# Efficient energies and algorithms for parametric snakes

Mathews Jacob, *Member, IEEE,*, Thierry Blu, *Member, IEEE,* and Michael Unser, *Fellow, IEEE.*

(e-mail:{mjacob@uiuc.edu, thierry.blu@epfl.ch, michael.unser@epfl.ch})

**Abstract**

Parametric active contour models are one of the preferred approaches for image segmentation because of their computational efficiency and simplicity. However, they have a few drawbacks which limit their performance. In this paper, we identify some of these problems and propose efficient solutions to get around them.

The widely-used gradient magnitude-based energy is parameter dependent; its use will negatively affect the parametrization of the curve and consequently its stiffness. Hence, we introduce a new edge-based energy that is independent of the parameterization. It is also more robust since it takes into account the gradient direction as well. We express this energy term as a surface integral, thus unifying it naturally with the region-based schemes. The unified framework enables the user to tune the image energy to the application at hand.

We show that parametric snakes can guarantee low curvature curves, but only if they are described in the curvilinear abscissa. Since normal curve evolution do not ensure constant arc-length, we propose a new internal energy term that will force this configuration.

The curve evolution can sometimes give rise to closed loops in the contour, which will adversely interfere with the optimization algorithm. We propose a curve evolution scheme that prevents this condition.

**Index Terms**

active contour, snake, segmentation, curve, spline

## I. Introduction

Snakes or active contour models have proven to be very effective tools for image segmentation. An active contour model is essentially a curve that evolves from an initial position towards the boundary of an object in such a way as to minimize some energy functional. The popularity of this semi-automatic approach may be attributed to its ability to aid the segmentation process with a-priori knowledge and user interaction.

Extensive research in this area have resulted in many snake variants [1], [2]; these are distinguished mainly by the type of curve representation used and the choice of the image energy term. The popular curve representation schemes in the snake literature are

- Point-based snakes, where the curve is an ordered collection of discrete points (also termed as snaxels) [3]–[5].

- Parametric snakes, where the curve is described continuously in a parametric form using basis functions such as B-splines [6]–[9], Fourier exponentials [10], [11] etc.
- Geometric snakes, where the planar curve is represented as a level set of an appropriate 2-D surface [12]–[16].

The point-based approach can be viewed as a special case of parametric curve representation where the basis functions are uniform translates of a B-spline of degree zero[1]; likewise, parametric approaches using smooth basis functions will tend to the point-based scheme as the number of basis functions increases. In general, however, representations using smooth basis functions require fewer parameters than point-based approaches and thus result in faster optimization algorithms [6], [10], [17]. Moreover, such curve models have inherent regularity and hence do not require extra constraints to ensure smoothness [9], [17].

Since both the above mentioned schemes represent the curve explicitly, it is easy to introduce a priori shape constraints into the snake framework [10], [18]–[20]. It is also straightforward to accommodate user interaction; this is often done by allowing the user to specify points through which the curve should go through [3]. However, these models offer less flexibility in accounting for topological changes during the curve evolution. One will have to perform some extra bookkeeping to accommodate changes in topology.

Geometric approaches offer great flexibility as far as the curve topology is considered; they presently constitute a very promising research area [14]–[16]. However, they tend to be computationally more complex since they evolve a surface rather than a curve. Also, since the curve representation is implicit, it is much more challenging to introduce shape priors into this framework [21].

In this paper, we focus on general parametric snakes due to its computational advantages and simplicity. We will start by taking a critical look at them, identifying some of their limitations, and propose some improvements to make them more attractive.

There are many different image energy terms that are used in practice. Most of the commonly used approaches fall into two broadly defined categories: (i) edge-based schemes which use local image information (typically gradient information) [3], [6], [9], [10], [17], [22], and, (ii)

[1]A B-spline of degree zero is defined as $\beta^0(x) = \begin{cases} 1 & \text{if } |x| < 0.5 \\ 0 & \text{else} \end{cases}$

region-based methods which uses global image features (eg. statistical formulation ) [8], [10], [11], [19], [23]–[27]. Since the best choice of the energy depends on the specific application at hand, we try to unify these approaches into a single framework; we obtain a general algorithm which can be tuned easily to the problem.

We propose a new edge energy term which is independent of the parametrization, unlike most of the commonly-used energies. The use of this energy will preserve the parametrization and consequently the curve stiffness. This energy is also more robust than the traditional gradient magnitude-based energy because it accounts for the direction of the gradient as well. We re-express this energy term as a region integral, thus unifying it with the region-based energies in a natural way. Thanks to the new approach, the choice of image energy is reduced to appropriately choosing the preprocessing.

We also clarify some earlier statements about splines by showing that parametric snakes can implicitly ensure smooth curves, but only if they are described in the curvilinear abscissa. Since general curve evolution approaches do not guarantee this configuration, we introduce a new internal energy term which forces the snake to the constant arc-length parametrization. We also propose efficient computational schemes for evaluating the partial differentials of the energy terms; thanks to the parametric curve representation in terms of finitely supported scaling functions, we can compute the differentials exactly and efficiently.

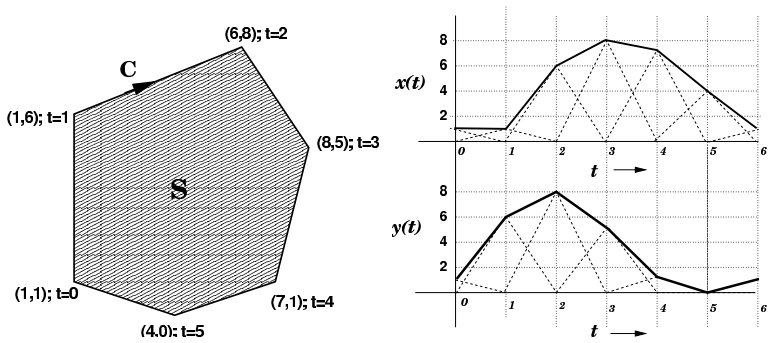
The paper is organized as follows. In the next section, we provide some mathematical preliminaries and formulate the parametric active contour problem. We deal with the image energy, internal energy and the external constraint energy in Sections 3, 4 and 5 respectively. In Section 6, we derive efficient expressions for the partial derivatives of the energy terms. In Section 7, we propose a practical solution for the detection and suppression of loops.

## II. Mathematical Preliminaries

### A. Parametric representation of closed curves

A curve in the $x$ — $y$ plane can be described in terms of an arbitrary parameter $t$ as $\mathbf{r}(t) = (x(t), y(t))$. When the curve is closed, the function vector $\mathbf{r}(t)$ is periodic.

We can represent $\mathbf{r}(t)$ efficiently as linear combinations of some basis functions. Here, we




(a) Polygon (b) Parametric representation

Fig. 1. Scaling function representation of a polygon. The dotted lines in (b) indicate the corresponding linear B-spline basis functions. Note that in this special case the knots are the vertices of the polygon themselves.

focus on bases derived from the integer shifts of a scaling function[2]; this type of representation includes most of the popular curve descriptors. Particular examples are band-limited, spline and polygonal schemes. The band-limited representation uses the sinc scaling function and is equivalent to the Fourier representation of a closed curve [29]. The scaling function representation of a curve is given by

$$\mathbf{r}(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \sum_{k=-\infty}^{\infty} \mathbf{c}_k \, \varphi(t-k), \tag{1}$$

where $\mathbf{c}_k = (c_{x,k}, c_{y,k})$ is the coefficient vector; they are often termed as knot points. We illustrate the parametric description of a polygon in terms of linear B-spline basis functions in Fig. 1.

If the period, $M$, is an integer, we have $\mathbf{c}_k = \mathbf{c}_{k+M}$. This reduces the infinite summation to

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

where $\varphi_p$ is the $M$ - periodization of $\varphi$:

$$\varphi_p(t) = \sum_{k=-\infty}^{\infty} \varphi(t - k\,M). \tag{3}$$

[2]Scaling functions are functions that satisfy a two-scale relation $\varphi\left(\frac{x}{2}\right) = \sum_k a(k)\,\varphi(x-k)$, where $a(k)$ is the two-scale mask [28].

### B. Active contour models: formulation

An active contour, as introduced by Kass et. al. [3], is a curve described as an ordered collection of points which evolves from its initial position to some boundary within the image. The curve evolution is formulated as an energy minimization; the snake energy is typically a linear combination of three terms:

1) the image energy, which is responsible for guiding the snake towards the boundary of interest.
2) the internal energy, which ensures that the segmented region has smooth boundaries.
3) the constraint energy, which provides a means for the user to interact with the snake.

The total energy of the snake is written as

$$E_{\text{snake}}(\mathbf{\Theta}) = E_{\text{image}}(\mathbf{\Theta}) + E_{\text{int}}(\mathbf{\Theta}) + E_{\text{c}}(\mathbf{\Theta}), \tag{4}$$

where $\mathbf{\Theta}$ is the collection of curve coefficients $\mathbf{\Theta} = (\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{M-1})$. The optimal curve parameters are obtained as

$$\overline{\mathbf{\Theta}} = \arg\min_{\mathbf{\Theta}} E_{\text{snake}}(\mathbf{\Theta}). \tag{5}$$

It is obvious that the quality of segmentation is dependent on the choice of the energy terms. We deal with them in detail in the following sections and they are listed in Table. I for easy reference. The energy is minimized iteratively by updating the snake coefficients.

## III. Image energy

The image energy is the most important of the three energy terms. In this section, we identify some limitations of the widely-used gradient magnitude energy and propose a new cost function that overcomes these problems. We also present a unified framework which includes the edge-based and region-based approaches as particular cases.

### A. Edge-based image energy

Traditional snakes rely on edge maps derived from the image to be guided to the actual contour. The most popular approach is based on the magnitude of the gradient.

