SEGMENTATION BY SMOOTHING B-SPLINE ACTIVE SURFACE

Jérôme VELUT, Hugues BENOIT-CATTIN, Christophe ODET

CREATIS, CNRS UMR 5515, Inserm U 630 INSA,
Bât. B. Pascal, 69621 Villeurbanne, France

ABSTRACT

In this paper, a new active surface based segmentation algorithm is proposed. We detail the regularization process that is conducted through a smoothing B-spline filtering of the mesh point displacements. The corresponding segmentation algorithm is presented and applied on medical data. The proposed segmentation method is fast and suitable to quadrangular meshes.

Index Terms— image segmentation, spline functions, IIR digital filters.

1. INTRODUCTION

Active contours [1] are one of the existing segmentation strategy well adapted to boundary detection in noisy data. Active contours, called snakes, have been widely investigated. Recently, smoothing B-spline filtering has been introduced in the regularization process and it improves the segmentation robustness as well as the convergence speed [2]. One of the applications is the medical imaging that needs robust, quick and reproducible segmentations. As medical imaging counts 3D data, a natural extension of the snakes are the active surfaces [3] that are able to detect the surface of an object in a 3D image.

In this paper we propose a new smoothing B-snake algorithm and its extension to an active surface regularized by smoothing B-spline filtering. Next section gives the theoretical background of snakes, B-Splines and their links. Section 3 presents our algorithm that is illustrated by some medical image segmentations in Section 4.

2. SNAKES AND B-SPLINES

We give in the following some basics on snakes and their smoothing B-spline extension that are the basis concepts of our smoothing B-spline active surface.

2.1. Snake basics.

A snake [1] is a parametric curve *g(s)=(x(s),y(s))* that evolves on an image *I(x,y)* and stops on the features of interest. The snake evolution is controlled by an energy minimization. The curve has to go where its total energy E_{snake}, defined in equation (1), is minimal.

$$E_{snake} = \int_s E_{int}(g(s)) + E_{ext}(g(s))ds \qquad (1)$$

where E_{int} has a regularization role and is the internal energy that traduces shape constraints on the curve; Where E_{ext} is the external energy that drives the snake toward image features.

Typically, E_{ext} is computed from the gradient of the image (Eq. 2) as the usual goal of a snake is to detect object boundaries.

$$E_{ext} = -\left|\nabla I(x,y)\right|^2 \qquad (2)$$

The variational method used in [1] to complete the minimization of equation (1) leads to a force balance given by:

$$A \cdot g(k) + f(k) = 0 \qquad (3)$$

where A is a pentadiagonal banded matrix built from E_{int}, where *g(k)=(x(k),y(k))* is the discrete version of the *g(s)* curve and where $f(k)=(f_x(k),f_y(k))$ constitutes the external forces computed at each *k* snake's point as follows:

$$f(k) = (f_x(k), f_y(k)) = \left(\frac{\partial E_{ext}}{\partial x(k)}, \frac{\partial E_{ext}}{\partial y(k)}\right) \qquad (4)$$

The gradient descent method determines (Eq. 5) the new positions of the snake points $(g_i(k))$ at each iteration.

$$g_i(k) = (A + \gamma \cdot I)^{-1}(\gamma \cdot g_{i-1}(k) - f(k)) \qquad (5)$$

where γ is a step-size parameter and *i* is the iteration index of the gradient descent evolution.

This method is used by Menet et al. [4] where *g(s)* is represented through B-spline. Such model offers a better local continuity control and a faster convergence. In [5], Brigger et al. show that the B-spline representation of a snake induces an built-in smoothness of the curve. They also propose a variant sample step of the curve to bring an implicit regularization. The internal energy and its regularization effect are implicit and the iterative process enounced in equation (5) is simplified to:

$$g_i(k) = g_{i-1}(k) - \gamma \cdot f(k) \qquad (6)$$

f(k) may be balloon forces [6], or gradient vector flow [7] or any other forces that lead the snake to the desired features.

