Proc. Of SPIE Vol 3458, Wavelet Applications in Signal and Image Processing VI, ed. A F Laine, M A Unser, A Aldroubi (1998) Copyright SPIE

Multi-scale B-spline Snakes for General Contour Detection

Patrick Brigger[1] and Michael Unser[2]

[1]*Biomedical Engineering and Physical Science*
National Institutes of Health (NIH)
Bethesda, MD, USA
[2]*Biomedical Imaging Group*
Swiss Federal Institute of Technology (EPFL)
CH-1015 Lausanne, Switzerland

ABSTRACT

Traditional snakes suffer from slow convergence speed (many control points) and difficult to adjust weighting factors for internal energy terms. We propose an alternative formulation using cubic B-splines, where the knot spacing is variable and controlled by the user. A larger knot spacing allows to reduce the number of parameters, which increases optimization speeds. It also eliminates the need for internal energies, which improves user interactivity. The optimization procedure is embedded into a multi-resolution image representation, where the number of snake points is adapted to the image grid spacing by correctly adjusting the spline knot spacing. Hence, the proposed method provides a multi-scale approach in both the image and parametric contour domain. Our technique provides fast optimization of the initial snake curve and leads to more stable algorithms in noisy imaging environments. Several biomedical examples of applications are included to illustrate the versatility of the method.

1. INTRODUCTION

The original snake by Kass et. al. [1] is the solution of a functional minimization. While it provides an elegant mathematical formulation, it has two main drawbacks: 1) a large number of control points subject to optimization, and, 2) an explicit formulation of the smoothness constraint, which requires the specification of weighting factors — these are often difficult to determine a priori. The associated problems of slow convergence speed, non-intuitive user-interaction because of weight factors, the necessity of representing the contour as a finite set of disconnected points, and the difficulty of obtaining reliable high order derivatives of a discrete curve have been tackled in a number of publications [2, 3, 4, 5]. For instance, Amini et. al. [2] proposed a dynamic programming strategy for the optimization, thus guaranteeing convergence, enabling the integration of hard constraints, and bypassing local minima. Several authors were able to improve convergence speed by using different types of greedy optimization algorithms [3, 4]. A way to overcome the difficulties of the proper choosing of the weight parameters is described in [5]. Here, the internal forces are controlled by approximating the contour by curvature arcs that compensate for the normal forces.

An alternative approach to snakes, which also circumvents some of the problems, is to use a parametric B-spline representation of the curve. Menet et. al. [6] introduced the concept of B-spline snakes, emphasizing the advantages of local control, compact representation, and the possibility to include corners. They also showed that B-spline snakes could offer improved convergence speed and stability. While internal curve energies were still used in [6], Flickner et. al. [7] recognized that the inherent smoothness built into the spline model combined with a fewer number of control points no longer requires the introduction of internal energies. Hence, a fast spline rendering algorithm based on fast Bezier curve rendering makes an optimization process at interactive speeds possible. The property of implicitly built-in smoothness into the B-spline has also been successfully exploited in [8], who proposed a multi-stage model for the optimization. Hence, the B-snake is mainly characterized by the following points: 1) few parameters, and 2) smoothness implicitly built into the model. In addition, the B-snake approach naturally permits the local control of the curve by controlling individual control points.

In this work, we propose to take advantage of some of the smoothness properties of the B-splines, thus improving optimization speed and user interactivity. Our main contributions are as follows: First, we show that the cubic B-spline is the optimal solution for cost functions involving a smoothness constraint that is based on the second derivative of the curve. Second, we introduce a scale parameter for the B-splines, which allows us to control the smoothness of the snake implicitly, thus eliminating the regularization term used in the traditional snake implementation. The proposed formulation eliminates the need for curve-internal energies and also allows the easy introduction of external hard-constraint points. Since increasing the

scale of the spline is related to reducing the number of node points, this kind of approach tends to simplify the optimization process. Third, we improve the speed and robustness of the optimization by making use of a multi-resolution technique (image pyramid). We employ a centered cubic spline pyramid which features improved PSNR values with respect to standard pyramid constructions [9]. The original image array is orthogonally projected onto a sequence of coarser spline spaces with a dyadic scale progression $m = 2^i$; this yields a series of images where the resolution is decreased by a factor of two from one level to the next. Accordingly, the resolution of the snake curve can be decreased by the same amount, which will provide faster computation speeds without compromising the approximation quality.

The paper is organized as follows. In Section 2, we start with a theoretical analysis that provides strong arguments in favor of the B-spline model. In Section 3, we justify our simplification of the conventional model, which is to impose smoothness constraints implicitly through the knot spacing of the spline rather than by regularization, and provide the mathematical formulas for the parametric B-spline snake. We also discuss some of the implementation issues; in particular, the specification of external forces. In Section 4, we consider optimization strategies and describe our multi-resolution approach. In Section 5, we test the performance of the proposed method. Finally, in Section 6, we present several examples for semi-automatic contour detection in biomedical image sequences.

2. SNAKES REVISITED: A FUNCTIONAL FORMULATION

The purpose of this section is to justify the use of splines for solving snake problems. In order to develop a mathematical model, we will consider a simple configuration where the contour to be detected can be represented by a single function. Even though this corresponds to a somewhat restricted situation (a general curve requires one function per coordinate dimension), it will provide us with important mathematical insights that are directly transposable to the more general case, which will be treated in Section 3. The present functional model may offer an interesting alternative to dynamic programming techniques, which are traditionally used in this context [10, 11].

