Disponible en ligne sur
SciVerse ScienceDirect
www.sciencedirect.com

Elsevier Masson France
EM|consulte
www.em-consulte.com

IRBM

IRBM 34 (2013) 235–243

Original article

Spline-based framework for interactive segmentation in biomedical imaging[☆]

R. Delgado-Gonzalo, M. Unser

Biomedical Imaging Group, École polytechnique fédérale de Lausanne (EPFL), Switzerland

Received 15 November 2012; received in revised form 7 April 2013; accepted 12 April 2013
Available online 24 May 2013

Abstract

Active contours constitute a computationally attractive framework for image segmentation. In this paper, we describe a fully parametric design that relies on B-spline bases and is specified by control points on the image. Our technique yields successful segmentation results even for challenging datasets where object contours are not well-defined. We can achieve this because our parametric approach uses few parameters and allows us to constrain the topology of the curve. We provide the details for an efficient implementation and discuss different design alternatives. This work comes with a companion software that gives the end user full control over the snakes. We demonstrate the features of our software which includes an interactive interface where the user can manipulate the position of control points by simple mouse actions.

1. Introduction

Segmentation is one of the key tasks in image analysis. In medicine, the anatomical structures that appear in magnetic resonance (MR) or computed tomography (CT) scans are often segmented from the image for use in surgical planning, navigation, diagnosis, and therapy evaluation. In biology, the extraction of accurate cell outlines allows one to perform quantitative statistical measurements within cell structures avoiding spurious fluctuations from the image background.

Active contours, also called snakes, have proved to be very effective tools in this field [1–3]. A snake is a curve that evolves within an image from an initial position, which is often specified by a user, toward the boundary of an object. The evolution of the curve is formulated as a minimization problem. The associated cost function is called the *snake energy*. Snakes have become popular because it is possible for the user to interact with them, not only when specifying their initial position, but also during the segmentation process. A practical issue with these methods is that they may overfit the image data in very noisy situations. Otherwise, they may leak out through gaps or holes in image structures when the image gradient separating the different parts of the image is very low or almost non-existing (*e.g.*, two regions that have identical mean and differ only in the value of the variance of the texture intensity values).

In this paper, we provide implementation details for spline-based snakes that use different kinds of energies (region and contour-based) and for an efficient optimization strategy of the energy functionals. Finally, we illustrate the features of our software including an interactive interface where the user can intuitively manipulate the curve by changing the position of control points.

2. Parametric representation of closed curves

2.1. Parametric representation

A curve in the plane, denoted as $\mathbf{r}(t) = (x(t), y(t))$ with $t \in \mathbb{R}$ a continuous parameter, can be described by a pair of Cartesian coordinate functions x and y. These two one-dimensional functions are efficiently parameterized by linear combinations of basis functions. Among all possible bases, we focus on those derived from a compactly supported generator φ and its integer shifts $\{\varphi(\cdot - k)\}_{k\in\mathbb{Z}}$. This allows us to take advantage of the availability of fast and stable interpolation algorithms [4].

[☆] This work was funded by the Swiss SystemsX.chinitiative under Grant 2008/005 and the Swiss National ScienceFoundation under Grant 200020-121763.

E-mail addresses: ricard.delgado@epfl.ch (R. Delgado-Gonzalo), michael.unser@epfl.ch (M. Unser).

1959-0318/$ – see front matter © 2013 Elsevier Masson SAS. All rights reserved.
http://dx.doi.org/10.1016/j.irbm.2013.04.002

We are interested in closed curves that are specified by M control points. The parametric representation of the curve is then given by the vectorial equation

$$\mathbf{r}(t) = \sum_{k=-\infty}^{\infty} \mathbf{c}[k]\, \varphi(M\, t - k). \tag{1}$$

where $\mathbf{c}[k] = \mathbf{c}[k + M]$ is an M-periodic sequence of control points $\{\mathbf{c}[k] = (c_x[k], c_y[k])\}_{k\in\mathbb{Z}}$. We show in Fig. 1 a curve parameterized by a few control points, and its corresponding coordinate functions. The number of control points M determines the degrees of freedom in the model (1). Small numbers lead to constrained shapes, while large numbers provide more flexibility and the ability to approximate arbitrarily complex curves.

Since the curve $\mathbf{r}$ is closed, each coordinate function is periodic, and the period is common for both. For simplicity, in (1) we normalized this period to be unity. Under these conditions, we can reduce the infinite summation in (1) to a finite one involving periodized basis functions as

$$\mathbf{r}(t) = \sum_{k=0}^{M-1} \mathbf{c}[k]\, \varphi_M(M\, t - k), \tag{2}$$

where $\varphi_M(\,\cdot\,) = \sum_{n=-\infty}^{\infty} \varphi(\,\cdot\, - M\, n)$ is the M-periodization of the basis function φ.

Using this model, we approximate any closed curve as accurately as desired by using a higher number of vector coefficients. Conversely, we can simplify any shape by decreasing the number of vector coefficients. We show in Fig. 2 the resulting curve when sampling an epitrochoid with different M. The particular choice of φ determines the properties of the parametric model such as smoothness, reproduction of interesting shapes, or computational efficiency [5].

2.2. Desirable properties for the basis functions

We now enumerate the required geometric conditions that lead to a stable algorithm, and translate them to the corresponding mathematical formalism on the basis functions.

2.2.1. Unique and stable representation

We want the coordinate functions of our parametric curve to be uniquely defined in terms of the coefficients. Moreover, we also want the interpolation procedure to be numerically stable. This is achieved by requiring φ to satisfy the Riesz basis condition. That is, there must exist two constants $0 < A \leq B < \infty$ such that

$$A\, \|\mathbf{c}\|_{\ell_2([0\ldots M-1])} \leq \sqrt{M}\, \|\mathbf{r}\|_{L_2([0,1])} \leq B\, \|\mathbf{c}\|_{\ell_2([0\ldots M-1])}$$

for all $\mathbf{c} \in \ell_2([0\ldots M-1])$ [5].