2.2. Regularization through Smoothing B-spline filtering

Although variant sampling step is a mean of regularization, the snake at low resolution will not be able to embed enough prior knowledge in an initial model, due to the lack of points. Precioso et al. [2] propose the use of a low-pass IIR filter on each parametric components of the curve to regularize the snake. This filter is the smoothing B-spline filter described in [8]. It computes a more or less smooth approximation $\hat{g}(k)$ of a finite set of points *g(k)*, where the continuous representation $\hat{g}(s)$ of $\hat{g}(k)$ minimizes the following functional:

$$\varepsilon_s^2 = \sum_{k=-\infty}^{+\infty}\left(g(k)-\hat{g}(k)\right)^2 + \lambda \int_{-\infty}^{+\infty}\left(\frac{\partial^2 \hat{g}(s)}{\partial s^2}\right)^2 ds \qquad (7)$$

where λ is the parameter that tunes the smoothness constraint of $\hat{g}(k)$.

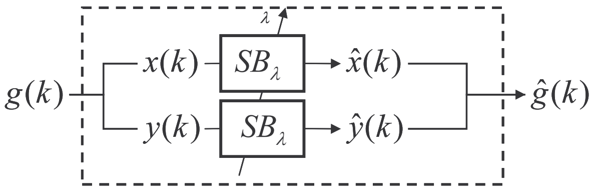

Figure 1. Smoothing B-spline filtering of a parametric discrete signal g(k).

Figure 1 shows the filtering process involved in [2]. It is based on the filtering of a parametric signal by the SB_λ filter which cut-off frequency is controlled by *λ*.

Even if this algorithm involved a region-based criterion, the evolution of the snake is similar to (6). Nevertheless, the way of using the smoothing B-spline filter in the Precioso's algorithm has some drawbacks. Indeed, the regularization amount increases as the iterations and consequently the final snake smoothness is dependent on the initial one. Precioso et al. [2] gives empiric values of suitable *λ* in their algorithm (*λ* can take value in [0.1,1]) to avoid these problems.

We propose in the next section another algorithm that regularizes a snake with a smoothing B-spline filter (Figure 2) without the drawbacks mentioned above and without any *λ* restriction. The proposed algorithm is finally extended to an active surface framework.

3. SMOOTHING B-SPLINE ACTIVE SURFACES

3.1. Smoothing B-Snake

Figure 2 shows the proposed smoothing B-snake algorithm. Note that a 2D locally regularized version of this algorithm and details regarding the filtering process are given in [9]. First of all, we consider the snake without any regularization *(g(k)=* $\hat{g}(k)$*)* and we define an iterative process equivalent to (Eq. 6):

$$g_i(k) = g_{i-1}(k) + \gamma \cdot d_i(k) \qquad (8)$$

where $g_i(k)$ is a snake point $(x_i(k),y_i(k))$ at iteration *i*, and where $d_i(k)$ is the deformation force under each snake point and is comparable to *f* computed from external energy.

Then, we introduce the regularization by smoothing the non regularized curve at iteration *i*:

$$\hat{g}_i(k) = sb_\lambda(k) * g_i(k) \qquad (9)$$

where sb_λ is the impulse response of the filter SB_λ.

From equations (8) and (9), we get:

$$\hat{g}_{i+1}(k) = \hat{g}_i(k) + (\gamma \cdot d_i(k)) * sb_\lambda(k) \qquad (10)$$

Equation (10) shows that the regularization is a smoothing B-spline filtering of the deformation forces, and it justifies the algorithm presented in Figure 2.