*1) Gradient magnitude energy:* Many of the parametric snakes described in the literature use the integral of the square of the gradient magnitude along the curve as the image energy [6], [9], [10], [17]. Mathematically, we have

$$E_{\text{mag}} = -\int_0^M \left|\nabla f\left(t\right)\right|^2 dt, \tag{6}$$

where $\nabla f\left(t\right)$ denotes the gradient of $f$ at the point $\mathbf{r}\left(t\right)$. As pointed out in [22], one disadvantage of this measure is that it does not use the direction of the gradient. At the boundary, the image gradient is perpendicular to the contour. This extra information can be incorporated into the external energy to make it more robust.

A more fundamental problem is the dependence of $E_{\text{mag}}$ on the parametrization; we obtain a different value of $E_{\text{mag}}$ if the curve is represented in terms of a parameter $t' = w\left(t\right)$, where $w$ is a monotonically increasing one to one warping function. The use of such an energy may therefore result in the curve re-adjusting its parametrization in trying to minimize $E_{\text{mag}}$ (e.g. with B-spline curves, the knots will move to regions of the contour where the gradient magnitude is relatively high). This problem is demonstrated in Fig. 3-b.

*2) New gradient-based image energy:* The gradient magnitude energy is the integral of a scalar field derived from the gradient vector field. We propose a new energy that uses the vector field directly:

$$E_{\text{grad}} = -\oint_{\mathcal{C}} \mathbf{k} \cdot \left(\nabla f\left(\mathbf{r}\right) \times d\mathbf{r}\right) \tag{7}$$

$$= -\oint_{\mathcal{C}} \nabla f\left(\mathbf{r}\right) \cdot \underbrace{\left(d\mathbf{r} \times \mathbf{k}\right)}_{||d\mathbf{r}||\,\hat{\mathbf{n}}(\mathbf{r})}, \tag{8}$$

where $\mathbf{k}$ is the unit vector orthogonal to the image plane. Here $\hat{\mathbf{n}}\left(\mathbf{r}\right)$ denotes the unit normal to the curve at $\mathbf{r}$. Note that this approach of accounting for the gradient direction is similar in philosophy to [22], even though the expression used by these authors is different and parameter dependent.

This integration process is illustrated in Fig. 2; with our convention, the vector $\hat{\mathbf{n}}\left(\mathbf{r}\right)$ is the inward unit normal to the curve[3] meaning that we are integrating the component of the gradient orthogonal to the curve. Note that (7) is independent of the parameter $t$, and hence does not

[3]The vector $\mathbf{k}$ is chosen depending on the direction in which the curve is described, such that $\hat{\mathbf{n}}\left(\mathbf{r}\right) = \frac{d\mathbf{r}\times\mathbf{k}}{||d\mathbf{r}||}$ is the inward unit normal.

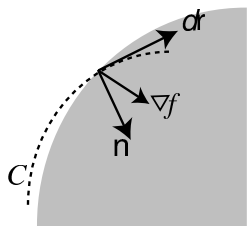



Fig. 2. Gradient and normal to the curve

depend on the parametrization. The improvement obtained by using the new energy instead of the parameter dependent magnitude-based energy is shown in Fig. 3-c.

*3) General edge-based image energy:* We consider a generalized form of (7) by substituting $\nabla f$ with other feature-enhancing vector fields. A promising approach is the use of optimal steerable filters to derive an appropriate edge enhancing vector field [30]. This method uses filters that are more directional than the $x$ and $y$ components of the conventional gradient operator to derive a noise-resilient field.

The general form of edge-based image energy can be expressed mathematically as

$$E_{\text{edge}} = -\oint_{\mathcal{C}} \mathbf{k} \cdot (\mathbf{e}_f(\mathbf{r}) \times d\mathbf{r}), \tag{9}$$

where $\mathbf{e}_f$ is an appropriate vector field derived from $f$. The magnitude of $\mathbf{e}_f(\mathbf{r})$ gives a measure of the edge strength at $\mathbf{r}$, while its direction specifies the edge orientation. We now show that the computation of this edge-based energy is equivalent to evaluating a region integral.

*Proposition 1:* The general edge-based image energy (9) can also be expressed as

$$E_{\text{edge}} = \int_{\mathcal{S}} \underbrace{\nabla \cdot \mathbf{e}_f(\mathbf{s})}_{T_e(f)} d\mathbf{s}, \tag{10}$$

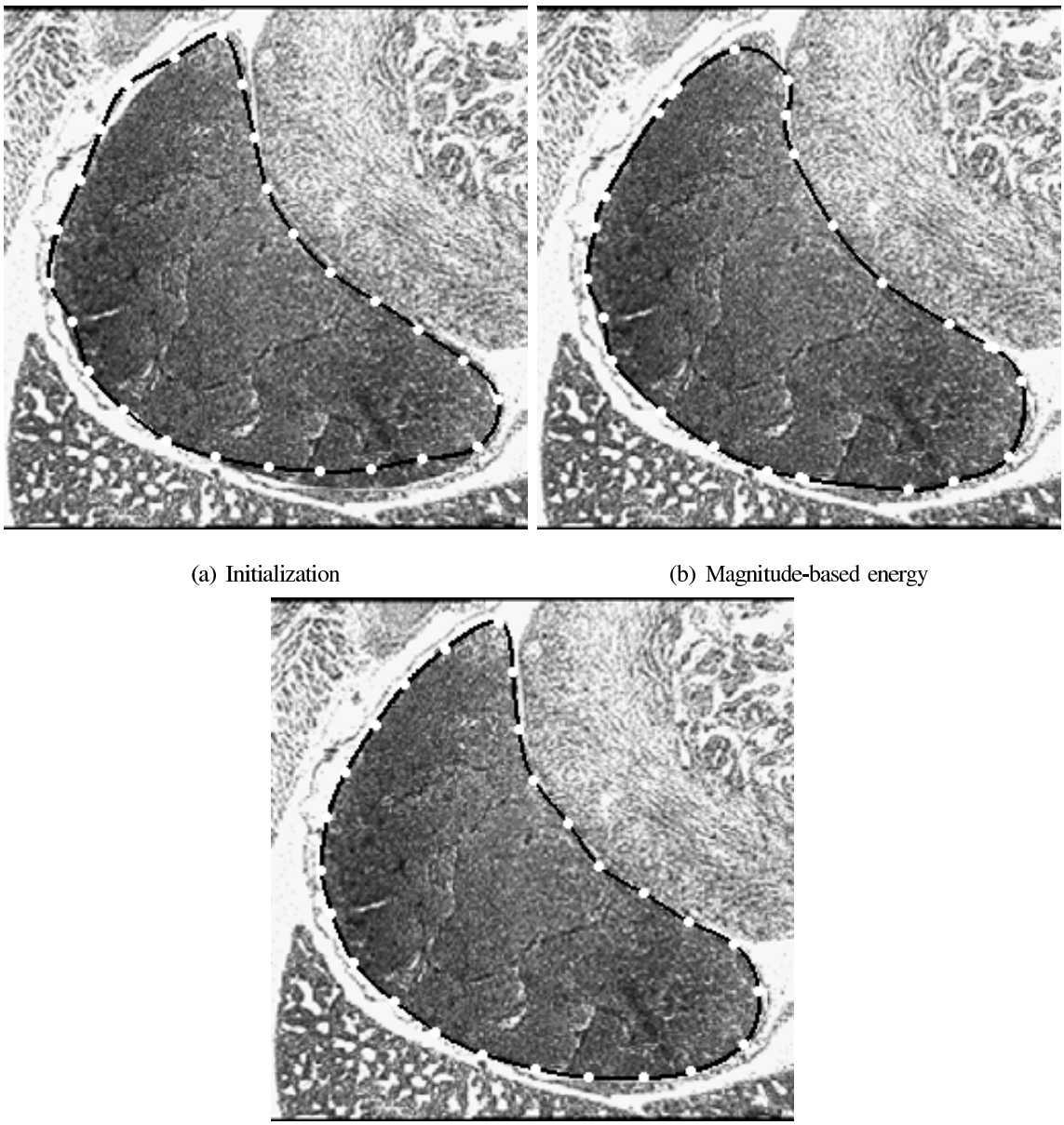
where $\nabla \cdot \mathbf{e}_f$ denotes the divergence of the vector field $\mathbf{e}_f$.

*Proof:* Green's theorem relates the volume integral of the divergence of a 3-D vector field $\mathbf{F}$ over a closed volume $\mathcal{V}$ bounded by the surface $\mathcal{S}$ to its integral over $\mathcal{S}$:

$$\int_{\mathcal{V}} (\nabla \cdot \mathbf{F})\, d\mathbf{v} = \int_{\mathcal{S}} \mathbf{F} \cdot d\mathbf{s}. \tag{11}$$

The restriction of Green's theorem to two dimensional space yields

$$\int_{\mathcal{S}} \left( \frac{\partial \mathbf{F}_x}{\partial x} + \frac{\partial \mathbf{F}_y}{\partial y} \right) dxdy = \oint_{\mathcal{C}} (\mathbf{F}_y dx - \mathbf{F}_x dy). \tag{12}$$



(a) Initialization (b) Magnitude-based energy

(c) New edge-based energy

Fig. 3. Segmentation of a mouse organ using edge-based energy (a) The knots (denoted by the white dots) are initialized so that the curve is approximately in the curvilinear abscissa. (b) Curve evolution based on the gradient magnitude-based energy. Note that the knots accumulate at some points along the curve in the final curve, thus restricting the flexibility of the curve. (c) Curve evolution based on our new edge-based energy; by better preserving the parametrization, it often result in a better segmentation.