2.2.2. 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, which we formalize as

$$\mathbf{A}\, \mathbf{r}(t) + \mathbf{b} = \sum_{k=-\infty}^{\infty} (\mathbf{A}\, \mathbf{c}[k] + \mathbf{b})\, \varphi(M\, t - k), \tag{3}$$

where $\mathbf{A}$ is a (2×2) matrix and $\mathbf{b}$ is a two-dimensional vector. From (3), it is easy to show that affine invariance is ensured if and only if

$$\sum_{k=-\infty}^{\infty} \varphi(M\, \cdot\, - k) = 1.$$

In the literature, this constraint is often named the *partition-of-unity* condition [4].

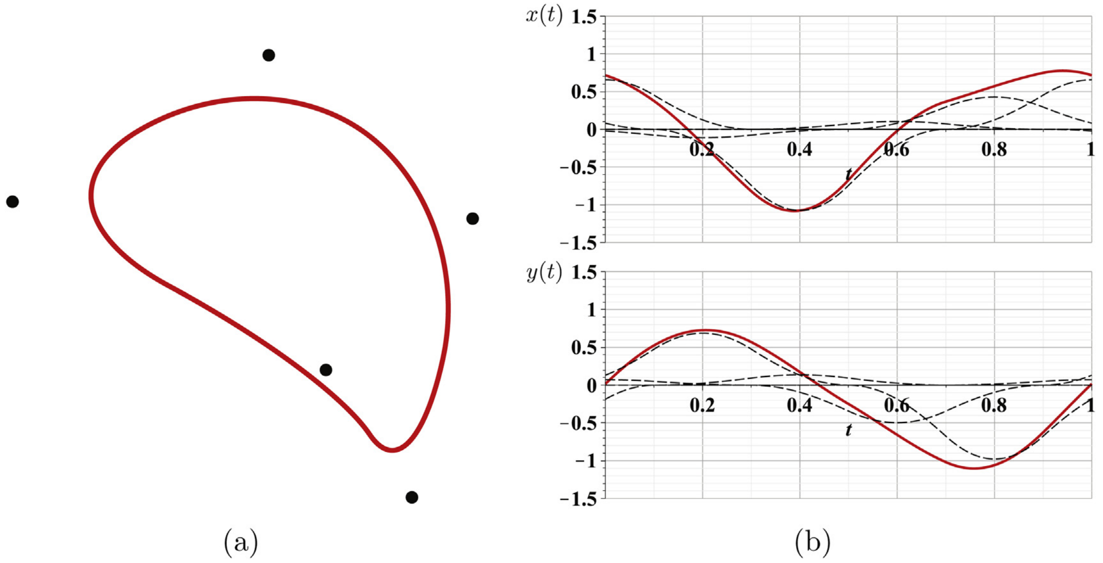


Fig. 1. Parametric representation of a closed curve with $M = 5$. The points in (a) correspond to the control points, and the dashed lines in (b) indicate the corresponding basis functions.

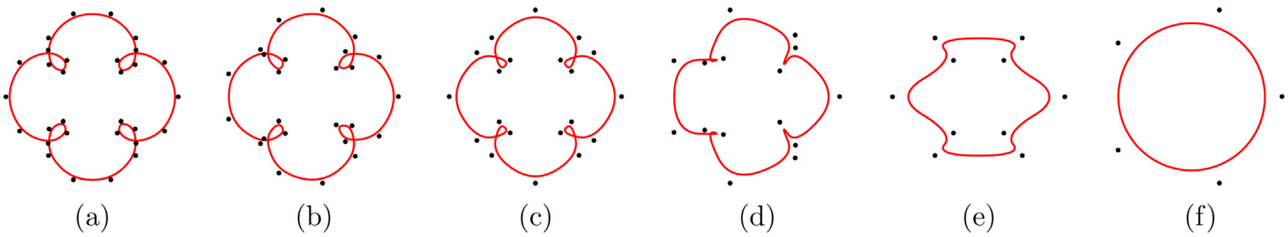


Fig. 2. Parametric curve obtained when sampling the parametric equations of an epitrochoid curve at different resolutions. (a) $M = 30$. (b) $M = 25$. (c) $M = 20$. (d) $M = 15$. (e) $M = 10$. (f) $M = 5$. Because an epitrochoid is made of a repetition of four structurally identical sections, the parametric curve may not capture well all symmetries when M is not evenly divisible by 4.

2.3. Choice of the basis function

The choice of the basis function φ under the constraints introduced in Section 2.2 is usually governed by two factors: computational complexity of the overall segmentation algorithm and capability of the snake to adopt specific shapes and to retain smoothness.

B-spline functions have traditionally been used as basis functions due to the existence of efficient interpolation algorithms [6], their finite support, and their good approximation properties [7]. The general expression of the B-spline of degree n is

$$\beta^n(t) = \frac{1}{n!}\sum_{k=0}^{n+1}\binom{n+1}{k}(-1)^k\left(t-k+\frac{n+1}{2}\right)_+^n,$$

where t_+^n is the causal side of the monomial t^n defined as

$$t_+^n = \begin{cases} t^n & t > 0 \\ 0 & t \leq 0. \end{cases}$$

The degree n of a B-spline is usually determined by the desired degree of smoothness for the proper representation of a particular class of objects. The length of the support of such basis functions is $W = n + 1$. It can be shown that the computational complexity of the snake energy, and the speed of the optimization algorithm is related to the size of the support of the basis functions [3]. Therefore, the computational load of the segmentation algorithm increases with the degree of smoothness.

A typical design constraint inherited from the time when snakes were built out of polygons (or, equivalently, linear B-splines) is to minimize the overall curvature. Using the well-known variational properties of splines [8], one can show that the minimization of the curvature subject to interpolation constraints yields a cubic-spline curve. Thus, the cubic B-spline β^3 appears to be the most natural choice for representing low-curvature and smooth parametric curves.