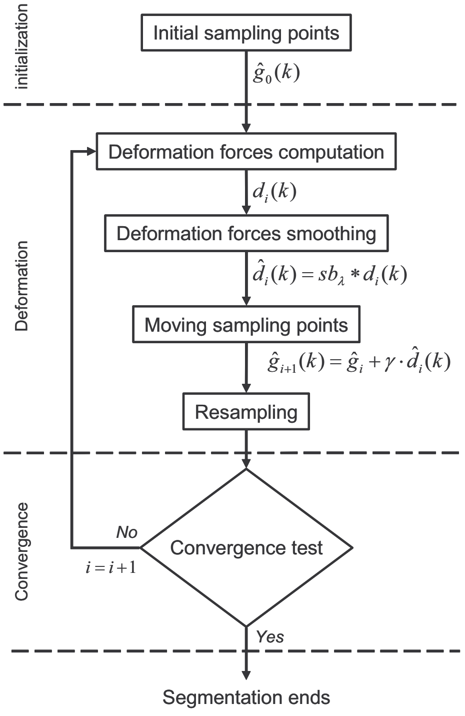

Figure 2. Smoothing B-Snake algorithm.

When $\hat{g}_{i+1}(k)$ is computed, the curve is resample to ensure a constant sample step. This is necessary to let the cut-off frequency associated to *λ* having the same smoothing effect along the curve.

In the next section, a smoothing B-snake is extended to an active surface framework.

3.2. Smoothing B-spline filter: 2D

One issue in generalizing the previous smoothing B-snake to an active surface is the topology property of the considered spaces. A snake being a parametric curve described with one parameter *s*, any point has only two neighbors that makes easy the digital filtering of such a signal. In the case of 2D surface meshes, we need to adapt the smoothing B-spline filtering process and we have to consider suitable surfaces.

To keep an efficient implementation, we want the smoothing spline filter to be separable, that means that the filter processes independently the *k* and *l* directions. Unser et al. [8] show that the 2D smoothing B-spline functional (Eq. 7) must comprise appropriate cross terms to conserve the separability:

$$\begin{aligned}\varepsilon_s^2 = &\sum_{k=-\infty}^{+\infty}\sum_{l=-\infty}^{+\infty}\left(g(k,l)-\hat{g}(k,l)\right)^2 \\ &+\lambda\int_{-\infty}^{+\infty}\sum_{l=-\infty}^{+\infty}\left(\frac{\partial^2\hat{g}(u,l)}{\partial u^2}\right)^2 du \\ &+\mu\sum_{k=-\infty}^{+\infty}\int_{-\infty}^{+\infty}\left(\frac{\partial^2\hat{g}(k,v)}{\partial v^2}\right)^2 dv \\ &+\lambda\mu\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\left(\frac{\partial^2\hat{g}(u,v)}{\partial u^2\partial v^2}\right)^2 dudv\end{aligned} \qquad (11)$$

where λ and μ are the smoothing parameters for both directions. In the following, we consider $\lambda = \mu$ because at the present time, there is no reason to favor one of the parametric direction on the surface.

Even if *ĝ(k,l)* is not the optimal solution of (11) [8], the smoothing effect is still suitable for our active surface regularization. The Figure 3 illustrates this smoothing effect on a cylindric and a torus mesh.

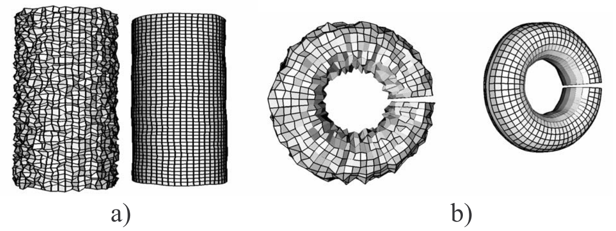

Figure 3. Noisy cylinder (a) and torus (b) smoothing B-spline filtering.

3.3. Plan-homeomorphic mesh filtering

The 2D smoothing B-spline filtering has to be applied on a Cartesian grid: Each filtered point is represented in Cartesian coordinate system. In a 2D surface mesh context, it means that the surface has to accept a bijective parameterization in the 2D domain.

