B-Spline Snakes: A Flexible Tool for Parametric Contour Detection

Patrick Brigger, *Associate Member, IEEE*, Jeff Hoeg, and Michael Unser, *Fellow, IEEE*

***Abstract*—We present a novel formulation for B-spline snakes that can be used as a tool for fast and intuitive contour outlining. We start with a theoretical argument in favor of splines in the traditional formulation by showing that the optimal, curvature-constrained snake is a cubic spline, irrespective of the form of the external energy field. Unfortunately, such regularized snakes suffer from slow convergence speed because of a large number of control points, as well as from difficulties in determining the weight factors associated to the internal energies of the curve. We therefore propose an alternative formulation in which the intrinsic scale of the spline model is adjusted a priori; this leads to a reduction of the number of parameters to be optimized and eliminates the need for internal energies (i.e., the regularization term). In other words, we are now controlling the elasticity of the spline implicitly and rather intuitively by varying the spacing between the spline knots. The theory is embedded into a multiresolution formulation demonstrating improved stability in noisy image environments. Validation results are presented, comparing the traditional snake using internal energies and the proposed approach without internal energies, showing the similar performance of the latter. Several biomedical examples of applications are included to illustrate the versatility of the method.**

I. Introduction

MANY applications require the extraction of salient image features such as edges, lines, subjective differences in gray-level. In medical applications, physicians are highly interested in computer assisted devices that help to detect and outline specific organs or other medical features of interest. The computer should help but not replace the physician. Ideally, the system performs an automatic analysis to eliminate human subjectivity and intra- and inter-human variability, while providing physicians with the possibility of an intuitive and easy user-interaction. In such a way, it is possible to incorporate knowledge and experience. For curve fitting, for example, the contour generated by the algorithm should be easily modifiable by the technician. Similar constraints may be found in many other applications as well. The snake as an energy minimizing "spline" has found wide acceptance and has proven extremely useful in applications for medical analysis [17], [21], [26], [31], feature tracking in video sequences [19], three-dimensional (3-D) object recognition [47], and stereo matching [20], [27].

The snake as originally introduced by Kass *et al.* [20] is described as an energy minimizing spline guided by external and internal forces. The external force is given as a user-supplied input function. The internal forces are determined solely by the shape of the curve and are usually defined in terms of first and second order derivatives. The snake gives an elegant method to simulate an elastic material, which can dynamically conform to local image features. This type of snake has two main features: 1) a large number of control points that define the curve and 2) an explicit formulation of the smoothness constraint. The following problems have been recognized in the literature:

1) slow convergence speed because of the large number of coefficients to optimize;
2) difficulty in determining the weights associated with the smoothness constraints;
3) description of the curve by a finite set of disconnected points;
4) high-order derivatives on the discrete curve may not be accurate in noisy environments [48].

Based on the original philosophy, different solutions have been suggested to render the snake more stable and to yield faster convergence results [2], [22], [49], [50]. An alternative approach to snakes, which also circumvents some of the problems, is to use a parametric B-spline representation of the curve, first introduced as B-snake [27], and improved in [18], [23], [48]. Such a formulation of an active contour allows local control, compact representation, and it 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. Another model that can handle multiple snakes, allows change of topology and is less sensitive to initialization is based on the level set interpretation of the Euclidean curve shortening equation [9], [25], [38]. Let us also mention the finite element approach [13], [14], where the solution of a differential equation allows a discretization sufficiently fine to ensure a negligible error between computed and true solution.

The motivation for our work is to extend the basic concept of B-spline snakes in order to improve their efficiency, speed, and applicability in an interactive environment. Our main contributions are as follows. First, we present a new theorem that states that the optimal solution for a conventional (curvature-constrained) snake cost function is a cubic spline, a curve that is most conveniently represented using B-spline basis functions.

Manuscript received May 28, 1998; revised May 30, 2000. This work was supported in part by the Swiss National Research Foundation. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Josiane B. Zerubia.

P. Brigger was with Biomedical Engineering and Instrumentation Program, National Center for Research Resources, National Institutes of Health, Bethesda, MD 20892 USA. He is now with getAbstract.com, CH-6003 Luzern, Switzerland (e-mail: patrick@brigger.com). He

J. Hoeg, deceased, was with the National Institutes of Health, Bethesda, MD 20892 USA.

M. Unser is with the Biomedical Imaging Group, Swiss Federal Institute of Technology CH-1015, Lausanne, Switzerland (http://bigwww.epfl.ch).

Publisher Item Identifier S 1057-7149(00)07226-2.

Second, we choose to specify an initial B-spline snake configuration using *node points* that are situated exactly on the snake curve, rather than the traditional B-spline *control points*; we believe that this approach provides a more practical, and intuitive user-interaction. Third, we introduce a scale parameter for the B-splines, which allows us to control the smoothness of the snake implicitly (as opposed to explicitly, through a regularization term). 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. Moreover, we find that the results of the method are similar to those obtained with conventional regularized snakes, which have considerably more parameters. Fourth, we improve the speed and robustness of the optimization by making use of a multiresolution technique (image pyramid). Finally, we propose to use digital filtering techniques to accelerate the process of curve rendering, which is by far the most costly part of the algorithm.

The paper is organized as follows. In Section II, we start with a theoretical analysis that provides strong arguments in favor of the B-spline model. In Section III, 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, and how to accelerate curve rendering. In Section IV, we consider optimization strategies and describe our multiresolution approach. In Section V, we test the performance of the proposed method. Finally, in Section VI, we present several examples for semiautomatic contour detection in biomedical image sequences.

II. 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 III. The present functional model may offer an interesting alternative to dynamic programming techniques, which are traditionally used in this context [4].

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

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

with the condition that $g(z; x) \geq g(0; x), \forall z$. For notational simplicity, we assume that $f(x)$ is defined over the entire real line.

A. 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 (1-D) cost function similar to the one introduced by Kass *et al.* [20]. The criterion in (2) 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 II-B).

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 [34]. 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

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

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 [1], which states that for any function $s(x)$ whose second derivative is square integrable [36, Lecture 6]

$$\begin{aligned} &\int_{-\infty}^{+\infty} \left(\frac{d^2 s(x)}{dx^2} \right)^2 dx \\ &\quad = \int_{-\infty}^{+\infty} \left(\frac{d^2 s_{\text{int}}(x)}{dx^2} \right)^2 dx \\ &\qquad + \int_{-\infty}^{+\infty} \left(\frac{d^2 s(x)}{dx^2} - \frac{d^2 s_{\text{int}}(x)}{dx^2} \right)^2 dx. \end{aligned} \tag{4}$$

Hence, we obtain

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

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. ■

The result of Theorem 1 is closely related to the classical minimum curvature property of spline interpolant [16], [51], which corresponds to the special case $\lambda \to 0$. Here, the main difference with the standard results on the optimality of splines is that we do not have an explicit solution available because of the generality of the cost function (2). However, we are still able to show that the optimal solution among all possible C^2-functions belongs to the space of cubic splines which makes the problem tractable numerically. We can also 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 [46]. 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 [32], [35]. Theorem 1 extends this result 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 remarkable property. In principle, it is possible to extend this kind of formulation using more general differential operators L (for instance, a linear combination between first and second derivatives) which leads to the so-called L-splines [37, Ch. 10]. However, we do not think that there is a great advantage in doing so since these representations are much more cumbersome than the cubic B-splines. In addition, they do not necessarily share their good approximation properties.

