wise sampling the union of two spheres of radius 50 pixel. We show a rendering of these shapes in Figure 8. They are parameterized with the distance  $d$ , in pixel units, between the centers of the spheres. For  $d < 100$ , the spheres intersect; for  $d = 100$ , the spheres share one single pixel; for  $d > 100$ , the spheres are disjoint. The grayscale values of the images are 255 for the shape and 0 for the background. We are interested in isolating each sphere. Without loss of generality, we focus on segmenting one of them.

We compared our snake to a traditional level-set method based on the formulation of Chan-and-Vese [1], and to the 3D active meshes of [3]. The implementation of the level-set method was taken from the free open-source image-processing package Fiji<sup>2</sup> implementing the algorithm described in [32]. The implementation of the active-meshes method was taken from the free open-source image-processing package Icy.<sup>3</sup>

We initialized the level-set method by providing a point seed (the only possible initialization afforded by the Fiji plugin). The initial positions were determined by a detector of local maxima applied over a version of the image that was smoothed with a Gaussian kernel of with  $\sigma = 10$ . A total number of 2 local maxima were detected in all images. We discarded the rightmost detection since we are interested in segmenting the

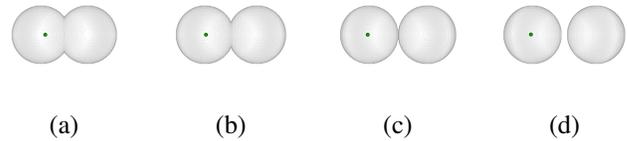


Fig. 8. Renderings of the test images used in the analysis of performance when segmenting overlapping objects. The dark small sphere represents the seed point for the level-set method, and the center of the initial position for our snake and the active meshes. (a)  $d = 80$ . (b)  $d = 90$ . (c)  $d = 100$ . (d)  $d = 110$ .

Method	$J$ [%]	Time [s]	Segmented shape
$d = 80$			
Spline Snake	3.06	0.93	Left sphere
Level-Set	48.66	2808.00	Merged spheres
Active Meshes	48.91	6.32	Merged spheres
$d = 90$			
Spline Snake	2.61	0.91	Left sphere
Level-Set	49.69	2862.03	Merged spheres
Active Meshes	50.11	6.01	Merged spheres
$d = 100$			
Spline Snake	0.83	0.93	Left sphere
Level-Set	50.00	2889.13	Merged spheres
Active Meshes	2.79	3.96	Left sphere
$d = 110$			
Spline Snake	0.71	0.93	Left sphere
Level-Set	0.25	1412.09	Left sphere
Active Meshes	1.98	4.44	Left sphere

TABLE II  
ACCURACY AND EFFICIENCY OF THE MENTIONED SEGMENTATION ALGORITHMS WHEN SEGMENTING OVERLAPPING OBJECTS.

leftmost object. We initialized our snake as a perfect sphere of radius 60 pixel units. Finally, the active meshes cannot take the form of an ideal sphere but can approximate it. We initialized this method using the automatic tessellation of the sphere of radius 60 provided by the plugin.

We chose  $M_1 = M_2 = 3$ , which are low values that favor ellipsoid-like shapes during the segmentation process. Then, we ran the optimization process until convergence using the edge-based energy (*i.e.*, we set the tradeoff parameter  $\alpha = 1$ ) and our internal energy with  $\lambda_1 = 0.1$  and  $\lambda_2 = 0.01$ .

We executed the level-set with an advection value of 220 and a propagation value of 100. These values were chosen to accelerate the propagation of the evolving level-set front and to obtain a faster convergence without losing accuracy.

For the active meshes, we set the mesh resolution to 10, the time-evolution step to 0.01, the window size to 100, and we evolved the snake using the gradient criterion with weight 0.5 and regularization weight 0.01.

We show in Table II a comparison across all mentioned methods of the Jaccard distance  $J$  reached at the end of the optimization process as well as the time it took the algorithms to converge for the different test datasets. The times shown in Table II exclude the preprocessing stages of the three methods.