The basic problem is to detect a contour described by a function $y_0 = f(x)$ in the x-y plane. Instead of an explicit definition, the unconstrained contour curve y is specified in terms of the minimum of a potential function $V(x, y)$

$$V(x, y) = g(y - f(x); x) \tag{1}$$

with the condition that $g(z; x) \geq g(o; x)$, $\forall z$. For notational simplicity, we assume that $f(x)$ is defined over the entire real line.

2.1 Regularized solution — cubic splines are optimal

The basic problem is to approximate $f(x)$ by a snake curve $s(x)$ that is constrained to be smooth. For this purpose, we consider the following optimization problem

$$s^*(x) = \arg\min_s \sum_{k \in Z} V(k, s(k)) + \lambda \int_{-\infty}^{+\infty} \left(\frac{d^2 s(x)}{dx^2} \right)^2 dx \tag{2}$$

which constitutes a one-dimensional cost function similar to the one introduced by Kass et al. [1]. This criterion involves two distinct terms. The first is the so-called data term, which forces the solution to be close to the minimum of the potential function $V(x, y)$. Note that $V(x, y)$ is evaluated at the discrete location $x = k$ and $y = s(k)$ to accommodate for the discrete nature of the input data (external force). The second term expresses a smoothness constraint, which will tend to privilege solutions that have a low average curvature. The amount of smoothness of the solution is controlled by the regularization factor λ; it typically reflects our a priori knowledge. We are assuming that the problem is well defined in the sense that a solution exists. Note that this solution is a snake function $s(x)$ that is continuously defined over R, even though our data are discrete.

In order to be able to find the solution, we now present our key result, and show that the optimal snake is indeed a spline.

Theorem 1: The solution of problem (2) is a cubic spline with knots at the integers.

Note that the solution is not necessarily unique because we did not impose any particular constraint on V(*x,y)* (e.g., convexity). However, an optimal snake curve is uniquely defined by its values at the knot points $s^*(k)$, $k \in Z$, or, equivalently, by the sequence of its B-spline coefficients (cf. Section 2.2).

Proof of Theorem 1: For any given snake candidate $s(x)$, we define its cubic spline interpolant $s_{\text{int}}(x)$, which is uniquely defined as in [12]. It is a C^2-curve that agrees with $s(x)$ at the integers (i.e., $s_{\text{int}}(k) = s(k),\ \forall k \in Z$), and is a cubic polynomial on each interval $[k, k+1),\ k \in Z$. We then rewrite the cost function as

$$\xi(s) = \sum_{k \in Z} V\left(k, s_{\text{int}}(k)\right) + \lambda \int_{-\infty}^{+\infty} \left(\frac{d^2 s(x)}{dx^2} \right)^2 dx, \tag{3}$$

where we have substituted $s(k)$ by $s_{\text{int}}(k)$ in the first part of the criterion while leaving the value of $V(\cdot,\cdot)$ unchanged. To manipulate the second term, we use the so-called first integral equation [13], which states that for any function $s(x)$ whose second derivative is square integrable [14, Lecture 6)]:

$$\int_{-\infty}^{+\infty} \left(\frac{d^2 s(x)}{dx^2} \right)^2 dx = \int_{-\infty}^{+\infty} \left(\frac{d^2 s_{\text{int}}(x)}{dx^2} \right)^2 dx + \int_{-\infty}^{+\infty} \left(\frac{d^2 s(x)}{dx^2} - \frac{d^2 s_{\text{int}}(x)}{dx^2} \right)^2 dx, \tag{4}$$

Hence, we obtain

$$\xi(s) = \sum_{k \in Z} V\left(k, s_{\text{int}}(k)\right) + \lambda \int_{-\infty}^{+\infty} \left(\frac{d^2 s_{\text{int}}(x)}{dx^2} \right)^2 dx + \lambda \int_{-\infty}^{+\infty} \left(\frac{d^2 s(x)}{dx^2} - \frac{d^2 s_{\text{int}}(x)}{dx^2} \right)^2 dx. \tag{5}$$

By hypothesis, there exists a solution s^*, not necessarily unique, for which $\xi(s^*)$ is minimal. This solution has a unique interpolator s^*_{int}, which fixes the first and second terms in the above expression. Finally, $\xi(s^*)$ can be minimal if and only if the third term is minimal as well, that is, when $s^{*(2)} - s^{*(2)}_{\text{int}} = 0$ *almost everywhere*; i.e., the set of x's such that $s^{*(2)} - s^{*(2)}_{\text{int}} \neq 0$ is of measure zero. If we integrate twice, we get that $s^* - s^*_{\text{int}} = ax + b$ *everywhere* (because a set of measure zero does not contribute to the integral). Finally, because of the interpolation condition, we conclude that $s^*(x) = s^*_{\text{int}}(x)$ *everywhere*. It follows that the optimal snake $s^*(x)$ is a cubic spline. ■

We can relate our problem to curve fitting by interpreting the function $g(\cdot, x)$ in (1) as a pseudo-metric in y, which is allowed to vary as we move along x (or increment k). In particular, if we consider a quadratic criterion of the form $g(s(k) - f(k); k) = w(k)[s(k) - f(k)]^2$, where $w(k)>0$ is a sequence of weighting factors, then the optimization task is mathematically equivalent to the well-known smoothing spline problem in statistics [15]. The goal there is to find a smooth (regularized) curve s(x) that is reasonably close (in the least squares sense) to a set of noisy data points $f(k)$. The fact that this leads to a spline solution is well known in this context (quadratic cost function); it was established independently by Reinsh and Schoenberg [16, 17]. Theorem 1 extends this result considerably because we did not make any hypothesis on the form of the potential function (or pseudo-metric) $g(s(k) - f(k); k)$. In other words, we have shown that splines are optimal, irrespective of the metric used, which is a rather remarkable property.