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

$$\beta^3(x) = \begin{cases} 2/3 + |x|^3/2 - x^2, & 0 \le |x| < 1 \\ (2 - |x|)^3/6, & 1 \le |x| < 2 \\ 0, & 2 \le |x|. \end{cases} \tag{7}$$

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. [44]), we obtain the explicit formula

$$\xi(s) = \sum_{k\in Z} V(k,\, (b_1^3 * c)(k)) + \lambda \sum_{k\in Z} (b_1^3 * d^{(2)} * c)(k)(d^{(2)} * c)(k) \tag{8}$$

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. [43]). 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 (8), 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 [46]. 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 $f(x)$. As we will see later, λ can eventually be dropped by using a variable size knot spacing, which still assures smoothness.

III. B-Spline Snakes: Parametric Formulation

A. 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 (8) 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

$$s^*(x) = \arg\min_{s_h(t)} \sum_{k\in Z} V(k,\, s_h(k)) \tag{9}$$

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

$$s_h(x) = \sum_{k \in Z} c_h(k) \beta^3(x/h - k). \tag{10}$$

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. The main advantages of this parametric formulation are as follows. First, we have reduced the number of parameters, which simplifies the implementation but also accelerates computation. Second, it is relatively easy to get an intuitive feeling for the smoothing effect of the parameter h. Specifically, we have the following error bound [24], [40], [45]

$$\|f - s_h^*\|_{L_2} \leq C \cdot h^{n+1} \cdot \left(\int_{-\infty}^{+\infty} |\omega^{n+1} \hat{f}(\omega)|^2 \, d\omega \right)^{1/2} \tag{11}$$

where

- s_h^* least squares spline approximation of $f(x)$ with knot spacing h;
- $n = 3$ degree of the spline;
- $\hat{f}(\omega)$ Fourier transform of $f(x)$.

Thus, the error can be made arbitrarily small provided that the generalized bandwidth of the signal (which is measured by the right most term) is sufficiently small when compared to h. Another way to understand the nature of this smoothing is to use the close relation that exists between spline and bandlimited approximations [42]. With this interpretation, s_h^* is more or less equivalent to the bandlimited version of $f(x)$ with a cutoff frequency at $\omega_0 = \pi/h$.

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:

$$\begin{aligned} \boldsymbol{s}_h(t) &= (s_x(t), s_y(t)) \\ &= \sum_{k=Z} \boldsymbol{c}(k) \cdot \beta^n \left(\frac{t}{h} - k \right) \\ & \qquad 0 \leq t \leq t_{\max} = hN) \end{aligned} \tag{12}$$

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 2-D spline snake is characterized by its vector-sequence of B-spline coefficients $\boldsymbol{c}(k) = (c_x(k), c_y(k))$. 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 (cf. Section III-E).

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 $M = t_{\max}$ 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.

B. B-Spline Snake Formulation: Node Points versus Control Points

In the literature, B-splines are defined by their control points [3], which correspond to the B-spline coefficients in (6). Previous B-spline snake formulations have used these control points for the definition of the curve [18], [27], [48]. In this paper, we prefer instead to work with node points, at least for the interactive part of the process. The node points are part of the snake and correspond to the knots of the spline curve. By positioning them appropriately, we have a very direct and intuitive way of controlling the shape of the curve. For B-splines of degree zero and one, the control points are identical to the curve points at that location. For higher degree splines, however, the control points are significantly distinct from the actual coordinates of the spline curve, especially for large values of h. An illustration is given in Fig. 1 for a B-spline curve and four node points using a B-spline of degree one and a B-spline of degree three. In the latter case, the control points do not belong to the curve and they are different from the node points. Clearly, for higher degree spline, user-interactivity is easier when the node points can be manipulated.

The node points are just another equivalent representation of the spline curve; they are directly related to the control points via a system of linear equations [16]. We will briefly show below how this system can be solved very efficiently using digital filtering techniques.

We impose a set of N discrete node points $\boldsymbol{n}(k) = (n_x(k), n_y(k))$, $0 \leq k < N$ that are required to be part of the curve $\boldsymbol{s}_h(t)$. Hence

$$\boldsymbol{n}(k) := \boldsymbol{s}_h(hk) = \sum_{j \in Z} \boldsymbol{c}(j) \cdot b^n(k - j), \qquad 0 \leq k < N \tag{13}$$

where b^n is the discrete B-spline kernel of degree n: $b^n(k) = \beta^n(x)|_{x=k}$. From (12) one can then compute the B-spline coefficients $\boldsymbol{c}(k)$ by inverse filtering

$$\boldsymbol{c}(k) = \sum_{j \in Z} \boldsymbol{n}(j) \cdot (b^n)^{-1}(k - j), \qquad 0 \leq k < N \tag{14}$$

where $(b^n)^{-1}$ denotes the sequence that is the convolution inverse of b^n. This is an IIR filter that can be efficiently implemented using recursive-filtering [44]. This step represents an important difference to other B-snake propositions. It has to be

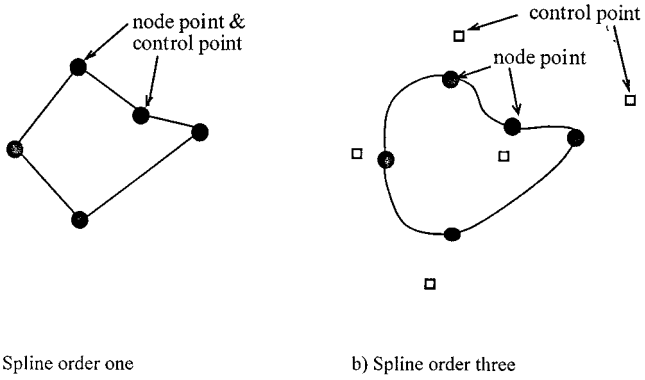

Fig. 1. Illustration of a B-spline snake curve, its node points (triangles) and its control points (squares). (a) B-spline of order zero (node points are equivalent to control points). (b) B-spline of order three. Here, control and node points differ significantly. User-interactivity is clearly improved if one uses node-points rather than control points.

evaluated once at the beginning of the snake optimization procedure and it provides the correct initialization for the B-spline coefficients. A correct initialization is mandatory such that the node points coincide with the actual curve.

Equation (14) provides us with N distinct coefficients. For the evaluation of (12)–(14), the finite sequence needs to be extended to infinite length. This is achieved through appropriate definition of boundary conditions and will be discussed in Section III-E.

C. Energy Formulation

With the introduction of variable knot spacing, we no longer require internal curve energies. Experimental tests given in Section V-A will demonstrate this point. The external potential function is typically given by a smoothed version of the gradient of the input data [8]

$$g(x, y) = \sqrt{\left(\frac{\partial}{\partial x}\,\varphi * f\right)^2 + \left(\frac{\partial}{\partial y}\,\varphi * f\right)^2} \qquad (15)$$

where f denotes the input image and φ is a smoothing kernel; for example, a Gaussian. Our cost function is the summation of the gradient (external force) over the path of the curve $\boldsymbol{s}(x)$ sampled at M consecutive points

$$\xi(\boldsymbol{c}(k)) = \sum_{i=0}^{M-1} -g(\boldsymbol{s}(i)). \qquad (16)$$

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 (16) is used because we employ a minimization technique for the optimization.

