

# Adaptive B-Splines and Boundary Estimation

Mário A. T. Figueiredo, José M. N. Leitão  
Instituto de Telecomunicações and DEEC  
Instituto Superior Técnico  
1096 Lisboa Codex, PORTUGAL  
mtf@lx.it.pt, jleitao@red.lx.it.pt

Anil K. Jain  
Department of Computer Science  
Michigan State University  
East Lansing, MI 48824, U.S.A.  
jain@cps.msu.edu

## Abstract

*This paper describes a boundary estimation scheme based on a new adaptive approach to B-spline curve fitting. The number of control points of the spline, their locations, and the observation parameters, are all considered unknown. The optimal number of control points is estimated via a new minimum description length (MDL) type criterion. The result is an adaptive parametrically deformable contour which also estimates the observation model parameters. Experiments on synthetic and real (medical) images confirm the adequacy and good performance of the approach.*

## 1. Introduction

### 1.1. Snakes and deformable templates/models

Snakes and their conceptual descendents have been often proposed to deal with contour/boundary estimation problems in several application contexts. A relevant example is medical imaging, where contour estimation is the fundamental first step of many automatic image analysis systems.

As proposed in [15], a snake is a virtual object (in the image plane) which can deform elastically (thus possessing *internal energy*) and which is under the influence of a potential field (thus having *external/potential energy*) function of image features of interest. The equilibrium (minimal energy) configuration 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 its “myopia” (use of strictly local data), have stimulated a great amount of research [1], [2], [4], [8], [17], [18], [21], [26]. One of the main problems is its non-adaptiveness, with parameters having to be specified by the user.

Deformable templates (DT) and deformable models (DM) constitute another class of approaches to con-

tour/object estimation. In DM, parametric shape representations are used, contrasting with the explicit descriptions used by snakes. In DT, some basic shape suffers deformations to fit the desired image features. Fundamental work on DT and DM is [11], [12]; see also [13], [14], [23], and [25]. Again, a critical difficulty is the lack of adaptiveness; parameters have to be previously specified.

### 1.2. Bayesian point of view

Snakes are interpretable as *maximum a posteriori* (MAP) Bayesian estimators, with the internal and external energies being associated with the prior and the *likelihood function*, respectively [8], [24]. The same is true for DT, where a prior biases the estimate towards the template shape [11], [13], [23]. In DM, however, it is common not to use a deformation energy (a prior) if the model itself guarantees regularity/smoothness of the represented shapes; a *maximum likelihood* (ML) interpretation is still valid. These probabilistic views have the advantage of giving precise meaning (not just physical analogies) to the involved entities; e.g., the likelihood function can be rigorously derived from an observation model rather than just from common sense.

### 1.3. Proposed approach

We propose an adaptive smooth boundary estimation scheme, based on B-spline representations, which can be classified as an adaptive DM. A statistical estimation framework is adopted, with all the parameters governing the involved models being taken as unknown.

When using B-spline-based contour descriptors, a key issue is the choice of the number of control points. We address it by viewing B-spline fitting as an ML estimation problem and introducing a suitable MDL-type criterion<sup>1</sup>. The resulting adaptive order B-spline descriptor is then used to build a DM which also estimates the observation parameters.

<sup>1</sup>This work was supported by the NATO Collaborative Research Grant #CRG 960010, and by PRAXIS XXI project 2/2.1/TIT/1580/95.

<sup>1</sup>Recently, MDL-type criteria have been proposed for several problems in computer vision and image processing (see [9] and references therein).

Although conventional potentials based on the image gradient are compatible with the proposed adaptive criterion, we adopt here a *region-based* approach [8], [21], [26]. This means that the likelihood function of the contour position depends on all the image (split into inner and outer regions), not just on a narrow stripe along the contour (as in *snakes*). This strategy works in situations where gradients fail (e.g., regions of equal mean) and is robust against local artifacts.

## 2. B-splines, curves, and closed curves

### 2.1. Splines and B-splines

Splines are piecewise polynomials widely used in computer graphics, computer vision, and image processing to build efficient parametric descriptions/approximations of functions, curves, and surfaces [3], [5], [7], [10], [22].