2.2 Computational solution

To solve the snake problem numerically, we express its cubic spline solution using the standard B-spline expansion

$$s^*(x) = \sum_{k \in Z} c(k) \beta^3(x - k) \tag{6}$$

where the $c(k)$ are the B-spline coefficients, and where the generating function is a cubic B-spline.

Thanks to (6), we can now manipulate (2) to obtain a discrete form of the criterion in terms of the B-spline coefficients $c(k)$. Using the basic convolution and differentiation rules of splines (cf. [18]), we obtain the explicit formula

$$\xi(s) = \sum_{k \in Z} V(k, (b_1^3 * c)(k)) + \lambda \sum_{k \in Z} (b_1^3 * d^{(2)} * c)(k)(d^{(2)} * c)(k) \tag{7}$$

where * denotes the discrete convolution operator and where the kernels b_1^3 (discrete cubic B-spline) and $d^{(2)}$ (second difference) are defined by their z-transform as follows: $B_1^3(z) = (z+4+z^{-1})/6$ and $D^{(2)}(z) = z-2+z^{-1}$ (cf. [19]). Note that we have now replaced the integral in the second term by a sum, which is much more tractable computationally. The task is then to minimize (7), which is typically achieved by differentiation with respect to $c(k)$. In the case of a quadratic potential function, this leads to a linear system of equations that can be solved using any of the standard techniques [15]. For the more general case when $V(x, y)$ is not quadratic, the solution may still be determined numerically, for example by using an iterative algorithm (steepest descent or conjugate gradient).

Note that the spline snake (6) has as many degrees of freedom (B-spline coefficients) as there are discrete contour points; i.e., one per integer grid point. If λ is sufficiently small, then the spline (5) will interpolate $f(k)$ exactly. Conversely, the use of larger values of λ will have the effect of stiffening the spline and smoothing out the discontinuities of the unconstrained contour curve $f(x)$.

3. B-SPLINE SNAKES: PARAMETRIC FORMULATION

3.1 Parametric solution

The previous section has provided the link between splines and the traditional variational formulation of snakes. In this section, we propose to impose smoothness constraints in a simpler and more economical fashion, and to give an intuitive B-spline snake formulation useful for images.

The idea is to eliminate the second term in (7) and to introduce a variable knot spacing between the knot points. An increased knot spacing will essentially have the same smoothing effect on the solution. Thus, we consider the simplified optimization problem

$$s^*(x) = \arg\min_{s_h(t)} \sum_{k\in Z} V(k, s_h(k)) \tag{8}$$

which is now constrained indirectly in the sense that $s_h(x)$ with $h>1$ is a coarser spline with knot spacing h:

$$s_h(x) = \sum_{k\in Z} c_h(k) \cdot \beta^3(x/h - k) \tag{9}$$

Hence, our new smoothness parameter is h rather than λ. Typically, we will take h to be an integer m, which will reduce the number of degrees of freedom (B-spline coefficients) in the same proportion. If we perform the same substitution as before, we find that in the case of a quadratic potential function the new solution corresponds to a weighted least square spline approximation of the unconstrained curve $f(x)$. In the general case where $V(x, y)$ is not quadratic, we still have some form of minimum error approximation, except that the "metric" is no longer Euclidean.

To differentiate this new solution from the previous one, we will call it a parametric spline. This terminology is justified by the fact that the smoothness constraint is entirely implicit and that the number of degrees of freedom is much less than the number of contour points. A reduced number of control points simplifies the implementation and accelerates computation.

The argument is essentially the same for more general curves in the plane, which are described using two splines instead of one. Specifically, we represent a general B-spline snake as follows:

$$\mathbf{s}_h(t) = (s_x(t), s_y(t)) = \sum_{k=Z} \mathbf{c}(k) \cdot \beta^n(\frac{t}{h} - k), \quad (0 \le t \le t_{\max} = hN - 1). \tag{10}$$

where $s_x(t)$ and $s_y(t)$ are the x and y spline components, respectively; these are both parameterized by the curvilinear variable t. The exact value of $t_{\max}$, which marks the end of the curve, is dictated by the desired resolution of the final discrete curve; by convention, we do only render the curve points for t integer. This 2D spline snake is characterized by its vector-sequence of B-spline coefficients $\mathbf{c}(k) = (c_x(k), c_y(k))$. Note that there are only $N = t_{\max}/h$ primary coefficient vectors, each corresponding to a spline knot on the curve; the other coefficient values are deduced using some prescribed boundary conditions. N node points that lie exactly on the snake define the initial curve. The B-spline coefficients $\mathbf{c}(k)$ are obtained by inverse filtering [18] from the N node points.

