Unsupervised Contour Representation and Estimation Using B-Splines and a Minimum Description Length Criterion

Mário A. T. Figueiredo, *Member, IEEE*, José M. N. Leitão, *Member, IEEE*, and Anil K. Jain, *Fellow, IEEE*

Abstract—This paper describes a new approach to adaptive estimation of parametric deformable contours based on B-spline representations. The problem is formulated in a statistical framework with the likelihood function being derived from a region-based image model. The parameters of the image model, the contour parameters, and the B-spline parameterization order (i.e., the number of control points) are all considered unknown. The parameterization order is estimated via a minimum description length (MDL) type criterion. A deterministic iterative algorithm is developed to implement the derived contour estimation criterion. The result is an unsupervised parametric deformable contour: it adapts its degree of smoothness/complexity (number of control points) and it also estimates the observation (image) model parameters. The experiments reported in the paper, performed on synthetic and real (medical) images, confirm the adequacy and good performance of the approach.

Index Terms—B-splines, counter estimation, deformable contours, image segmentation, minimum description length, snakes.

I. INTRODUCTION

IMAGE segmentation and contour estimation are among the most challenging, important, and frequently addressed fundamental problems in image analysis. When no assumptions are made about the morphology of the objects/regions to be estimated, we have an *image segmentation* problem, in the common usage of the term. When the problem is more confined to that of finding some individual image region, it is commonly referred to as *contour estimation*; a typical example is organ boundary location in medical images.

A. Snakes and Related Approaches

Rooted in the seminal work [30], *snakes* (or *active contours*) and their descendants constitute the most often used class of approaches to smooth boundary estimation. As originally proposed [30], a snake is a virtual object $\mathbf{v}$ (living on the image plane) which can deform elastically (thus possessing *internal energy*, $E_{\text{int}}(\mathbf{v})$) and which is immersed in a potential field (thus

having *external/potential* energy, $E_{\text{ext}}(\mathbf{v}, \mathbf{I})$) which is a function of certain features of the image $\mathbf{I}$. The equilibrium (minimal total energy) configuration

$$\mathbf{v}^* = \arg\min_{\mathbf{v}}\{E_{\text{int}}(\mathbf{v}) + E_{\text{ext}}(\mathbf{v}, \mathbf{I})\} \quad (1)$$

is a compromise between *smoothness* (enforced by the elastic nature of the model) and proximity to the desired image features (by action of the external potential).

Several drawbacks of conventional *snakes*, such as their "myopia" (i.e., use of image data strictly along the boundary), have stimulated a great amount of research; although most limitations of the original formulation have been successfully addressed (see, e.g., [6], [9], [10], [34], [38], [43], [49], and [52]), nonadaptiveness (in the sense that some or all parameters have to be set *a priori*) remains to be solved.

B. Deformable Templates/Models

In parametrically deformable models and templates, as used for contour estimation, the contour itself, or deformations applied to a given template, are parametrically described [25], [50]. Techniques used include Fourier descriptors [22], [28], [46], spline models [1], [31], [35], [44], wavelets [7], and polygons [29] (see a recent review in [27]). The parametric approach contrasts with the explicit, i.e., nonparametric, contour descriptions used in snakes. By using low-order parameterizations, the possible shapes may be implicitly placed under some regularity constraint; this allows dropping the (snake-type) internal energy term and formulating contour location as a parameter estimation problem. For example, a few low-order Fourier coefficients can only describe smooth curves [22], [46]. However, as in snakes, one of the main difficulties of these techniques is their lack of adaptiveness; namely, the order of the parameterization has to be specified *a priori*.

C. Bayesian Viewpoint

It is commonly stated that, from a Bayesian perspective, snakes are interpretable as *maximum a posteriori* (MAP) contour estimators, where the internal and external energies are associated with the *a priori* probability function (or *prior*) and the likelihood function (observation model), respectively [18], [47]. The same is true for deformable templates where the prior biases the estimate toward the template shape [25], [28], [36], [46]. As referred above, in deformable models it is possible not to include a deformation energy (i.e., a prior, from the Bayesian viewpoint) when the parameterization itself guarantees regularity/smoothness of the represented shape.

However, a *maximum likelihood* (ML) estimation interpretation is still valid, since this can be seen as a limiting case of the MAP criterion with a uniform prior on the set of valid parameter values.

Formally, let $\mathbf{v}$ be the contour to be estimated on the observed image $\mathbf{I}$ (a $w_x \times w_y$ array of gray levels). A Bayesian approach requires the following steps:

1) specification of a prior $p(\mathbf{v})$ capturing *a priori* information/constraints on $\mathbf{v}$;
2) derivation of a likelihood function $p(\mathbf{I}|\mathbf{v})$ modeling the observed image conditioned on the true contour;
3) specification of a *loss function* $L(\mathbf{v}, \mathbf{v}')$ measuring how much loss is incurred by an estimate $\mathbf{v}'$ when the true contour is in fact $\mathbf{v}$.

Once these elements are in place, an optimal Bayes rule is the function of the data (called an estimator, and denoted $\hat{\mathbf{v}}(\mathbf{I})$) that minimizes the *a posteriori* expected loss (see, e.g., [42])

$$\hat{\mathbf{v}}(\mathbf{I}) = \arg\min_{\mathbf{v}'} \int L(\mathbf{v}, \mathbf{v}') p(\mathbf{v}|\mathbf{I}) \, d\mathbf{v} \tag{2}$$

where $p(\mathbf{v}|\mathbf{I})$ is the *a posteriori* probability density function obtained via Bayes law $p(\mathbf{v}|\mathbf{I}) = p(\mathbf{I}|\mathbf{v})p(\mathbf{v})/p(\mathbf{I})$. A particular choice of loss function leads to the well known maximum *a posteriori* (MAP) rule $\hat{\mathbf{v}}_{\text{MAP}} = \arg\max_{\mathbf{v}} p(\mathbf{v}|\mathbf{I})$. If the prior and the likelihood are written as

$$p(\mathbf{v}) = \frac{1}{Z_{\text{int}}} e^{-E_{\text{int}}(\mathbf{v})} \tag{3}$$

$$p(\mathbf{I}|\mathbf{v}) = \frac{1}{Z_{\text{ext}}(\mathbf{v})} e^{-E_{\text{ext}}(\mathbf{v}, \mathbf{I})} \tag{4}$$

(where the Z's are normalizing constants) then, the MAP estimator can be written as (1)

$$\begin{aligned} \hat{\mathbf{v}}_{\text{MAP}} &= \arg\max_{\mathbf{v}} \{p(\mathbf{I}|\mathbf{v})p(\mathbf{v})\} \\ &= \arg\min_{\mathbf{v}} \{E_{\text{int}}(\mathbf{v}) + E_{\text{ext}}(\mathbf{v}, \mathbf{I})\} \end{aligned} \tag{5}$$

if and only if $Z_{\text{ext}}(\mathbf{v})$ is a constant. In other words, this simple equivalence between the energy-minimization formulation and a MAP criterion is only possible if the normalizing constant of the likelihood function does not depend on $\mathbf{v}$. This crucial condition is usually not mentioned.

The Bayesian interpretation gives meaning to all the involved entities, e.g., the form of the (external) energy term that links the contour with the image contents (i.e., the likelihood function, in Bayesian terms) can be derived from knowledge about the observation model rather than simply from common sense arguments [15], [18]. The main difficulty in this approach is still the choice of the parameters involved in the definition of the *a priori* probability function and of the observation model.

D. Optimization Problem

Regardless of their theoretical/conceptual interpretation, both classical *snake*-type approaches and deformable templates/models lead to difficult optimization problems for which several computational techniques have been proposed: deterministic iterative schemes [18], [30] (see [9], for a comprehensive and integrative review); dynamic programming [2],

[15], [19], [24]; multi-resolution algorithms [15], [24], [28]; and stochastic methods [3], [25], [29], [47].

E. Proposed Approach

In this paper, we propose an adaptive contour estimation strategy based on parametrically deformable models. Our main goal is an unsupervised technique which does not require any kind of parameter adjustment by the user.

Although other choices do fit well in our formulation (e.g., Fourier descriptors, see [22]) this paper considers only B-spline representations since they are paradigmatic of the proposed approach. The problem is formulated in a statistical estimation framework, with the contour parameters and the observation model parameters all being considered unknown. A key issue arising in parametric contour descriptions is the choice of the parameterization order, e.g., the number of spline control points, or the number of coefficients in Fourier descriptors. In less formal terms, this problem can be restated as: how smooth, or how complex, should the contour be? This model order selection problem is an instance of the underfitting/overfitting tradeoff, present in many pattern recognition and image analysis problems [39]; briefly, if the selected order is too low, the model will not be able to represent the underlying shape (underfitting), if the order is too high, the representation will fit irrelevant (noisy) features (overfitting). We address this problem by using a *minimum description length* (MDL) type criterion. MDL is a criterion due to Rissanen [40], based on coding theoretical considerations; although it was not conceived within a Bayesian framework, MDL can be interpreted as corresponding to the adoption of a certain prior [12]. Recently, MDL-type criteria have been successfully used for several problems in computer vision and image processing (see [20] and references therein).

Concerning the likelihood function (image model) from which the parameters (i.e., the contour) are to be estimated, we adopt a *region-based* approach [18], [43], [52]. This means that the likelihood function of the contour position depends on all the image data (split into inner and outer regions), not just on a narrow stripe along the contour (as in *snakes*). This approach is adequate to situations where gradients do not make sense (e.g., inner and outer regions with the same mean). Moreover, by using all the image, this approach is robust against any small local artifacts that, if having high gradients, can strongly attract classical snakes.

The resulting contour estimation criterion is implemented via an iterative deterministic scheme. User intervention is limited to providing an initial contour; as will be shown in the examples, the adopted region-based model is very robust with respect to the initialization which may even be far from the final estimate.

To our knowledge, this is the first/only fully unsupervised (with respect to parameter adjustments) formulation for the problem of deformable boundary location.

II. B-SPLINE CONTOUR DESCRIPTORS

Splines are a widely used function approximation tool [13], [16]. In particular, they have been used in computer graphics and, more recently, in computer vision and image analysis;

spline representations of curves/contours were addressed in [1], [5], [8], [9], [23], [35], and [45].

For the sake of self-containedness and clarity, we now present a brief review of splines and B-splines; for a more detailed account, see [13] and [16].