D. Fast Curve Rendering Through Digital Filtering

The problem consists in evaluating (12) at M discrete points. Such an evaluation is necessary for the computation of the energy function (16) and for the display of the curve (where M may typically be chosen larger). Therefore, the continuous variable t is replaced by a discrete variable i, $0 \le i < M$. The value of M and the number N of given node points directly determines the knot spacing h. The discrete B-spline snake with N node points and M curve points is given as

$$\boldsymbol{s}(i) = \sum_{k \in Z} \boldsymbol{c}(k) \cdot \beta^n\left(\frac{i}{h} - k\right), \qquad h = \frac{M}{N}. \qquad (17)$$

Below, we present two different ways for fast curve rendering by digital filtering.

Interpolation: The most straightforward way is by interpolation. The B-spline function is evaluated at every position $(i/h - k)$ multiplied by the corresponding B-spline coefficient and summed. B-splines are of compact support, and therefore, the summing needs only to be carried out over a subset of all coefficients. To interpolate the curve at a point i, only the coefficients

$$\boldsymbol{c}(k);\ \left[\frac{i}{h} - \frac{n+1}{2}\right] \le k \le \left[\frac{i}{h} + \frac{n+1}{2}\right]$$

need to be included in the sum ($[\cdot]$ denotes integer truncation). The main computational drawback of this procedure is that the function (7) needs to be evaluated for each term in the sum.

Digital Filtering: The above described algorithm works for any combination of values of M and N. If we can impose M such that h is an integer value, a much more efficient algorithm can be described. In general, this requirement is easily met, since M is not critical and can be loosely chosen. The simplification is based on a convolution property for B-splines [43, Eq. (3.5)]. It states that any spline of degree n and knot spacing h (integer) can be represented as the convolution of $n+1$ moving average filters of size h followed by a spline of knot spacing one. Hence, the curve points can be obtained by three successive steps:

1) upsampling of the B-spline coefficients;
2) averaging by $(n+1)$ moving average filters of size h;
3) filtering by a unit B-spline kernel of degree n.

This algorithm can be implemented with as few as two multiplications and two additions per node point plus $(2n)$ adds per computed contour coordinate. Generally, it is faster than method one and also at least a factor of two better than the Oslo knot insertion algorithm commonly used in computer graphics [3].

E. Border Conditions

Appropriate boundary conditions are necessary for the computation of (12), (14), and (17) [6], [51]. In the following, we distinguish the cases of a closed snake and an open snake.

Closed Snake Curve: For a set of node points $\boldsymbol{n}(k)$, $k = 0, \cdots, N-1$, we require that $\boldsymbol{n}(N) = \boldsymbol{n}(0)$ and $\boldsymbol{n}(-1) = \boldsymbol{n}(N-1)$. The corresponding boundary conditions are periodic [Fig. 2(a)]. The extended signal $\boldsymbol{n}_s(k)$ of infinite length can be described by

$$\boldsymbol{n}_s(k) = \boldsymbol{n}(k \bmod N). \qquad (18)$$

Open Snake Curve: Different choices can be implemented for the open snake, such as mirror or anti-mirror boundary conditions [Fig. 2(b) and (c)]. In this application, the anti-mirror conditions with a pivot at the boundary value are the most suitable

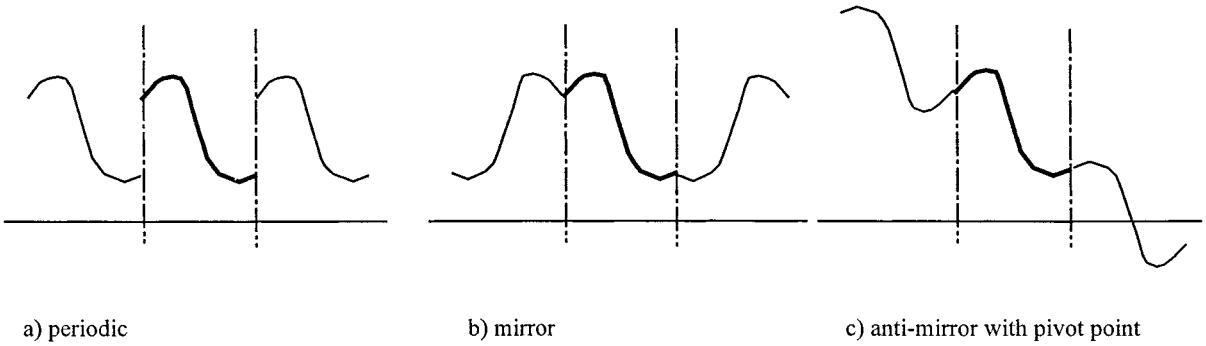

Fig. 2. Illustration of boundary conditions.

choice because they allow us to lock the end points of the curve. These anti-mirror conditions are such that

$$(\boldsymbol{n}(k_0+k)-\boldsymbol{n}(k_0))=(\boldsymbol{n}(k_0)-\boldsymbol{n}(k_0-k)) \qquad (22)$$

where $k_0 \in \{0,\, N-1\}$. Since the extended signal has a center of anti-symmetry at the boundary value, this value will be preserved exactly whenever the filter applied is symmetric, which turns out to be the case here. However, a new boundary value can not be defined as the lookup of an existing signal value, which makes the implementation slightly more complicated.

IV. Optimization Strategies

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 multidimensional 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

$$\begin{aligned}\frac{\partial \xi(\boldsymbol{c}(k))}{\partial \boldsymbol{c}(k)} &= \sum_{i=0}^{M-1} \frac{\partial(\boldsymbol{g}(s(i))}{\partial \boldsymbol{c}(k)} \\ \frac{\partial(\boldsymbol{g}(s(i))}{\partial \boldsymbol{c}(k)} &= \left.\frac{\partial(\boldsymbol{g}(s(i))}{\partial \boldsymbol{s}(i)}\right|_{\boldsymbol{s}=\boldsymbol{s}(\boldsymbol{c}(k))} \cdot \frac{\partial \boldsymbol{s}}{\partial \boldsymbol{c}(k)} \\ &= \left.\frac{\partial(\boldsymbol{g}(s(i))}{\partial \boldsymbol{s}(i)}\right|_{\boldsymbol{s}=\boldsymbol{s}(\boldsymbol{c}(k))} \cdot \beta^n\left(\frac{i}{h}-k\right). \end{aligned} \qquad (23)$$

The term $(\partial(\boldsymbol{g}(s(i))/\partial \boldsymbol{s}(i))|_{\boldsymbol{s}=\boldsymbol{s}(\boldsymbol{c}(k))}$ 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.

We have tested both the steepest descent algorithm and the conjugate gradient algorithm. The former optimizes in the direction of the local downhill gradient, which may result in an optimization procedure along the same direction many times, which is inefficient. The conjugate gradient algorithm performs the optimization along *A-orthogonal* directions. In the case of a quadratic potential, the procedure leads to a scheme where exactly one step is done in every search direction, which is much more efficient than the steepest descent approach. In our case, we make use of an extensions of the method to other, non-quadratic potential functions [30].

A. Multiresolution 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 multiscale representation. Multiscale processing is an old but powerful idea [7], [33]. 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 multiresolution 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. To minimize the loss of information from one level to the next we used a centered least squares pyramid as described in [5].