Clearly, if we specify N, the above automatically defines the knot-spacing h and therefore the smoothness constraint for the curve. Assuming a curve representation by $M = t_{\max} + 1$ discrete points, we obtain $h = M / N$. The freedom of the spline curve has been reduced by the same amount, resulting in a smoothing and stiffening of the curve. Increasing the number N of node points will reduce the knot spacing, and consequently it will reduce the smoothing effect on the curve.

3.3 Energy formulation

With the introduction of variable knot spacing, we no longer require internal curve energies. Experimental tests elsewhere demonstrate this point experimentally [20]. In the general case, the external potential function is defined by the gradient of the input data. While there are many possibilities to compute the gradient, we have chosen the morphological gradient:

$$\nabla I(x, y) = Dilation(I) - Erosion(I) \tag{11}$$

This gradient is smooth and has proven useful in other applications [21]. Our cost function is the summation of the gradient (external force) over the path of the curve $\mathbf{s}(x)$ sampled at M consecutive points:

$$\xi(\mathbf{c}(k)) = \sum_{i=0}^{M-1} -\nabla I(\mathbf{s}(i)) \tag{12}$$

For the cost function to be a good approximation of the curvilinear integral, we will typically select M sufficiently large so that the curve points are connected (i.e. within a distance of one pixel of each other). We note, however, that the exact value of M is not critical; a less dense sampling may be used to increase optimization speed. The negative sign in (12) is used because we employ a minimization technique for the optimization.

4. OPTIMIZATION STRATEGIES

4.1 Function minimization

The literature is rich on minimization strategies, including methods such as golden section search, steepest descent methods and conjugate gradient methods, as well as heuristic techniques tailored to a particular problem. In two dimensional optimization problems, one may advantageously use the information of the gradient of the energy function if it is available. The B-spline formulation allows an easy computation of the gradient function of the energy term. The parameters subject to optimization are the B-spline coefficients (and hence indirectly the node-points), yielding:

$$\begin{aligned} \frac{\partial \xi(\mathbf{c}(k))}{(\mathbf{c}(k)} &= \sum_{i=0}^{M-1} \frac{\partial(\nabla I(\mathbf{s}(i))}{\partial \mathbf{c}(k)} \\ \frac{\partial(\nabla I(\mathbf{s}(i))}{\partial \mathbf{c}(k)} &= \frac{\partial(\nabla I(\mathbf{s}(i)))}{\partial \mathbf{s}(i)}\bigg|_{\mathbf{s}=\mathbf{s}(\mathbf{c}(k))} \cdot \frac{\partial \mathbf{s}}{\partial \mathbf{c}(k)} = \frac{\partial(\nabla I(\mathbf{s}(i)))}{\partial \mathbf{s}(i)}\bigg|_{\mathbf{s}=\mathbf{s}(\mathbf{c}(k))} \cdot b^n\left(\frac{i}{h} - k\right) \end{aligned} \tag{13}$$

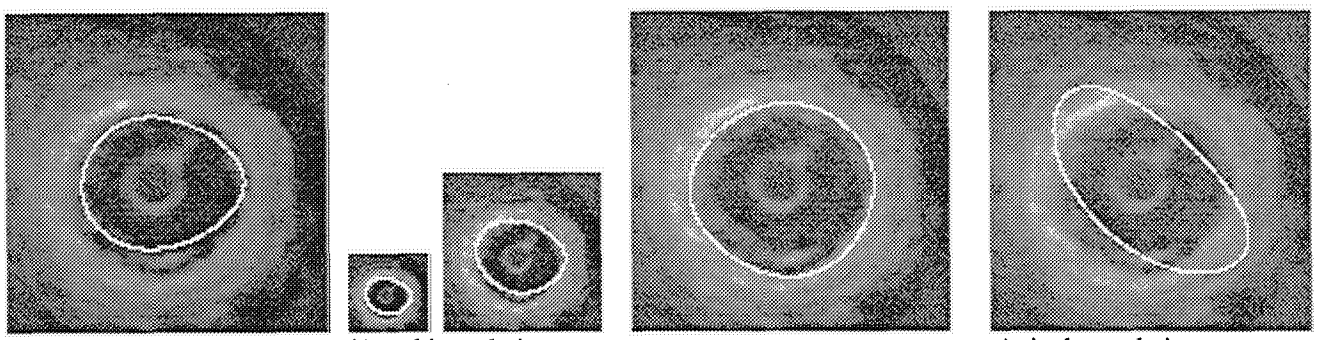
The term $\left.\frac{\partial(\nabla I(\mathbf{s}(i)))}{\partial \mathbf{s}(i)}\right|_{\mathbf{s}=\mathbf{s}(\mathbf{c}(k))}$ is the derivative of the force function in the spatial domain. It can be obtained by computing a direct B-spline transform followed by a spline interpolation from a derivative B-spline. The direct transform only has to be done once at the beginning of the optimization. The use of splines throughout the entire development allows us to define a consistent B-spline snake framework. Equipped with the gradient, we can proceed to find the minimum of the energy function. Two algorithms have been studied: steepest descent and conjugate gradient. The former starts at an initial snake contour with parameters $\mathbf{c}_0(k)$. Consecutive values of $\mathbf{c}_i(k)$ are changed to $\mathbf{c}_{i+1}(k)$ by minimizing along a line going from $\mathbf{c}_i(k)$ in the direction of the local downhill gradient $-\lambda \frac{\partial \xi(\mathbf{c}_i(k))}{\partial \mathbf{c}_i(k)}$. Lambda is found by performing a line search minimizing $\xi(\mathbf{c}_i(k))$ along the direction of the gradient. This algorithm is often not efficient, because it potentially minimizes along the same line many times. The conjugate gradient algorithm provides a better solution, because it employs search directions that are *A-orthogonal*, which – in theory and for a quadratic potential function – perform exactly one minimization along the same line [22]. For non-quadratic potential functions, the algorithm is iterated, and still outperforms the steepest descent algorithm in the majority of cases.