A. Splines, B-Splines, and Closed Curves

Let $\{t_0 < t_1 < \cdots < t_k\} \subset [t_0, t_k] \subset \mathbb{R}$ be the set of so-called *knots*.[1] By definition, spline functions are polynomial inside each interval $[t_{i-1}, t_i]$ and exhibit a certain degree (say $m-1$) of continuity at the knots. The set of all splines on $[t_m, t_{k-m}]$, which are C^{m-1} continuous at the knots is a linear space of dimension $(k-m)$. The set of so-called B-splines, denoted $\{\mathcal{B}_k^m(t), k = 0, \cdots, k-m-1\}$, constitute a basis (though a nonorthogonal one) for this linear space. Accordingly, each spline $f(t)$ in this space has a unique representation (where $\{c_0, c_1, \cdots, c_{k-m-1}\}$ is a set of coefficients/weights)

$$f(t) = \sum_{i=0}^{k-m-1} c_i \mathcal{B}_i^m(t), \qquad t \in [t_m, t_{k-m}]. \tag{6}$$

The B-splines are nonnegative, $\mathcal{B}_i^m(t) \geq 0$, and verify the so-called *partition of the unity* property

$$\sum_i \mathcal{B}_i^m(t) = 1, \quad \text{for } t \in [t_m, t_{k-m}]. \tag{7}$$

Planar curves are simply the $\mathbb{R}^2$ version of (6)

$$\mathbf{v}(t) \equiv [x(t) \quad y(t)] = \sum_{i=0}^{k-m-1} \mathbf{c}_i \mathcal{B}_i^m(t) \tag{8}$$

where the $\mathbf{c}_i \equiv [c_i^x \quad c_i^y]$ are now points in $\mathbb{R}^2$, called *control points*. To describe closed curves, the periodic extension of the knot sequence, $\{\tilde{t}_j, j \in \mathcal{Z}\}$ with $\tilde{t}_j = t_{j \mod k}$, is defined [23]. The basis functions also have to be periodic [23], which is achieved by defining the periodic expansions

$$\tilde{\mathcal{B}}_i^m(t) = \sum_{j=-\infty}^{+\infty} \mathcal{B}_{i+j(t_k-t_0)}^m(t) \tag{9}$$

which still verify (7). A k-knots closed spline curve is then a $\mathbb{R} \to \mathbb{R}^2$ periodic function (of period $t_k - t_0$) representable as a linear combination of k periodic basis functions

$$\mathbf{v}(t) \equiv [x(t) \quad y(t)] = \sum_{i=0}^{k-1} \mathbf{c}_i \tilde{\mathcal{B}}_i^m(t), \qquad t \in \mathbb{R}. \tag{10}$$

In computer graphics and image analysis, it is common to use $m = 2$ or $m = 3$ [17]. In all the examples presented below, cubic $(m = 3)$ B-splines will be used (and we will drop the superscript m); nevertheless, everything is valid for any m. Also, we will only address the periodic case, which is the one of interest for boundary representation.

B. Discrete Contours

A discretized spline curve (a contour on a digital image) is a set of N equispaced samples of $\mathbf{v}(t)$ collected as a $(N \times 2)$-vector

$$\mathbf{v} = [\mathbf{v}_0^T \cdots \mathbf{v}_{N-1}^T]^T = [\mathbf{x} \quad \mathbf{y}] \tag{11}$$

[1]For simplicity, in this paper we exclude the possibility of multiple knots; see, e.g., [16] for the consequences of this option.

where we assume that $N > k$ (usually, $N \gg k$), and

$$\mathbf{v}_i = [x_i \quad y_i] = \left[x\left(t_0 + \frac{i(t_k-t_0)}{N}\right) \quad y\left(t_0 + \frac{i(t_k-t_0)}{N}\right) \right]$$

for $i = 0, 1, \cdots, N-1$.

If we let the coordinates of the control points be arranged into a parameter vector $\boldsymbol{\theta}_{(k)} = [\mathbf{c}_0^T \cdots \mathbf{c}_{k-1}^T]^T = [\boldsymbol{\theta}_{(k)}^x \quad \boldsymbol{\theta}_{(k)}^y]$, where subscript (k) is used to emphasize that there are k control points, the discretized closed spline $\mathbf{v}$ can be obtained by a matrix product

$$\mathbf{v} = \mathbf{B}_{(k)} \boldsymbol{\theta}_{(k)} \Leftrightarrow \{\mathbf{x} = \mathbf{B}_{(k)} \boldsymbol{\theta}_{(k)}^x \text{ and } \mathbf{y} = \mathbf{B}_{(k)} \boldsymbol{\theta}_{(k)}^y\} \tag{12}$$

where the elements of $\mathbf{B}_{(k)}$ are given by $[\mathbf{B}_{(k)}]_{ij} = \tilde{\mathcal{B}}_j(t_0 + (((t_k - t_0)i)/N))$. In the absence of constraints on the control points, the set of all splines with k control points and a given set of basis functions (i.e., a given set of knots) is a linear space: the range of matrix $\mathbf{B}_{(k)}$, $\mathcal{R}(\mathbf{B}_{(k)})$.

C. B-Spline Fitting: 1-D Case with Known Knots

Consider the set of pairs $\{(s_i, f(s_i)), i = 0, 1, \cdots, N-1\}$, and consider the problem of finding the periodic spline (of a certain degree m, on a given periodic set of knots $\{\tilde{t}_0 < \tilde{t}_1 < \cdots < \tilde{t}_{k-1}\}$, with $k \leq N$; usually $k \ll N$) that best fits this set. Since the knots are known, a set of k periodic B-splines can be defined, $\{\tilde{\mathcal{B}}_j(t), j = 0, 1, \cdots, k-1\}$, and the elements of the $N \times k$ matrix $\mathbf{B}_{(k)}$ computed. The unknown spline can then be described as a linear combination of these B-splines and so the problem is that of finding the corresponding weights $\mathbf{c} \equiv [c_0 \quad c_1 \cdots c_{k-1}]$.

With $\mathbf{f} = [f(s_0) \quad f(s_1) \cdots f(s_{N-1})]^T$, the usual least squares criterion leads to

$$\hat{\mathbf{c}}_{(k)} = \arg\min_{\mathbf{c}} \|\mathbf{f} - \mathbf{B}_{(k)}\mathbf{c}\|^2$$
$$= (\mathbf{B}_{(k)}^T \mathbf{B}_{(k)})^{-1} \mathbf{B}_{(k)}^T \mathbf{f} \equiv \mathbf{B}_{(k)}^\dagger \mathbf{f} \tag{13}$$

where $\mathbf{B}_{(k)}^\dagger$ is the *pseudo-inverse* of $\mathbf{B}_{(k)}$. Notice that the null-space of $\mathbf{B}_{(k)}$ is $\mathcal{N}(\mathbf{B}_{(k)}) = \{\mathbf{0}\}$ and, because all its columns are linearly independent, $(\mathbf{B}_{(k)}^T \mathbf{B}_{(k)})$ is invertible. The vector of fitted spline values, at the same coordinates $\{s_0, s_1, \cdots, s_{N-1}\}$, denoted $\mathbf{g} = [g(s_0) \cdots g(s_{N-1})]^T$, is

$$\mathbf{g} = \mathbf{B}_{(k)}\hat{\mathbf{c}} = \mathbf{B}_{(k)}\mathbf{B}_{(k)}^\dagger \mathbf{f} \equiv \mathbf{B}_{(k)}^\perp \mathbf{f} \tag{14}$$

where $\mathbf{B}_{(k)}^\perp \equiv \mathbf{B}_{(k)}\mathbf{B}_{(k)}^\dagger$ is the *projection matrix*. That is, $\mathbf{g}$ is the projection of $\mathbf{f}$ onto the (k-dimensional) range space of $\mathbf{B}_{(k)}$, $\mathcal{R}(\mathbf{B}_{(k)})$.

D. B-Spline Fitting: 2-D Case with Known Number of Control Points

Now consider the situation where N points on the image plane are given

$$\mathbf{v} = \begin{bmatrix} x_0 & y_0 \\ \vdots & \vdots \\ x_{N-1} & y_{N-1} \end{bmatrix} = \begin{bmatrix} x(s_0) & y(s_0) \\ \vdots & \vdots \\ x(s_{N-1}) & y(s_{N-1}) \end{bmatrix}$$
$$= [\mathbf{x} \quad \mathbf{y}] \tag{15}$$

and the spline (described on a k-dimensional B-spline basis, with given k) that best fits them is sought; however, two key elements are missing: 1) the values of the s_i's to which the x_i's and the y_i's correspond, which are necessary to build matrix

$\mathbf{B}_{(k)}$ and 2) the knots, which are also indispensable in computing $\mathbf{B}_{(k)}$. Of the proposed strategies to address these issues (e.g., [8], [17], [26]), we adopt the simplest one, *uniform assignment*: $t_j = j$, for $j = 0, \cdots, k-1$ and $s_i = ik/N$, for $i = 0, \cdots, N-1$. Of course, it would be more interesting to consider the *free-knot problem*, i.e., to estimate the knot locations; however, this is a much harder problem to which there currently exists no general optimal solution [37], though there are several proposed practical techniques [16], [17], [26].

Summarizing, given a vector $\mathbf{v}$ with N data points and a choice for k, matrix $\mathbf{B}_{(k)}$ (which, with a *uniform assignment*, only depends on k and N) can be built and its pseudo-inverse $\mathbf{B}_{(k)}^{\dagger}$ computed. The estimated control points are then given by

$$\widehat{\boldsymbol{\theta}_{(k)}} = [\widehat{\boldsymbol{\theta}_{(k)}^{x}} \quad \widehat{\boldsymbol{\theta}_{(k)}^{y}}] = [\mathbf{B}_{(k)}^{\dagger}\mathbf{x} \quad \mathbf{B}_{(k)}^{\dagger}\mathbf{y}] = \mathbf{B}_{(k)}^{\dagger}\mathbf{v}. \quad (16)$$

Finally, the discretized spline curve corresponding to the estimated control points is given by $\mathbf{s} = [\mathbf{s}^{x} \quad \mathbf{s}^{y}] = [\mathbf{B}_{(k)}^{\perp}\mathbf{x} \; \mathbf{B}_{(k)}^{\perp}\mathbf{y}] = \mathbf{B}_{(k)}\widehat{\boldsymbol{\theta}_{(k)}}$.

III. ESTIMATING THE NUMBER OF CONTROL POINTS

It is well known that least squares fitting is equivalent to maximum likelihood (ML) estimation when the observations are modeled as being perturbed by white Gaussian noise. Specifically, let the observations $\mathbf{x}$ and $\mathbf{y}$ be independent white Gaussian noise contaminated versions of $\mathbf{B}_{(k)}\theta_{(k)}^{x}$ and $\mathbf{B}_{(k)}\theta_{(k)}^{y}$, with variances σ_x^2 and σ_y^2, respectively; i.e., $p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta}_{(k)}^{x}, \boldsymbol{\theta}_{(k)}^{y}, \sigma_x^2, \sigma_y^2) = p(\mathbf{x}|\boldsymbol{\theta}_{(k)}^{x}, \sigma_x^2)p(\mathbf{y}|\boldsymbol{\theta}_{(k)}^{y}, \sigma_y^2)$, where

$$p(\mathbf{x}|\boldsymbol{\theta}_{(k)}^{x}, \sigma_x^2) = (2\pi\sigma_x^2)^{-(N/2)}$$
$$\cdot \exp\left\{-\frac{1}{2\sigma_x^2}\|\mathbf{x} - \mathbf{B}_{(k)}\boldsymbol{\theta}_{(k)}^{x}\|^2\right\},$$
$$p(\mathbf{y}|\boldsymbol{\theta}_{(k)}^{y}, \sigma_y^2) = (2\pi\sigma_y^2)^{-(N/2)}$$
$$\cdot \exp\left\{-\frac{1}{2\sigma_y^2}\|\mathbf{y} - \mathbf{B}_{(k)}\boldsymbol{\theta}_{(k)}^{y}\|^2\right\}.$$

Then, the ML estimate of $\boldsymbol{\theta}_{(k)}$ is

$$\widehat{\boldsymbol{\theta}_{(k)}} \equiv [\widehat{\boldsymbol{\theta}_{(k)}^{x}} \quad \widehat{\boldsymbol{\theta}_{(k)}^{y}}]$$
$$= \left[\arg\max_{\boldsymbol{\theta}_{(k)}^{x}} p(\mathbf{x}|\boldsymbol{\theta}_{(k)}^{x}, \sigma_x^2) \quad \arg\max_{\boldsymbol{\theta}_{(k)}^{y}} p(\mathbf{y}|\boldsymbol{\theta}_{(k)}^{y}, \sigma_y^2)\right]$$
$$= [\mathbf{B}_{(k)}^{\dagger}\mathbf{x} \quad \mathbf{B}_{(k)}^{\dagger}\mathbf{y}] = \mathbf{B}_{(k)}^{\dagger}\mathbf{v} \quad (17)$$

as in (16) (see also (13). Notice that the estimates $\widehat{\boldsymbol{\theta}_{(k)}^{x}}$ and $\widehat{\boldsymbol{\theta}_{(k)}^{y}}$ do not depend on σ_x^2 and σ_y^2.