Clearly, the level-set method is the slowest. This is in agreement existing experiments for the 2D case [7]. Meanwhile, our snake and the active meshes demonstrate a similar level

<sup>2</sup><http://fiji.sc/>

<sup>3</sup><http://icy.bioimageanalysis.org/>

of performance in terms of speed and accuracy, even though the active-meshes method takes advantage of the GPU present in the hardware, while our snake and the level-set method do not.

We can also see from Table II that the level-set method extracts a merged version of the two spheres as long as there exists a single pixel that connects them. This is due to the fact that there is no constrain on the evolution of standard level-sets, and that these can leak through holes. Some special energy functionals have been proposed in order to constrain the topology of the solutions at the expense of computational performance [33], [34]. On the other hand, the active meshes and our snake succeed in segmenting the left sphere alone even in the presence of some overlap. In addition, the proposed method is also the fastest.

#### D. Approximation of Shapes

In this section, we move away from numerical simulations, and we investigate the capabilities of our snake when approximating realistic shapes as a function of  $M_1$  and  $M_2$ . We quantify its accuracy at outlining the wall of a spleen within slices of a 3D CT-scan image sequence.

The data we used are part of the 3D-IRCADb (3D image reconstruction for comparison of algorithm database). It includes several sets of medical images of patients and the manual segmentation of the various structures of interest, performed by clinical experts.<sup>4</sup> For every patient under analysis, the ground truth is available as a triangular mesh where the vertex locations correspond to pixel positions. Moreover, the database provides a 3D voxel mask with the interior of the mesh. In the case of the mask, the volume consists of 166 slices with a spacing between slices of 1.8 mm. Each slice is a  $(512 \times 512)$  image with a pixel spacing of 0.961 mm.

To approximate the spleen with our snake, we first detect the boundary pixels of the spleen mask for each slice. Then, for each slice, we fit a spline corresponding to a circle of latitude of  $\sigma$ . Using this approach we obtain a snake  $\sigma$  aligned in the vertical direction. The north pole is located at the apex of the spleen, and the south pole is located at its bottom. The circles of latitude are adapted to the shape of the spleen in the  $XY$  plane. Then, we refine the final fit with a global 3D optimization led by the edge-based energy (*i.e.*, we set the tradeoff parameter  $\alpha = 1$ ), and our internal energy with  $\lambda_1 = 0.1$  and  $\lambda_2 = 0.01$ . For computational purposes, we discretized each integral with approximately the same number of samples. Specifically, we set the number of points along a meridian and a circle of latitude to the multiple of  $M$  that is the closest to  $N_{\text{sampl}} = 39$ .

In the simulations of Figure 9, we investigated the dependence of the Jaccard distance  $J$  and the total computation time on  $M_1 = M_2 = M$ . Our results show that the error decreases quadratically, which demonstrates the ability of the proposed model to segment objects with an ellipsoid-like topology. We also observe that the computation time is increasing quadratically with  $M$  plus some fluctuations. This is due to the fact that each iteration of the algorithm involves the

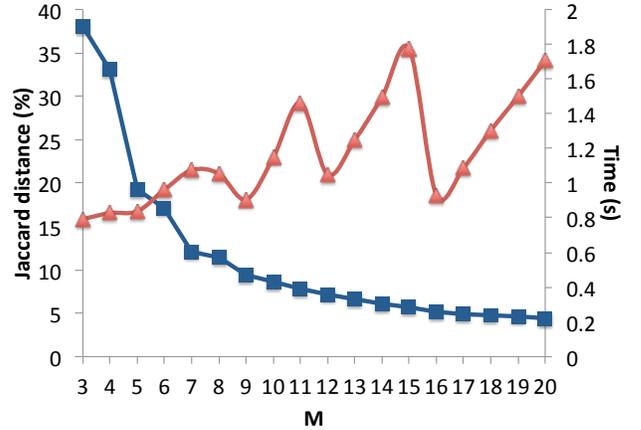


Fig. 9. Evolution of the Jaccard distance as a function of  $M = M_1 = M_2$  when approximating a spleen.

computation of the energy over  $\mathcal{O}(M^2 [N_{\text{sampl}}/M]^2)$  surface points plus the update of  $\mathcal{O}(M^2)$  control points. This sums up to a total complexity of  $\mathcal{O}(M^2 [N_{\text{sampl}}/M]^2 + M^2)$ , where the first term is the dominant one within the range of interest of  $M$ .