4.2 Multi-resolution procedure

Snakes may get trapped in local minima. Hence, a good initialization close to the desired contour is necessary. It is obtained based on *a priori* information or based on a manual placement. This requirement can be loosened through the use of a multi-scale representation. Multiscale processing is an old but powerful idea [23, 24]. It is usually applicable whenever one wishes to implement an image processing algorithm that is iterative in nature and requires many successive updates. The basic principle is to construct an image pyramid and to start applying the procedure at the coarsest level on a very small version of the image. Upon convergence, the solution is propagated to the next finer level where it is used as starting condition. One then proceeds with this coarse-to-fine iteration strategy until one reaches the finest level of the pyramid which corresponds to the image itself. This type of multi-resolution approach has two advantages: first, on reduced versions of the image, a smaller number of curve points can be used thereby improving optimization speed. Second, it usually also improves robustness; the pyramid has a smoothing effect on the criterion to be optimized which often reduces the likelihood of getting trapped in local optima.

We have presented a pyramid decomposition for improved up-projection of labels [25]. In this framework, the pyramid has a common axis of symmetry at all resolution levels. The image at a resolution r is orthogonally projected on a subspace $V_{r+1}(b_w^n(k))$, which may be defined by spline basis functions at different scales (the subscript w defines the scale of the spline, i.e. the knot spacing). If one uses powers of two for w, then for every step in the pyramid the image size is reduced by two in each dimension. Hence, the original set of node points $\mathbf{n}(k)$ is down projected onto a smaller version of the original image. The reduced image contains fewer image details, and thus less noise. The snake is potentially able to converge to the correct result from an initial contour farther away from the final contour. An example demonstrating the increased robustness of the multi-resolution approach is given in Fig. 1. In Fig. 1a), the initial starting contour is shown, which was drawn manually. The result of the direct optimization is shown in Fig. 1c). Then, a three-level pyramid was computed using cubic spline basis functions. The optimized contours for the different levels are shown in Fig. 1b). Clearly, the multi-resolution decomposition eliminates small (undesired) image details and the snake is able to converge to the true result.

At spatially coarser resolution levels, the discrete curve points are spaced closer together. Accordingly, the resolution of the snake curve can be decreased by the same amount, which provides faster computation speeds without compromising the approximation quality. Given a snake scale parameter h_0 at the original resolution, we obtain $h_m = h_0 / m$ for the coarser pyramid level.



a) initial contour b) multi-resolution c) single-resolution

Figure 1: Identical initial contour a) for both b) and c), but only correct convergence with the multi-resolution approach.

5. PERFORMANCE EVALUATION

We have already shown elsewhere that our B-spline snake formulation without internal energies produces results similar to those of the traditional snake using internal energies [20], while providing faster optimization speeds and providing more intuitive user interaction. Here, in addition, we would like to assess whether the proposed approach provides more consistent, and therefore in average more precise optimization results. The smaller the variance of multiple measures of the same quantity, the more consistent the analysis is. We propose to measure intra-observer consistency, as well as algorithm consistency.

Intra-observer consistency is assessed from five different manual measurements of the surface-area of the corpus callosum and the associated standard errors. The five measurements were repeated using two different drawing tools; first using the pencil tool of a public domain software (NIH Image, V. 161), and then using the B-spline interface. For the latter, the user clicked and positioned node points along the contour, which were linked by a cubic spline. The manual tracer was free in selecting the number of desired node points. In average, about 20 node points were used.

Algorithm consistency was assessed by measuring surface-area and standard error of the mean of the results obtained after optimization of the initial contours from the B-spline interface. Results are presented in Fig. 2 and in Table I. We note that manual contour outlining using the pencil tool or using the B-spline snake essentially produces the same quantitative results. Uncertainty values in terms of standard error of the mean are comparable for both approaches. Time requirements were similar, slightly favoring the B-spline approach. The latter, however, provides a visually better looking and more appealing contour (see Fig. 3). Surface-areas obtained after optimization of the initial curve prove to be more reliable, expressed by the lower value of standard error. The mean surface-area is smaller than in the previous two cases. It seems that the human tracer has a tendency to place contours somewhat outside of the transition, so that these transitions remain visible. An experiment with a phantom object would be adequate here to determine which of the two contours is closer to the true contour. Note also that the subjective impression of the manual tracer was that the B-spline snake offered a more agreeable way of contour outlining.