Let *G* being our surface described by the function *g(u,v)*:

$$g(u,v) = (g_x(u,v), g_y(u,v), g_z(u,v)) \qquad (12)$$

and the discrete representation of *G* (that is called a mesh) is given by:

$$g(k,l) = (g_x(k,l), g_y(k,l), g_z(k,l)) \qquad (13)$$

where *k* and *l* are the index of the points constituting the mesh surface *G*.

The bijective parametrization condition is translated into a topology constraint: Suitable surfaces are the one homeomorphic to the infinite plan as in Figure 3 (torus, infinite cylinders and infinite plans). One property of these surface meshes is that they can be represented only with square cells (that is the Cartesian grid), and we talk about quadrangular surface meshes. Each point has 4 neighbours (2 neighbours on each direction *k* and *l*) and each edge has 2 adjacent faces.

3.4. Genus zero surface filtering

Genus zero surfaces gather every surface that is homeomorphic to a sphere and can be represented with square cell almost everywhere. Indeed, only 2 points, the sphere poles, are not 4-connected. Their number of neighbours depends on the resolution along the parallel lines. As a sphere whose poles have been removed is equivalent to a cylinder, we can apply the smoothing B-spline filter 2D as in §3.3. The poles have to be processed differently: In [10], Taubin et al. gives an iterative method to smooth surfaces with a signal processing formalism. Desbrun et al. extend this surface smoothing in [11]. Although these methods are somewhat computationally costly, we apply them to filter the 2 sphere poles as illustrated Figure 4.

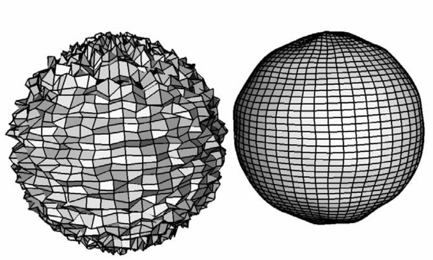

Figure 4. Noisy sphere smoothing B-spline filtering.

3.5. Deformation algorithm

As explained in section 3.1, our regularisation is done through a smoothing B-spline filtering of the deformation forces. The smoothing B-spline active surface is based on the same equations (Eq. 8-9-10) except that *ĝ(k)* is now *ĝ(k,l)* and that *d(k)* becomes *d(k,l)* with 3 components:

$$\hat{g}_{i+1}(k,l) = \hat{g}_i(k,l) + (\gamma \cdot d_i(k,l)) * sb_\lambda(k,l) \qquad (14)$$

$$d(k,l) = \left(d_x(k,l), d_y(k,l), d_z(k,l)\right) \qquad (15)$$

d(k,l) may be seen as a vector field that is recomputed and smoothed at each iteration as in the smoothing B-snake algorithm (Figure 2).

The smoothing B-spline active surface is then a direct generalization of smoothing B-snake described in section 3.1. The algorithm is not modified as long as the initial surface is represented by a quadrangular mesh everywhere. Singular points (also call “zero points”, like the poles of a sphere) are processed using the Taubin et al method [10].

4. RESULTS

In this section, we illustrate the smoothing B-spline active surface segmentation method on 3D MRI images.

Figure 5-a shows an MRI angiography of a human aorta. The initial surface is a cylindrical mesh initialized roughly near the real object. In Figure 5-c, a low regularization (*λ=2*) and 37 iterations allows the surface to be close to the real data but the segmentation is sensitive to noise. In Figure 5-b (*λ=30*) the segmentation is not perturbed by noise while being close to the aorta, with 33 iterations.