A. MDL Criterion

To estimate k, we adopt a *minimum description length* (MDL) criterion [40], adapted to the current problem. The first key observation behind MDL-type criteria is that looking for an ML estimate is equivalent to looking for the Shannon code for which the observations have the shortest code-length [40]; this is because Shannon's optimal code-length,[2] for data $\mathbf{v}$, obeying a probability density function $p(\mathbf{v}|\boldsymbol{\theta}_{(k)}, \sigma_x^2, \sigma_y^2)$, is simply [11], [40]

$$L(\mathbf{v}|\boldsymbol{\theta}_{(k)}, \sigma_x^2, \sigma_y^2) = -\log p(\mathbf{v}|\boldsymbol{\theta}_{(k)}, \sigma_x^2, \sigma_y^2). \quad (18)$$

[2]In bits or *nats*, respectively, if base-2 or natural logarithms are used [11].

It can be claimed that only discrete data have finite code-lengths, with a Gaussian density not being allowed here. However, as argued in [41], finite values may be obtained by discretizing a density to an arbitrary precision; a loose usage of the term "code-length" is convenient and harmless.

The second fundamental fact is that the parameters themselves are also part of the code, in the following sense: a code word representing $\mathbf{v}$ can not be decoded by itself; only full knowledge of $p(\mathbf{v}|\boldsymbol{\theta}_{(k)}, \sigma_x^2, \sigma_y^2)$ (i.e., of its parameters) allows reconstructing the code and respective decoder. Accordingly, the MDL criterion states that the description code-length to be minimized by the estimate must include not only the data code-length but also the code-lengths of the parameters.

In our case, the joint MDL estimate of k and $\boldsymbol{\theta}_{(k)}$, which we will denote by $\widehat{\boldsymbol{\theta}_{(\hat{k})}}$, is then

$$(\widehat{\boldsymbol{\theta}_{(\hat{k})}}, \widehat{\sigma_x^2}, \widehat{\sigma_y^2}) = \arg\min_{k, \boldsymbol{\theta}_{(k)}, \sigma_x^2, \sigma_y^2}\{L(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta}_{(k)}, \sigma_x^2, \sigma_y^2)$$
$$+ L(\boldsymbol{\theta}_{(k)}, \sigma_x^2, \sigma_y^2)\} \quad (19)$$

where $L(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta}_{(k)}, \sigma_x^2, \sigma_y^2) = -\log p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta}_{(k)}, \sigma_x^2, \sigma_y^2)$, and $L(\boldsymbol{\theta}_{(k)}, \sigma_x^2, \sigma_y^2)$ is the parameters description length. Notice that, unlike the ML estimate of $\boldsymbol{\theta}_{(k)}$ for fixed k, $\widehat{\boldsymbol{\theta}_{(\hat{k})}}$ does depend on σ_x^2 and σ_y^2, this justifying why they were explicitly included as unknowns. Furthermore, we will write $L(\boldsymbol{\theta}_{(k)}, \sigma_x^2, \sigma_y^2) = L(\boldsymbol{\theta}_{(k)})$, i.e., we assume that σ_x^2 and σ_y^2 have constant description lengths which can be dropped from (19). Now, since, for a given k, the minima w.r.t. $\boldsymbol{\theta}_{(k)}^{x}$ and $\boldsymbol{\theta}_{(k)}^{y}$ do not depend on σ_x^2 and σ_y^2, we can write

$$\hat{k} = \arg\min_{k}\left\{L(\boldsymbol{\theta}_{(k)}) + \min_{\sigma_x^2}\left[\min_{\boldsymbol{\theta}_{(k)}^{x}}(-\log p(\mathbf{x}|\boldsymbol{\theta}_{(k)}^{x}, \sigma_x^2))\right]\right.$$
$$\left. + \min_{\sigma_y^2}\left[\min_{\boldsymbol{\theta}_{(k)}^{y}}(-\log p(\mathbf{y}|\boldsymbol{\theta}_{(k)}^{y}, \sigma_y^2))\right]\right\}. \quad (20)$$

After some simple manipulation, we obtain

$$\hat{k} = \arg\min_{k}\left\{L(\boldsymbol{\theta}_{(k)}) + N\log\left(\sqrt{\widehat{\sigma_x^2}(k)\widehat{\sigma_y^2}(k)}\right)\right\}, \quad (21)$$

where $\widehat{\sigma_x^2}(k)$ and $\widehat{\sigma_y^2}(k)$ are the residual error variances (which are functions of k)

$$\widehat{\sigma_x^2}(k) = \frac{\|\mathbf{x} - \mathbf{B}_{(k)}^{\perp}\mathbf{x}\|^2}{N}, \quad \widehat{\sigma_y^2}(k) = \frac{\|\mathbf{y} - \mathbf{B}_{(k)}^{\perp}\mathbf{y}\|^2}{N}.$$

Since matrix $\mathbf{B}_{(k)}$ is fully specified by k and N, from $\hat{k}$, we also immediately obtain $\widehat{\boldsymbol{\theta}_{(\hat{k})}^{x}} = \mathbf{B}_{(\hat{k})}^{\dagger}\mathbf{x}$, and $\widehat{\boldsymbol{\theta}_{(\hat{k})}^{y}} = \mathbf{B}_{(\hat{k})}^{\dagger}\mathbf{y}$.

B. Parameter Description Length

Specifying $L(\boldsymbol{\theta}_{(k)})$ is a crucial aspect of MDL criteria. The commonly used $L(\boldsymbol{\theta}_{(k)}) = \frac{1}{2}\log N$ is an asymptotically (large sample) optimal value, valid for (real) parameters that depend on all the data values [40]. Since this is not true for B-spline control points, we propose what may be described as the *natural code-length for discrete contours*.

Let $\widetilde{\boldsymbol{\theta}_{(k)}} = [\widetilde{\boldsymbol{\theta}_{(k)}^{x}} \quad \widetilde{\boldsymbol{\theta}_{(k)}^{y}}]$ be a finite precision version of $\boldsymbol{\theta}_{(k)} = [\boldsymbol{\theta}_{(k)}^{x} \quad \boldsymbol{\theta}_{(k)}^{y}]$, and let $\delta_{(k)}^{x} = (\widetilde{\boldsymbol{\theta}_{(k)}^{x}} - \boldsymbol{\theta}_{(k)}^{x})$ and $\delta_{(k)}^{y} = (\widetilde{\boldsymbol{\theta}_{(k)}^{y}} - \boldsymbol{\theta}_{(k)}^{y})$

be the associated error vectors. The error between the curve represented by vector $\widetilde{\boldsymbol{\theta}_{(k)}}$, $\tilde{\mathbf{v}} = [\tilde{\mathbf{x}} \ \tilde{\mathbf{y}}] = [\mathbf{B}_{(k)}\widetilde{\boldsymbol{\theta}_{(k)}^x} \ \mathbf{B}_{(k)}\widetilde{\boldsymbol{\theta}_{(k)}^x}]$, and the exact one $[\mathbf{x} \ \mathbf{y}] = [\mathbf{B}_{(k)}\boldsymbol{\theta}_{(k)}^x \ \mathbf{B}_{(k)}\boldsymbol{\theta}_{(k)}^y]$, is, by linearity

$$\varepsilon^x = \mathbf{B}_{(k)}\delta_{(k)}^x \quad \text{and} \quad \varepsilon^y = \mathbf{B}_{(k)}\delta_{(k)}^y. \tag{22}$$

Now, let ξ be the maximal absolute error allowed in any of the coordinates of the contour, i.e., let us impose that[3] $\|\varepsilon^x\|_\infty \leq \xi$ and that $\|\varepsilon^y\|_\infty \leq \xi$. In view of (22), it is clear that

$$\left(\|\boldsymbol{\theta}_{(k)}^x\|_\infty \leq \frac{\xi}{\|\mathbf{B}_{(k)}\|_\infty}\right) \Rightarrow \|\varepsilon^x\|_\infty \leq \xi \tag{23}$$

$$\left(\|\boldsymbol{\theta}_{(k)}^y\|_\infty \leq \frac{\xi}{\|\mathbf{B}_{(k)}\|_\infty}\right) \Rightarrow \|\varepsilon^y\|_\infty \leq \xi. \tag{24}$$

Since $\mathbf{v}$ is a discrete curve, $\xi = 1$ is the natural choice. Concerning $\|\mathbf{B}_{(k)}\|_\infty$, invoking the *partition of the unity* property (7), and the positivity of the B-splines, $\tilde{\mathcal{B}}_j(t) \geq 0$, leads to

$$\|\mathbf{B}_{(k)}\|_\infty = \max_i \sum_j |[\mathbf{B}_{(k)}]_{ij}|$$

$$= \max_i \sum_j \tilde{\mathcal{B}}_j\left(t_0 + \frac{(t_k - t_0)i}{N-1}\right) = 1.$$

It is also clear that if all the control points are inside the $w^x \times w^y$ image plane, then, $\mathbf{v}$ will also be inside the image plane (a spline is inside the convex hull of its control points [17]). Accordingly

$$L(\boldsymbol{\theta}_{(k)}) = k\left(\log\frac{w^x\|\mathbf{B}_{(k)}\|_\infty}{\xi} + \log\frac{w^y\|\mathbf{B}_{(k)}\|_\infty}{\xi}\right)$$

$$= k\log(w^x \ w^y). \tag{25}$$

The final estimation criterion, obtained by inserting $L(\boldsymbol{\theta}_{(k)}) = k\log(w^x \ w^y)$ into (21), has the following intuitively reasonable features:

1) for given N and $(w^x \ w^y)$, increasing k decreases $N\log(\sqrt{\widehat{\sigma_x^2}(k) \ \widehat{\sigma_y^2}(k)})$ but increases $L(\boldsymbol{\theta}_{(k)})$, forcing an MDL-typical compromise between these two terms;
2) larger N (more data) gives more relative weight to the error variance, i.e. more control points will be allowed;
3) when the coordinate ranges w^x and/or w^y increase, the variance term receives a smaller relative weight, i.e. smaller fitting precision is imposed.