Let  $\{t_0 \leq t_1 \leq \dots \leq t_k\} \subset [t_0, t_k] \subset \mathbb{R}$ , the set of so-called *knots*, be given. By definition, spline functions are polynomial inside each interval  $[t_{i-1}, t_i]$ , and exhibit a certain degree of continuity at the knots [5].

The set  $\{\mathcal{B}_k^m(t), k = 0, \dots, k-m-1\}$  of B-splines of degree  $m$  constitutes a basis (though a non-orthogonal one) for the  $(k-m)$ -dimensional linear space of all piecewise polynomials (on  $[t_m, t_{k-m}]$ ) which are  $C^{m-1}$  continuous at the knots; the elements of this space (the splines) can then be represented as

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

each basis function (B-spline)  $\mathcal{B}_i^m(t)$  depends on a set of knot values  $\{t_k, \dots, t_{k+m+1}\}$  (for details, see, e.g., [5], [7]).

The  $\mathbb{R}^2$  version of Eq. (1),

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

describes an open curve on the plane, where the  $\mathbf{c}_i \equiv [c_i^x \ c_i^y]$  are now 2D vectors called *control points*.

Expressing closed curves requires the periodic extension of the knot sequence,  $\{\tilde{t}_j, j \in \mathcal{Z}\}$  with  $\tilde{t}_j = t_{j \bmod k}$  [10]. These knots can be seen as points along a circumference of perimeter  $(t_k - t_0)$ . Also, a set of  $k$  periodic B-splines  $\{\tilde{\mathcal{B}}_k^m(t), k = 0, \dots, k-1\}$  is obtained by the periodic expansion (of period  $(t_k - t_0)$ ) of the aperiodic ones [10]. If the knots are equispaced ( $\tilde{t}_i - \tilde{t}_{i-1} = \delta$ ), the B-splines are termed *uniform* and only differ by shifts,  $\tilde{\mathcal{B}}_{i+1}^m(t) = \tilde{\mathcal{B}}_i^m(t - \delta)$ . Closed  $k$ -knots spline curves are then functions of period  $(t_k - t_0)$ ; they can be written as linear combinations of  $k$  periodic basis function (the space is now  $k$ -dimensional):

$$\mathbf{f}(t) = \sum_{i=0}^{k-1} \mathbf{c}_i \tilde{\mathcal{B}}_i^m(t), \quad t \in \mathbb{R}. \quad (2)$$

Although it is a common practice to use quadratic or cubic ( $m = 2, 3$ ) B-splines [7], our technique is valid for any degree; thus, we will drop the superscript  $m$ . Also, we only address the periodic case (of interest for boundary representation), although the concepts also apply to aperiodic B-splines.

### 2.2. B-spline curve fitting

Take the problem of finding the degree- $m$  periodic spline, on a given set of knots  $\{t_0, t_1, \dots, t_k\}$ , that best fits  $N$  pairs  $\{(s_i, f(s_i)), i = 0, \dots, N-1\}$ , with  $k \leq N$  (usually  $k \ll N$ ). With  $\mathbf{f} = [f(s_0) \ \dots \ f(s_{N-1})]^T$ ,  $\mathbf{c} = [c_0 \ \dots \ c_{k-1}]^T$ , and  $\mathbf{B} = [B_{ij}]$  with  $B_{ij} = \tilde{\mathcal{B}}_j(s_i)$ , the least squares criterion reads

$$\hat{\mathbf{c}} = \arg \min_{\mathbf{c}} \|\mathbf{f} - \mathbf{B}\mathbf{c}\|^2 = (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T \mathbf{f} \equiv \mathbf{B}^\dagger \mathbf{f}, \quad (3)$$

where  $\mathbf{B}^\dagger$  is the *pseudo-inverse* of  $\mathbf{B}$ ; its null-space is  $\mathcal{N}(\mathbf{B}) = \{0\}$ , so  $(\mathbf{B}^T \mathbf{B})^{-1}$  exists. Let  $\mathbf{g} = \mathbf{B}\hat{\mathbf{c}} = \mathbf{B}\mathbf{B}^\dagger \mathbf{f} \equiv \mathbf{B}^\perp \mathbf{f}$ , where  $\mathbf{B}^\perp$  is the *orthogonal projector* onto the  $(k-1)$ -dimensional range space of  $\mathbf{B}$  (denoted  $\mathcal{R}(\mathbf{B})$ ).