Hence, the original set of node points $\boldsymbol{n}(k)$ 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 multiresolution approach is given in Fig. 3. In Fig. 3(a), the initial starting contour is shown, which was drawn manually. The result of the direct optimization is shown in Fig. 3(c). Then, a three-level pyramid was computed using cubic spline basis functions. The optimized contours for the different levels are shown in Fig. 3(b). Clearly,

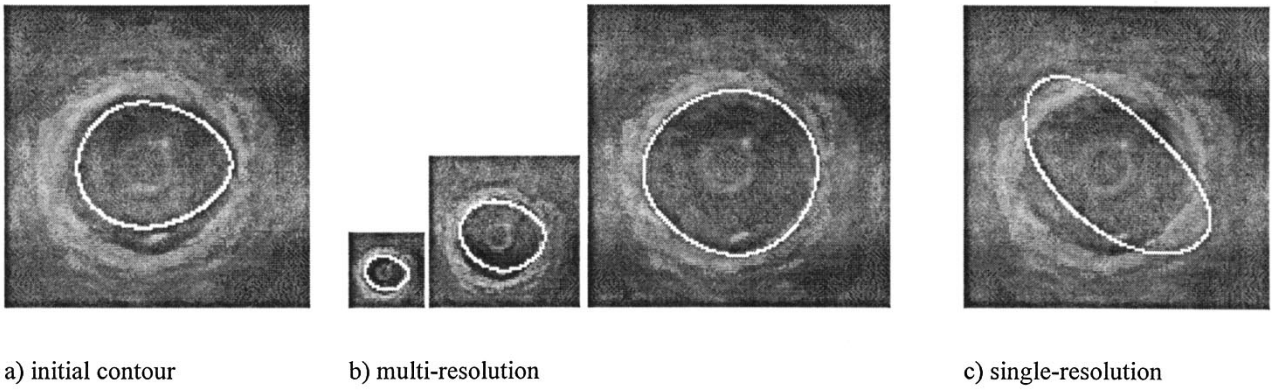

Fig. 3. Identical initial contour (a) for both (b) and (c), but only correct convergence with the multiresolution approach.

the multiresolution decomposition eliminates small (undesired) image details and the snake is able to converge to the true result.

V. Performance Evaluation

A. Comparison: Variable Knot Spacing vs. Internal Energies

In this section, we shall demonstrate the effect of a variable knot spacing on the stiffness of the curve, and its ability to conform to smooth image features. In particular, we will compare the proposed snake implementation to the traditional snake model by Kass, and show that similar results can be obtained without the use of internal energies.

The comparison is based on a binary test image consisting of a vertical line, of which a small part has been displaced to the left (see Fig. 4, initial contour). In order to obtain a smooth force function, the binary image is smoothed by a 2-D Gaussian, with $\sigma = 5$. Optimization is formulated as a minimization problem, and hence the optimal snake position is on the line. Eleven node points that have been set manually at unequal length intervals characterize the initial snake curve. Depending on the smoothness requirements of the final curve, two different results can be anticipated from the optimization. 1) The resulting curve is vertically centered on the longer line, being unaffected by the small displaced part. Such an outcome corresponds to an important smoothing constraint. 2) The resulting curve has a "bump" and is attracted toward the small displacement on the left. This outcome reflects a less severe smoothness requirement.

First, the traditional snake is computed with various weights for the internal energies. Each discrete curve point is independently optimized and attracted to the closest minimum by setting weights for the stretching and bending energy to zero [Fig. 4(a)]. A weight of $\alpha_{stretch} = 0.1$ and $\alpha_{bend} = 0.1$ tends to pull the "bump" toward the right [Fig. 4(b); however, this does not produce a straight curve yet. A weight of $\alpha_{stretch} = 0.2$ and $\alpha_{bend} = 0.2$ produces an almost flat curve [Fig. 4(c)]. This type of snake proves to be very flexible in that the user can choose among a large number of smoothness requirements by adjusting $\alpha_{stretch}$ and α_{bend}. The feature may also represent a drawback for certain applications, because of the associated difficulties in choosing the correct weighting factors, by either empirical of automatic means.

The B-spline snake incorporates smoothness through different knot spacings. The knot spacing, $h = M/N$, where M is the number of interpolated points and N is the number of node points, can be changed by either varying M or N. For this example, we have decided to employ the same number of node (control) points as for the traditional snake, and hence h is changed by changing the number of interpolated points M. A knot spacing of $h = 1$ signifies that no points are interpolated between node points. A B-spline of degree one corresponds exactly to the above experiment with zero weights, and the result is identical [Fig. 5(a)]. Using a B-spline of higher degree, node points are no longer completely independent. For all remaining experiments, we have used a B-spline of degree three, because it leads to visually pleasant curve representations. The optimized result with $h = 2$ is shown in Fig. 5(b). Note that there is one interpolated point between two node points. The point helps attracting the curve toward the longer line. An increased knot-spacing with $h = 3$ uses two interpolated values between two node points for computation of energy [Fig. 5(c)]. In this configuration, these points manage to fully attract the curve toward the longer line, and the "bump" disappears. The experiment demonstrates the similar effect of a variable knot spacing and of internal energies on the smoothness of the final snake curve.

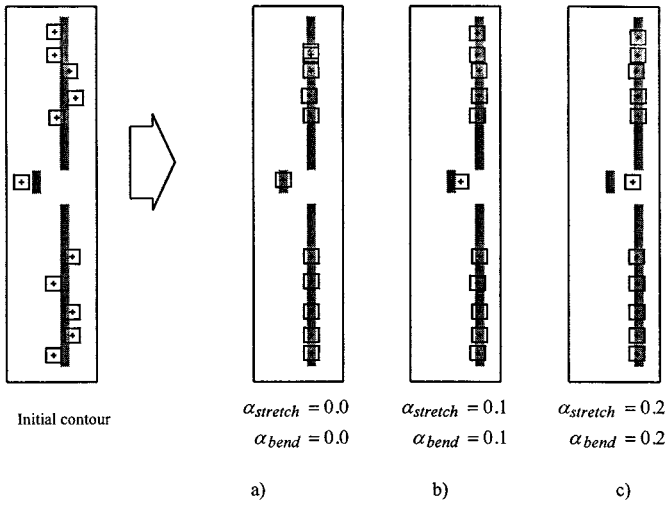

Fig. 4. Traditional snake: initial contour and optimization with different internal energies.