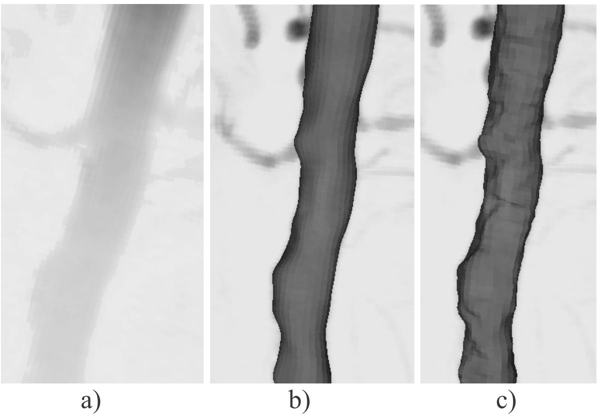

Figure 5. Aorta MRI angiography segmentation a) Original data and final segmentation with b) λ=30 c) λ=2.

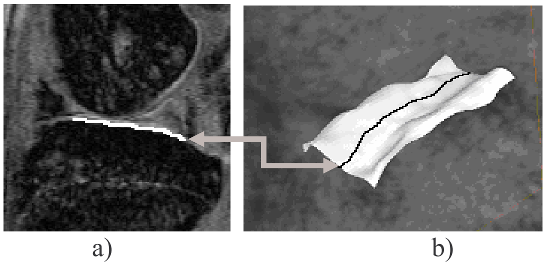

Figure 6. MRI of a guinea pig knee. a) Slice extracted from the data volume. b) final segmentation with λ=10 and planar initial surface.

Figure 6 illustrates the segmentation of the tibia’s plate from a guinea-pig 3D MRI after 23 iterations. In this case, the initial surface is a planar mesh. The black line on Figure 6-b represents the intersection between the surface and the slice of Figure 6-a.

5. CONCLUSIONS

In this paper, a new active surface based segmentation algorithm is proposed. The regularization process is done trough smoothing B-spline filtering of the mesh point displacements. Such a regularization is controlled by a unique parameter λ that tunes the cut-off frequency of the smoothing B-spline filter. This approach is fast due to the 1D digital filtering used. It is well adapted to quadrangular meshes homeomorphic to a plan (plan, torus, cylinder). Furthermore, by using a specific management of singular points, genus zero surfaces can be used in the segmentation process.

6. ACKNOWLEDGEMENTS

This work is in the scope of the scientific topics of the PRC-GdR ISIS research group of the French National Center for Scientific Research CNRS.

7. REFERENCES

[1] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," presented at ICCV, London, 1987.

[2] F. Precioso, M. Barlaud, T. Blu, and M. Unser, "Robust real-time segmentation of images and videos using a smooth-spline snake-based algorithm," IEEE Transactions on Image Processing, vol. 14, pp. 910-924, 2005.

[3] X. Chen and E. K. Teoh, "3D object segmentation using B-Surface," Image and Vision Computing, vol. 23, pp. 1237-1249, 2005.

[4] S. Menet, P. Saint-Marc, and G. r. Medioni, "B-Snakes : Implementation and Application to Stereo," Image Understanding Workshop., pp. 720-726, 1990.

[5] P. Brigger, J. Hoeg, and M. Unser, "B-spline snakes: a flexible tool for parametric contour detection," IEEE Transactions on Image Processing, vol. 9, pp. 1484-1496, 2000.

[6] L. D. Cohen, "On Active Contour Models," INRIA, Rocquencourt, Programmation, Calcul Symbolique et Intelligence Artificielle 1075, 1989.

[7] C. Xu and J. L. Prince, "Generalized gradient vector flow external forces for active contours," Elsevier - Signal Processing., vol. 71, pp. 131-139, 1998.

[8] M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing. Part I. Theory," IEEE Transactions on Signal Processing, vol. 41, pp. 821-833, 1993.

[9] J. Velut, H. Benoit-Cattin, and C. Odet, "Locally Regularized Snake Through Smoothing B-Spline Filtering," presented at EUSIPCO, Firenze, 2006.

[10] G. Taubin, "A signal Processing Approach to Fair Surface Design," presented at SIGGRAPH, 1995.

[11] M. Desbrun, M. Meyer, P. Schroder, and A. H. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," Proceedings of the ACM SIGGRAPH Conference on Computer Graphics, pp. 317-324, 1999.