This last property, which at first sight may look strange, does make sense if one admits that the same data points, on a larger image, are relatively less meaningful and can then be fitted with less precision; in other words, we are talking about relative, and not absolute, precision. Anyway, notice that this effect is very weak; because it appears inside a logarithm, only orders of magnitude changes in the product $w^x \ w^y$ will have a strong influence on the result.

It is well known that MDL-type criteria can be interpreted under a Bayesian perspective [12], [40]. Equation (19) can be seen as a MAP estimator

$$\left(\widehat{\boldsymbol{\theta}_{(k)}^x}, \widehat{\boldsymbol{\theta}_{(k)}^y}, \widehat{\sigma_x^2}, \widehat{\sigma_y^2}\right)$$

$$= \arg \max_{k, \boldsymbol{\theta}_{(k)}^x, \boldsymbol{\theta}_{(k)}^y, \sigma_x^2, \sigma_y^2}$$

$$\{p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta}_{(k)}^x, \boldsymbol{\theta}_{(k)}^y, \sigma_x^2, \sigma_y^2)p(\boldsymbol{\theta}_{(k)}^x, \boldsymbol{\theta}_{(k)}^y, \sigma_x^2, \sigma_y^2)\} \tag{26}$$

---

[3]Here, $\|\mathbf{u}\|_\infty$ is the l_∞ vector norm, $\|\mathbf{u}\|_\infty = \max_i |u_i|$; the induced matrix norm is $\|\mathbf{B}_{(k)}\|_\infty = \max_i \sum_j |[\mathbf{B}_{(k)}]_{ij}|$.
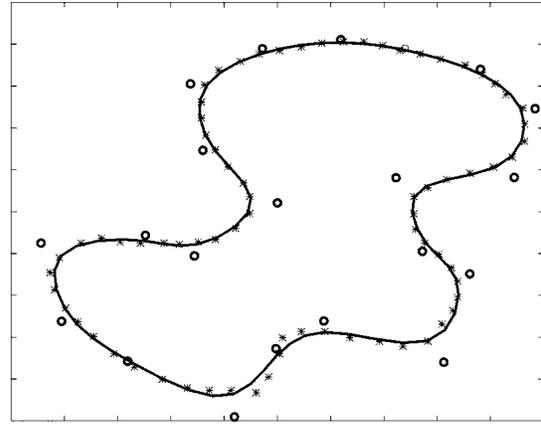

Fig. 1.   Hand-drawn points (marked "∗"), estimated spline (solid line), and control points (small circles).
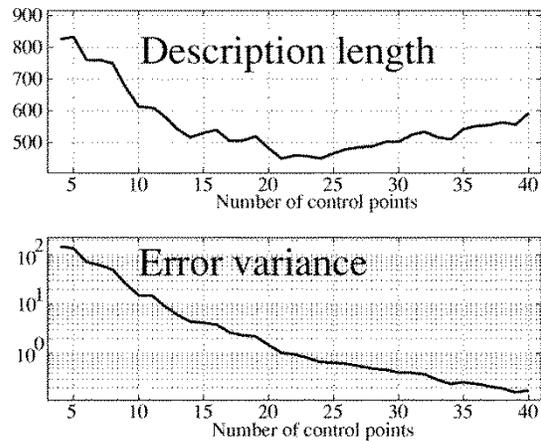

Fig. 2.   Plots of the description length (minimum at 21) and error variance (relative to Fig. 1).

with the prior

$$p(\boldsymbol{\theta}_{(k)}^x, \boldsymbol{\theta}_{(k)}^y, \sigma_x^2, \sigma_y^2) \propto \exp\{-k\log(w^x \ w^y)\}. \tag{27}$$

This prior can be interpreted as a (parametric) "*smoothing*-type" prior which favors "*simpler*" (in the sense of having fewer control points) contours. Another prior for the number of B-spline knots was proposed in [8].

C. Some Examples

Adaptive spline fitting, although not our final goal, is an important problem in itself. Moreover, to the authors' knowledge, there exist no published automatic general criteria to determine the optimal number of knots. Accordingly, we present now a few examples. Figs. 1 and 3 show two sets of data points ("∗"), the fitted splines (solid lines) and the corresponding control points (small circles). The evolution of the error variances (we plot $(\sigma_x^2(k)\sigma_y^2(k))^{1/2}$) and the description lengths as functions of k are shown in Figs. 2 and 4; notice that, the first shape, being more complex, requires more control points ($\hat{k} = 21$) than the second one ($\hat{k} = 7$). In Fig. 5, the data from Fig. 1 was perturbed with additive noise; as a result, a smaller number of control points was found optimal; this shows that the criterion is able to distinguish intrinsic curve complexity from noisy data, thus behaving as an adaptive smoother.
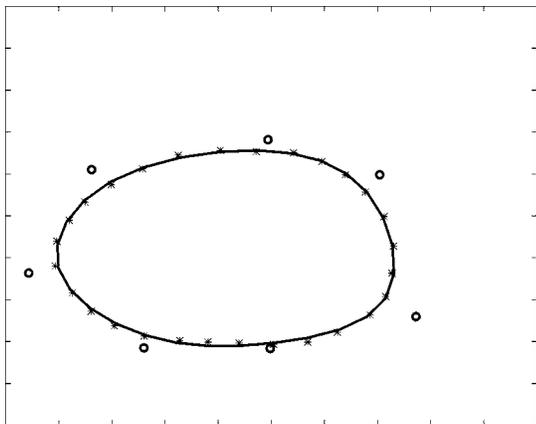
Fig. 3. Hand-drawn points (marked "$*$"), estimated spline (solid line), and control points (small circles).
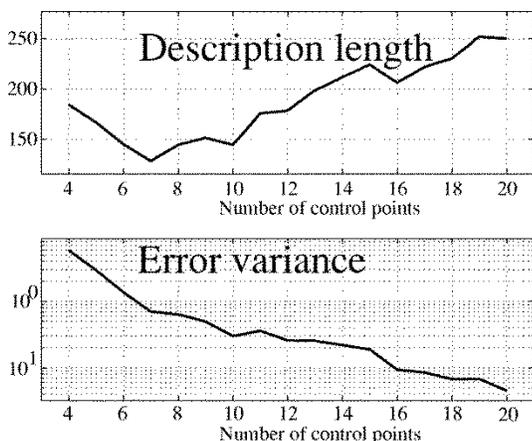

Fig. 4. Plots of the description length (minimum at 7) and error variance (relative to Fig. 3).
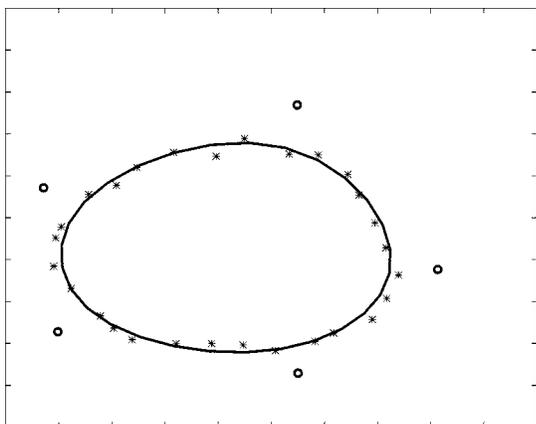

Fig. 5. Noise contaminated points (marked "$*$"), estimated spline (solid line), and control points (small circles); see text for comment.

IV. CONTOUR ESTIMATION

We will now address our main problem: contour estimation, not from a set of points, but from an observed image. To do so, we need an image model (or observation model), i.e., a probabilistic formulation of how the observed image is related to the underlying contour. In this paper, we adopt region-based models for two types of reasons. First, we have medical imaging appli-

cations in mind, and region based models are known to be robust with respect to local artifacts and poor image quality. Secondly, region-based models allow an easy derivation of the likelihood function, which is a main ingredient of our formulation. Of course it is also possible to write models and derive likelihood functions for other (e.g., gradient-based) approaches.

A. Observation Model

Let the observed image $\mathbf{I}$ (a $w^x \times w^y$ array of gray levels) be a random function of an ideal (in the sense of [25]) object whose (closed) boundary $\mathbf{v} = \mathbf{B}_{(k)}\boldsymbol{\theta}_{(k)}$ is a discretized uniform periodic spline, with k and $\boldsymbol{\theta}_{(k)}$ both unknown. An image observation model is, in probabilistic terms, a likelihood function $p(\mathbf{I}|\boldsymbol{\theta}_{(k)}, \boldsymbol{\phi})$, where the image plays the role of observed data, while the control points play the role of unknown parameters; other parameters characterizing the observation mechanism, which we will also consider unknown, are collected into vector $\boldsymbol{\phi}$. Although other choices would be possible, let us consider the region-based model characterized by the two following hypotheses:

1) Conditional Independence: Given the contour, the image pixels are independently distributed.

2) Region Homogeneity: The conditional probability function of each pixel depends only on whether it belongs to the inside or outside region of the contour, i.e., all pixels inside (resp. outside) have a common distribution characterized by a parameter vector $\boldsymbol{\phi}_{\mathrm{in}}$ (resp. $\boldsymbol{\phi}_{\mathrm{out}}$), with $\boldsymbol{\phi} = [\boldsymbol{\phi}_{\mathrm{in}}, \boldsymbol{\phi}_{\mathrm{out}}]$. From these assumptions, the likelihood function is

$$p(\mathbf{I}|\boldsymbol{\theta}_{(k)}, \boldsymbol{\phi}) = \left(\prod_{(i,j) \in \mathcal{I}(\mathbf{v})} p(I_{(i,j)}|\boldsymbol{\phi}_{\mathrm{in}}) \right) \cdot \left(\prod_{(i,j) \in \mathcal{O}(\mathbf{v})} p(I_{(i,j)}|\boldsymbol{\phi}_{\mathrm{out}}) \right) \quad (28)$$

with $\mathbf{v} = \mathbf{B}_{(k)}\boldsymbol{\theta}_{(k)}$, and $I_{(i,j)}$ denoting the value of pixel (i, j), while $\mathcal{I}(\mathbf{v})$ and $\mathcal{O}(\mathbf{v})$ are, respectively, the inside and outside regions of contour $\mathbf{v}$. Finally, $p(I_{(i,j)}|\boldsymbol{\phi}_{\mathrm{in}})$ and $p(I_{(i,j)}|\boldsymbol{\phi}_{\mathrm{out}})$ are the pixel-wise probability functions, of the inner and outer regions, respectively. Notice that there is no guarantee that the normalizing constant of $p(\mathbf{I}|\mathbf{v}, \boldsymbol{\phi})$, if written as in (4), does not depend on $\mathbf{v}$.

The conditional independence assumption leads to a simple expression for the joint probability of all the image pixels. For other region models, such as the inner and outer regions being characterized by two different textures, exact expressions for the joint probability are, in general, very difficult or even impossible to obtain. In those situations, one may resort to approximations, with obvious candidates being the mean field approximation (see, e.g., [51] and references therein), the pseudo-likelihood approximation [4], [20], or the technique proposed in [48]. However, notice that this is not a fundamental limitation of our approach, but just a computational/implementation issue.