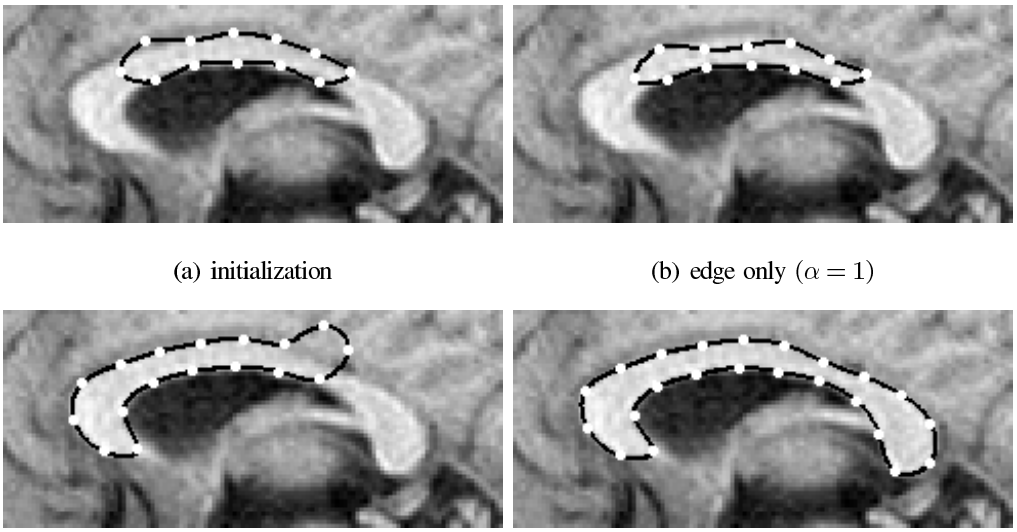
The integral on the left is computed over the area $\mathcal{S}$ bounded by the curve $\mathcal{C}$ while the one on

the right is over $\mathcal{C}$. Using the vector notation, we rewrite (12) as

$$\int_{\mathcal{S}} (\nabla \cdot \mathbf{F})\, d\mathbf{s} = - \oint_{\mathcal{C}} \mathbf{k} \cdot (\mathbf{F} \times d\mathbf{r})\,, \tag{13}$$

where $\mathbf{k}$ is the unit normal to the two dimensional space. Using this identity, we simplify (9) to the form (10). ■

Note that in the special case when $\mathbf{e}_f = \nabla f$, we get $T_e(f) = \nabla^2 f$. This means that our new gradient-based energy (7) is equivalent to integrating the Laplacian of the image in the region bounded by the curve.



(a) initialization (b) edge only ($\alpha = 1$)

(c) region only ($\alpha = 0$) (d) unified ($\alpha = 0.5$)

Fig. 4. Illustration of the use of the unified image energy in the segmentation of a corpus-callossum image (b) The use of the gradient based energy fails to converge in regions where the gradient information is absent (c) The region-based energy is misled by the lack of image contrast (d) The unified energy leads to a good segmentation.

### *B. Region-based image energy*

Recent research in active contours is increasingly focusing on the use of statistical region-based image energy [11], [19], [24], [25]. This type of energy can provide the snake with vital boundary information, especially while it is far away from the real contour, thus resulting in a larger basin of attraction.

The use of this energy assumes two main regions in image[4], with different probability distributions. A simple example is the case where we have to segment a white object from a dark background; the regions will have different means and possibly different variances. We use the statistical formulation of Staib et. al. [10] to specify the region likelihood function:

$$\begin{aligned} E_{\text{region}} &= -\int_{\mathcal{S}} \log\left(\mathrm{P}\left(f\left(\mathbf{s}\right)|\mathbf{s}\in\mathcal{R}\right)\right) d\mathbf{s} \\ &\quad -\int_{\mathcal{S}'} \log\left(\mathrm{P}\left(f\left(\mathbf{s}\right)|\mathbf{s}\in\mathcal{R}'\right)\right) d\mathbf{s}, \end{aligned} \tag{14}$$

where $\mathcal{R}$ and $\mathcal{R}'$ denote the different image regions. We denote the regions in the curve and outside by $\mathcal{S}$ and $\mathcal{S}'$ respectively. It is easy to see that (14) attains a maximum when $\mathcal{R}=\mathcal{S}$ and $\mathcal{R}'=\mathcal{S}'$. We rewrite the above integral as

$$\begin{aligned} E_{\text{region}} &= -\int_{\mathcal{S}} \log\left(\mathrm{P}\left(f\left(\mathbf{s}\right)|\mathbf{s}\in\mathcal{R}\right)\right) d\mathbf{s} \\ &\quad -C+\int_{\mathcal{S}} \log\left(\mathrm{P}\left(f\left(\mathbf{s}\right)|\mathbf{s}\in\mathcal{R}'\right)\right) d\mathbf{s}, \end{aligned} \tag{15}$$

where $C=\int_{\mathcal{S}'\bigcup\mathcal{S}} \log\left(\mathrm{P}\left(f\left(\mathbf{s}\right)|\mathbf{s}\in\mathcal{R}'\right)\right) d\mathbf{s}$. Since $C$ does not depend on the position of the curve and hence we remove it from the cost function. Thus, the region-based cost function is simplified to

$$E_{\text{region}}=\int_{\mathcal{S}} \underbrace{-\log\left(\frac{\mathrm{P}\left(f\left(\mathbf{s}\right)|\mathbf{s}\in\mathcal{R}\right)}{\mathrm{P}\left(f\left(\mathbf{s}\right)|\mathbf{s}\in\mathcal{R}'\right)}\right)}_{T_r(f)} d\mathbf{s}. \tag{16}$$

We now give a few examples to illustrate (16).

1) The regions $\mathcal{R}$ and $\mathcal{R}'$ have Gaussian distributions with the same variance. In this case, we obtain

$$T_r\left(f\right)=-\frac{2\left(\mu_{\mathcal{R}}-\mu_{\mathcal{R}'}\right)}{\sigma^2}\left(f-\underbrace{\frac{\mu_{\mathcal{R}}+\mu_{\mathcal{R}'}}{2}}_{\mu_{\mathcal{R},\mathcal{R}'}}\right), \tag{17}$$

where $\mu_{\mathcal{R}}>\mu_{\mathcal{R}'}$ are the means of the regions $\mathcal{R}$ and $\mathcal{R}'$ and and $\sigma$ the standard deviation. The regions of $f$ with values above $\mu_{\mathcal{R},\mathcal{R}'}$ are mapped to negative values while the ones below are assigned positive values. Hence, evolving the contour using (16) will result in the curve adjusting itself to have regions of $f$ above $\mu_{\mathcal{R},\mathcal{R}'}$ inside while excluding the ones

[4]This can be generalized to $n>2$ regions.

below $\mu_{\mathcal{R},\mathcal{R}'}$. The assumption of the variances of the regions being the same is appropriate if we have piecewise constant images corrupted by additive Gaussian noise.

2) The regions inside and outside the contour have Gaussian distributions with different variances. In this case, we obtain

$$T_r\left(f\right) = a\, f^2 + b\, f + c, \tag{18}$$

where $a = \left(\frac{1}{\sigma^2_{\mathcal{R}'}} - \frac{1}{\sigma^2_{\mathcal{R}}}\right)$, $b = -2\left(\frac{\mu_{\mathcal{R}'}}{\sigma^2_{\mathcal{R}'}} - \frac{\mu_{\mathcal{R}}}{\sigma^2_{\mathcal{R}}}\right)$ and $c = \left(\frac{\mu^2_{\mathcal{R}'}}{\sigma^2_{\mathcal{R}'}} - \frac{\mu^2_{\mathcal{R}}}{\sigma^2_{\mathcal{R}}}\right) + \log\left(\frac{\sigma_{\mathcal{R}'}}{\sigma_{\mathcal{R}}}\right)$. Here, $\sigma_{\mathcal{R}}$ and $\sigma_{\mathcal{R}'}$ are the standard deviations of the regions inside and outside the curve respectively. Since the snake uses the information from the variances as well, it can resolve the boundaries even when both regions have identical means but different variances [8].

In the absence of prior knowledge of the probability distributions $\mathrm{P}\left(f\left(\mathbf{s}\right)|\mathbf{s} \in \mathcal{R}\right)$ and $\mathrm{P}\left(f\left(\mathbf{s}\right)|\mathbf{s} \in \mathcal{R}'\right)$, the statistical parameters are estimated from the image $f$ themselves as the snake evolves; we assume the current position of the contour to define the regions (i.e. $\mathcal{S} = \mathcal{R}$ and $\mathcal{S}' = \mathcal{R}'$) and estimate the parameters as discussed in Section VI-C.

The extension of this method for the segmentation of multi-component images (e.g color images) is straightforward. For a $n$-D vector image $\mathbf{f} : \mathbb{R}^2 \rightarrow \mathbb{R}^n$, we have

$$E_{\text{region}} = \int_{\mathcal{S}} \underbrace{-\log\left(\frac{\mathrm{P}\left(\mathbf{f}\left(\mathbf{s}\right)|\mathbf{s} \in \mathcal{R}\right)}{\mathrm{P}\left(\mathbf{f}\left(\mathbf{s}\right)|\mathbf{s} \in \mathcal{R}'\right)}\right)}_{T_r(\mathbf{f})} d\mathbf{s}. \tag{19}$$

Note that the region information from the vector data is efficiently concatenated into the scalar image $T_r\left(\mathbf{f}\right)$. This framework is used for the segmentation of textures in [31]. They obtain an appropriate vector image from the gray level image using a Gabor filterbank.