In Figure 10, we show the voxelized mask we used as ground truth and the successive approximations of our snake for different values of  $M$ . We see that, for small values  $M$  (such as  $M = 3$ ), the snake takes an almost ellipsoidal shape and is not capable of capturing every detail of the spleen structure. As we increase the number of control points, the snake captures the structure of the organ while providing a smooth surface.

#### E. Segmentation of 3D Confocal Microscopic Images

We finally illustrate the behavior of our snake and provide further insights into its capabilities in real-world applications. In this section, the ground truth is missing, so we must relinquish quantitative assessments in favor of qualitative ones. Here, we initialized our snake manually using the interaction capabilities of our software.

1) *Cell-Body Segmentation*: We processed a stack ( $576 \times 504 \times 200$ ) of confocal ( $\times 60$  magnification) images from the brain cortex of a rat, with YFP labeling for the neurons and GFP for the microglia.<sup>5</sup> The challenge was to segment the body of the neuronal cells, despite their non-spherical shape, the lack of clear borders, and the presence of several surrounding objects. We set  $M_1 = M_2 = 7$ , roughly initializing the snake position around each cell body as a sphere, and ran the optimization process until convergence using exclusively the edge-based energy (*i.e.*, we set the tradeoff parameter  $\alpha = 1$ ). One example of the resulting segmentation is shown in Figure 5. We observe that the snake was able to adapt well to the 3D cell shape: the surface is accurately fitted despite the limited degrees of freedom of the model, while irregularities are properly smoothed out. The optimization process took only 0.74 s, which is faster than