B. Complete Estimation Criterion

Any unsupervised scheme has to estimate, from the observed image $\mathbf{I}$, not only the number of control points k, and their positions $\boldsymbol{\theta}_{(k)}$, but also the observation parameters $\boldsymbol{\phi}$. Combining
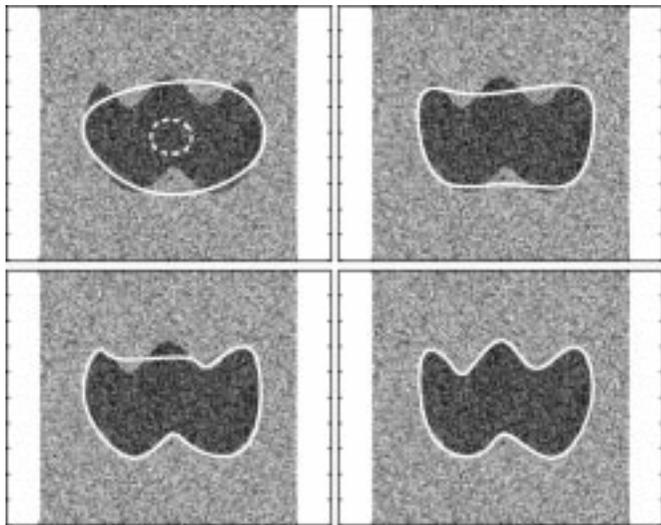
Fig. 6. Example with synthetic image; evolution of the contour shape for several numbers of control points 5, 8, 10, and 11 (chosen estimate). The dashed line is the initial contour.
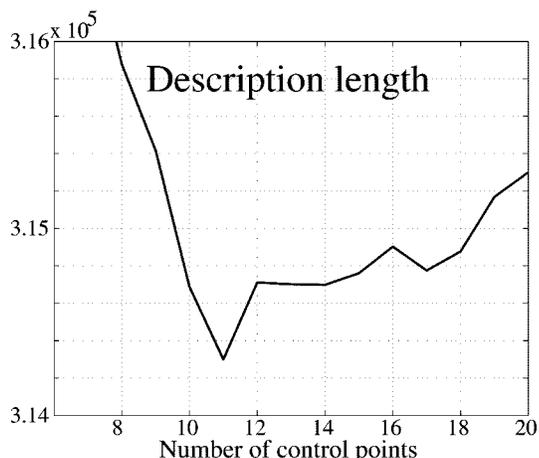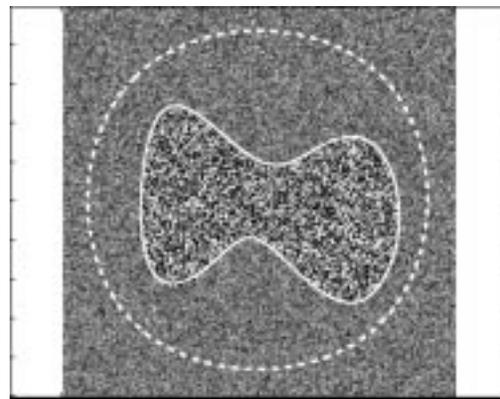

Fig. 8. Synthetic image (same mean, different variances) and estimated contour (see text for parameter values). The dashed line is the user-provided initial contour.
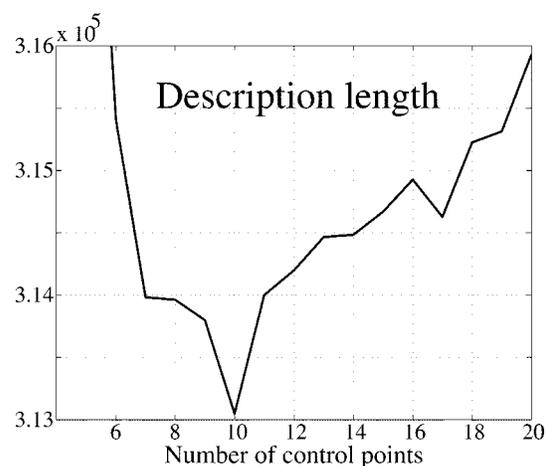

Fig. 7. Evolution of the description length for the example of Fig. 6.

Fig. 9. Evolutions of the description length for the example of Fig. 8.

the likelihood function in (28) with the MDL criterion derived above yields

$$(\widehat{\boldsymbol{\theta}_{(k)}}, \boldsymbol{\phi}) = \arg \min_{k, \boldsymbol{\theta}_{(k)}, \boldsymbol{\phi}} \{- \log p(\mathbf{I}|\boldsymbol{\theta}_{(k)}, \boldsymbol{\phi}) + k \log(w^x w^y)\}.$$

As in the previous section, this minimization can be rearranged into a nested pair of minimizations

$$\hat{k} = \arg \min_k \left\{ k \log(w^x w^y) - \max_{\boldsymbol{\phi}, \boldsymbol{\theta}_{(k)}} \left\{ \log p(\mathbf{I}|\boldsymbol{\theta}_{(k)}, \boldsymbol{\phi}) \right\} \right\} \tag{29}$$

however, unlike in the fitting problem studied above, the inner joint maximization is not simple and can not be split and analytically solved. We postpone this issue to the following subsections simply by denoting the result, necessarily a function of k and $\mathbf{I}$, as $G(\mathbf{I}, k)$. With this notation

$$\hat{k} = \arg \min_k \{k \log(w^x w^y) - G(\mathbf{I}, k)\}. \tag{30}$$

Once this minimization problem is solved, both $\boldsymbol{\phi}$ and $\widehat{\boldsymbol{\theta}_{(\hat{k})}}$ will be natural byproducts.

V. ALGORITHMS

We address the minimization in (29) with a three-level hierarchical scheme of nested algorithms: the inner scheme (Algorithm 1) solves w.r.t. $\boldsymbol{\theta}_{(k)}$, with k and $\boldsymbol{\phi}$ fixed; Algorithm 1 is then used by Algorithm 2 to solve w.r.t. both $\boldsymbol{\theta}_{(k)}$ and $\boldsymbol{\phi}$, still with k held constant. Finally, the solution w.r.t. k is found by exhaustive search over a range of values.

A. Algorithm 1: Solving for $\boldsymbol{\theta}_{(k)}$, given k and $\boldsymbol{\phi}$

Our first building block implements the inner maximization in (29) for fixed k and $\boldsymbol{\phi}$. The maximization w.r.t. $\boldsymbol{\theta}_{(k)}$ can be rewritten as a constrained maximization with respect to $\mathbf{v}$

$$\max_{\boldsymbol{\theta}_{(k)}} \{\log p(\mathbf{I}|\boldsymbol{\theta}_{(k)}, \boldsymbol{\phi})\}$$
$$= \text{solution of} \begin{pmatrix} \max_{\mathbf{v}} \{\log p(\mathbf{I}|\mathbf{v}, \boldsymbol{\phi})\} \\ \text{subject to: } \mathbf{v} \in \mathcal{R}(\mathbf{B}_{(k)}) \end{pmatrix} \tag{31}$$

where $\mathcal{R}(\mathbf{B}_{(k)})$ is the range space of $\mathbf{B}_{(k)}$, i.e., the search is constrained to those contours that can be written as $\mathbf{v} = \mathbf{B}_{(k)}\boldsymbol{\theta}_{(k)}$, for some $\boldsymbol{\theta}_{(k)}$. To solve it, we use a form of the *gradient projection* method (i.e., *gradient ascent* where each gradient is projected onto the constraint space [33]), also related to the iterative techniques considered in [9]. A description of the algorithm follows.
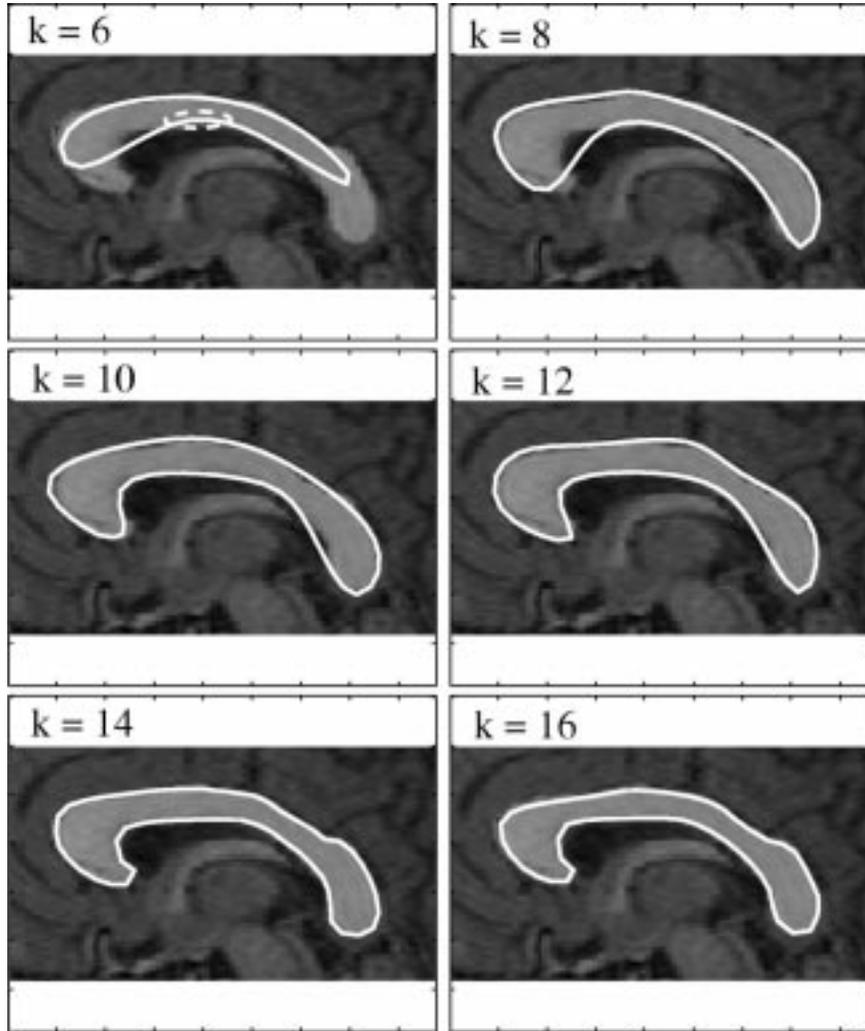
Fig. 10. Example with a real (magnetic resonance) image; the task is to find the boundary of the *corpus callosum*. We show estimates for 6, 8, 10, 12, 14, and 16 (the chosen number) control points. As above, the dashed line is the initial contour.