Traditional B-splines can only generate curves that can be expressed as piecewise polynomials. An important shape that is not included in this family is the ellipse. In medical imaging, it is usually necessary to segment arteries and veins within tomographic slices [9]. Because those objects are physiological tubes, their section shows up as ellipses in an image. Ellipse-like objects are also present at microscopic scales. For instance, cell nuclei are known to be nearly circular [10] and free-floating water drops are similarly spherical thanks to surface-tension forces [11]. However, these elements deform and become elliptical when they are subjected to stress forces. In [5] we characterized all minimal-support bases for the representation of closed geometric curves capable of reproducing exponential polynomials. In particular, the specific result for minimum-support basis functions that reproduce ellipses is summarized in the next theorem.

Theorem 1 (Delgado-Gonzalo et al. 2012). *The centered generating function with minimal support that satisfies all conditions in Section 2.2 and reproduces sinusoids of unit period with M coefficients is*

$$\varphi(t) = \frac{1}{2}\sum_{k=0}^{3}(-1)^k\, c_M[k]\, \phi_M(t+\frac{3}{2}-k), \tag{4}$$

where the coefficients are $c_M = [1, 1+2\cos\frac{2\pi}{M}, 1+2\cos\frac{2\pi}{M}, 1]$, *and where*

$$\phi_M(t) = \frac{1}{2}\begin{cases} \dfrac{\sin^2(\dfrac{\pi}{M}t)}{\sin^2\dfrac{\pi}{M}} & t > 0 \\ 0 & t \leq 0. \end{cases}$$

We show in Fig. 3 some members of this family for several values of M. We observe that they are continuous, with finite support of length $W = 3$, and bump-like. Every member of the family is shorter than the cubic B-spline, and therefore results in a more

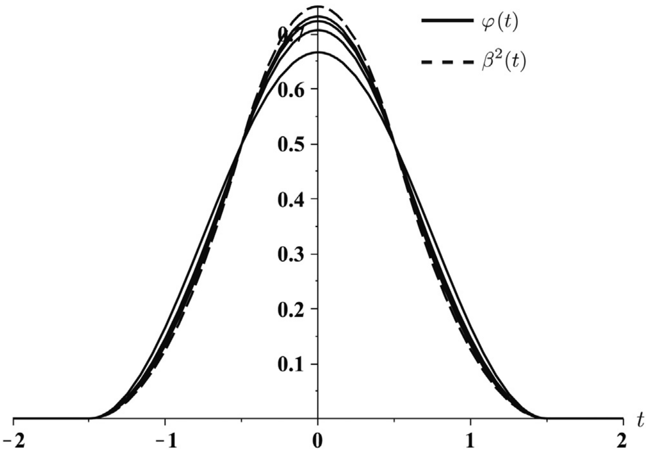


Fig. 3. Plot of a quadratic B-spline β^2 and of the generating functions in (4) for $M = 3$, 4, 5, and 6. The function with the lowest peak at $t = 0$ corresponds to $M = 3$, and, as M increases, the central peak increases as well.

efficient algorithm. Moreover, when $M \to \infty$, they converge to the quadratic B-spline. By a Maclaurin series expansion, we have that $\lim_{M\to\infty} \varsigma_M(t) = \frac{1}{2} t_+^2$. Then, $\lim_{M\to\infty} c_M = [1, 3, 3, 1]$ immediately implies that $\lim_{M\to\infty} \varphi = \beta^2$. This is because the quadratic B-spline can be written as

$$\beta^2(t) = \frac{1}{2}\left(\left(t+\frac{3}{2}\right)_+^2 - 3\left(t+\frac{1}{2}\right)_+^2 + 3\left(t-\frac{1}{2}\right)_+^2 - \left(t-\frac{3}{2}\right)_+^2\right).$$

Note that the convergence of φ to β^2 is pointwise. A piecewise expression of φ can be obtained by expanding (4) into

$$\varphi(t) = \frac{1}{1-\cos\frac{2\pi}{M}} \times \begin{cases} \cos\frac{2\pi\,|t|}{M}\cos\frac{\pi}{M} - \cos\frac{2\pi}{M} & 0 \le |t| < \frac{1}{2} \\ \left(\sin\frac{\pi\,(3/2-|t|)}{M}\right)^2 & \frac{1}{2} \le |t| < \frac{3}{2} \\ 0 & \frac{3}{2} \le |t|\,. \end{cases} \qquad (5)$$

The theory we present in the rest of the paper is general enough to accommodate for these as well as any other basis function φ satisfying the properties of Section 2.2.

3. Snake energies

We follow the standard paradigm introduced by Kass et al. [12] and formulate the snake evolution as an energy minimization. The snake energy is typically a linear combination of three terms:

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

The total energy of the snake is written as

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

The optimal spline parameters are obtained as

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

The energy-minimization process is nothing but an optimization procedure, where we iteratively update the snake representation so as to reach the minimum of the energy function from a starting position. Many methods exist to minimize the energy functional (gradient descent, PDEs, DP, among others).

The quality of segmentation is determined by the choice of the energy terms; it is generally agreed that specific image energies need to be defined for each particular imaging modality. In our model we obviated the constraint energy since we accommodate the user interaction as a hard constraint by leaving the desired coefficient out of the optimization routine.

3.1. Image energy

The image energy is usually application-dependent and is the most important of the three terms. It is usually designed with the purpose of finding particular features within the image. There are many strategies which can broadly be categorized in two families: 1) edge-based schemes, which use gradient information to detect contours [3,12–14] and 2) region-based methods, which use statistical information to distinguish different regions [3,15–17]. Most region energies are expressed as surface integrals of an integrand g over the domain $\mathcal{S}$ enclosed by the curve [3], like in

$$E_{\text{image}} = \int_{\mathcal{S}} g(\mathbf{x})\,\mathrm{d}\mathbf{x}.$$

An efficient way to implement the computation of the energy function is by using pre-integrated images [18]. By Green's theorem, we rewrite the surface integral as the line integrals

$$\begin{aligned}\int_{\mathcal{S}} g(\mathbf{x})\,\mathrm{d}\mathbf{x} &= -\oint_{\mathcal{C}} g^y(x,y)\,\mathrm{d}x \\ &= \oint_{\mathcal{C}} g^x(x,y)\,\mathrm{d}y,\end{aligned} \qquad (6)$$

where $\mathcal{C}$ is the boundary of $\mathcal{S}$, and

$$g^y(x,y) = \int_{-\infty}^{y} g(x,\xi)\,\mathrm{d}\xi \qquad (7)$$

$$g^x(x,y) = \int_{-\infty}^{x} g(\xi,y)\,\mathrm{d}\xi. \qquad (8)$$

The use of Green's theorem to rewrite the surface integrals as line integrals reduces dramatically the computational load. This can only be achieved if the curve is defined continuously.