<sup>4</sup><http://www.ircad.fr/software/3Dircadb/3Dircadb.php>

<sup>5</sup><http://www.cellimagelibrary.org/images/27155/>

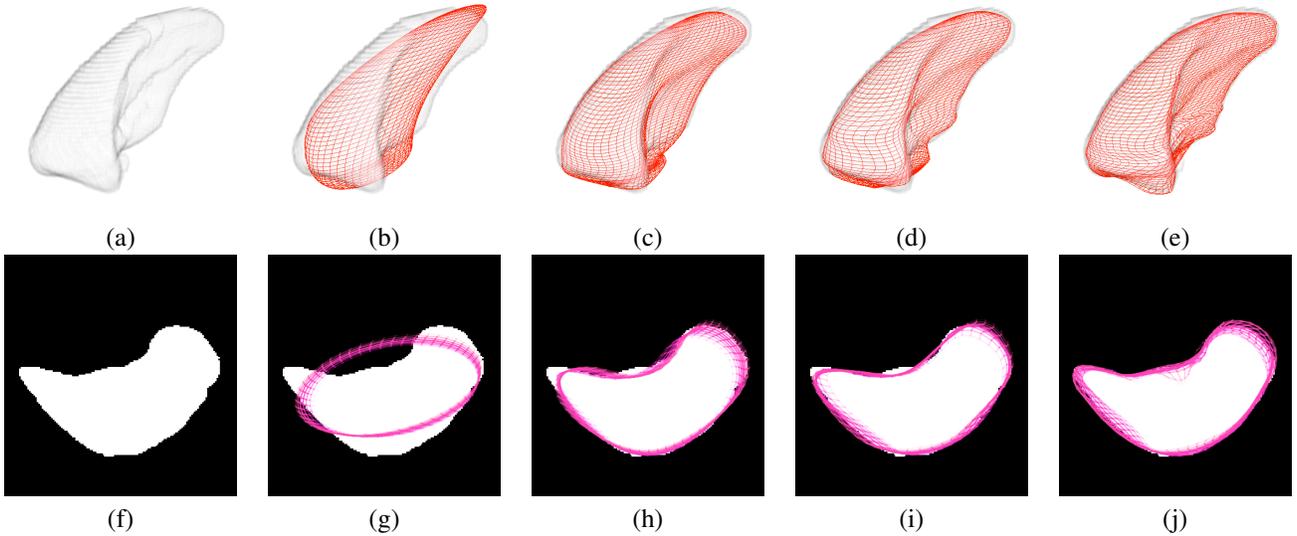


Fig. 10. Approximation of a 3D CT-scan spleen for different values of  $M = M_1 = M_2$ . (a)-(e) Rendering of the spleen where the 3D snake has been overlaid with different number of control points: (b)  $M = 3$ , (c)  $M = 7$ , (d)  $M = 10$ , and (e)  $M = 20$ . (f) Cross-section of the spleen where the neighboring 3D mesh has been projected to the slice of interest. (g)  $M = 3$ , (h)  $M = 5$ , (i)  $M = 7$ , and (j)  $M = 10$ . In the 2D views, we color the mesh depending on the  $z$ -axis coordinate and set its transparency proportional to the distance of the displayed slice.

the duration of the acquisition of such data (usually, much longer than 1 s).

2) *Segmentation of Glomeruli*: We investigated the segmentation of olfactory glomeruli in the mouse brain which represent neuroglial functional units in olfactory information processing [35]. With Topro staining, glomeruli correspond to the dark areas delimited by fluorescent cell bodies. In Figure 11, they are visualized as bright areas with an inverted look-up-table. However, the surrounding fluorescent cells do not form continuous boundaries. This penalizes nonparametric active contours, as the snake may ‘leak out’ between the neighboring cells. By contrast, we show in Figure 11 that the proposed snake method (with  $M_1 = M_2 = 7$ ) is able to accurately identify the glomerulus border. This is a key advantage of the proposed parametric model, as it allows one to exert *a priori* control over the regularity and topology of the snake. Here again, the optimization process was performed using the edge-based energy exclusively (*i.e.*,  $\alpha = 1$ ). Convergence was reached after 1.74 s, which is remarkably fast.

## V. CONCLUSION

Our contribution in this paper is a new fully parametric snake with a sphere-like topology. It is constructed using exponential B-splines and it is therefore capable of reproducing any ellipsoid, irrespective of its position or orientation. Our snake is characterized by fewer control points than nonparametric snakes and can approximate any blob-like structure with arbitrary precision. The modification of one control point affects a limited region of the snake surface, which favors intuitive interactions with the user. Since our shape model is fully characterized by few control points, the design of customized shapes becomes possible by simple manipulation of these points in the same way that control points are used in the NURBS meshes typical of computer-aided industrial designs [36]. Moreover, the control points