Algorithm 1
Inputs: k, ϕ, and an initial valid contour $\hat{\mathbf{v}}^{(0)} \in \mathcal{R}(\mathbf{B}_{(k)})$.
Output: a contour estimate $\hat{\mathbf{v}}$.
   Step 0: Build $\mathbf{B}_{(k)}$ and compute $\mathbf{B}_{(k)}^{\perp}$. Let $n = 0$.
   Step 1: Compute the gradient with respect to the contour

$$\delta\mathbf{v} = (\nabla(\log p(\mathbf{I}|\mathbf{v}, \phi))|_{\mathbf{v}=\hat{\mathbf{v}}^{(n)}}). \tag{32}$$

   Step 2: Update the contour estimate according to $\hat{\mathbf{v}}^{(n+1)} = \hat{\mathbf{v}}^{(n)} + \varepsilon\mathbf{B}_{(k)}^{\perp}\delta\mathbf{v}$, i.e., taking a step in the direction of the projection of the gradient onto $\mathcal{R}(\mathbf{B}_{(k)})$; parameter ε controls the step-size.
   Step 3: If a stopping criterion is met, stop and output $\hat{\mathbf{v}} = \hat{\mathbf{v}}^{(n+1)}$; if not, increment n and go back to Step 1.

The gradient w.r.t. the contour coordinates (Step 1), since these are (integer) pixel locations, is approximated by discrete differences. It can be shown that this gradient is normal to the contour [52]; this fact can be used to speed up the computations. Parameter ε should be kept small to avoid instabilities near the minima.

B. Solving for ϕ and $\boldsymbol{\theta}_{(k)}$, with fixed k

   To implement this joint maximization, we use an iterative estimation/maximization scheme having Algorithm 1 as one of its steps. Although resembling the *expectation–maximization* (EM) algorithm [14], notice that this is not an EM algorithm. Moreover, it is not guaranteed to converge to a joint global maximum. This type of schemes have been used by several authors, in different areas, under different names; see, e.g., the *adaptive segmentation algorithm* in [32], or adaptive versions of the *iterated conditional modes* (ICM) algorithm [4], [18]. Formally, the algorithm is as follows.

Algorithm 2
Inputs: k, and an initial valid contour $\hat{\mathbf{v}}^{(0)} \in \mathcal{R}(\mathbf{B}_{(k)})$.
Outputs: estimates ϕ and $\hat{\mathbf{v}}$
   Step 0: Let $q = 0$.
   Step 1: Given $\hat{\mathbf{v}}^{(q)}$, compute the ML estimate $\phi^{(q)} \equiv [\widehat{\phi_{\text{in}}}^{(q)} \ \widehat{\phi_{\text{out}}}^{(q)}]$ according to

$$\widehat{\phi_{\text{in}}}^{(q)} = \arg\max_{\phi_{\text{in}}}\left\{\prod_{(i,j)\in\mathcal{I}(\hat{\mathbf{v}}^{(q)})} p(I_{(i,j)}|\phi_{\text{in}})\right\} \tag{33}$$

$$\widehat{\phi_{\text{out}}}^{(q)} = \arg\max_{\phi_{\text{out}}}\left\{\prod_{(i,j)\in\mathcal{O}(\hat{\mathbf{v}}^{(q)})} p(I_{(i,j)}|\phi_{\text{out}})\right\}. \tag{34}$$
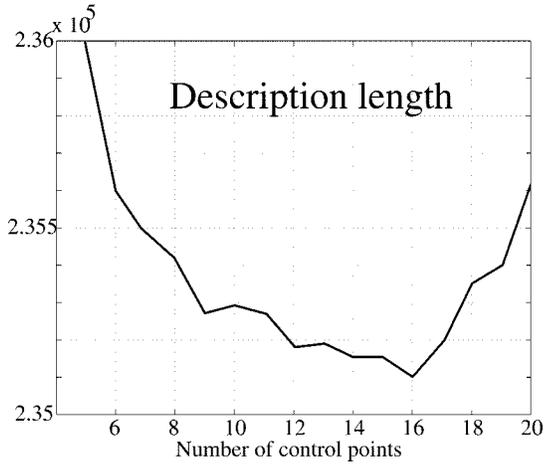
Fig. 11. Evolution of the description length corresponding to the example in Fig. 10.

Step 2: Run Algorithm 1, providing k, $\boldsymbol{\phi}^{(q)}$, and $\hat{\mathbf{v}}^{(q)}$ as inputs. The output is an updated contour estimate $\hat{\mathbf{v}}^{(q+1)}$ (which is still in $\mathcal{R}(\mathbf{B}_{(k)})$, by construction of Algorithm 1).

Step 3: If some stopping criterion is met, stop, and provide as outputs $\hat{\boldsymbol{\phi}} = \hat{\boldsymbol{\phi}}^{(q+1)}$ and $\hat{\mathbf{v}} = \mathbf{v}^{(q+1)}$; otherwise, increment q and return to Step 1.

The particular form of (33) and (34) depends on the image model. In the experiments presented in this paper, we will assume the two following image models:

1) Gaussian: All pixels are independent and Gaussian distributed with means μ_{in} and μ_{out} and variances σ_{in}^2 and σ_{out}^2, for the inside and outside regions, respectively. In this case, $\boldsymbol{\phi}_{\mathrm{in}} = [\mu_{\mathrm{in}} \ \sigma_{\mathrm{in}}^2]$, $\boldsymbol{\phi}_{\mathrm{out}} = [\mu_{\mathrm{out}} \ \sigma_{\mathrm{out}}^2]$, and **Step 1** consists simply of computing the (inside and outside) sample mean and variance (which are the ML estimates).

2) Rayleigh: In this case, which adequately models ultrasound images [15], the pixels are Rayleigh distributed. For the inside pixels, we have

$$p(I_{(i,j)}|\sigma_{\mathrm{in}}^2) = \frac{I_{(i,j)}}{\sigma_{\mathrm{in}}^2} \exp\left\{ -\frac{I_{(i,j)}^2}{2\sigma_{\mathrm{in}}^2} \right\} \qquad (35)$$

and a similar expression (with σ_{out}^2) for the outside. The parameter vector is now $\boldsymbol{\phi} = [\sigma_{\mathrm{in}}^2 \ \sigma_{\mathrm{out}}^2]$ whose ML estimates are simply one half of the sample means of squares [15].

We are also now in a position to obtain particular expressions for the function

$$G(\mathbf{I}, k) = \max_{\boldsymbol{\phi}, \boldsymbol{\theta}_{(k)}} \{ \log p(\mathbf{I}|\boldsymbol{\theta}_{(k)}, \boldsymbol{\phi}) \} = \log p(\mathbf{I}|\widehat{\boldsymbol{\theta}_{(k)}}, \hat{\boldsymbol{\phi}}) \quad (36)$$

which, of course, depend on the particular image model being adopted. For both the Gaussian and Rayleigh image models, the result is the same (up to an additive constant) and given by

$$G(\mathbf{I}, k) \propto -\frac{N_{\mathrm{in}}(k)}{2} \log(\widehat{\sigma_{\mathrm{in}}^2}(k)) - \frac{N_{\mathrm{out}}(k)}{2} \log(\widehat{\sigma_{\mathrm{out}}^2}(k))$$

where $N_{\mathrm{in}}(k)$ and $N_{\mathrm{out}}(k)$ are, respectively, the number of image pixels inside and outside the estimated contour.

C. Solving with Respect to k

Finally, the full scheme simply proceeds as follows. For each k in a given set $\mathcal{K} = \{k_{\mathrm{min}}, k_{\mathrm{min}} + 1, \cdots, k_{\mathrm{max}}\}$, run Algo-
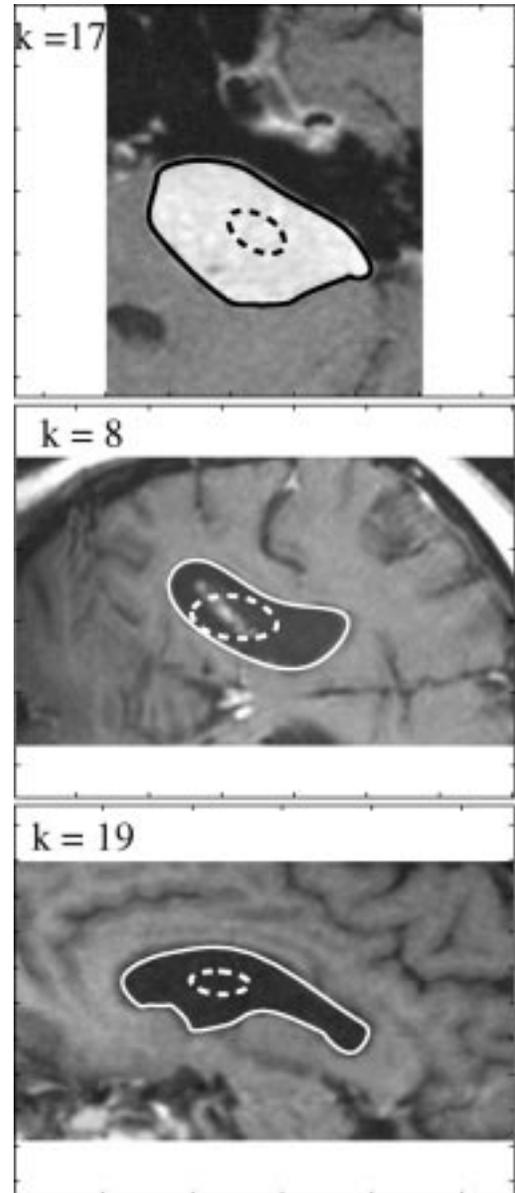

Fig. 12. Three other examples on magnetic resonance brain images; the estimated numbers of control points are indicated in the figures.

rithm 2 using each obtained contour estimate to initialize the next run (of course, an initial estimate is needed for the first run) and storing the output estimates. With these, we compute $\{k \log(w_x w_y) - G(\mathbf{I}, k), \text{ for } k \in \mathcal{K}\}$ and find the minimum with respect to k, i.e.,

$$\hat{k} = \arg \min_{k \in \mathcal{K}} \{ k \log(w^x w^y) - G(\mathbf{I}, k), \text{ for } k \in \mathcal{K} \}.$$

Although this exhaustive search may seem inefficient, on a Pentium-II (300 MHz) personal computer, the complete algorithm usually takes less than 1 min.

VI. EXPERIMENTS

The first two examples (Figs. 6–9) use synthetic images obeying the Gaussian model above defined. In Fig. 6, the inner and outer regions have the same variance ($\sigma_{\mathrm{in}} = \sigma_{\mathrm{out}} = 60$), but different means ($\mu_{\mathrm{in}} = 80$, $\mu_{\mathrm{out}} = 160$). The initial
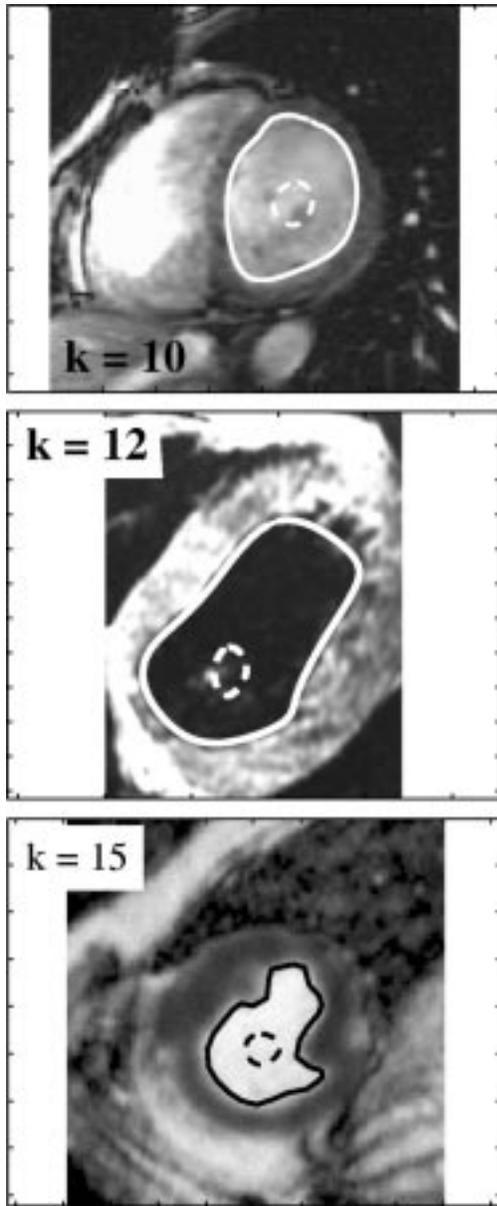
Fig. 13. Three more examples, now with cardiac MR images. The estimated number of control points are indicated in the figures.