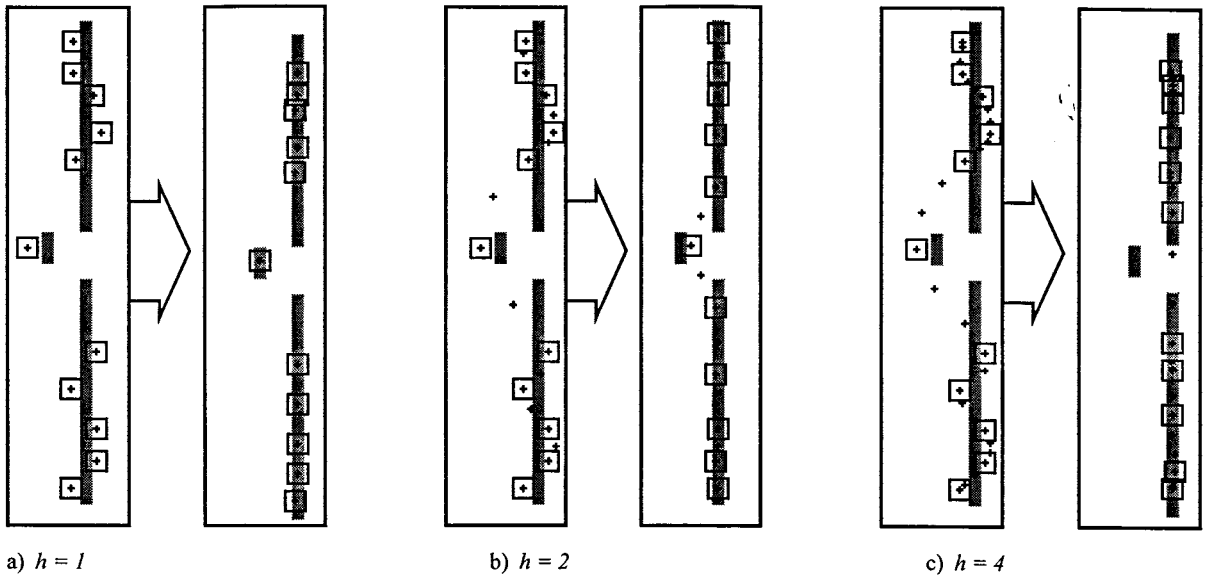

Fig. 5. B-spline snake: initial contour and optimization with different knot spacings.

B. Consistency Assessment

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, 20.2 node points were used ($n = 5$, number of node points for each experiment: 17, 21, 23, 19, 21).

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. Different number of node points were used for the manual placement of the initial curve, and the results are also an indication of the sensitivity of the method to manual placement of the curve, and to the number of node points chosen. Results are presented in Fig. 6 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. 7). 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. We therefore also note that the locations of the initial curve as well as the number of node points are not critical for reproducible optimization results. It

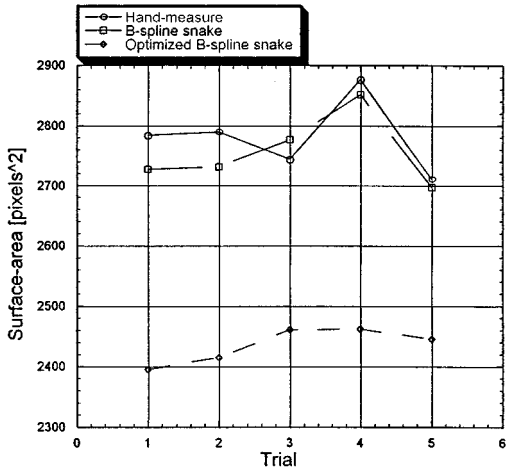

Fig. 6. Comparison of surface areas obtained by 1) manual tracing, 2) B-spline outlining, and 3) B-spline snake with optimization

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.

C. Stability Assessment

The test image consists of an arbitrary curve with a Gaussian shaped gray level profile. We illustrate the algorithm's performance in the presence of partially distorted contours. A gap of increasing dimension was inserted in the test image (Fig. 8) to illustrate the algorithm's interpolation efficacy.

The results of these tests are illustrated in Fig. 8. Fig. 8(a) displays the initial contour, and Fig. 8(b)–(e) show the optimization result when the contours are partially distorted. The B-spline snake tries to follow the contour piece that is being extracted. At

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

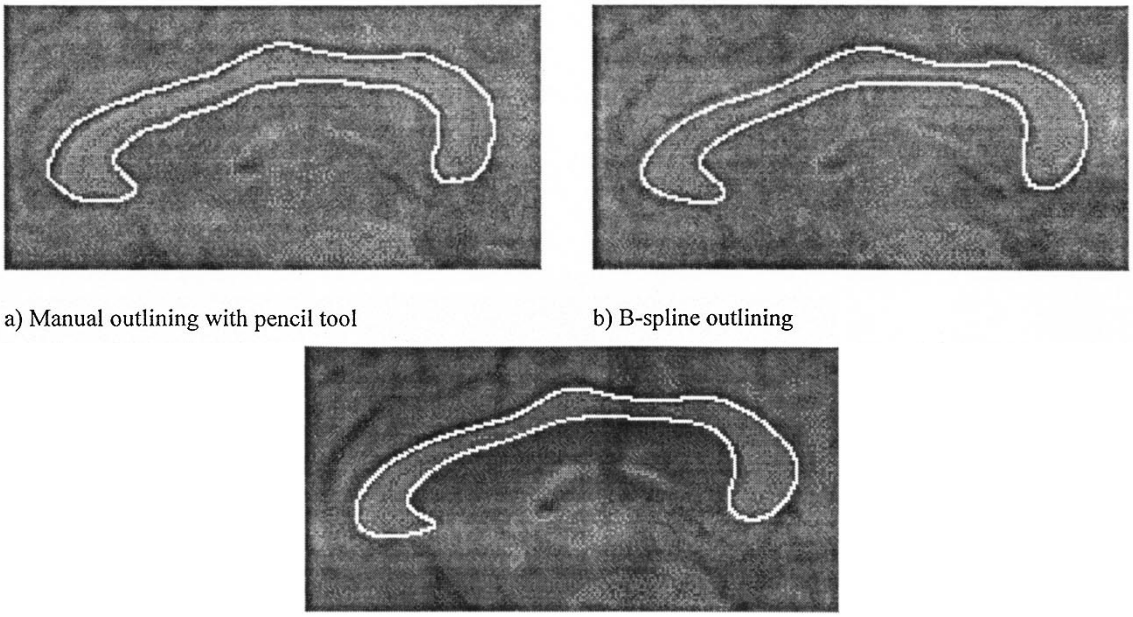

Fig. 7. Comparison of different contour outlining techniques.

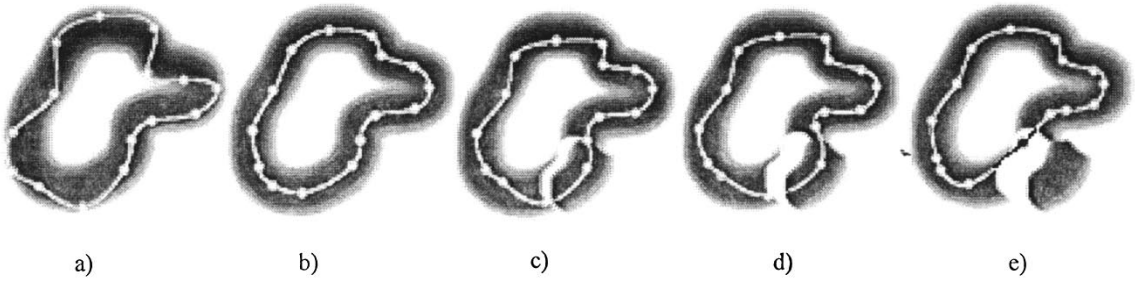

Fig. 8. (a) Initial contour and (b)–(e) optimization with an emerging contour piece at increasing distances. The dots indicate the location of the node points.

a sufficient distance of the extracted contour piece to the original contour position, the snake then snaps back to interpolate the missing contour.

The proposed algorithm is also stable in noisy imaging environments. We have employed the method extensively on a quantification problem for plaque characterization in coronary ultrasound images. To test the algorithm's accuracy, coronary arteries were imaged post-mortem. The same arteries underwent automatic analysis by our algorithm, as well as manual inspection by dissectioning. The automatically obtained results have been compared to those obtained by histological dissectioning. A total of $n = 189$ sections from 20 different segments has been compared, and those results are summarized in Fig. 9. There is an excellent linear correlation between the two measures.