Now consider a common situation in many applications where  $N$  points (in the image plane) are given:

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

The periodic spline that best fits them is sought, but two key elements (essential to obtain  $\mathbf{B}$ ) are missing: the  $s_i$ 's to which the  $x_i$ 's and the  $y_i$ 's correspond, and the knots  $\{t_0, \dots, t_k\}$ . Several strategies have been proposed, each having specific advantages and drawbacks [3], [5], [7]. The simplest one, and most convenient for our purposes, is the *uniform assignment*: take  $t_j = j$ , for  $j = 0, \dots, k-1$  and  $s_i = ik/N$ , for  $i = 0, \dots, N-1$ . We can see the knots and the  $s_i$ 's as defining uniform partitions of a circumference of perimeter  $k$ . Given  $\mathbf{v}$  and  $k$ , we build  $\mathbf{B}$  (it only depends on  $k$  and  $N$ ), compute  $\mathbf{B}^\dagger$ , and estimate the control points as

$$\hat{\mathbf{c}} = \begin{bmatrix} \hat{c}_0^x & \hat{c}_0^y \\ \vdots & \vdots \\ \hat{c}_{k-1}^x & \hat{c}_{k-1}^y \end{bmatrix} = \mathbf{B}^\dagger \mathbf{v}. \quad (4)$$

Finally, notice that Eq. (3) (and (4)) can be interpreted as an ML estimator if the observations  $\mathbf{f}$  are modelled as white Gaussian noise (variance  $\sigma^2$ ) contaminated versions of an "ideal" discretized spline  $\mathbf{B}\mathbf{c}$ , i.e.  $\mathbf{f} = \mathbf{B}\mathbf{c} + \mathbf{n}$ . Formally:

$$p(\mathbf{f}|\mathbf{c}, \sigma^2) = (2\pi\sigma^2)^{-\frac{N}{2}} \exp\left\{-\frac{\|\mathbf{f} - \mathbf{B}\mathbf{c}\|^2}{2\sigma^2}\right\} \quad (5)$$

$$\hat{\mathbf{c}} = \arg \max_{\mathbf{c}} \{\log p(\mathbf{f}|\mathbf{c}, \sigma^2)\} = \mathbf{B}^\dagger \mathbf{f} \quad (6)$$

Notice that  $\hat{\mathbf{c}}$  does not depend on  $\sigma^2$ .

### 3. Estimating the number of knots

We now focus on how to choose  $k$ , the number of control points (knots). To simplify the notation, we address the scalar case; curves on a plane are simply a pair of scalar splines. Clearly, minimizing the error variance is not a useful criterion; the (trivial) solution would be the largest possible  $k$ . (Note: in the sequel, writing  $\mathbf{c}_{(k)}$ ,  $\mathbf{B}_{(k)}$ , and  $\sigma^2_{(k)}$ , is meant to stress their dependency on  $k$ .)

#### 3.1. MDL criterion

To estimate  $k$ , we adopt an MDL-type criterion (see [19] and references therein) to the current problem.

The first key fact behind MDL is: the ML estimate corresponds to the Shannon code for which the observations have the shortest code-length [19]; in fact, Shannon's optimal code-length for  $\mathbf{f}$ , given  $\mathbf{c}_{(k)}$  and  $\sigma^2$ , is simply  $L(\mathbf{f}|\mathbf{c}_{(k)}, \sigma^2) = -\log p(\mathbf{f}|\mathbf{c}_{(k)}, \sigma^2)$  [19]. It can be argued that only discrete data have finite code-lengths. However, as recently noted in [20], these may be obtained by discretizing a density to an arbitrary precision; abuse of the term "code-length" is convenient and harmless.