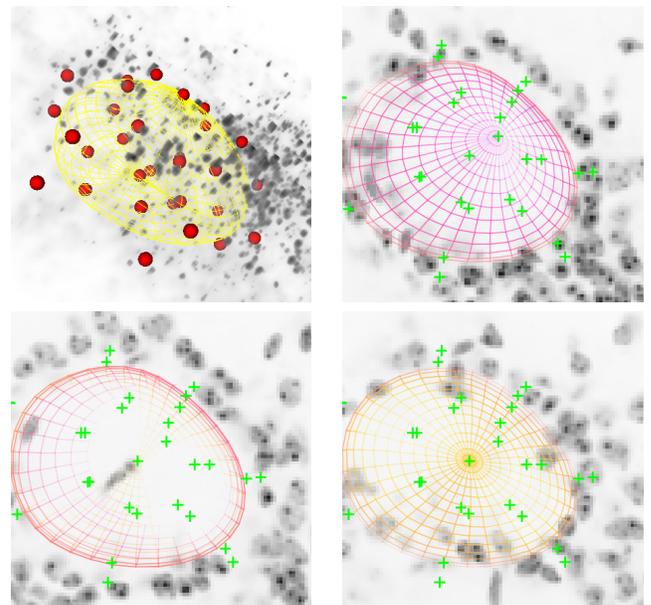


Fig. 11. Segmentation of an olfactory glomerulus in a 3D confocal image ( $256 \times 256 \times 67$ ) of a mouse brain (image courtesy of Lisa Roux at the Collège de France). Top-left: 3D view of the segmented glomerulus. The solid spheres represent the control points of the model and their location define the effective contour of the snakes. From left to right and top to bottom: snake at different  $z$ -axis positions ( $z = 10, 17$ , and  $24 \mu\text{m}$ ) of the 3D volume overlaid on the original 2D images. We color the mesh depending on the  $z$ -axis coordinate and set its transparency proportional to the distance of the slice displayed in the 2D viewer. The ‘+’ elements are the projection to the XY plane of the control points of the model.

may be used to perform statistical learning/analysis of the segmented objects [37].

We designed an edge-based energy that is capable of maintaining the consistency of the segmentation in the presence of clutter. This is accomplished by penalizing mismatches in the

directions of the image gradients. Furthermore, we combined it with a robust region-based energy. We were able to accelerate the implementation by taking advantage of Gauss' theorem, which was facilitated by the availability of explicit expressions of our bases. Moreover, we introduced a novel technique to detect self-intersection in order to know when the snake loses the sphere-like topology. We have applied our snakes to a variety of problems that involve synthetic simulations and challenging real datasets, where the object contours were not well defined. As a result, various experiments have shown that the proposed 3D snake can approximate blob-like objects with good accuracy. Moreover, the optimization process is remarkably fast as we have designed our bases to have the shortest-possible support.