Fig. 14. Two examples on real (ultrasound) images: (top) a cardiac (left ventricle) image and (bottom) an intravascular image.

contour provided by the user is represented by a dashed line, while the final estimates, for each value of k, are drawn as solid lines. Contour estimates for five, eight, ten, and 11 control points are shown, revealing that the algorithm adequately chose $k = 11$, the smallest number of control points that is enough to represent the contour shape; more control points would be clearly unnecessary. The evolution of the description length as a function of the number of control points is plotted in Fig. 7. The image model parameter estimates obtained were $\hat{\mu}_{in} = 80.74$, $\hat{\mu}_{out} = 159.48$, $\hat{\sigma}_{in} = 60.91$, and $\hat{\sigma}_{out} = 60.32$, whose closeness to the true values also testifies for the good performance of the method.

In the example of Fig. 8, the inner and outer regions have the same mean ($\mu_{in} = \mu_{out} = 120$) but different variances ($\sigma_{in} = 100$, $\sigma_{out} = 50$). The description length evolution (with minimum for $k = 10$) is plotted in Fig. 9. The final parameter

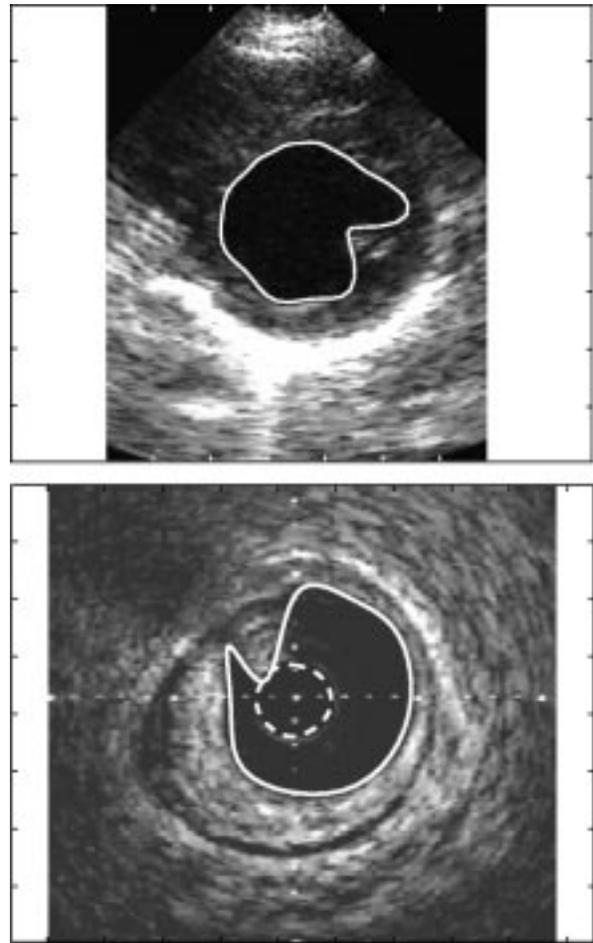estimates are $\hat{\mu}_{in} = 119.53$, $\hat{\mu}_{out} = 120.67$, $\hat{\sigma}_{in} = 99.12$, $\hat{\sigma}_{out} = 51.02$, again very close to the true values. Notice that in this example the initial contour is outside the true contour, and still the correct solution is obtained.

The next example considers the task of estimating the contour of the *corpus callosum* on a magnetic resonance (MR) brain image. Fig. 10 displays the evolution of the contour estimate (again, the initialization is shown as a dashed line) for increasing values of k (6, 8, 10, 12, 14, and 16). As shown in Fig. 11, the minimum of the description length corresponds to $k = 16$.

Figs. 12 and 13 show six more examples on real medical images (all using the Gaussian regions model). Notice how, even with nonhomogeneous regions (specially the outer ones), the obtained contours are excellent. The final pair of examples (in Fig. 14) deal with ultrasound data: one cardiac (left ventricle) image and an intravascular image. Here the Rayleigh model was used.

It is important to stress the fact that, in all the previous examples, the exact same algorithm was applied (apart from the selection of the Rayleigh model for the ultrasound images) without any parameter tuning (in fact, there are no parameters to tune). These examples also testify for the robustness of our method against poor initializations. The key factor controlling the success of a given initial contour is the resulting initial parameter estimate; more precisely, the first estimate of ϕ must be such that
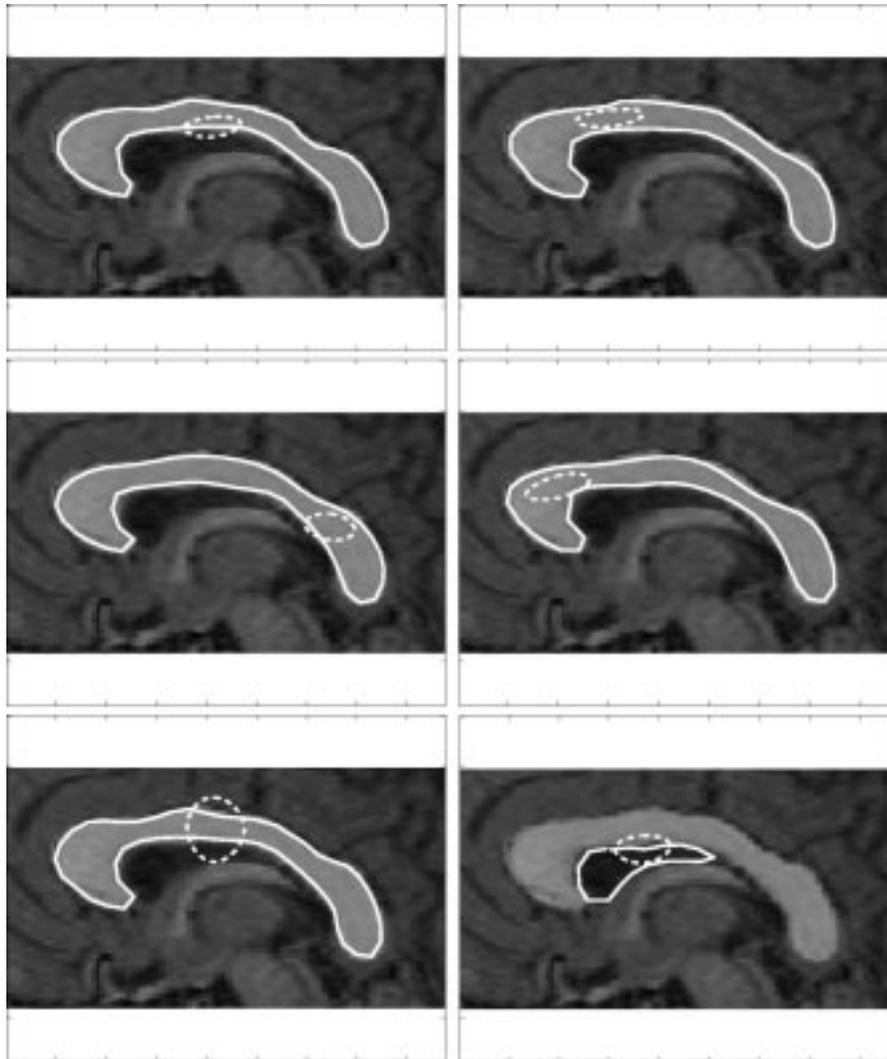
Fig. 15. Outlining the *corpus callosum* in a MR brain image. The first five cases are successful estimates from very different initializations, while the last one failed due to inappropriate initialization (as explained in the text).

the likelihood function will pull the contour in the right direction. In the absence of model mismatch (i.e., if the image really follows the assumed model), a good initial contour is one whose inside shares a larger area with the inside of the true contour than with its outside; in this way, initial parameter estimates will respect the order relations between the true parameters (e.g., if $\mu_{\text{in}} > \mu_{\text{out}}$, the same should be true for their first estimates) and the contour will move in the right direction. Of course real (medical) images do not exactly follow the region-based model, namely the outside regions tend to be somewhat nonhomogeneous, and the initialization may be more problematic. For practical applications (and in all the examples presented), the user indicates a point inside the boundary and the algorithm is initialized with a small contour around that point (which is guaranteed to be completely inside the inner region). In Fig. 15, we illustrate the robustness of our method with respect to initialization using the MR image from Fig. 10; we show five successful contour estimates from five very different initializations. Still, in Fig. 15, we present an example of a wrong estimate due to the fact that the initial contour had more overlap with the external region than with the inner one (as mentioned above).

We do not report comparisons of our method versus other deformable contour estimation schemes since we are not claiming that our approach achieves estimates that other methods can not. We are simply stating that we can do it in an unsupervised way and with a high degree of robustness with respect to initialization. For example, we have tried the method proposed in [49],[4] which is representative of the state of the art in snake-type methods, with most of the images considered in this paper (a notable exception in the one in Fig. 8, for which gradient based methods do not work). In most tests, that method was able to find good contour estimates, at the cost of careful tuning of the involved parameters (which are at least three, α, β, and μ, in the notation of [49], controlling aspects like the resistance of the contour to stretching and bending, and the relative weight of the internal and external forces); some images require stiffer contours, while some others demand more flexible ones, etc. In contrast, as already mentioned above, our method found all the contours in a totally unsupervised manner (apart from initialization) without any tuning or parameter adjustments.

[4]Available at http://iacl.ece.jhu.edu/projects/gvf.

VII. Concluding Remarks

This paper described a new approach to unsupervised smooth contour estimation based on a new adaptive B-spline representation. All the model parameters are considered unknown and estimated from the observed image. Examples presented, using synthetic and real (medical) images, showed the ability of the proposed method to estimate contours in an unsupervised manner, i.e., adapting to unknown shape and observation parameters. With the synthetic images, the good match between the estimated and the true parameters also shows the good performance of the approach.

References

[1] A. Amini, R. Curwen, and J. Gore, "Snakes and splines for tracking nonrigid heart motion," in *Proc. Eur. Conf. Computer Vision—ECCV'96*, Cambridge, U.K., 1996, pp. 251–261.