VI. BIO-MEDICAL APPLICATIONS

A. 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 as-

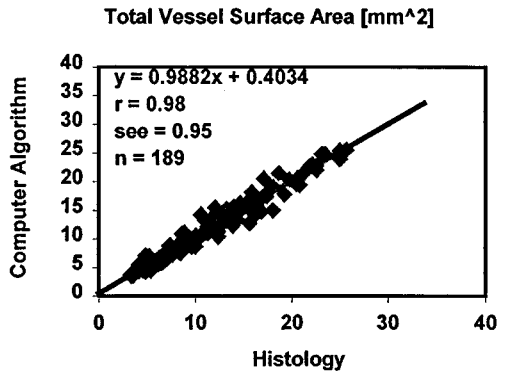

Fig. 9. Comparison of automatically obtained surface areas to those obtained by dissectioning. The data is from ultrasound images of left coronary arteries.

sessed 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. 7). 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 [15]. We would also like to mention the work performed in [12], that uses B-spline for contour description in the context of an identification and registration procedure of neuroanatomical atlases.

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.

B. 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. 10(a). Here, researchers 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. 10(a)] by a contrast operator highlighted contours of interest [Fig. 10(d)]. Manual placement of the initial contour in the first frame of the sequence using four node points [Fig. 10(b)] 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. 10(c)]. This contour was optimized using the conjugate gradient algorithm and a multiresolution decomposition of two levels. The optimized contour [Fig. 10(e)] 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. 10(f).

C. 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. 4. 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.

The above experiments show the good performance of the proposed method in various, different imaging environments. Also, they illustrate the robustness of the method with respect to different input noise levels and characteristics.

VII. Discussion

Improvements of the traditional snake have made it a very useful tool in many applications. 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. Greedy algorithms and some derived versions [22], [49] have been described to provide improved convergence speed. A way to overcome the difficulties of the proper choosing of the weight parameters has been proposed in [50]. Here, the internal forces are controlled by approximating the contour by curvature arcs that compensate for the normal forces. The B-spline snakes build on the initial concept, but rely on a somewhat modified underlying philosophy. When first published by Menet *et al.* [27], the authors emphasized 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 [27],

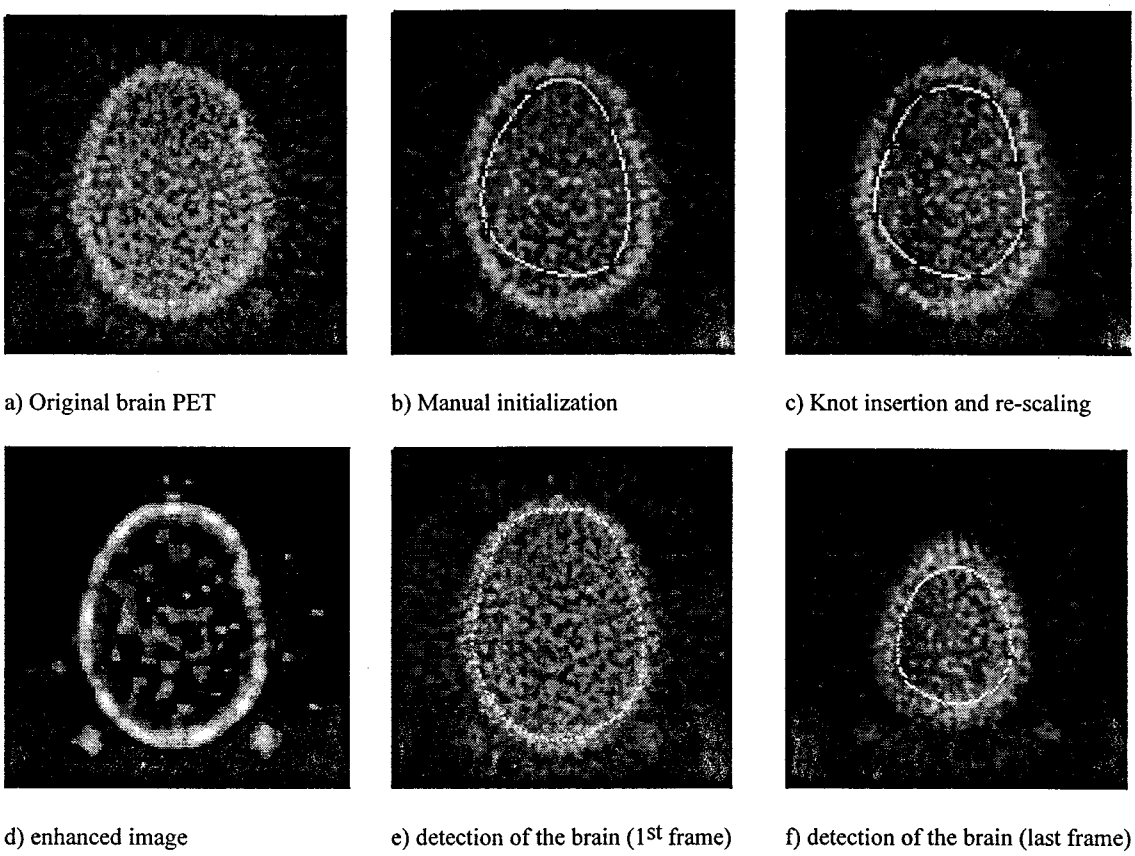

Fig. 10. PET-scan of the brain.

Flickner *et al.* [18] 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 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 [48], who proposed a multistage model for the optimization. One notices the link to another technique for smooth boundary representation: boundary description by Fourier descriptors [28], [29], [39], [41]. Similarly, a reduced number of parameters combined with an implicit smooth data model provides a robust description of a contour. With such an approach, a small perturbation Δc is uniformly distributed over the entire curve. Thus, the approach is more stable with respect to noisy objective functions. Also, the B-spline snake formulation is related to the wavelet approach described by Chuang and Kuo [10]. This representation leads to a multiscale curve description, and the authors demonstrate among others the nonshrinking property when only a few wavelet coefficients are used, and the stability of the representation.

In this work, we have provided some fundamental justification for the use of B-splines for solving snakes problems. We started by proving that cubic splines are optimal for cost functions involving a smoothness constraint based on the second derivative of the curve. Hence, by considering a simplified version of the cost function introduced by Kass *et al.* (which also involves a first order derivative term), we were able to give a rigorous justification for the B-spline discretization of the snake problem as initially proposed by Menet *et al.* While this gives a strong argument in favor of B-spline snakes, the main idea that we have pushed here is the possibility of simplifying the traditional snake formulation by both reducing the number of parameters and dropping the regularization term (internal energies). We have achieved this by introducing a global scaling parameter h, which controls the spacing between the spline nodes.