The second fundamental fact is that the parameters themselves are also part of the code. A code word alone can not be decoded by itself; only a full knowledge of  $p(\mathbf{f}|\mathbf{c}_{(k)}, \sigma^2)$  (i.e., of its parameters) allows reconstructing the code and respective decoder. Accordingly, the MDL estimate (which unlike in Eqs. (5)-(6), depends on the unknown  $\sigma^2$ ) is

$$\left(\widehat{\mathbf{c}}_{(k)}, \widehat{\sigma}^2\right) = \arg \min_{\mathbf{c}_{(k)}, \sigma^2} \left\{ L(\mathbf{f}|\mathbf{c}_{(k)}, \sigma^2) + L(\mathbf{c}_{(k)}, \sigma^2) \right\}, \quad (7)$$

where  $L(\mathbf{c}_{(k)}, \sigma^2)$  is the parameter description length. We assume that each parameter has a constant description length, i.e.  $L(\mathbf{c}_{(k)}, \sigma^2) = \lambda k$ . The minimization in Eq. (7) can be split into three nested ones as

$$\min_k \left\{ \lambda k + \min_{\sigma^2} \left\{ \min_{\mathbf{c}_{(k)}} \left\{ -\log p(\mathbf{f}|\mathbf{c}_{(k)}, \sigma^2) \right\} \right\} \right\}.$$

The inner minimization corresponds to the ML estimate in Eq. (6). The one w.r.t.  $\sigma^2$ , yields its ML estimate, given the obtained estimate  $\widehat{\mathbf{c}}_{(k)}$ ; it is thus a function of  $k$  and we will denote it as  $\widehat{\sigma}^2_{(k)}$ . After some manipulation,

$$\min_{\sigma^2} \left\{ \min_{\mathbf{c}_{(k)}} \left\{ -\log p(\mathbf{f}|\mathbf{c}_{(k)}, \sigma^2) \right\} \right\} = \frac{N}{2} \log \left( 2\pi\epsilon \widehat{\sigma}^2_{(k)} \right)$$

where

$$\widehat{\sigma}^2_{(k)} = \|\mathbf{f} - \mathbf{B}_{(k)}^\dagger \mathbf{f}\|^2 / N \quad (8)$$

(recall that  $\mathbf{B}_{(k)}^\dagger = \mathbf{B}_{(k)} \mathbf{B}_{(k)}^\dagger$ ). Finally, dropping all additive constants,

$$\widehat{k} = \arg \min_k \left\{ \frac{N}{2} \log \widehat{\sigma}^2_{(k)} + \lambda k \right\}. \quad (9)$$

From  $\widehat{k}$ , we also immediately have  $\widehat{\mathbf{c}}_{(\widehat{k})} = (\mathbf{B}_{(\widehat{k})})^\dagger \mathbf{f}$ .

#### 3.2. The parameter description length

Specifying  $\lambda$  is one of the critical features of MDL-type criteria. Originally, MDL used the (asymptotical) code-length  $\lambda = \frac{1}{2} \log N$  for real parameters [19]; it makes sense when all parameters are estimated from all the data, i.e., their precision should/can increase with the data set size. Control points are local parameters, i.e., they only depend on a few data points; then, it is senseless for  $\lambda$  to grow with the number of data points. Alternatively, let  $\epsilon$  be the numerical accuracy with which the elements of  $\mathbf{c}_{(k)}$  are written (the minimal possible difference between values), and let  $w$  be the range they span; then, clearly,

$$\lambda k = k \log \frac{w}{\epsilon} = k (\log(w) - \log(\epsilon)). \quad (10)$$