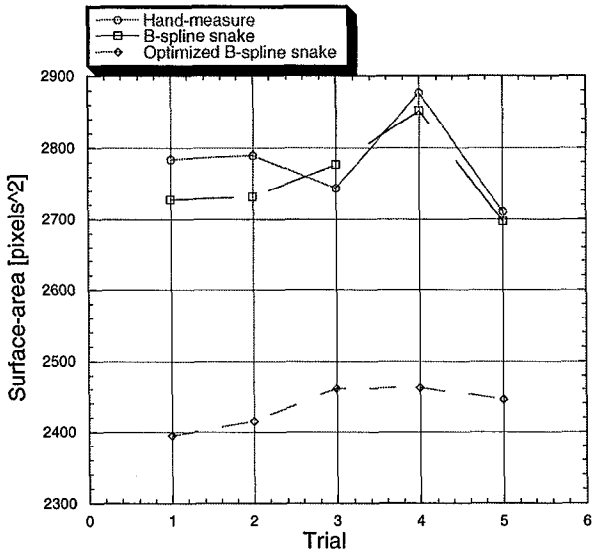
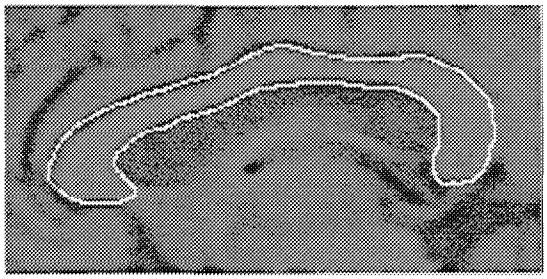
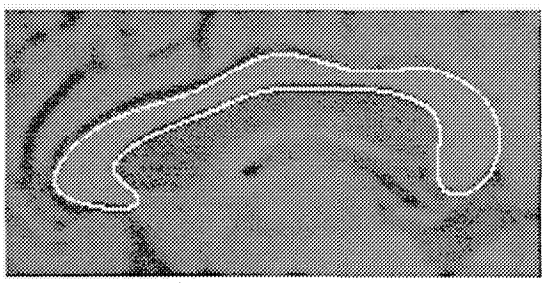


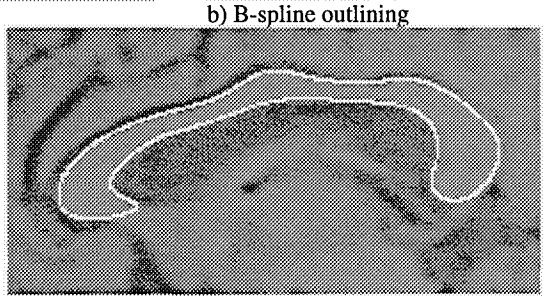
Figure 2: Comparison of surface areas obtained by 1) manual tracing, 2) B-spline outlining, and 3) B-spline snake with optimization



a) Manual outlining with pencil tool



b) B-spline outlining



c) B-spline outlining with optimization

Figure 3: Comparison of different contour outlining techniques

Table I:

Surface-area and standard errors obtained for repetitive contour outlining of the corpus callosum by 1) manual tracing, 2) B-spline outlining, and 3) optimized B-spline snake

Manual	B-spline outlining	Optimized B-spline snake
$s = 25;\ a = 2784$	$s = 20;\ a = 2727$	$s = 27;\ a = 2395$
$s = 23,\ a = 2789$	$s = 20;\ a = 2732$	$s = 24;\ a = 2415$
$s = 23;\ a = 2744$	$s = 21;\ a = 2777$	$s = 25;\ a = 2461$
$s = 22;\ a = 2877$	$s = 20;\ a = 2852$	$s = 25;\ a = 2462$
$s = 22;\ a = 2710$	$s = 23;\ a = 2697$	$s = 24;\ a = 2445$
$\bar{s} = 23;\ \bar{a} = 2781$	$\bar{s} = 21;\ \bar{a} = 2757$	$\bar{s} = 25;\ \bar{a} = 2436$
$std\ \ err = 28.0$	$std\ \ err = 27.0$	$std\ \ err = 13.2$

6. BIO-MEDICAL APPLICATIONS

6.1 Segmentation of corpus callosum in an MRI image

The corpus callosum is a relevant brain feature, and physicians are often interested in its size. Images are typically assessed by MRI. For this and other cases as well, it is routine medical practice for a technician to outline the boundaries manually. Often, the standard imaging tools consist of an interface that allows manual border tracing using the computer mouse. Technicians face several problems related to that task. First, the hand drawn boundary is subject to small, uncontrollable hand movements that result in a noisy boundary. Second, the technician often does not have the possibility to pause in the outlining process. Also, a curve that is not considered satisfactory often has to be redrawn in its entirety.