Despite the fact that we are assuming a continuously defined model for our functions, in a real-world implementation we only have at our disposal a sampled version of the function g we want to pre-integrate. To solve this inconsistency, we store the values of (7) or (8) at integer locations in auxiliary image arrays. Then, the energies are obtained using a bilinear interpolation of the sampled values in the auxiliary images.

3.2. Internal energy

The internal energy is responsible for ensuring the smoothness of the contour. Our parametric schemes rely on the smoothness of the representation induced by the differentiability of φ [13–15]. Therefore, the design of an internal energy functional reduces to selecting a basis function that is at least twice differential in order to have a well-defined curvature.

3.3. Snake optimization

The optimization of the snake is carried out efficiently by a Powell-like line-search method [19]. The algorithm proceeds as follows: first, a promising direction within the parameter space is chosen depending on the partial derivatives of the energy. Second, a one-dimensional minimization is performed along the selected direction. Finally, a new direction is chosen using the partial derivatives of the energy function once more, while enforcing conjugation properties. This scheme is repeated until convergence. This method requires either the partial derivatives of the energy function with respect to the parameters or, in our case, the control points.

3.3.1. Partial derivatives of the image energy

We compute the partial derivatives of the image energy as

$$\begin{aligned}
&\frac{\partial E_{\text{image}}}{\partial c_x[k]}\\
&= \frac{\partial}{\partial c_x[k]} \oint_C g^x(x, y)\mathrm{d}y\\
&\overset{(a)}{=} \frac{\partial}{\partial c_x[k]} \int_0^1 g^{x(t)}(x(t), y(t))y'(t)\mathrm{d}t\\
&\overset{(b)}{=} \int_0^1 \underbrace{\frac{\partial}{\partial x}\{g^{x(t)}(x(t), y(t))\}}_{g(x(t),y(t))} \underbrace{\frac{\partial x(t)}{\partial c_x[k]}}_{\varphi_M(Mt-k)} y'(t)\mathrm{d}t\\
&\overset{(c)}{=} \int_0^1 g(x(t), y(t))\varphi_M(Mt-k)M\sum_{i=0}^{M-1} c_y[i]\varphi'_M(Mt-i)\mathrm{d}t\\
&\overset{(d)}{=} \sum_{i=0}^{M-1} c_y[i]M \underbrace{\int_0^1 g(x(t), y(t))\varphi_M(Mt-k)\varphi'_M(Mt-i)\mathrm{d}t}_{Q_g[k,i]},
\end{aligned}$$

where equality (a) is obtained by the definition of line integral, (b) is obtained thanks to the chain rule, (c) is obtained by (2), and (d) is obtained by reordering the terms. Thus, we obtain the simplified expression

$$\frac{\partial E_{\text{image}}}{\partial c_x[k]} = \sum_{i=0}^{M-1} c_y[i]\, Q_g[k, i].$$

In a similar manner, we get

$$\frac{\partial E_{\text{image}}}{\partial c_y[k]} = -\sum_{i=0}^{M-1} c_x[i]\, Q_g[k, i].$$

To summarize, the computation of the partial derivatives of the image energy reduces to the trivial computation of the finite sequence Q_g. Since φ and φ' are compactly supported, $Q_g[k, i]$ differ from zero if and only if $\varphi_M(Mt-k)$ and ${\varphi'}_M(Mt-i)$ overlap. Formally, $Q_g[k, i] \neq 0$ if $\min\{|k-i|, |i-k|\} < W$, where W is the common support length of φ and φ'. Then, if M is large compared to W or if the length of the support of the basis functions is short, then most of the elements of $Q_g[k, i]$ are zero.

3.3.2. Partial derivatives of the internal energy

Since we rely on the smoothness of the basis functions to induce the smoothness of the curve, we provide now explicit expressions for the basis functions discussed in Section 2.3.

- The derivative of the quadratic polynomial B-spline is computed via the recurrence relation

$$\beta^{2'}(t) = \beta^1\left(t + \frac{1}{2}\right) - \beta^1\left(t - \frac{1}{2}\right).$$

- The derivative of the minimum-support function that allows the curve to reproduce ellipses is written compactly as

$$\varphi'(t) = \frac{-\frac{2\pi}{M}\operatorname{sgn}(t)}{1 - \cos\frac{2\pi}{M}} \begin{cases} \sin\frac{2\pi\,|t|}{M}\cos\frac{\pi}{M} & 0 \le |t| < \frac{1}{2}\\ \frac{1}{2}\sin\frac{2\pi\,(3/2 - |t|)}{M} & \frac{1}{2} \le |t| < \frac{3}{2}\\ 0 & \frac{3}{2} \le |t|\,. \end{cases}$$

4. Software

Our segmentation technique has been programmed as a plug-in for ImageJ. This is a free, open-source, multi platform, image-processing software[1]. Our plug-in[2] is independent of any imaging hardware. Moreover, by virtue of ImageJ, any common file format may be used.

In our implementation, we use the exponential B-spline (5). This choice ensures that the snake can perfectly reproduce circular and elliptical shapes. This property is highly relevant for delineating cross sections of cylindrical-like conduits and outlining blob-like objects. Other than that, the snake is versatile enough to provide a good approximation of any closed curve in the plane. As shown in Fig. 4, the M control points (crosses) define a control polygon (brighter line) that specifies the continuous representation of the snake (darker line).