## REFERENCES

- [1] T. Chan and L. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, February 2001.
- [2] H. Delingette, "General object reconstruction based on simplex meshes," *International Journal of Computer Vision*, vol. 32, no. 2, pp. 111–146, September 1999.
- [3] A. Dufour, R. Thibeaux, E. Labruyere, N. Guillen, and J.-C. Olivo-Marin, "3-D Active meshes: Fast discrete deformable models for cell tracking in 3-D time-lapse microscopy," *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1925–1937, July 2011.
- [4] Q. Duan, E. Angelini, and A. Laine, "Surface function actives," *Journal of Visual Communication and Image Representation*, vol. 20, no. 7, pp. 478–490, October 2009.
- [5] D. Barbosa, T. Dietenbeck, J. Schaerer, J. D'hooge, D. Friboulet, and O. Bernard, "B-Spline explicit active surfaces: An efficient framework for real-time 3-D region-based segmentation," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 241–251, January 2012.
- [6] R. Delgado-Gonzalo, N. Chenouard, and M. Unser, "Fast parametric snakes for 3D microscopy," in *Proceedings of the Ninth IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI'12)*, Barcelona, Spain, May 2–5, 2012, pp. 852–855.
- [7] R. Delgado-Gonzalo, P. Thévenaz, C. Seelamantula, and M. Unser, "Snakes with an ellipse-reproducing property," *IEEE Transactions on Image Processing*, vol. 21, no. 3, pp. 1258–1271, March 2012.
- [8] P. Dierckx, "Algorithms for smoothing data on the sphere with tensor product splines," *Computing*, vol. 32, no. 4, pp. 319–342, April 1984.
- [9] R. Gmelig Meyling and P. Pfluger, "B-spline approximation of a closed surface," *IMA Journal of Numerical Analysis*, vol. 7, no. 1, pp. 73–96, January 1987.
- [10] L. Schumaker and C. Traas, "Fitting scattered data on spherelike surfaces using tensor products of trigonometric and polynomial splines," *Numerische Mathematik*, vol. 60, no. 1, pp. 133–144, January 1991.
- [11] W. Boehm, "On de Boor-like algorithms and blossoming," *Computer Aided Geometric Design*, vol. 5, no. 1, pp. 71–79, June 1988.
- [12] P. Costantini, "An algorithm for computing shape-preserving interpolating splines of arbitrary degree," *Journal of Computational and Applied Mathematics*, vol. 22, no. 1, pp. 89–136, April 1988.
- [13] M. Unser, "Sampling—50 Years after Shannon," *Proceedings of the IEEE*, vol. 88, no. 4, pp. 569–587, April 2000.
- [14] M. Do Carmo, *Differential Geometry of Curves and Surfaces*, 1st ed. Prentice-Hall, 1976.
- [15] A. Aldroubi and M. Unser, "Sampling procedures in function spaces and asymptotic equivalence with Shannon's sampling theory," *Numerical Functional Analysis and Optimization*, vol. 15, no. 1–2, pp. 1–21, 1994.
- [16] C. de Boor and R. DeVore, "Partitions of unity and approximation," *Proceedings of the American Mathematical Society*, vol. 93, no. 4, pp. 705–709, April 1985.
- [17] R. Delgado-Gonzalo, P. Thévenaz, and M. Unser, "Exponential splines and minimal-support bases for curve representation," *Computer Aided Geometric Design*, vol. 29, no. 2, pp. 109–128, February 2012.
- [18] L. Boydston, T. Armstrong, and F. Bookstein, "A comparison of three dimensional maximum reach estimation techniques," *Journal of Biomechanics*, vol. 13, no. 8, pp. 717–724, 1980.
- [19] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, January 1987.
- [20] L. Staib and J. Duncan, "Boundary finding with parametrically deformable models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1061–1075, November 1992.
- [21] P. Brigger, J. Hoeg, and M. Unser, "B-Spline snakes: A flexible tool for parametric contour detection," *IEEE Transactions on Image Processing*, vol. 9, no. 9, pp. 1484–1496, September 2000.
- [22] M. Figueiredo, J. Leitão, and A. Jain, "Unsupervised contour representation and estimation using B-splines and a minimum description length criterion," *IEEE Transactions on Image Processing*, vol. 9, no. 6, pp. 1075–1087, June 2000.
- [23] M. Jacob, T. Blu, and M. Unser, "A unifying approach and interface for spline-based snakes," in *Proceedings of the SPIE International Symposium on Medical Imaging: Image Processing (MI'01)*, San Diego, CA, USA, February 19–22, 2001, pp. 340–347.
- [24] —, "Efficient energies and algorithms for parametric snakes," *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1231–1244, September 2004.
- [25] P. Thévenaz, R. Delgado-Gonzalo, and M. Unser, "The ovuscule," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 382–393, February 2011.
- [26] F. Precioso and M. Barlaud, "B-spline active contours for fast video segmentation," in *Proceedings of the 2001 International Conference on Image Processing*, Thessaloniki, Greece, October, 7–10, 2001, pp. 777–780.
- [27] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. Cambridge University Press, 1986.
- [28] C. Hoffmann, *Geometric and Solid Modeling*, 1st ed. Morgan Kaufmann, 1989.
- [29] T. McInerney and D. Terzopoulos, "T-snakes: Topology adaptive snakes," *Medical Image Analysis*, vol. 4, no. 2, pp. 73–91, June 2000.
- [30] F. de Chaumont, S. Dallongeville, N. Chenouard, N. Hervé, S. Pop, T. Provoost, V. Meas-Yedid, P. Pankajakshan, T. Lecomte, Y. Le Montagner, T. Lagache, A. Dufour, and J.-C. Olivo-Marin, "Icy: An open bioimage informatics platform for extended reproducible research," *Nature Methods*, vol. 9, no. 7, pp. 690–696, July 2012.
- [31] L. Zhang, E. Hoffman, and J. Reinhardt, "Atlas-driven lung lobe segmentation in volumetric X-ray CT images," *IEEE Transactions on Medical Imaging*, vol. 25, no. 1, pp. 1–16, January 2006.
- [32] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.
- [33] X. Han, C. Xu, and J. Prince, "A topology preserving level set method for geometric deformable models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, pp. 755–768, June 2003.
- [34] O. Duchenne, J.-Y. Audibert, R. Keriven, J. Ponce, and F. Ségonne, "Segmentation by transduction," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*, Anchorage, AK, USA, June 23–28, 2008, pp. 852–855.
- [35] L. Roux, K. Benchenane, J. Rothstein, G. Bonvento, and C. Giaume, "Plasticity of astroglial networks in olfactory glomeruli," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 108, no. 45, pp. 18442–18446, 2011.
- [36] L. Piegl and W. Tiller, *The NURBS Book*, 2nd ed. Springer Berlin Heidelberg, 2010.
- [37] D. Cremers and C. Schnörr, "Statistical shape knowledge in variational motion segmentation," *Image and Vision Computing*, vol. 21, no. 1, pp. 77–86, January 2003.