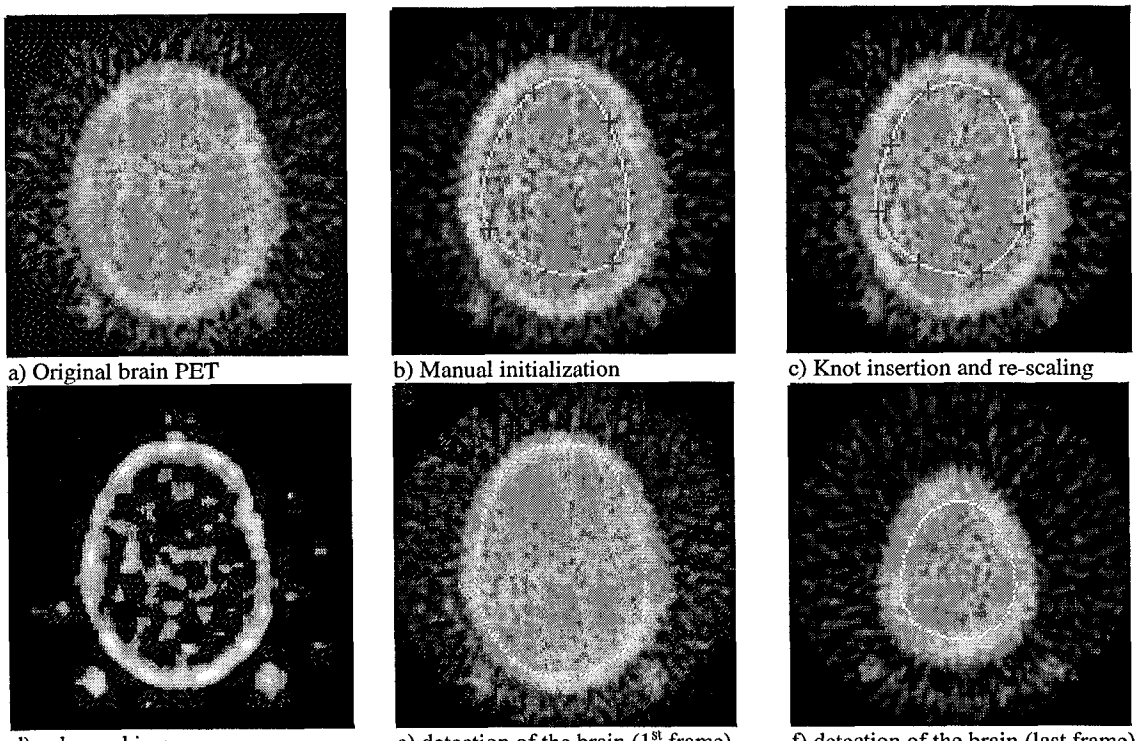
We have asked an experienced technician to outline the boundaries of the corpus callosum using the B-spline snake concept. She found the tools appealing, and stressed the fact that it is much easier and more accurate to click on a certain number of points, instead of having to follow the desired contour using a pencil drawing tool. We have presented this example in detail during the algorithm description (see Fig. 3). In this context, note also that elastic deformation transformations have been studied as a quantitative description of the callosal shape with respect to the Talairach atlas [26].

For this particular example, we also tried the standard Kass-type regularized snake but had considerable difficulties to find a set of the stiffness parameters that would make the snake converge to an appropriate solution. The main difficulty there is that

even if we assume that those parameters exist, it can be quite time consuming to adjust them interactively by trial and error. For the B-spline approach, the user interaction is much more natural; the snake can be constructed easily to conform to less standard shapes by simply entering more nodes in areas where the contour is not so smooth.

6.2 Segmentation of a PET image sequence

The purpose of this application is to demonstrate the vast range of applicability of the B-spline snake. PET-scans are much noisier than MRI scans or CT scans, and of much different nature than ultrasound images (coronary artery). An example of one frame of an 18-frame sequence of the brain is shown in Fig. 4a). Here, researches are interested in finding the boundaries of the brain. The B-spline snake is not suited for separating gray and white matter, because the desired contour is not smooth and would require the introduction of a large number of node points. However, on the PET scan, it may be an interesting tool for a gross separation of the brain from the scull.
Enhancement of the original image (Fig. 4a) by a contrast operator highlighted contours of interest (Fig. 4d). Manual placement of the initial contour in the first frame of the sequence using four node points (Fig. 4b) was followed by automatic knot insertion at a scale h twice as fine, yielding a modified initial contour with eight equidistantly spaced node points (Fig. 4c). This contour was optimized using the conjugate gradient algorithm and a multi-resolution decomposition of two levels. The optimized contour (Fig. 4e) is automatically propagated to the next neighboring frame, where it served as new initial contour (i.e. no more user interaction necessary). The final result for frame 18 is shown in Fig. 4f.



a) Original brain PET
b) Manual initialization
c) Knot insertion and re-scaling
d) enhanced image
e) detection of the brain (1st frame)
f) detection of the brain (last frame)

Figure 4: PET-scan of the brain.

6.3 Segmentation of ultrasound images

In this application, the B-spline snake is applied on coronary artery ultrasound images for the detection of the endothelial wall. From the resulting curve, the treating physician is then capable to compute surface-area of the coronary artery, which is an important measure for the analysis of atherosclerosis.
The segmentation has been obtained by manually positioning the initial curve in the first frame of the sequence at the approximate correct location using four node points and a cubic B-spline snake. The curve was then automatically optimized to attract it toward the endothelial wall using the conjugate gradient algorithm and a three-level pyramid. The final

segmentation result has been shown in Fig. 1. The entire image sequence is segmented based on a single initial contour for the first frame, with forward propagation of the initial solution to consecutive frames.

7. CONCLUSIONS

In this paper, an alternate B-spline snake formulation has been presented. Its main characteristic is a variable knot spacing, which allows to implicitly control the smoothness of the curve. We have given theoretical justification for B-splines to be optimal in a snake formulation where internal energies are no longer necessary. Also, we have presented an optimization strategy using a multi-resolution approach in both the image domain (image pyramid) and the parametric contour domain (larger spline knot spacing). Such an approach ensures fast execution speeds, as well as robustness in noisy image environments. We have presented an experiment to assess intra operator variability, demonstrating that the proposed approach produces consistent results and provides a mean for intuitive user-interaction.
The method has been demonstrated on a variety of problems, featuring different image modalities. The proposed algorithms can be applied on a vast range of images with varying signal to noise ratios and image resolutions. We believe that the proposed technique could be a valuable tool for the outlining of image contours in bio-medical applications.