By contrast with many other implicit and global parametric snakes, only a few control points are required to specify the proposed snake, which eases the interactions with the user. As illustrated in Fig. 4, the user can intuitively manipulate the position of any point by selecting it. Editing the points is performed with simple mouse actions. The live update of the snake is very fast. This is due to the fact that changing one parameter only affects the structure locally. Thus, only a limited region of the snake curve has to be recomputed. As a result, a fast and intuitive semi-automatic segmentation procedure that alternates between snake initialization, optimization, and correction, is made possible.

We make use of a convex combination of contour and region energies for the image energy $E_{\text{image}} = \alpha\, E_{\text{contour}} + (1-\alpha)\, E_{\text{region}}$. The contour energy

[1] http://rsb.info.nih.gov/ij/

[2] http://bigwww.epfl.ch/algorithms/esnake/

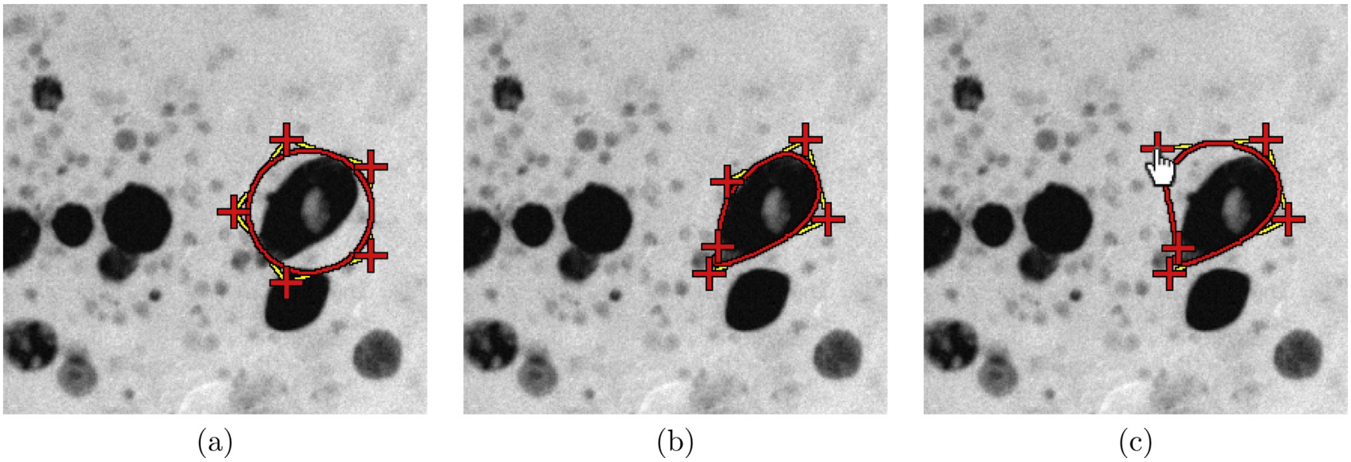


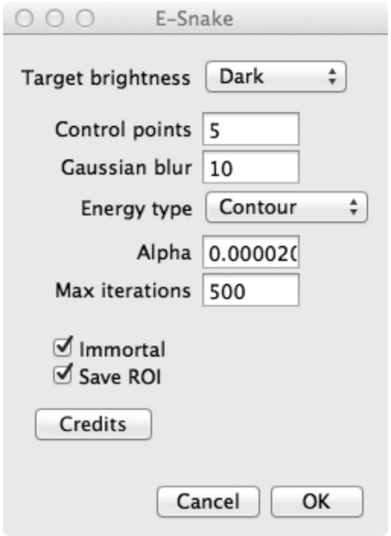
Fig. 4. Outlining of a cell with our snake using $M = 5$. (a) Initialization of the snake. (b) Snake after the minimization of the energy. (c) Interactive edition of a control point of the snake.

corresponds to the integral of the projection of the image gradient onto the curve [3]. This energy uses a version of the input image smoothed with a Gaussian kernel of standard deviation σ. The region energy maximizes the difference of intensity between the region enclosed by the curve and the intensity of a region within an elliptical shell [18].

4.1. Main interface

The plug-in requires at least one open image at launch. All image types are accepted, but the processing is performed on the luminance channel in the case of RGB Color images. In the case of a RGB Stack or a HSB Stack, the processing is performed on the current channel instead. In the case of 8-bit Color images, the underlying color indices are treated as intensity values.

At launch, the settings window appears (see Fig. 5). This window allows one to determine 1) the target to segment (bright or dark), 2) the number of control points M, 3) the standard deviation of the Gaussian blur σ, 4) energy type (contour, region, or a mixture of both), 5) the energy trade-off weight α, and 6) the maximum number of iterations of the snake optimizer. Moreover, two checkboxes determine if the snake is iterated until full convergence, and if the resulting outline is saved to ImageJ's ROI Manager.



Target brightness: Dark
Control points: 5
Gaussian blur: 10
Energy type: Contour
Alpha: 0.00002(
Max iterations: 500
☑ Immortal
☑ Save ROI

Fig. 5. Plug-in settings window.

The input parameters are:

- Target brightness: This option determines if the snake is attracted to dark or to bright targets.
- Control points: Number $M \geq 3$ of control points that define the control polygon.
- Gaussian blur: Standard deviation σ of the Gaussian kernel that is used to smooth the image when computing the contour energy.
- Energy type: This option determines if the snake evolves following the *contour energy*, the *region energy*, or a *mixture* of both.
- Alpha: Trade-off parameter that balances the contribution of the contour energy and the region energy when their mixture is used. The values of α are inside the interval [0, 1].
- Max iterations: Number of iterations that the optimizer will not exceed when evolving the snake.
- Immortal: If checked, the optimizer stops only when the snake has fully converged.
- Save ROI: If checked, the resulting outline is stored in ImageJ's ROI Manager.

If a ROI is present in the image at start, the plug-in uses it as initialization. Otherwise, the snake is initialized in a circular manner in the center of the image.

The interface of our plug-in is compatible with the macro language of ImageJ. Therefore, the command recorder can be used to produce macros for batch analyses.