### *C. Unified image energy*

Both of the above mentioned energies (edge-based and region-based) have their own advantages and disadvantages. The edge-based energy can give a good localization of the contour near the boundaries. Unfortunately, it has a small basin of attraction, thus requiring a good initialization or a balloon force [32]. On the other hand, the region-based energy have a large basin of attraction and can converge even if explicit edges are not present [25]. However, it does not give as good a localization as the edge-based energy at the image boundaries. Motivated by the complementary features of these schemes and the similarity of the expressions (10) and

(16), we propose a unified form of image energy. We choose a convex combination of the two energies to obtain an extended class, which inherits the advantages of both. The new image energy is given by

$$E_{\text{image}} = \int_{\mathcal{S}} \underbrace{T_{\text{u}}\left(f\right)}_{f_{\text{u}}}(\mathbf{s})\, d\mathbf{s}, \tag{20}$$

where $f_{\text{u}} = T_u\left(f\right) = \alpha T_e\left(f\right) + \left(1-\alpha\right) T_r\left(f\right)$. This unification is similar is philosophy to the approaches in [11], [15]. However, our scheme is more natural and yields a simpler expression since it combines the two energies into a single region integral. The simplicity of the expression will lead to computational advantages as will be discussed later on. Note that $f_{\text{u}}$ and hence $E_{\text{image}}$ is a function of the parameter $\alpha$. This enables us to tune the image energy to the problem at hand. For example, in the case of Fig. 7, the ultrasound images are very noisy and hence the gradient information is unreliable. In this case, we set $\alpha = 0$ to use only the region-based energy. In the case of less noisy MR images, the best results are obtained when $\alpha = 0.5$ (see Fig. 4)

## IV. Internal Energy

The internal energy is responsible for ensuring the smoothness of the contour. Kass proposed an internal energy which is the linear combination of the length of the contour and the integral of the square of the curvature along the contour. This smoothness term is the most widely used one in applications [1], [3], [5]. Its direct extension to parametric curves gives

$$\begin{aligned} E_{\text{int}} \;=\; \lambda_1 & \underbrace{\int_0^M \left(x'\left(t\right)^2 + y'\left(t\right)^2\right)^{\frac{1}{2}} dt}_{\text{Length}} \;+ \\ & \lambda_2 \int_0^M \underbrace{\left(\frac{x''\left(t\right) y'\left(t\right) - y''\left(t\right) x'\left(t\right)}{\left(x'\left(t\right)^2 + y'\left(t\right)^2\right)^{\frac{3}{2}}}\right)^2}_{|\kappa(\mathbf{r})|^2} dt, \end{aligned} \tag{21}$$

where $\kappa\left(\mathbf{r}\right)$ is the curvature of the curve at the point $\mathbf{r}\left(t\right)$. The first integral in (21) can be computed, while the second one is more complicated. We show in the Appendix-A that the

second term reduces to

$$\int_0^M |\kappa(\mathbf{r})|^2 \, dt \;=\; \frac{1}{c^2} \int_0^M \underbrace{\left( |x''(t)|^2 + |y''(t)|^2 \right)}_{|\mathbf{r}''(t)|^2} dt. \tag{22}$$

provided that

$$|x'(t)|^2 + |y'(t)|^2 = c; \quad \forall t. \tag{23}$$

that is, when the curve is parametrized by its curvilinear abscissa. Here

$$c = \frac{1}{M^2} \left( \underbrace{\int_0^M \left( x'(t)^2 + y'(t)^2 \right)^{\frac{1}{2}} dt}_{\text{Length}} \right)^2 . \tag{24}$$

It is justified to use $\oint_{\mathcal{C}} |\mathbf{r}''|^2$ as the curvature term in point-based snakes since the snake points (snaxels) are almost equally spaced. For parametric snakes described in the curvilinear abscissa, the curvature term is inversely proportional to the fourth power of the distance between the knots along the curve (c.f. (22) and (23)). We will have a smooth curve if its knots are well separated.

Most parametric schemes rely on the smoothness of the representation, thus eliminating the need for an explicit internal energy term [6], [9]–[11], [17]. However, these approaches can only ensure a low value of $\oint_{\mathcal{C}} |\mathbf{r}''|^2$; they can guarantee low curvature curves only when (23) hold. For example, a spline curve may be rough even with a small value of $\oint_{\mathcal{C}} |\mathbf{r}''|^2$ if some of the spline knots accumulate at one section of the curve. Similar problems exist with Fourier and other parametric representations. To counter this problem, we propose to add a new term to the criterion that will force the snake to satisfy (23).

### *A. Curvilinear reparametrization energy*

Our new energy term that penalizes the curve for not being in the curvilinear abscissa is given by

$$E_{\text{curv}} = \int_0^M \left| |\mathbf{r}'(t)|^2 - c \right|^2 dt, \tag{25}$$

where $c$ is given by (24). Evolving the curve with such a term will cause the curve knots to move tangential to the curve, thus bringing it to the curvilinear abscissa. An example of the type of improvement that can be obtained in this way is shown in Fig. 5.

Precioso et. al [33] proposed to reparametrize the curve to the constant arc-length representation after each step of the optimization algorithm to avoid the curves from looping. This scheme would yield the same results as our approach, but is computationally much more expensive.

### *B. Choice of the scaling basis function*

As mentioned before, the parametric representations can guarantee a small value of $\oint_{\mathcal{C}} |\mathbf{r}''|^2$. Using the well-known variational properties of splines [34], we can show that the minimization of $\oint_{\mathcal{C}} |\mathbf{r}''|^2$ subject to interpolation constraints yields a cubic spline curve with knots at the integers. Thus, the cubic B-spline model appears to be the most natural choice for representing parametric curves; it will yield a minimum curvature curve provided the parametrization is the curvilinear abscissa (i.e. the knots are uniformly spaced on the curve). The use of spline curves also brings in additional gains due to the existence of efficient algorithms [35], the local control of the contour due to the finite support of the B-spline basis function and their good approximation properties [36].

Due to these nice properties, we choose cubic spline curves in our implementation. However, the theory we present in this paper is general enough to accommodate for any other representation in terms of scaling function or wavelets.

### *C. New internal energy term*

In practice, the curve will almost be parametrized in the curvilinear abscissa after a few iterations with the internal energy term as $E_{\text{curv}}$. With this assumption, if we choose $c = \gamma \frac{\text{Length}^2}{M^2}$ in (25) instead of (24), we get

$$E_{\text{curv}}(\gamma) = E_{\text{curv}} + \frac{(1-\gamma)^2 \,\text{Length}^2}{M^2}. \tag{26}$$

This equation implies that we can also account for the $\text{Length}$ term in (21) by choosing $\gamma < 1$. We choose to perform this approximation since the partial derivatives of the $\text{Length}$ term cannot be computed exactly.We thus simplify the internal energy to

$$E_{\text{int}} = E_{\text{curv}}(\gamma). \tag{27}$$

In practice, we found it better not to minimize the length of the curve under normal circumstances; in other words, we usually set $\gamma = 1$. However, when the curve is detected to be looping, we decrease the length of the curve by choosing $\gamma = 0$. We discuss this issue in Section VII-B.

## V. External constraint energy.

As mentioned before, external constraint energy provides a means for the user to interact with the snake; he can guide the snake to the boundary when image information is too weak or ambiguous.

We introduce a point constraint mode, where the user has the option to specify a few points that should lie on the contour to be detected. We constrain the snake by adding an energy term which is the distance between these points and the corresponding closest points on the curve. The constraint energy is given by

$$E_{\mathrm{c}} = \sum_{i=0}^{N_{\mathrm{c}}-1} \min_{t\in[0,M]} \left|\mathbf{r}\left(t\right) - \mathbf{r}_{\mathrm{c},i}\right|^2, \tag{28}$$

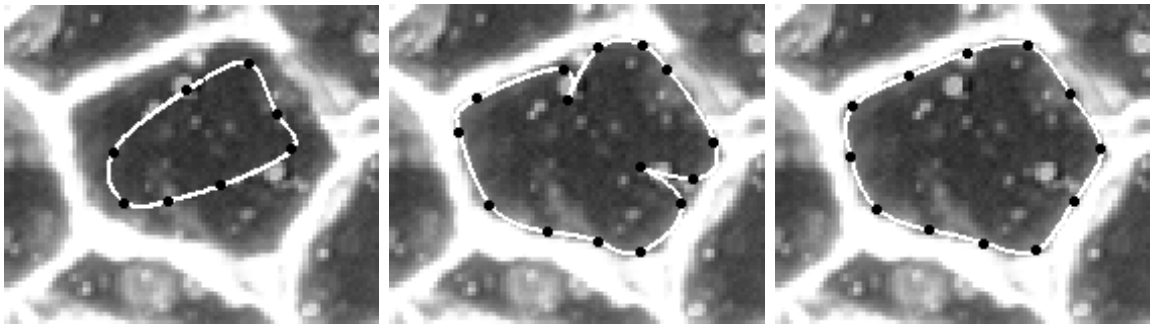
where $\mathbf{r}_{\mathrm{c},i}; i = 0..N_{\mathrm{c}} - 1$ are the constraints. This approach can be thought of as introducing virtual springs that pulls the curve towards the desired points: One end of the spring is fixed to the constraint point while the other end slides on the curve.

## VI. Evaluation of the partial derivatives

In this section, we express the partial derivatives of the component energies of the snake. These are used by the optimization algorithm to converge to the minimum of the energy function.

The theory mentioned so far is valid for general scaling function representations ranging from band-limited curves (Fourier series representation) to polygons. In order to derive efficient numerical schemes, we now make the additional assumption that the basis function is finitely supported in the interval $[0, N]$. Note that this class is still very rich as it includes most of the known scaling function families. The interesting cases for our purpose are the cubic B-spline function, which is finitely supported in the interval $[0, 4]$, and the linear B-spline function with the support $[0, 2]$.

*1) Partial derivatives of the magnitude-based image energy:* Following the work of Flickner et. al. [17], we locally optimize the snake during the initialization process (when the user is in the process of entering the initial curve), thus providing the user with a visual feedback. For this optimization we use the simple gradient magnitude-based energy mainly because it is applicable even when the curve is not yet closed and also because it is simple and computationally efficient. However, we only perform few iterations with this energy as it tends to bring the curve knots closer as mentioned before (c.f. Fig. 3).



(a) Initialization (b) No curvilinear energy (c) With curvilinear energy

Fig. 5. Without the curvilinear energy, the parametric representation cannot guarantee low curvature curves. Note that for the same initialization, the curve with the curvilinear reparametrization energy leads to smoother curves. Without the energy, the curve knots accumulate at some regions of the curve, thus leading to sharp edges; low energy curves are ensured only if the arc length is constant on the curve.