[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] Y. Amit, U. Grenander, and M. Piccioni, "Structural image restoration through deformable templates," *J. Amer. Statist. Assoc.*, vol. 86, no. 4, pp. 376–387, 1991.

[4] J. Besag, "On the statistical analysis of dirty pictures," *J. R. Statist. Soc. B*, vol. 48, no. 3, pp. 259–302, 1986.

[5] A. Blake, R. Curwen, and Z. Zisserman, "A framework for spatiotemporal control in the tracking of visual contours," *Int. J. Comput. Vis.*, vol. 11, no. 2, pp. 127–145, 1993.

[6] A. Chakaraborty, L. Staib, and J. Duncan, "Deformable boundary finding influenced by region homogeneity," in *Proc. IEEE Comput. Soc. Conf. Computer Vision Pattern Recognition—CVPR'94*, Seattle, WA, 1994, pp. 624–627.

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

[8] F. Cohen and J. Wang, "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.

[9] L. Cohen, "Auxiliary variables and two-step iterative algorithms in computer vision problems," *Int. J. Comput. Vis.*, vol. 6, no. 1, pp. 59–83, 1996.

[10] L. Cohen and I. Cohen, "Finite-element methods for active contour models and baloons for 2-D and 3-D images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 1131–1147, Nov. 1993.

[11] T. Cover and J. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[12] A. Dawid, "Prequential analysis, stochastic complexity, and Bayesian inference," in *Bayesian Statistics 4: Proceedings of the Fourth Valencia International Meeting*, J. Bernardo, J. Berger, A. David, and F. Smith, Eds. Oxford, U.K.: Oxford Univ. Press, 1992, pp. 1019–1125.

[13] C. deBoor, *A Practical Guide to Splines*. New York: Springer-Verlag, 1978.

[14] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood estimation from incomplete data via the EM algorithm," *J. R. Statist. Soc. B*, vol. 39, no. 1, pp. 1–38, 1977.

[15] J. Dias and J. Leitão, "Wall position and thickness estimation from sequences of echocardiograms images," *IEEE Trans. Med. Imag.*, vol. 15, pp. 25–38, Jan. 1996.

[16] P. Dierchx, *Curve and Surface Fitting with Splines*. Oxford, U.K.: Oxford Univ. Press, 1993.

[17] G. Farin, *Curves and Surfaces for Computer Aided Geometrical Design*. New York: Academic, 1990.

[18] M. Figueiredo and J. Leitão, "Bayesian estimation of ventricular contours in angiographic images," *IEEE Trans. Med. Imag.*, vol. 11, pp. 416–429, Mar. 1992.

[19] ——, "A nonsmoothing approach to the estimation of vessel contours in angiograms," *IEEE Trans. Med. Imag.*, vol. 14, pp. 162–172, Jan. 1995.

[20] M. Figueiredo and J. Leitão, "Unsupervised image restoration and edge location using compound Gauss-Markov random fields and the MDL principle," *IEEE Trans. Image Processing*, vol. 6, pp. 1089–1102, Aug. 1997.

[21] M. Figueiredo, J. Leitão, and A. K. Jain, "Adaptive B-splines and boundary estimation," in *Proc. IEEE Comput. Soc. Conf. Computer Vision Pattern Recognition—CVPR'97*, San Juan, PR, 1997, pp. 724–729.

[22] ——, "Adaptive parametrically deformable contours," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, M. Pellilo and E. Hancock, Eds. Berlin, Germany: Springer-Verlag, 1997, pp. 35–50.

[23] M. Flickner, J. Hafner, E. Rodriguez, and J. Sanz, "Periodic quasiorthogonal spline basis and applications to least squares over curve fitting of digital images," *IEEE Trans. Image Processing*, vol. 5, pp. 71–88, Jan. 1996.

[24] D. Geiger, A. Gupta, L. Costa, and J. Vlontzos, "Dynamic programming for detecting, tracking, and matching deformable contours," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 294–302, Mar. 1995.

[25] U. Grenander, *General Pattern Theory: A Mathematical Study of Regular Structures*. Oxford, U.K.: Oxford Univ. Press, 1993.

[26] Z. Huang and F. Cohen, "Affine-invariant B-spline moments for curve matching," *IEEE Trans. Image Processing*, vol. 5, pp. 1473–1480, Oct. 1996.

[27] A. K. Jain, Y. Zhong, and M. P. Dubuisson-Jolly, "Deformable template models: A review," *Signal Process.*, vol. 71, no. 1, pp. 109–129, 1998.

[28] A. K. Jain, Y. Zhong, and S. Lakshmanan, "Object matching using deformable templates," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 267–277, Mar. 1996.

[29] M. D. Jolly, S. Lakshmanan, and A. K. Jain, "Vehicle segmentation and classification using deformable templates," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 293–308, Mar. 1996.

[30] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, pp. 259–268, 1987.

[31] A. Klein, F. Lee, and A. Amini, "Quantitative coronary angiography with deformable spline models," *IEEE Trans. Med. Imag.*, vol. 16, pp. 468–482, Apr. 1997.

[32] S. Lakshmanan and H. Derin, "Simultaneous parameter estimation and segmentation of Gibbs random fields using simulated annealing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 799–813, Aug. 1989.

[33] D. Luenberger, *Linear and Nonlinear Programming*. Reading, MA: Addison–Wesley, 1984.

[34] T. McInerney and D. Terzopoulos, "Topologically adaptable snakes," in *Proc. Int. Conf. Computer Vision—ICCV'95*, Cambridge, MA, 1995, pp. 840–845.

[35] S. Menet, P. Saint-Marc, and G. Medioni, "Active contour models: Overview, implementation and applications," in *Proc. IEEE Int. Conf. Systems, Man, Cybernetics*, 1990, pp. 194–199.

[36] D. Mumford, "The Bayesian rationale for the variational formulations," in *Geometry-Driven Diffusion in Computer Vision*, B. ter Haar Romeny, Ed. Dordrecht, The Netherlands: Kluwer, 1994, pp. 135–147.

[37] G. Nurnberger, "Bivariate segment approximation and free knot splines," *Construct. Approx.*, vol. 12, pp. 555–558, 1996.

[38] P. Radeva, J. Serrat, and E. Marti, "A snake for model-based segmentation," in *Proc. Int. Conf. Computer Vision—ICCV'95*, Cambridge, MA, 1995, pp. 816–821.

[39] B. Ripley, *Pattern Recognition and Neural Networks*. Cambridge, U.K.: Cambridge Univ. Press, 1996.

[40] J. Rissanen, *Stochastic Complexity in Statistical Inquiry*, Singapore: World Scientific, 1989.

[41] ——, "Fisher information and stochastic complexity," *IEEE Trans. Inform. Theory*, vol. 42, pp. 40–47, Jan. 1996.

[42] C. Robert, *The Bayesian Choice: A Decision Theoretic Motivation*. New York: Springer-Verlag, 1994.

[43] R. Ronfard, "Region-based strategies for active contour models," *Int. J. Comput. Vis.*, vol. 13, no. 2, pp. 229–251, 1994.

[44] D. Rueckert and P. Burger, "Contour fitting using an adaptive spline model," in *Proc. Brit. Machine Vision Conf.*, vol. 1, Birmingham, U.K., 1995, pp. 207–216.

[45] P. Saint-Marc, H. Rom, and G. Medioni, "B-spline contour representation and symmetry detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 1191–1197, Nov. 1993.

[46] L. Staib and J. Duncan, "Boundary finding with parametrically deformable models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 1061–1075, Nov. 1992.

[47] G. Storvik, "A Bayesian approach to dynamic contours through stochastic sampling and simulated annealing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 10, pp. 976–986, 1994.

[48] Z. Wu, "An approximate method of evaluating the joint likelihood function for first-order GMRF's," *IEEE Trans. Image Processing*, vol. 2, pp. 520–522, Apr. 1993.

[49] C. Xu and J. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Trans. Image Processing*, vol. 7, pp. 359–369, Mar. 1998.
[50] A. Yuille and P. Hallinan, "Deformable templates," in *Active Vision*, A. Blake and A. Yuille, Eds. Cambridge, MA: MIT Press, 1992, pp. 21–38.
[51] J. Zhang, "The mean field theory in EM procedures for blind Markov random field restoration," *IEEE Trans. Image Processing*, vol. 2, pp. 27–40, Jan. 1993.
[52] S. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, energy/Bayes/MDL for multi-band image segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 884–900, Sept. 1996.

Mário A. T. Figueiredo (S'87–M'95) received the E.E., M.S., and Ph.D. degrees in electrical and computer engineering, all from Instituto Superior Técnico (IST), Technical University of Lisbon, Lisbon, Portugal, in 1985, 1990, and 1994, respectively.

Since 1994, he has been an Assistant Professor with the Department of Electrical and Computer Engineering, IST. He is also a Researcher with the Communication Theory and Pattern Recognition Group, Institute of Telecommunications, Lisbon. In 1998, he held a visiting position with the Department of Computer Science and Engineering, Michigan State University, East Lansing. His research interests include image analysis, computer vision, unsupervised learning, and information and coding theory.

Dr. Figueiredo received the Portuguese IBM Scientific Prize in 1995.

José M. N. Leitão (M'95) received the E.E. and Ph.D degrees in electrical engineering and the "Agregado" degree in electrical and computer engineering, all from Instituto Superior Ténico (IST), Technical University of Lisbon, Lisbon, Portugal, in 1970, 1983, and 1992, respectively.

He was with the Laboratory of Physiology, Gulbenkian Institute of Science, Oeiras, Portugal, from 1970 to 1972. After three years at the University of Tübingen, Germany, he joined the faculty of IST in 1976, where he is currently a Full Professor with the Department of Electrical and Computer Engineering. He is also the Coordinator of the Communication Theory and Pattern Recognition Group, Institute of Telecommunications, Lisbon. His main research interests are communication and information theory, pattern recognition, signal and image processing.

Anil K. Jain (S'70–M'72–SM'86–F'91) is a University Distinguished Professor with the Department of Computer Science and Engineering at Michigan State University, East Lansing. His research interests include statistical pattern recognition, Markov random fields, texture analysis, neural networks, document image analysis, fingerprint matching, and 3-D object recognition. He is the co-author of *Algorithms for Clustering Data* (Englewood Cliffs, NJ: Prentice-Hall, 1988), edited the book *Real-Time Object Measurement and Classification* (Berlin, Germany: Springer-Verlag, 1988), and co-edited the books, *Analysis and Interpretation of Range Images* (Berlin, Germany: Springer-Verlag, 1989), *Markov Random Fields* (New York: Academic, 1992), *Artificial Neural Networks and Pattern Recognition* (Amsterdam, The Netherlands: Elsevier, 1993), *3D Object Recognition* (Amsterdam, The Netherlands: Elsevier, 1993), and *BIOMETRICS: Personal Identification in Networked Society* (Norwell, MA: Kluwer, 1998).

Dr. Jain received the Best Paper Awards in 1987 and 1991 and certificates for outstanding contributions in 1976, 1979, 1992, and 1997 from the Pattern Recognition Society. He also received the 1996 IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE from 1990 to 1994. He is a Fellow of IAPR. He received a Fulbright Research Award in 1998.