4.2. User interaction

The snake can be manipulated thanks to a revised set of tools from ImageJ's toolbar, which we now discuss below (see Fig. 6). ImageJ's toolbar is restored as soon as the plug-in is terminated.

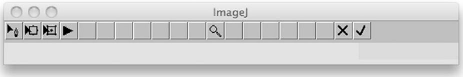


Fig. 6. Modified ImageJ's toolbar incorporating tools to interact with the snake. From left to right: Move crosses, Resize snake, Rotate snake, Optimize, Zooming tool, and Return to ImageJ with or without cancellation of the operation.

- Moving a point: To select a point, the user can click anywhere in the image. The point closest to the click is then highlighted. Points can be moved by dragging them within the image. The point closest to where the user clicks is moved to the position where the user releases the mouse. Alternatively, one can use the arrow keys of the keyboard to get a finer control over the position of the points. To translate the snake all at once, the user can use standard key modifiers while dragging the mouse.
- Resizing the curve: When this tool is selected, a bounding box appears on the image enclosing the control points of the snake. To move the edges of the bounding box, the user can drag them. The curve rescales accordingly to the new bounding box. To resize the snake all at once, the user can use standard key modifiers while dragging the mouse.
- Rotating the curve: When this tool is selected, a bounding box appears on the image enclosing the control points of the snake. To rotate the bounding box, the user can drag it. The curve is rotated accordingly to the rotated bounding box. The center of rotation is the center of the bounding box.
- Optimizing the snake: The 'play' button (a right-pointing black triangle) is used to start optimizing the snake.
- Image magnification/minification: To magnify an image, the user can select the magnification tool and click in the image. The minification can be done in the same way using standard key modifiers.
- Returning to ImageJ: There are two possible ways of leaving the plug-in and returning to ImageJ. The 'tick' button is used to return to ImageJ adding the ROI to the ROI Manager if 'Save ROI' is selected in the settings windows; the 'cross' button exits the plug-in without saving.

4.3. *Performance*

We present in this section two experimental setups to illustrate the performance of our plug-in. In the first one, we work with synthetic data and perform an objective validation of the segmentation properties of our snake. In the second experiment, we perform a quantitative evaluation by segmenting real cardiac MRI data.

4.3.1. *Simulated data*

To validate objectively the performance of our method, we used a phantom of size (2000 × 2000) pixels built out of known ellipses. Specifically, we wanted to recover Ellipse d of the widely used Shepp–Logan phantom [20].

We used the active contour model (2) for different values of M and several basis functions. Specifically, we used linear B-splines, quadratic B-splines, and our function (4) that we refer to as ellipse-reproducing spline. The linear B-spline basis function has the smallest support. However, the snake can only adopt the form of a polygon. The quadratic B-spline basis function has the same support and regularity as (4), but lacks the ellipse-reproducing property.

Table 1
Error percentage of our snake when segmenting synthetic data.

M	Linear B-spline β^1	Quadratic B-spline β^2	Ellipse-reproducing spline φ
3	59.42	20.23	0.17
4	35.59	5.32	0.20
5	23.99	0.26	0.07

We used the Jaccard distance J, expressed as a percentage, as objective measure of accuracy. We approximated J with a pixel-wise discretization of the images using the expression

$$J = 1 - \frac{|\Theta \cap \Omega|}{|\Theta \cup \Omega|},$$

where Θ corresponds to the ground-truth region (Ellipse d), and Ω is the region enclosed by the snake.

We provide in Table 1 a quantitative evaluation of this experiment, and illustrate the outcome in Fig. 7. We see that the Jaccard distance decreases monotonically for linear and quadratic B-splines as the value of M increases. This result is expected since, as claimed in Section 2.1, the parametric contour model (2) can approximate any closed curve as accurately as desired by using a higher number of vector coefficients. The use of the ellipse-reproducing spline as the basis function allows the snake to succeed ($J \approx 0.1\,\%$) at segmenting the ellipse for every M. This is a consequence the ellipse-reproducing property of the spline, which ensures that the snake can take the shape of any ellipse. Thus, an increase in the number of control points does not bring any further improvement when the shape to segment is a perfect ellipse.

4.3.2. *Medical data*

In this section, we quantify the accuracy of our snake at outlining the temporal evolution of the ascending aorta within a 2D cardiac MR image sequence.

The data we used are MRI measurements obtained in a 3T clinical scanner acquired at the Geneva University Hospitals (HUG), Switzerland[3]. The sequence was acquired using the retrospective technique with the systolic peak as trigger for synchronizing the MR measurements with the cardiac cycle [21]. The data consists of a time-series of 20 images with a temporal resolution of 55.5 ms. Each image is of size (256 × 176) with a pixel spacing of 1.25 mm. We show in Fig. 8(a)-(e) a cropped version of some frames of the dataset. Note how the aorta boundaries become noisy and blurred when the contrast is poor. The ground truth was obtained by manual annotation. In each segmented image, 300 points (named landmark points) define a closed polygon outlining the ascending aorta.