We consider the integral in (6) and differentiate it with respect to the coefficients using the chain rule (using (1)):

$$\begin{bmatrix} \partial E_{\text{mag}}/\partial c_{x,k} \\ \partial E_{\text{mag}}/\partial c_{y,k} \end{bmatrix} = \int_0^M \nabla g\,(t)\;\varphi_p\,(t-k)\,dt \tag{29}$$

where $g = |\nabla f|^2$. We approximate the inner-product as a discrete sum:

$$\begin{bmatrix} \partial E_{\text{mag}}/\partial c_{x,k} \\ \partial E_{\text{mag}}/\partial c_{y,k} \end{bmatrix} \approx \frac{1}{R} \sum_{i=0}^{NR} \nabla g \left( \frac{[k\,R+i]_{MR}}{R} \right) \varphi \left( \frac{i}{R} \right), \tag{30}$$

where $R$ is the sampling rate and $[k]_M$ stands for $k$ mod $M$. In the above expression, we used the finite support of the scaling function to limit the range of the summation. Also note that we have transferred the periodicity from the kernel to $\nabla g$; this means that the summation is evaluated assuming periodic boundary conditions on $\nabla g$. Thus, if $\nabla g$ and $\varphi\left(\frac{i}{R} - k\right)$ are precomputed, the evaluation of the partial derivatives just involves a weighted sum. The computational complexity is therefore proportional to $RMN$.

*2) Partial derivatives of the unified image energy:* For closed curves, we preferentially use the unified energy to optimize the curve. In line with the work of [10], [11], [37], we now use Green's theorem (12) to convert region integrals (over the region bounded by a closed curve) to integrals over the curve; our main motivation is computational efficiency. (20) can be efficiently

computed as the curve integral

$$\int_{\mathcal{S}} f_{\mathrm{u}}(x,y)\,dxdy = \oint_{\mathcal{C}} f_{\mathrm{u}}^{y}(x,y)\,dx \tag{31}$$

$$= -\oint_{\mathcal{C}} f_{\mathrm{u}}^{x}(x,y)\,dy, \tag{32}$$

where

$$f_{\mathrm{u}}^{y}(x,y) = \int_{-\infty}^{y} f_{\mathrm{u}}(x,\tau)\,d\tau \tag{33}$$

$$f_{\mathrm{u}}^{x}(x,y) = \int_{-\infty}^{x} f_{\mathrm{u}}(\tau,y)\,d\tau. \tag{34}$$

Applying the chain rule of differentiation on (32), we obtain $\partial E_{\mathrm{image}}/\partial c_{x,k}$ as

$$\begin{aligned}
\frac{\partial}{\partial c_{x,k}}(E_{\mathrm{image}}) &= \frac{\partial}{\partial x}(E_{\mathrm{image}}) \cdot \frac{\partial}{\partial c_{x,k}}(x(t)) \\
&= -\int_0^M \underbrace{\frac{\partial f_{\mathrm{u}}^{x}}{\partial x}}_{f_{\mathrm{u}}} \varphi_p(t-k) \underbrace{\sum_{l=0}^{M} c_{y,l}\,\varphi_p{}'(t-l)}_{y'(t)} \\
&= -\sum_{l=0}^{M} c_{y,l} \int_0^M f_{\mathrm{u}}(t)\,\varphi_p(t-k)\,\varphi_p{}'(t-l)\,dt \\
&= -\sum_{l=-\infty}^{\infty} c_{y,l}^{p} \underbrace{\int_{-\infty}^{\infty} f_{\mathrm{u}}(t)\,\varphi(t-k)\,\varphi'(t-l)\,dt}_{Q_{f_{\mathrm{u}}}(k,l)}.
\end{aligned} \tag{35}$$

In the last step we expanded $\varphi_p(t-k)$ using (3) and made a change of variable, thus extending the integral from $-\infty$ to $\infty$. We also transferred the periodicity of $Q_{f_{\mathrm{u}}}$ to the coefficient sequence. Since $Q_{f_{\mathrm{u}}}(k,l)$ is a finite sequence, the evaluation of (35) amounts to an appropriate finite sum. In a similar manner, using (31) we obtain

$$\frac{\partial E_{\mathrm{image}}}{\partial c_{y,k}} = \sum_{l=-\infty}^{\infty} c_{x,l}^{p} \underbrace{\int_{-\infty}^{\infty} f_{\mathrm{u}}(t)\,\varphi(t-k)\,\varphi'(t-l)\,dt}_{Q_{f_{\mathrm{u}}}(k,l)}.$$

The main steps in the computation of the partial derivatives are:

1) The evaluation of the sequence $Q_{f_{\mathrm{u}}}(k,l)\,;\ \ |k-l| < N$. (With a change of variables we obtain $Q_{f_{\mathrm{u}}}(k,l) = \int_{-\infty}^{\infty} f_{\mathrm{u}}(t+k)\,\varphi(t)\,\varphi'(t+k-l)\,dt$. Since $\varphi(t)$ is finitely supported in the interval $[0,N]$, $Q_{f_{\mathrm{u}}}(k,l)$ is zero if $|k-l| \geq N$). Approximating the integral as a

discrete sum, we obtain

$$Q_{f_{\mathrm{u}}}(k,l) = \frac{1}{R}\sum_{i=0}^{NR} f_{\mathrm{u}}\left(\frac{[kR+i]_M}{R}\right) \underbrace{\varphi\left(\frac{i}{R}\right)\varphi'\left(\frac{i}{R}+k-l\right)}_{b_{k-l}(i)}. \tag{36}$$

Provided we precompute[5] the sequence $b_m(i)\,; m = \{-N+1 \ldots N-1\}$, the computation of $Q_{f_{\mathrm{u}}}(k,l)\,; 0 < k,l < M$ involves an weighted sum of length $Ns$.

2) The evaluation of the partial derivatives, which are obtained as

$$\begin{bmatrix} \partial E_{\mathrm{image}}/\partial c_{x,k} \\ \partial E_{\mathrm{image}}/\partial c_{y,k} \end{bmatrix} = \sum_{l=0}^{M-1} \begin{bmatrix} -c^p_{y,l} \\ c^p_{x,l} \end{bmatrix} Q_{f_{\mathrm{u}}}(k,l)\,. \tag{37}$$

Here, the computational complexity is of the order of $RM^2N^2$. Note that there is a factor of 2 advantage in implementing the partial derivatives of $E_{\mathrm{grad}}$ as in (37) rather than its direct evaluation from (7). The performance improvement in the implementation of the unified energy is even better as compared to the one in [11], where the energy is the sum of two integrals.

### A. *Partial derivatives of the internal energy*

Differentiating the expression of $E_{\mathrm{int}} = E_{\mathrm{curv}}$ and simplifying further, we obtain the partial derivatives as a simple multidimensional filtering of the scaling function coefficients. We show in the appendix that the partial derivatives of the the term $E_{\mathrm{int}}$ can be computed as

$$\begin{aligned} \frac{\partial}{\partial c_{x,k}}(E_{\mathrm{int}}) &= \sum_{|l|,|m|,|n|<N} c^p_{x,k-l}\, c^p_{x,k-m}\, c^p_{x,k-n}\, h_1(l,m,n) + \\ &\quad \sum_{|l|,|m|,|n|<N} c^p_{x,k-l}\, c^p_{y,k-m}\, c^p_{y,k-n}\, h_1(l,m,n) - \\ &\quad 4c \sum_{|l|<N} c^p_{x,k-l}\, h_2(l) \\ \frac{\partial}{\partial c_{y,k}}(E_{\mathrm{int}}) &= \sum_{|l|,|m|,|n|<N} c^p_{y,k-l}\, c^p_{y,k-m}\, c^p_{y,k-n}\, h_1(l,m,n) + \\ &\quad \sum_{|l|,|m|,|n|<N} c^p_{y,k-l}\, c^p_{x,k-m}\, c^p_{x,k-n}\, h_1(l,m,n) - \\ &\quad 4c \sum_{|l|<N} c^p_{y,k-l}\, h_2(l)\,, \end{aligned} \tag{38}$$

[5]The samples of $\varphi$ can be computed by solving for its values at the integers as shown in [28] and using the two-scale relation to refine it .

where

$$h_1(l,m,n) = \int_{-\infty}^{\infty} \varphi'(t)\,\varphi'(t+l)\,\varphi'(t+m)\,\varphi'(t+n)\,dt \quad (39)$$

$$h_2(l) = \int_{-\infty}^{\infty} \varphi'(t)\,\varphi'(t+l)\,dt. \quad (40)$$

Note that the multidimensional filtering is performed assuming periodic boundary conditions. The computational complexity is small, since the sum depends only on the coefficient sequence whose number is typically much lesser than the number of curve samples. The computational complexity in evaluating the above sum is $N^3M$. The filter coefficients (39) and (40) are precomputed as shown in Appendix-C.

### *B. Partial derivatives of the constraint energy*

Computing the partial derivatives of (28), in all its generality, would give a very complicated expression. To make the problem more tractable and to reduce its computational complexity, we make the assumption that the optimal parameters $t_i;\ i = 0 \ldots N_{\mathrm{c}} - 1$ are known. In this case, (28) gets simplified to

$$E_{\mathrm{c}} = \sum_{i=0}^{N_{\mathrm{c}}-1} \left|\mathbf{r}(t_i) - \mathbf{r}_{\mathrm{c},i}\right|^2 \quad (41)$$

and its partial derivatives are given by:

$$\begin{bmatrix} \partial E_{\mathrm{c}}/\partial c_{x,k} \\ \partial E_{\mathrm{c}}/\partial c_{y,k} \end{bmatrix} = \sum_{i=0}^{N_{\mathrm{c}}-1} \left( \begin{bmatrix} x_{\mathrm{c},i} \\ y_{\mathrm{c},i} \end{bmatrix} - \begin{bmatrix} x(t_i) \\ y(t_i) \end{bmatrix} \right) \varphi(t_i - k). \quad (42)$$

Using the finite support of the scaling functions, we limit the sum to the relevant indices (we need to evaluate it only for $\{i|\ 0 < (t_i - k) < N_{\mathrm{c}}\}$). In practice, we resort to a two-step optimization where the snake is first evolved using the above formulas for the derivatives with the current set of $t_i$'s. The optimal parameters $t_i$ are then re-estimated within the loop as:

$$t_i = \arg \min_{t \in [0,M]} \left|\mathbf{r}(t) - \mathbf{r}_{\mathrm{c},i}\right|;\ \ i = 0 \ldots N_{\mathrm{c}} - 1 \quad (43)$$

### *C. Estimation of the probability distribution functions*

The evaluation of (14) requires the specification of the probability distribution functions $P(f(\mathbf{s})|\mathbf{s} \in \mathcal{R})$ and $P(f(\mathbf{s})|\mathbf{s} \in \mathcal{R})$. If we do not have any a priori knowledge of these distributions, these are estimated iteratively from the image data itself assuming $\mathcal{R} = \mathcal{S}$ and

$\mathcal{R}' = \mathcal{S}'$. Note that these assumptions are valid if the snake is close to the real boundary. We use densities such as the Gaussian distribution which are represented by a few parameters (mean and variance). The estimation of these parameters require integrating the image and its square in the region bounded by $\mathcal{S}$. We compute the integrals efficiently using (31) with the corresponding integrated functions (similar to (33)) precomputed.