The error formula (11) is a standard result in approximation theory (Strang–Fix conditions) that also applies to finite element models [40]. This formula is especially important in our context because it provides a quantification of the implicit smoothing effect that can be achieved through the adjustment of h—the idea that we are promoting here. The problematic of the standard snake formulation as well as Cohen and Cohen's finite element approach [13], [14] is quite different: one seeks a solution of a differential equation and one wishes to chose a discretization that is sufficiently fine so that the error between the computed and true solution is negligible. Here we don't want to track the maximum of our unregularized external forces exactly; rather, we want to filter it to ensure that the solution is sufficiently smooth. In other words, we need to work with a rather large h, while the finite element method ideally calls for a small h, the smoothing being controlled through the coefficients of the differential equation. The advantages of such a parametric formulation are two-fold. First, it simplifies the model because of the reduction in the number of parameters (use of fewer, but broader basis functions). Second, the use of a variable spacing

provides a very natural and intuitive way to control the stiffness of the curve. A larger spacing implies a smoother curve; hence the reasoning that internal energies become superfluous. We have tried to experimentally demonstrate the point by comparing the traditional snake with internal energies to B-spline snakes without internal energies, showing that it essentially produces the same effect.

Choosing weights for internal energies may sometimes be quite an impossible undertaking. The proposed approach, however, has the great advantage of being user-friendlier, because only one parameter, the value of N, has to be determined, which is most intuitive. The node placement is straightforward by simple point and click. In the case of an automatic initialization, one has to determine *a priori* how many node points are required for a sufficiently accurate curve representation. Also, automatic optimization may move node points to high curvature features, thereby locally adapting the smoothness ability of the curve. An approximate estimate of a first contour can be easily resampled at a finer knot spacing h, which is then subject to optimization. The proposed approach is based on optimization of a small number of node points and is therefore computationally efficient. Moreover, the convenient form of a B-spline curve represents a coded version of the detected contour without redundancy. Our method does not rely on derivative operators of discrete variables, which makes it robust with respect to noise. We have introduced the idea of a multiresolution approach in the snake optimization process, which further enhances algorithm stability. Experimental results have shown that a correct convergence is possible from initial contours, which otherwise lead to wrong optimization results. Hence, accuracy and closeness of the initial contour to the desired contour can be further relaxed. Finally, note that the B-spline snake is a parametric curve. Tasks of computing surface-areas of a closed curve are thereby significantly simplified. For instance, algorithms from computational geometry can be applied that compute surface-area by subdividing the area into triangles, of which surface areas are summed. Accuracy can be chosen as a convenient function of the scale parameter h of the B-spline snake curve. In our work, we require N (i.e., the number of node points) to be given as input. Minimum mean-square estimation techniques or other measure could be introduced to automatically determine this number [11].

Finally, we have presented semiautomatic optimization schemes for various biomedical applications. For image sequences, the user provides the initial contour for the first frame only, and this information is then used for the segmentation of the entire image sequence by forward propagation. Note that the proposed B-spline snake approach potentially offers the possibility of a true 3-D analysis, by introducing *node frames*, which are linked by cubic spline functions.

VIII. Conclusions

In this paper, a B-spline snake formulation has been presented based on the use of node points, a variable knot spacing, and the use of a multiresolution optimization strategy. The presented algorithms are characterized by fast execution speeds and the possibility for an intuitive user-interaction. Speed has been obtained essentially by two means. First, we have considerably reduced the number of free parameters of the snake curve. It not only simplifies the problem, but also makes it possible to apply some of the more powerful optimization techniques (e.g., conjugate gradient as opposed to steepest descent). Second, we have introduced fast digital filtering techniques for the rendering of the snake curve. Intuitive user-interaction is given by the fact that the node points are situated exactly on the curve. We have demonstrated that it is no longer necessary to include curve related energy terms for the optimization of the snake, such as first and second order derivatives.

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 resolutions. We believe that the proposed technique could be a valuable tool for the outlining of image contours in bio-medical applications.

In all of the applications presented, the user had to specify an initial contour. For a specific application, it may be advantageous to perform dedicated pre-processing so as to obtain an automatic initialization.

Acknowledgment

The authors would like to thank A. Aldroubi and T. Blu for their numerous inputs with respect to mathematically related problems, M. Eden for his valuable discussions, and P. Thévenaz for his suggestions for algorithm testing and help on B-spline implementation.

References

[1] J. H. Ahlberg, E. N. Nilson, and J. L. Wash, *The Theory of Splines and Their Applications*. New York: Academic, 1967.

[2] A. Amini, T. Weymouth, and R. Jain, "Using dynamic programming for solving variational problems in vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 855–867, Sept. 1990.

[3] R. Bartels, J. Beatty, and B. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*. Los Altos, CA: Morgan Kaufmann, 1987.

[4] P. Brigger, S. L. Bacharach, G. Srinivasan, K. A. Nour, J. Carson, V. Dilsizian, A. Aldroubi, and M. Unser, "Segmentation of gated Tl-201-SPECT images and computation of ejection fraction: A different approach," *J. Nucl. Cardiol.*, vol. 6, no. 3, pp. 286–297, 1999.

[5] P. Brigger, F. Mueller, K. Illgner, and M. Unser, "Centered pyramids," *IEEE Trans. Image Processing*, vol. 6, no. 9, pp. 1254–1264, 1999.

[6] C. M. Brislawn, "Preservation of subband symmetry in multirate signal coding," *IEEE Trans. Signal Processing*, vol. 43, pp. 3046–3050, Dec. 1995.

[7] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact code," *IEEE Trans. Commun.*, vol. COM-31, pp. 337–345, Apr. 1983.

[8] J. F. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 679–697, 1986.

[9] V. Caselles, F. Catte, T. Coll, and F. Dibos, "A geometric model for active contours in image processing," *Num. Math.*, vol. 66, pp. 1–31, 1993.

[10] G. Chuang and C. Kuo, "Wavelet descriptor of planar curves: Theory and applications," *IEEE Trans. Image Processing*, vol. 5, pp. 56–70, Jan. 1996.

[11] F. S. Cohen and W. Jin-Yinn, "Part I: Modeling image curves using invariant 3-D object curve models—A path to 3-D recognition and shape estimation from image contours," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 1–12, Jan. 1994.

[12] F. S. Cohen, Y. Zhengwei, Z. Huang, and J. Nianov, "Automatic matching of homologous hisological sections," *IEEE Trans. Biomed. Eng.*, vol. 45, no. 5, pp. 642–649, 1998.

[13] L. D. Cohen and I. Cohen, "A finite element method applied to new acitve contour models and 3D reconstruction from cross sections," in *Proc. 3rd Int. Conf. Computer Vision*, Los Alamitos, CA, 1990, pp. 587–591.