[3] http://www.hug-ge.ch/

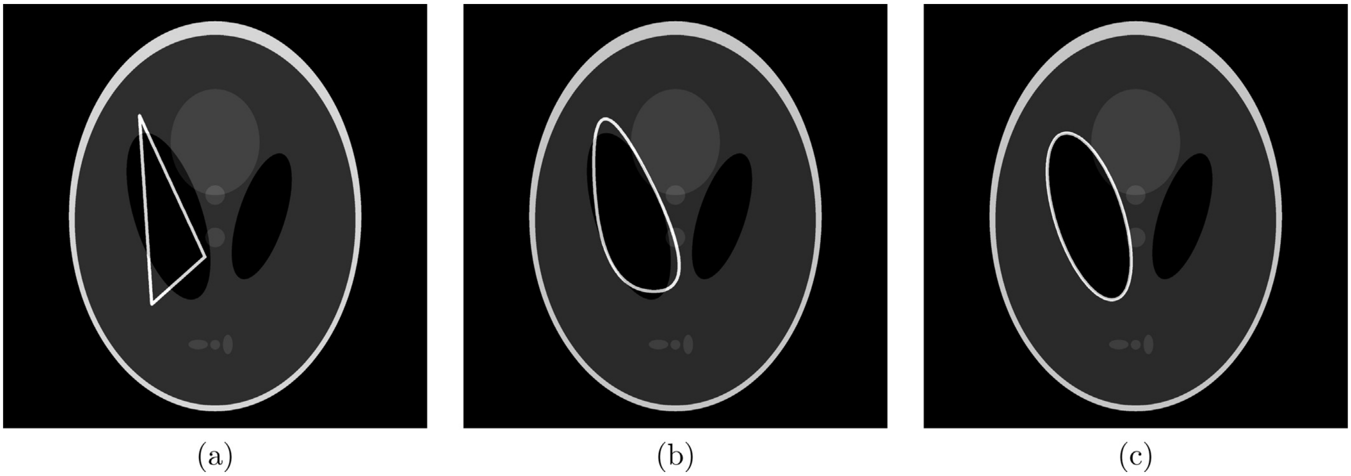


Fig. 7. Application of parametric snakes to synthetic data. (a) Polygonal snake with $M = 3$. (b) Quadratic-spline snake with $M = 3$. (c) Ellipse-reproducing-spline snake with $M = 3$.

We manually initialized the active contour in the first frame of the sequence, and we evolved the snake, guided exclusively by the edge energy on a smoothed version of the image. The smoothing was Gaussian, with a kernel of variance $\sigma^2 = 3$. We then measured the Jaccard distance and the landmark error. We computed the landmark error as the mean distance of the snake to the landmark points given by the ground truth. Then, we used the result of the optimization to initialize another snake in the subsequent frame. The process was repeated untill the end of the sequence.

We report in Table 2 several statistical figures for the Jaccard distance and the landmark error across the whole time sequence. The analysis was performed for several basis functions and several values of M. We observe that the polygonal snake does not provide an accurate segmentation for any value of M. However, the accuracy improved as the value of M increases. This is

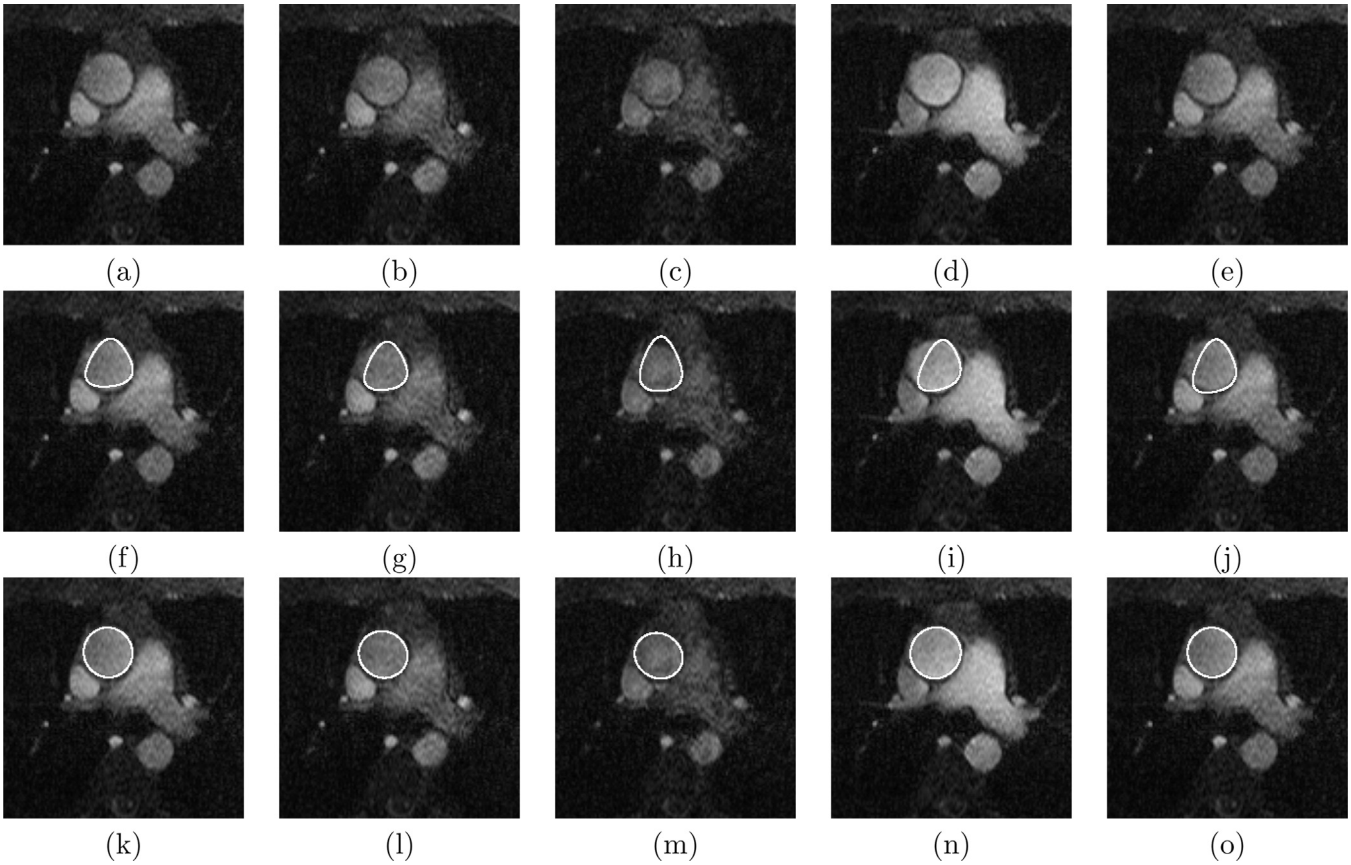


Fig. 8. Application of parametric snakes to the segmentation of MRI data. (a)–(e) Original data. (f)–(j) Quadratic-spline snake with $M = 3$. (k)–(o) Ellipse-reproducing-spline snake with $M = 3$.