The estimation of the distributions are followed by a non-linear transformation which maps $f$ into $f_{\mathbf{u}}$. Since this transformation is time consuming, the estimation of the distributions and the updating of $f_{\mathbf{u}}$ is only performed periodically, typically once every 10 iterations.

### *D. Computation of the length and area*

The computation of the internal energy requires the estimation of the current length of the curve. We compute the length as a discrete approximation of the integral $\int_0^M \left(x'(t)^2 + y'(t)^2\right)^{\frac{1}{2}} dt$ as

$$\text{Length} = \frac{1}{R} \sum_{i=0}^{MR-1} \sqrt{x'\left(\frac{i}{R}\right)^2 + y'\left(\frac{i}{R}\right)^2}. \tag{44}$$

The area of the curve is obtained by Green's theorem as $\oint_{\mathcal{C}} y dx$, which when expanded gives

$$\text{Area} = \sum_{k=0}^{M-1} \sum_{l=-N+1}^{N-1} c_{y,k}\, c^p_{x,l}\, q(k-l), \tag{45}$$

where $q(m) = \int_{-\infty}^{\infty} \varphi(t)\, \varphi'(t-m)\, dt$ is obtained as in [38]. Note that the area obtained by the above expression is signed; its sign is utilized to determine the direction (clockwise or anti-clockwise) of the curve.

## VII. Evolving the curve

### *A. Optimization Algorithm*

As mentioned before, the active contour algorithm extracts the final contour by finding the minimum of the energy function. Having obtained the partial derivatives, we can use an efficient optimization algorithm to evolve the contour. Here, we implemented the conjugate gradient and steepest descend algorithms. The conjugate gradient algorithm resulted in slightly faster convergence, but was less flexible for loop recovery and knot addition/deletion discussed later. Hence in our final implementation we reverted to the simpler steepest descend algorithm, which was found to be entirely satisfactory for our purpose.

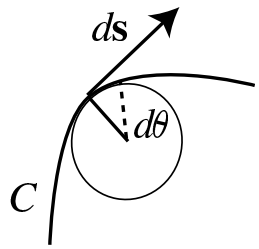



Fig. 6. Computation of the elemental angle

### *B. Loop detection and recovery*

The optimization process can sometimes lead to looping curves. The probability of loops is greatly reduced by the introduction of the curvilinear reparameterization energy; without this term, the knots tend to bunch together, eventually resulting in loops (c.f. Fig. 5).

Despite the use of the new internal energy, looping may still arise occasionally[6] when the image energy forces some knots to move faster than the others. This compromises our approach since we use Green's theorem which assumes simply connected regions. In the case of polygonal representation (linear spline curve), Chesnaud et. al. proposed to perform crossing tests to detect the presence of loops [24]. Unfortunately, this method is time consuming and not directly applicable to general scaling function curves. Hence, we devised a fast method for loop detection. We compute the total tangential angle[7]:

$$\theta_{\text{total}} = \int_0^M d\theta\,(t)\,dt, \tag{46}$$

where

$$d\theta\,(t) = \frac{x'\,(t)\,y''\,(t) - y'\,(t)\,x''\,(t)}{x'\,(t)^2 + y'\,(t)^2}. \tag{47}$$

We show in the Appendix-D that the value of the integral (46) is $2\,(n-m)\,\pi$, where $m$ and $n$ are the number of loops in the clockwise and anti-clockwise direction respectively. Hence, for a simply connected curve, we expect $\pm 2\pi$ (depending on the direction in which the curve is described). We approximate (46) by a discrete sum over the parameter $t$.

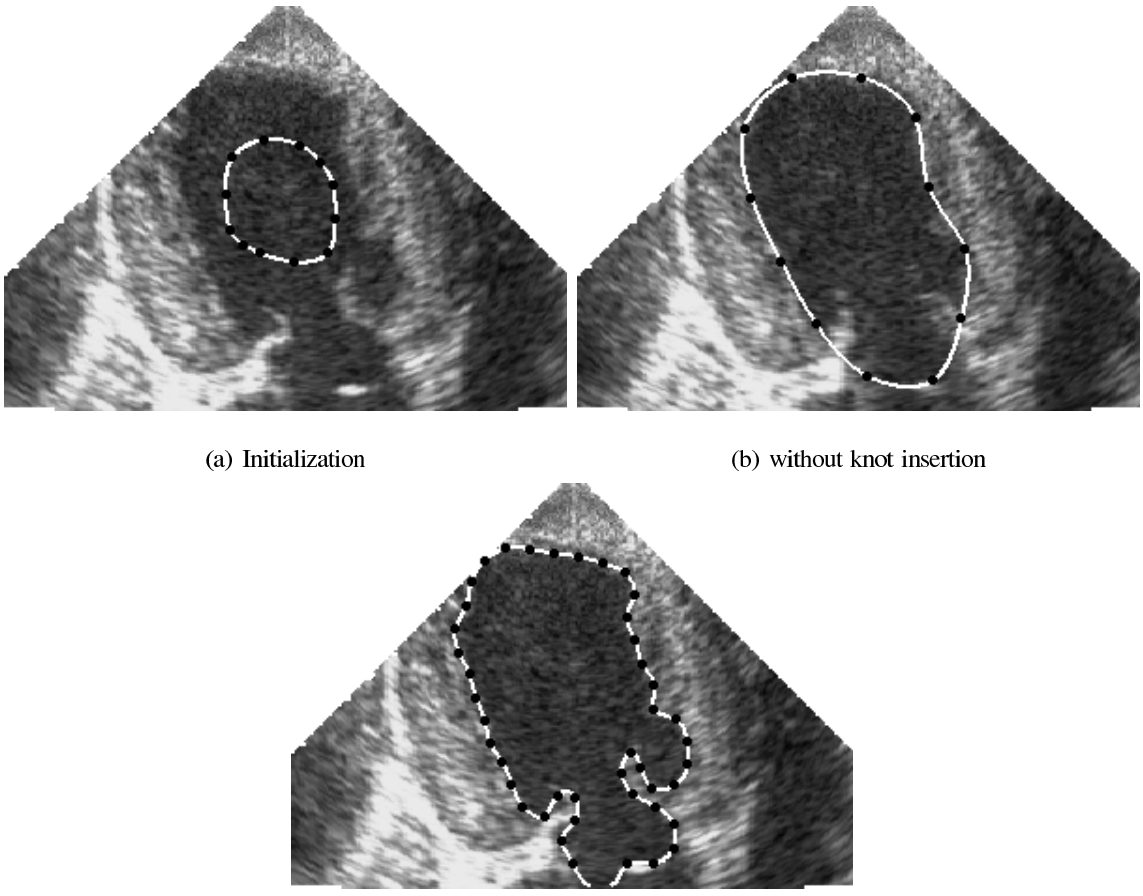
Note that our criterion can give a value $2\pi$ even if the curve is looping (when $n+1=m$), which implies that it is not completely foolproof. In principle, it is possible to detect these cases

[6]In our experiments, looping arise in about 10% of the cases.

[7]For a plane curve, the tangential angle $\theta$ is defined by $d\theta = \kappa\,|d\mathbf{r}|$ [39].

by splitting the integral (46) over a series of smaller intervals and checking if there is a loop 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.

Once we detect a loop, we evolve the curve with only the $E_{\text{int}}$ term with $\gamma = 0$, thus decreasing its length. In practice, the curvature of the curve at the loops are high. Since minimizing the length corresponds to evolving the curve at every point depending on its curvature [14], the loops tend to disappear very rapidly. This scheme may result in regions of the contour with high curvature getting smoothed out during the loop-elimination process. However, in most cases the image energy guides the contour back to the edges once the loops are eliminated.



(a) Initialization (b) without knot insertion

(c) with knot insertion

Fig. 7. Segmentation of the inner wall of the heart of a dog from its ultrasound image. Only the region-based energy is used in this case ($\alpha = 0$). Note that knot (knots are denoted by black dots) insertion and loopcheck is indispensable in this case. Time taken for (c): 3.2s on a 667MHz Macintosh G4.

### C. Shrinking/growing snakes

If the snake is initialized away from the actual boundary, it has to shrink/grow to reach the boundary. This changes the average spacing of the knots, which in turn controls the average curvature of the curve (cf. (22)).