[14] ——, "Finite-element methods for active contour models and balloons for 2-D and 3-D images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 1131–1147, Nov. 1993.
[15] 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. Assist. Tomogr.*, vol. 20, no. 1, pp. 88–97, 1996.
[16] C. deBoor, *A Practical Guide to Splines*. New York: Springer-Verlag, 1978.
[17] H. Eviatar and R. Somorjai, "A fast, simple active contour algorithm for biomedical images," *Pattern Recognit. Lett.*, vol. 17, pp. 969–974, 1996.
[18] M. Flickner, H. Sawhney, D. Pryor, and J. Lotspiech, "Intelligent interactive image outlining using spline snakes," in *28th Asilomar Conf. Signals, Systems, Computers*, vol. 1, 1994, pp. 731–735.
[19] M. Hoch and P. Litwinowicz, "A semi-automatic system for edge tracking with snakes," *Vis. Comput.*, vol. 12, pp. 75–83, 1996.
[20] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, pp. 321–331, 1988.
[21] A. Klein, T. K. Egglin, J. S. Pollak, F. Lee, and A. Amini, "Identifying vascular features with orientation specific filters and B-spline snakes," *Comput. Cardiol.*, pp. 113–116, 1994.
[22] K. Lam and H. Yan, "Fast greedy algorithm for active contours," *Electron. Lett.*, vol. 30, no. 1, pp. 21–23, 1994.
[23] C. W. Liao and G. Medioni, "Surface approximation of a cloud of 3D points," *Graph. Models Image Process.*, vol. 57, no. 1, pp. 67–74, 1995.
[24] W. A. Light, "Recent developments in the Strang-Fix theory for approximation orders," in *Curves and Surfaces*, P. J. Laurent, A. L. Méhauté, and L. L. Schumaker, Eds. New York: Academic, 1991, pp. 285–292.
[25] R. Malladi, J. Sethian, and B. Vemuri, "Shape modeling with fron propagation: A level set approach," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 158–175, 1995.
[26] T. McInerney and D. Terzopoulos, "Topologically adaptable snakes," in *5th Int. Conf. Computer Vision*, 1995, pp. 840–845.
[27] S. Menet, P. Saint-Marc, and G. Medioni, "B-snakes: Implementation and application to stereo," in *Image Understanding Workshop*, Sept. 1990, pp. 720–726.
[28] E. Persoon and K. Fu, "Shape discrimination using Fourier descriptors," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, pp. 170–179, Mar. 1977.
[29] B. Pinkowski, "Multiscale Fourier descriptors for classifying semivowels in spectrograms," *Pattern Recognit.*, vol. 26, no. 10, pp. 1593–1602, 1993.
[30] E. Polak, *Computational Methods in Optimization*. New York: Academic, 1971.
[31] S. Ranganath, "Contour extraction from cardiac MRI studies using snakes," *IEEE Trans. Med. Imag.*, vol. 14, pp. 328–338, June 1995.
[32] C. H. Reinsh, "Smoothing by spline functions," *Numer. Math.*, vol. 10, pp. 177–183, 1967.
[33] A. Rosenfeld, *Multiresolution Image Processing*. New York: Springer-Verlag, 1984.
[34] I. J. Schoenberg, "Contribution to the problem of approximation of equidistant data by analytic functions," *Quart. Appl. Math.*, vol. 4, pp. 112–141, 1946.
[35] ——, "Spline functions and the problem of graduation," *Proc. Nat. Acad. Sci.*, vol. 52, pp. 947–950, 1964.
[36] ——, *Cardinal Spline Interpolation*. Philadelphia, PA: SIAM, 1973.
[37] L. L. Schumaker, *Spline Functions: Basic Theory*. New York: Wiley, 1981.
[38] J. A. Sethian, *Level Set Methods*. Cambridge, U.K.: Cambridge Univ. Press, 1996.
[39] L. Staib, "Boundary finding with parametrically deformable models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 1061–1075, Nov. 1992.
[40] G. Strang and G. Fix, *A Fourier Analysis of Finite Element Variational Method*. Rome, Italy: Edizioni Cremonese, 1971, pp. 796–830.
[41] R. Tello, "Fourier descriptors for computer graphics," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, pp. 861–865, May 1995.
[42] M. Unser, A. Aldroubi, and M. Eden, "Polynomial spline signal approximations: Filter design and asymptotic equivalence with Shannon's sampling theorem," *IEEE Trans. Inform. Theory*, vol. 38, pp. 95–103, Jan. 1992.
[43] ——, "B-spline signal processing. Part I: Theory," *IEEE Trans. Signal Processing*, vol. 41, pp. 821–833, Feb. 1993.
[44] ——, "B-spline signal processing. Part II: Efficient design and applications," *IEEE Trans. Signal Processing*, vol. 41, pp. 834–848, Feb. 1993.
[45] M. Unser and I. Daubechies, "On the approximation power of convolution-based least squares versus interpolation," *IEEE Trans. Signal Processing*, vol. 45, pp. 1697–1711, July 1997.
[46] G. Wahba, *Spline Models for Observational Data*. Philadelphia, PA: SIAM, 1990.
[47] J. Wang and F. Cohen, "Part II: 3-D object recognition and shape estimation from image contours using B-splines, shape invariant matching, and neural networks," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 13–23, Jan. 1994.
[48] M. Wang, J. Evans, L. Hassebrook, and C. Knapp, "A multistage, optimal active contour model," *IEEE Trans. Image Processing*, vol. 5, pp. 1586–1591, Nov. 1996.
[49] D. Williams and M. Shah, "A fast algorithm for active contours and curvature estimation," *Comput. Vis. Graph. Image Process.*, vol. 55, no. 1, pp. 14–26, 1992.
[50] G. Xu, E. Segawa, and S. Tsuji, "Robust active contours with insensitive parameters," *Pattern Recognit.*, vol. 27, no. 7, pp. 879–884, 1994.
[51] F. Yamaguchi, *Curves and Surfaces in Computer Aided Geometric Design*. Berlin, Germany: Springer-Verlag, 1988.

Patrick Brigger (S'92–A'96) was born in Luzern, Switzerland, on May 5, 1966. He received the M.S. and Ph.D. degrees in electrical engineering from the Swiss Federal Institute of Technology in Lausanne, Switzerland, in 1992 and 1995, respectively.

From 1996 to 1998, he was with the Biomedical Engineering and Instrumentation Program, National Institutes of Health, Bethesda, MD, where guided several projects related to biomedical image processing. In 1998, he became Principal Investigator for a Small Business Innovation Research Grant, and was co-founder and partner of ImageMinds, an imaging and Internet technology company. In 1999, he co-founded getAbstract.com, an Internet-based content provider in the field of compressed knowledge (http://www.getabstract.com). He is currently the CTO of the company and directs all technical related activities. His main fields of interest are emerging and new internet and e-commerce technologies.

Jeff Hoeg, photograph and biography not available at time of publication.

Michael Unser (M'88–SM'94–F'99) was born in Zug, Switzerland, on April 9, 1958. He received the M.S. (summa cum laude) and Ph.D. degrees in electrical engineering from the Swiss Federal Institute of Technology, Lausanne, Switzerland, in 1981 and 1984, respectively.

From 1985 to 1997, he was with the Biomedical Engineering and Instrumentation Program, National Institutes of Health, Bethesda, MD, where he was heading the Image Processing Group. He is now Professor and Head of the Biomedical Imaging Group at the Swiss Federal Institute of Technology, Lausanne. His main research area is biomedical image processing. He has a strong interest in sampling theories, multiresolution algorithms, wavelets, and the use of splines for image processing. He is the author of more than 70 published journal papers in these areas. He is on the editorial boards of *Signal Processing*, the *Journal of Visual Communication and Image Representation*, and *Pattern Recognition*.

Dr. Unser is a member of the Image and Multidimensional Signal Processing Committee of the IEEE Signal Processing Society. He is a former Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING (1992–1995) and the IEEE SIGNAL PROCESSING LETTERS (1994–1998). He co-organized the 1994 IEEE-EMBS Workshop on Wavelets in Medicine and Biology, and serves as conference chair for SPIE's Wavelet Applications in Signal and Image Processing, which has been held annually since 1993. He received the Dommer Prize for Excellence from the Swiss Federal Institute of Technology in 1981, the research prize of the Brown-Boveri Corporation, Switzerland, for his thesis in 1984, and the IEEE Signal Processing Society's 1995 Best Paper Award (IMDSP technical area) for a paper with A. Aldroubi and M. Eden on B-spline signal processing.