8. ACKNOWLEDGMENTS

The authors would like to thank Akram Aldroubi for his numerous inputs with respect to mathematically related problems, Murray Eden for his valuable discussions, and Philippe Thévenaz for his suggestions for algorithm testing and help on B-spline implementation.

9. REFERENCES

[1] M. Kass, A. Witkin and D. Terzopoulos, "Snakes: Active contour models", *Internat. J. of Computer Vision*, pp. 321-331, 1988.
[2] A. Amini, T. Weymouth and R. Jain, "Using dynamic programming for solving variational problems in vision", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, no. 9, pp. 855-867, September, 1990.
[3] D. Williams and M. Shah, "A fast algorithm for active contours and curvature estimation", *CVGIP: Image Understanding*, vol. 55, no. 1, pp. 14-26, 1992.
[4] K. Lam and H. Yan, "Fast greedy algorithm for active contours", *Electronics Letters*, vol. 30, no. 1, pp. 21-23, 1994.
[5] G. Xu, E. Segawa and S. Tsuji, "Robust active contours with insensitive parameters", *Pattern Recognition*, vol. 27, no. 7, pp. 879-884, 1994.
[6] S. Menet, P. Saint-Marc and G. Medioni, "B-snakes: Implementation and application to stereo", in *Image Understanding Workshop*, pp. 720-726, Darpa, September, 1990.
[7] 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*, vol. 1, pp. 731-735, 1994.
[8] M. Wang, J. Evans, L. Hassebrook and C. Knapp, "A multistage, optimal active contour model", *IEEE Trans. on Image Processing*, vol. 5, no. 11, pp. 1586-1591, November, 1996.
[9] P. Brigger and M. Unser, "General discrete image pyramids", in *SPIE*, vol. 3169, pp. 212-223, San Diego, July, 1997.
[10] M. Unser, G. Pelle, P. Brun and M. Eden, "Automated extraction of serial myocardial borders from M-mode echocardiograms", *IEEE Transactions on Medical Imaging*, vol. MI-8, no. 1, pp. 96-103, 1989.
[11] P. Brigger, S.L. Bacharach, G. Srinivasan, K.A. Nour, V. Dilsizian, A. Aldroubi and M. Unser, "Segmentation of gated Tl-201-SPECT images for automatic computation of myocardial volume and ejection fraction", *Journal of Nuclear Cardiology*, 1998. In press.
[12] I.J. Schoenberg, "Contribution to the problem of approximation of equidistant data by analytic functions", *Quart. Appl. Math.*, vol. 4, pp. 112-141, 1946.
[13] J.H. Ahlberg, E.N. Nilson and J.L. Wash, *The theory of splines and their applications*. New York: Academic Press, 1967.
[14] I.J. Schoenberg, *Cardinal spline interpolation*. Philadelphia, PA: Society of Industrial and Applied Mathematics, 1973.
[15] G. Wahba, *Spline models for observational data*. Philadelphia: 1990.
[16] I.J. Schoenberg, "Spline functions and the problem of graduation", in *Proc. Nat. Acad. Sci.*, vol. 52, pp. 947-950, 1964.
[17] C.H. Reinsh, "Smoothing by spline functions", *Numer. Math.*, vol. 10, pp. 177-183, 1967.

[18] M. Unser, A. Aldroubi and M. Eden, "B-spline signal processing. Part II : efficient design and applications", *IEEE Trans. Signal Processing*, vol. 41, no. 2, pp. 834-848, February, 1993.
[19] M. Unser, A. Aldroubi and M. Eden, "B-spline signal processing. Part I : theory", *IEEE Trans. Signal Processing*, vol. 41, no. 2, pp. 821-833, February, 1993.
[20] P. Brigger, R. Engel and M. Unser, "B-spline Snakes and a JAVA interface: An Intuitive Tool for General Contour Outlining", in *ICIP'98*, Chicago, October 4-7, 1998, 1998. In press.
[21] P. Brigger, "Morphological plant cell analysis", in *First Workshop on Mathematical Morphology and its Applications to Signal Processing*, pp. 101-106, Barcelona, May, 1993.
[22] E. Polak, *Computational methods in optimization*. New York: Academic Press, 1971.
[23] A. Rosenfeld, *Multiresolution Image Processing*. New, York: Springer-Verlag, 1984.
[24] P.J. Burt and E.H. Adelson, "The Laplacian pyramid as a compact code", *IEEE Trans. Commun.*, vol. COM-31, no. 4, pp. 337-345, April, 1983.
[25] P. Brigger, F. Mueller, K. Illgner and M. Unser, "Centered pyramids", *IEEE Transactions on Image Processing*, 1996. Submitted for publication.
[26] C. Davatzikos, M. Vaillant, S.M. Resnick, J.L. Prince, S. Letovsky and R.N. Bryan, "A computerized approach for morphological analysis of the corpus callosum", *J Comput Assit Tomography*, vol. 20, no. 1, pp. 88-97, 1996.