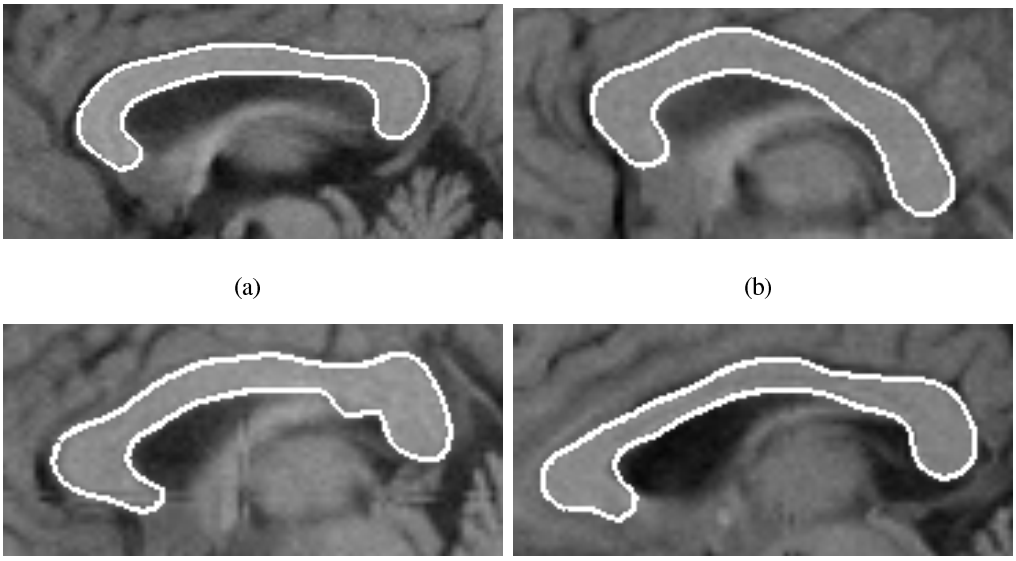
We monitor the length of the curve as it evolves in order to eventually add/delete knots as required. If the average length per knot is greater than the desired value (set by the user), a knot is added to the curve. The distances between the knots are evaluated and a knot is added at the longest interval. Similarly, a knot is deleted if the length per knot is less than the user set value. In this case, the knot whose sum of the distances to its neighbors is the shortest is deleted. The addition/deletion of knots temporarily destroys the uniform spacing of knots. But, thanks to the reparametrization energy term, it returns to the curvilinear abscissa in a few iterations (without the reparametrization energy, knot insertion is a tricky issue as close knots may eventually lead to looping curves). The performance improvement in adopting this strategy is illustrated in Fig. 7.

## VIII. Discussion and Conclusion

We have successfully applied the snake algorithm to a variety of cases including the segmentation of corpus-callossum from MR images and segmentation of the inner heart wall from ultrasound data. Some examples of the segmented corpus-callossum images are shown in Fig. 8. Thanks to the unified image energy, the snake gives a good segmentation even if it is not initialized very close to the actual boundary. This approach also makes the algorithm less sensitive to the initial shape of the snake.

The curvilinear reparametrization energy ensures that the curves are smooth. Without this term, the segmentation of the heart data (see from Fig 7) is impossible; the curves often resulted in loops. The knot insertion/deletion procedure ensures that the evolving curve has the same stiffness as the initialization.

To conclude, we have presented several enhancements over classical parametric snakes. We have identified some limitations of the conventional gradient magnitude image energy and proposed a new energy that eliminates these problems. We have shown that a general form of this energy can be expressed as a region integral, thus unifying it naturally with the region-based approaches. The unification yields a powerful class of image energies that combines the



(a) (b)

(c) (d)

Fig. 8. Segmentation of corpus-callossum of 4 different subjects from their MR images. The initialization was a small curve at the center similar to Fig. 4. We gave equal weight to the region and gradient terms ($\alpha = 0.5$). Average time taken: 1.9 s on a 667MHz Macintosh G4.

advantages of edge and region-based approaches. We have shown that the spline representation can guarantee smooth curves if these are described in the curvilinear abscissa. Since the curve evolution process can negatively affect the reparametrisation of the curve, we proposed a new internal energy which forces the knot points to remain equally spaced. The various energy terms that we have proposed are summarized in Table I.

The evolution of the curve may lead to looping curves that violate our assumption of the region to be simply connected. Hence, we introduced a simple loop detection test. We also proposed an efficient curve evolution-based algorithm for recovery from the loops. We introduced efficient computational schemes for the evaluation of the partial differentials used in the optimization; we converted all the quantities as curve integrals and simplified the expressions making use of the properties of scaling function curve representation.

The implementation of this algorithm is available as a java plugin for ImageJ [40] at http://bigwww.epfl.ch/jaco

## ACKNOWLEDGEMENTS

This work was supported by the Swiss National Science Foundation under grant 2100-053540.

| Energy Type | General Expression | Special Cases |
|---|---|---|
| Image Energy | $\int_S T_{\mathrm{u}}(f)\,d\mathrm{s}$ | Gradient-based energy: Eq. (7)<br>General edge-based energy: Eq. (9)<br>$(\alpha = 1)$ |
| | | Region-based energy: Eq. (16)<br>$(\alpha = 0)$ |
| | | Unified energy: Eq. (20)<br>$(0 \leq \alpha \leq 1)$ |
| Internal Energy | $\int_0^M \left\|\left\|\mathbf{r}'(t)\right\|^2 - \gamma\,\frac{\mathrm{Length}}{M}\right\|^2$ | Curvilinear reparametrizaton energy: Eq. (25)<br>$(\gamma = 1)$ |
| | | Length energy: Eq. (26)<br>$(\gamma = 0)$ |
| Constraint Energy | $\sum_{i=0}^{N_{\mathrm{c}}-1} \min_{t\in[0,M]} \left\|\mathbf{r}(t) - \mathbf{r}_{\mathrm{c,i}}\right\|$ | Point constraint: Eq. (28) |

TABLE I

DIFFERENT ENERGY TERMS USED IN THE SNAKE OPTIMIZATION

## APPENDIX - A: SIMPLIFICATION OF THE CURVATURE TERM IN THE INTERNAL ENERGY

The square of the curvature of the curve at a point $\mathbf{r}(t)$ can be expressed in the vector form as

$$\left|\kappa(\mathbf{r})\right|^2 = \frac{(\mathbf{r}' \times \mathbf{r}'') \cdot (\mathbf{r}' \times \mathbf{r}'')}{\left|\mathbf{r}'\right|^6} \tag{48}$$

Assuming the parameter $t$ to be in the curvilinear abscissa, we have $|\mathbf{r}'(t)| = c,\ \forall t$. Making use of the vector identity $\mathbf{a}.(\mathbf{b} \times \mathbf{c}) = \mathbf{c}.(\mathbf{a} \times \mathbf{b})$, the numerator of (48) can be rewritten as

$$\begin{aligned}
(\mathbf{r}' \times \mathbf{r}'') \cdot (\mathbf{r}' \times \mathbf{r}'') &= \mathbf{r}'' \cdot (\mathbf{r}' \times \mathbf{r}'' \times \mathbf{r}') \\
&= \mathbf{r}'' \cdot (\mathbf{r}''(\mathbf{r}' \cdot \mathbf{r}') - \mathbf{r}'(\mathbf{r}' \cdot \mathbf{r}'')) \\
&= |\mathbf{r}''|^2 |\mathbf{r}'|^2 - |\underbrace{\mathbf{r}'' \cdot \mathbf{r}'}_{d(\mathbf{r}'^2)=0}|^2.
\end{aligned}$$

In the second step, we make use of the identity $\mathbf{a}\times\mathbf{b}\times\mathbf{c}=(\mathbf{a}\cdot\mathbf{c})\mathbf{b}-(\mathbf{b}\cdot\mathbf{c})\mathbf{a}$. So the expression for the curvature can be written as

$$|\kappa(\mathbf{r})|^2=\frac{|\mathbf{r}''|^2}{|\mathbf{r}'|^4}=\frac{|\mathbf{r}''|^2}{c^2}. \tag{49}$$

## APPENDIX - B: PARTIAL DERIVATIVES OF THE CURVILINEAR REPARAMETRIZATION TERM

Expanding (25) we obtain

$$\begin{aligned} E_{\text{curv}} &= \int_0^M \left(x'(t)^4+y'(t)^4+2x'(t)^2y'(t)^2\right)dt+ \\ &\quad c^2M-2c\int_0^M\left(x'(t)^2+y'(t)^2\right)dt. \end{aligned} \tag{50}$$

Differentiating $E_{\text{curv}}$ with respect to $c_{x,k}$, we get

$$\begin{aligned} \frac{\partial E_{\text{curv}}}{\partial c_{x,k}} &= \int_0^M\left(4x'(t)^3+4x'(t)\,y'(t)^2\right)\frac{\partial}{\partial c_{x,k}}\left(x'(t)\right)dt \\ &\quad -4c\int_0^M x'(t)\,\frac{\partial}{\partial c_{x,k}}\left(x'(t)\right)dt. \end{aligned} \tag{51}$$

Now substituting for $x(t)$ and $y(t)$ from (1), yields

$$\begin{aligned} \frac{\partial E_{\text{curv}}}{\partial c_{x,k}} &= \sum_{l,m,n\in\mathbb{Z}} c_{x,l}^p\;c_{x,m}^p\;c_{x,n}^p\;h_1(k-l,k-m,k-n)+ \\ &\quad \sum_{l,m,n\in\mathbb{Z}} c_{x,l}^p\;c_{y,m}^p\;c_{y,n}^p\;h_1(k-l,k-m,k-n)- \\ &\quad 4c\sum_{l,m,n\in\mathbb{Z}} c_{x,l}^p\;h_2(k-l)\,. \end{aligned} \tag{52}$$

The filters $h_1$ and $h_2$ are given by

$$h_1(l,m,n) = \int_{-\infty}^{\infty}\varphi'(t)\,\varphi'(t+l)\,\varphi'(t+m)\,\varphi'(t+n)\,dt \tag{53}$$

$$h_2(l) = \int_{-\infty}^{\infty}\varphi'(t)\,\varphi'(t+l)\,dt. \tag{54}$$

With a change of variables and using the finite support of $h_1$ and $h_2$, we can simplify (52) to (38).