**Ricard Delgado-Gonzalo** was born in 1983 in Barcelona, Spain. He received two diplomas in Telecommunications Engineering and in Mathematics from the Universitat Politècnica de Catalunya (UPC) in 2006 and 2007, respectively. In 2008, he joined the Biomedical Imaging Group at the École polytechnique fédérale de Lausanne (EPFL), Switzerland, where he obtained this Ph.D. in biomedical image processing in 2013. He currently works on applied problems related to image reconstruction, segmentation, and tracking, as well as on the mathematical foundations of signal processing and spline theory. His other important professional interests include technology management and transfer.



**Nicolas Chenouard** (M'08) received the M.S. degree in engineering and bioinformatics from the Institut National des Sciences Appliquées, Lyon, France, and the M.S. degree in artificial intelligence and data mining from the University Pierre et Marie Curie (Paris VI), Paris, France, both in 2006, and the Ph.D. degree in image and signal processing from Institut Pasteur and Telecom ParisTech, Paris, in 2010. He was a Post-Doctoral Fellow with the Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, and

the Center for Biomedical Imaging, Université de Lausanne, Lausanne, from 2010 to 2012. He was engaged in research on the design of steerable wavelet frames in multiple dimensions, learning techniques for adaptive frame design, and bioimaging. Since March 2012, he has been a Post-Doctoral Fellow with the Langone Medical Center, New York University, New York. His current research interests include neuroscience, in particular, synaptic mechanisms by using advanced microscopy and image processing techniques.



**Michael Unser** (M'89, SM'94, F'99) received the M.S. (summa cum laude) and Ph.D. degrees in Electrical Engineering in 1981 and 1984, respectively, from the École Polytechnique Fédérale de Lausanne (EPFL), Switzerland. From 1985 to 1997, he worked as a scientist with the National Institutes of Health, Bethesda USA. He is now full professor and Director of the Biomedical Imaging Group at the EPFL. His main research area is biomedical image processing. He has a strong interest in sampling theories, multiresolution algorithms, wavelets, and

the use of splines for image processing. He has published 200 journal papers on those topics.

Dr. Unser has held the position of associate Editor-in-Chief (2003-2005) for the IEEE Transactions on Medical Imaging and has served as Associate Editor for the same journal (1999-2002; 2006-2007), the IEEE Transactions on Image Processing (1992-1995), and the IEEE Signal Processing Letters (1994-1998). He is currently member of the editorial boards of Foundations and Trends in Signal Processing, and Sampling Theory in Signal and Image Processing. He co-organized the first IEEE International Symposium on Biomedical Imaging (ISBI2002) and was the founding chair of the technical committee of the IEEE-SP Society on Bio Imaging and Signal Processing (BISP).

Dr. Unser received the 1995 and 2003 Best Paper Awards, the 2000 Magazine Award, and two IEEE Technical Achievement Awards (2008 SPS and 2010 EMBS). He is an EURASIP Fellow and a member of the Swiss Academy of Engineering Sciences.