Table 2
Accuracy statistics of our snake when segmenting real MRI data. The best mean accuracy is highlighted for each M.

	Linear B-spline β^1			Quadratic B-spline β^2			Ellipse-reproducing spline φ		
	$M=3$	$M=4$	$M=5$	$M=3$	$M=4$	$M=5$	$M=3$	$M=4$	$M=5$
Mean Jaccard (%)	58.55	33.13	20.46	21.62	4.49	2.84	**4.87**	**2.91**	**2.39**
Std. Jaccard (%)	6.89	9.68	5.39	2.20	1.59	1.69	4.28	1.53	1.66
Max. Jaccard (%)	66.77	64.64	29.16	24.41	7.49	8.98	22.46	7.73	7.70
Min. Jaccard (%)	40.21	22.86	10.48	17.78	1.66	1.58	2.69	1.16	0.53
Mean Landmark (mm)	6.74	3.54	1.79	1.79	0.37	0.24	**0.42**	**0.24**	**0.20**
Std. Landmark (mm)	1.39	1.11	0.47	0.21	0.12	0.13	0.40	0.12	0.13
Max. Landmark (mm)	10.38	7.68	2.57	2.06	0.59	0.72	2.07	0.59	0.63
Min. Landmark (mm)	4.34	2.34	0.98	1.50	0.16	0.15	0.21	0.11	0.08

expected from out active contour model, and supports the simulations in Section 4.3.1. The quadratic-spline snake and the ellipse-reproducing snake converge to similar accuracy starting with $M=5$. This was expected, since we showed in Section 2.3 that the ellipse-reproducing spline converges to the quadratic B-spline when M increases. However, for low values of M, the difference is noticeable, and the quadratic-spline snake produce shapes that are not compatible with the region of interest. Finally, we show in Fig. 8 the visual outcome of this experiment for some frames of the dataset.

5. Conclusion

We have described a flexible computational framework that is specifically adapted to spline-like parametric active contours that can be tuned to given applications. We have discussed the different design options and provided optimized solutions for various components. We have been able to accelerate the implementation of our snakes by taking advantage of Green's theorem, which was facilitated by the availability of the explicit expressions of our basis. We also provided explicit expressions of the partial derivatives of the energy functionals and described the user interface of our associated software.

References

[1] McInerney T, Terzopoulos D. Deformable models in medical image analysis: a survey. Medical Image Analysis 1996;1:91–108.
[2] Jain A, Zhong Y, Dubuisson-Jolly M-P. Deformable template models: a review. Signal Processing 1998;71:109–29.
[3] Jacob M, Blu T, Unser M. Efficient energies and algorithms for parametric snakes. IEEE Transactions on Image Processing 2004;13:1231–44.
[4] Unser M. Sampling–50 years after Shannon. Proceedings of the IEEE 2000;88:569–87.
[5] Delgado-Gonzalo R, Thévenaz P, Unser M. Exponential splines and minimal-support bases for curve representation. Computer Aided Geometric Design 2012;29:109–28.
[6] Unser M. Splines: a perfect fit for signal and image processing. IEEE Signal Processing Magazine 1999;16:22–38.
[7] Blu T, Unser M. Quantitative Fourier analysis of approximation techniques: part II–Wavelets. IEEE Transactions on Signal Processing 1999;47:2796–806.
[8] Schoenberg I. Spline functions and the problem of graduation. Proceedings of the National Academy of Sciences of the United States of America 1964;52:947–50.
[9] Adame I, van der Geest R, Wasserman B, Mohamed M, Reiber J, Lelieveldt B. Automatic segmentation and plaque characterization in atherosclerotic carotid artery MR images. Magnetic Resonance Materials in Physics, Biology and Medicine 2004;16:227–34.
[10] Alberts B, Johnson A, Lewis J, Raff M, Roberts K, Walter P. Molecular Biology of the Cell. Fifth edition. Garland Science; 2007.
[11] Stalder A, Kulik G, Sage D, Barbieri L, Hoffmann P. A snake-based approach to accurate determination of both contact points and contact angles. Colloids And Surfaces A: Physicochemical and Engineering Aspects 2006;286:92–103.
[12] Kass M, Witkin A, Terzopoulos D. Snakes active contour models. International Journal of Computer Vision 1987;1:321–31.
[13] Staib L, Duncan J. Boundary finding with parametrically deformable models. IEEE Transactions on Pattern Analysis and Machine Intelligence 1992;14:1061–75.
[14] Brigger P, Hoeg J, Unser M. B-Spline snakes: a flexible tool for parametric contour detection. IEEE Transactions on Image Processing 2000;9:1484–96.
[15] Figueiredo M, Leitão J, Jain A. Unsupervised contour representation and estimation using B-splines and a minimum description length criterion. IEEE Transactions on Image Processing 2000;9:1075–87.
[16] M Jacob, T Blu, M Unser. A unifying approach and interface for spline-based snakes. Proceedings of the SPIE International Symposium on Medical Imaging: Image Processing (MI'01). San Diego, CA, USA 2001;340–347.
[17] Thévenaz P, Delgado-Gonzalo R, Unser M. The ovuscule. IEEE Transactions on Pattern Analysis and Machine Intelligence 2011;33:382–93.
[18] Delgado-Gonzalo R, Thévenaz P, Seelamantula C, Unser M. Snakes with an ellipse-reproducing property. IEEE Transactions on Image Processing 2012;21:1258–71.
[19] Press W, Teukolsky S, Vetterling W, Flannery B. Numerical Recipes: The Art of Scientific Computing. Third edition Cambridge, UK: Cambridge University Press; 1986, 818 p.
[20] Shepp L, Logan B. The Fourier reconstruction of a head section. IEEE Transactions on Nuclear Science 1974;21:21–43.
[21] Lotz J, Meier C, Leppert A, Galanski M. Cardiovascular flow measurement with phase-contrast MR imaging: basic facts and implementation. RadioGraphics 2002;22:651–71.