## APPENDIX - C: PRECOMPUTATION OF THE KERNEL

We use the property that the derivative of a scaling function $\varphi$ can be written as $\varphi'(t) = \varphi^{\{1\}}(t) - \varphi^{\{1\}}(t-1)$, where $\varphi^{\{1\}}$ is the scaling function whose mask (scaling filter) is $A^{\{1\}}(z) = \left(\frac{2}{1+z^{-1}}\right) A(z)$; $A(z) \leftrightarrow a_k$ is the mask of $\varphi$. Using this relation, we rewrite the filter coefficients (39) and (40) as

$$h_1(l,m,n) = -\triangle^f \triangle_1^b \triangle_2^b \triangle_3^b g_1(l,m,n) \tag{55}$$

$$h_2(l) = -\triangle^f \triangle_1^b g_2(l), \tag{56}$$

where

$$\triangle_i^b g(l_1,..,l_i,..,l_n) = g(l_1,..,l_i,..,l_n) - g(l_1,..,l_i-1,..,l_n)$$

$$\triangle^f g(l_1,l_2..,l_n) = g(l_1+1,l_2+1,..,l_n+1) - g(l_1,l_2,..,l_n)$$

and

$$g_1(l,m,n) = \int_{-\infty}^{\infty} \varphi^{\{1\}}(t)\,\varphi^{\{1\}}(t+l)\,\varphi^{\{1\}}(t+m)\,\varphi^{\{1\}}(t+n)\,dt \tag{57}$$

$$g_2(l) = \int_{-\infty}^{\infty} \varphi^{\{1\}}(t)\,\varphi^{\{1\}}(t+l)\,dt. \tag{58}$$

The scaling function $\varphi^{\{1\}}$ satisfies the twoscale relation

$$\varphi^{\{1\}}(t) = \sum_{k=0}^{N} a_k^{\{1\}}\,\varphi^{\{1\}}(2t-k). \tag{59}$$

Consequently, the kernels $g_1$ and $g_2$ satisfy the two-scale relations

$$g_1(\mathbf{k}) = \sum_{\mathbf{l}} h_1(\mathbf{l})\,g_1(2\mathbf{k}-\mathbf{l}) \tag{60}$$

$$g_2(k) = \sum_{\mathbf{l}} h_2(l)\,g_2(2k-l), \tag{61}$$

where the two-scale masks $H_1$ and $H_2$ are given by

$$h_1(l,m,n) = \frac{1}{2}\left(\sum_k a^{\{1\}}(k)\,a^{\{1\}}(k-l)\,a^{\{1\}}(k-m)\,a^{\{1\}}(k-n)\right) \tag{62}$$

$$h_2(l) = \frac{1}{2}\left(\sum_k a^{\{1\}}(k)\,a^{\{1\}}(k-l)\right). \tag{63}$$

Using the two-scale relation, the sequences $g_1$ and $g_2$ are exactly computed as in [38].

## APPENDIX - D: INTEGRAL OF THE TANGENTIAL ANGLE.

We start by observing that the integral (46) can be expressed as

$$\theta_{\text{total}} = \text{Im}\left(\int_0^M \frac{d\left(x'\left(t\right) + \mathbf{j}\, y'\left(t\right)\right)}{x'\left(t\right) + \mathbf{j} y'\left(t\right)} dt\right), \tag{64}$$

where $\text{Im}\left(z\right)$ gives the imaginary part of $z$ and $\mathbf{j} = \sqrt{-1}$. This can be rewritten as the curve integral

$$\theta_{\text{total}} = \text{Im}\left(\oint_{\mathcal{C}'} \frac{dz}{z}\right), \tag{65}$$

where $\mathcal{C}'$ is the curve described $\left(x'\left(t\right), y'\left(t\right)\right)$ and $z = x' + iy'$. Using Cauchy's integral formula, we obtain the value of this integral as $2\pi$ times the winding number[8] of the contour $\mathcal{C}'$ about the origin. Since each loop in $\mathcal{C}$ corresponds to one in $\mathcal{C}'$ in the same direction, but around the origin, the winding number of $\mathcal{C}'$ is $(m - n)$, where $m$ and $n$ are the number of times $\mathcal{C}$ loops in the anticlockwise and clockwise direction respectively.

## REFERENCES


[1] A.K. Jain, Y. Zhong, and M.P.D. Jolly, "Deformable template models: A review," *Signal Processing*, vol. 76, pp. 109–129, 1998.

[2] T. McInerney and D. Terzopoulos, "Deformable models in medical image analysis: a survey," *Medical Image Analysis*, vol. 1, pp. 91–108, 1996.

[3] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, pp. 321–332, 1988.

[4] C. Xu and J. L. Prince, "Snakes, shapes and gradient vector flow," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 359–369, 1998.

[5] J. Gao, A. Kosaka, and A. Kak, "A deformable model for human organ extraction," in *ICIP*, Chicago, 1998, pp. 323–327.

[6] S. Menet, P. Saint-Mark, and G. Medioni, "B-snakes: implementation and application to stereo," in *Image Understanding workshop*, 1990, pp. 720–726.

[7] M. Gebhard, J. Mattes, and R. Eils, "An active contour model for segmentation based on cubic B-splines and gradient vector flow," in *MICCAI*, 2001.

[8] M. A. Figueiredo and J. M. N. Leitao, "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, 2000.

[9] 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, 2000.

[10] L. H. Staib and J. S. Duncan, "Boundary fitting with parametrically deformable models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1061–1075, 1992.


[8]The winding number of a contour about a point $z_0$ is the number of times the contour passes around $z_0$ in the counterclockwise direction [39].

[11] A. Chakraborthy, L. H. Staib, and J. S. Duncan, "Deformable boundary finding in medical images by integrating gradient and region information," *IEEE Transactions on Medical Imaging*, vol. 15, no. 6, pp. 859–870, 1996.

[12] V. Casselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," in *ICCV*, 1995.

[13] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 2, pp. 158–174, 1995.

[14] G. Sapiro, *Geometric partial differential equations and image analysis*, Cambridge University Press, 2001.

[15] N. Paragios and R. Deriche, "Unifying boundary and region-based information for geodesic active tracking," in *IEEE Computer Vision and Pattern Recognition*, Forth Collins, Colarado, 1999.

[16] O. Amadieu, E. Debreuve, M. Barlaud, and G. Aubert, "Simeltaneous inward and outward curve evolution," in *ICIP*, Kobe, Japan, 1999.

[17] M. Flickner, H. Sawhney, D. Pryor, and J. Lotspiech, "Intelligent interactive image outlining using spline snakes," in *28th Asilomar Conference on Signals, Systems and Computers*, 1994.

[18] T.F. Cootes, A. Hill, C.J. Taylor, and J. Haslam, "Use of active shape models for locating structures in medical images," *Image and Vision computing*, vol. 12, pp. 355–365, 1994.

[19] M. Jacob, T. Blu, and M. Unser, "A unifying approach and interface for spline-based snakes," *SPIE International Symposium on Medical Imaging: Image Processing*, vol. 4322, pp. 340–347, 2001.

[20] C. Davatzikos, X. Tao, and D. Shen, "Hierarchical active shape models, using the wavelet transform," *IEEE Transactions on Medical Imaging*, 2003.

[21] M. E. Leventon, W. E. L. Grimson, and O. Faugeras, "Statistical shape influence in geodesic active contours," in *Computer Vision and Pattern Recognition*, 2000.

[22] H. Park, T. Schoepflin, and Y. Kim, "Active contour model with gradient directional information: Directional snake," *IEEE Transactions on Circuits and systems for video technology*, vol. 11, no. 2, pp. 252–256, 2001.

[23] M. Figueiredo and J. Leitao, "Bayesian estimation of ventricular contours in angiographic images," *IEEE Transactions on Medical Imaging*, vol. 11, no. 3, pp. 416–429, 1992.

[24] C. Chesnaud, P. Refregier, and V. Boulet, "Statistical region snake-based segmentation adapted to different physical noise models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1145–1157, 1999.

[25] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266276, 2001.

[26] S. C. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing and Bayes/MDL for multiband image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 884–900, 1996.

[27] S. Jehan-Besson, M. Barlaud, and G. Aubert, "A 3-step algorithm using region based active contours for video objects detection," *EURASIP Journal on Applied Signal Processing*, vol. 6, pp. 572–581, 2002.

[28] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, 1996.

[29] F. Candocia and J. C. Prince, "Comments on sinc interpolation of discrete periodic signals," *IEEE Transactions on Signal Processing*, vol. 11, pp. 2044–2047, 1998.

[30] M. Jacob and M. Unser, "Design of steerable filters for feature detection using a Canny-like criterion," *Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[31] N. Paragios and R. Deriche, "Geodesic active regions for supervised texture segmentation," in *International Conference on Image Processing*, Corfou, Greece, 1999, pp. 688–694.

[32] L.D. Cohen and I. Cohen, "Finite-element methods for active contour models and baloons for 2-D and 3-D images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1131–1147, 1993.

[33] F. Precioso and M. Barlaud, "Regular B-spline active contours for fast video segmentation," in *IEEE International Conference on Image Processing*, Rochester, USA, 2003.

[34] I.J. Schoenberg, "Spline functions and the problem of graduation," *Proc. Nat. Acad. Sci.*, vol. 52, pp. 947–950, 1964.

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

[36] T. Blu and M. Unser, "Quantitative Fourier analysis of approximation techniques: Part II—Wavelets," *IEEE Transactions on Signal Processing*, vol. 47, no. 10, pp. 2783–2795, 1999.

[37] G. Aubert, M. Barlaud, O. Faugeras, and S.J. Besson, "Image segmentation using active contours: calculus of variations or shape gradients," *SIAM Journal of Applied Mathematics*, 2003.

[38] M. Jacob, T. Blu, and M. Unser, "An exact method for computing the area moments of wavelet and spline curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 633–642, 2001.

[39] E. Weisstein, "Eric Weisstein's world of mathematics," *http://mathworld.wolfram.com*.

[40] W. Rasband, "ImageJ web site," *http://rsb.info.nih.gov/ij/*.