We neglect the necessary rounding up to the closest integer. Eq. (10) is still not useful unless we know  $w$  and the optimal  $\epsilon$ . However, for curves on *digital planes* (digital images), a natural choice arises: since all coordinates are integers,  $\epsilon = 1$ , and since a curve is expressed by two scalar splines,  $\lambda = \log w_x + \log w_y = \log(w_x w_y)$ , where  $w_x$  and  $w_y$  are the width and height (in pixels) of the image. Finally, with  $\sigma^2_{x,(k)}$  and  $\sigma^2_{y,(k)}$  denoting the variances for the  $x$  and  $y$  coordinates,  $\sigma^2_{(k)} \equiv (\sigma^2_{x,(k)} + \sigma^2_{y,(k)})/2$ , and  $w \equiv w_x w_y$ , we have

$$\widehat{k} = \arg \min_k \left\{ N \log \widehat{\sigma}^2_{(k)} + k \log w \right\}, \quad (11)$$

which has the following reasonable properties: **(a)** for given  $N$  and  $w$ , increasing  $k$  decreases  $N \log \widehat{\sigma}^2_{(k)}$  but increases  $k \log w$ , thus forcing a compromise between the two terms; **(b)** a larger  $N$  (more data points) gives more relative weight to the error variance, i.e., more control points are allowed to try to decrease it; **(c)** when the coordinates range ( $w_x$ ,  $w_y$ , or both) increases, the variance term has less relative weight, i.e., a smaller fitting precision is imposed.

Finally, we point out that although MDL is not (conceptually) a Bayesian criterion [19], Eq. (7) is interpretable as a MAP estimator with the prior  $p(\mathbf{c}_{(k)}, \sigma^2) \propto \exp\{-k \log w\}$ .

In [3],  $p(k) \propto \exp\{k(1-m^2+m) \log(\frac{N}{2\pi})\}$  was proposed; there, the penalty on  $k$  grows with  $N$ , which (see above) is not reasonable. Moreover, it makes no sense for (perfectly valid) degree-1 splines (i.e., polygons) because for  $m = 1$ ,  $p(k)$  becomes an increasing function of  $k$ .

#### 3.3. Two examples

Figs. 1 – 4 report two examples of adaptive cubic (i.e.,  $m = 3$ ) B-spline fitting which, although not the final goal of this paper, is a useful tool in itself. Notice how the second

(more complex) shape requires more control points ( $\hat{k} = 20$ ) than the first one ( $\hat{k} = 13$ ).

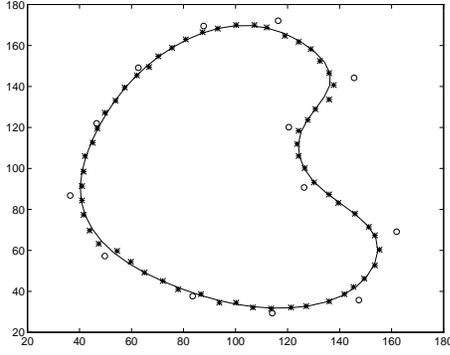


Figure 1. Data points (\*), estimated spline (solid line), and control points (o).

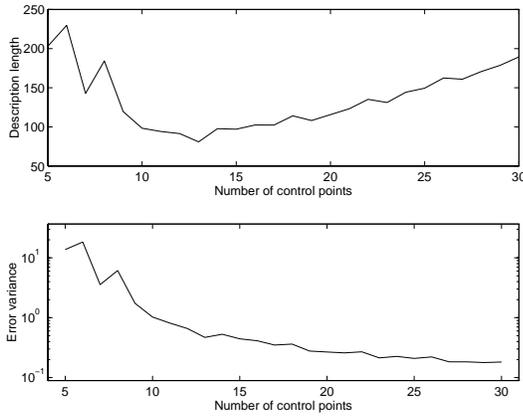


Figure 2. Description length (minimum at 13) and error variance plots (relative to Fig. 1).

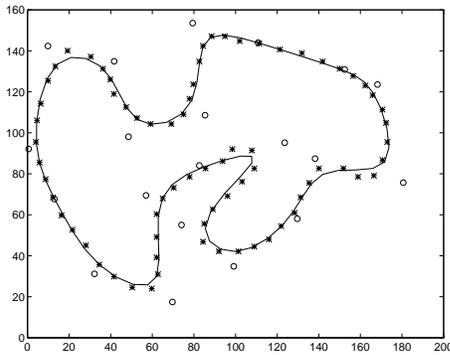


Figure 3. Data points (\*), estimated spline (solid line), and control points (o).

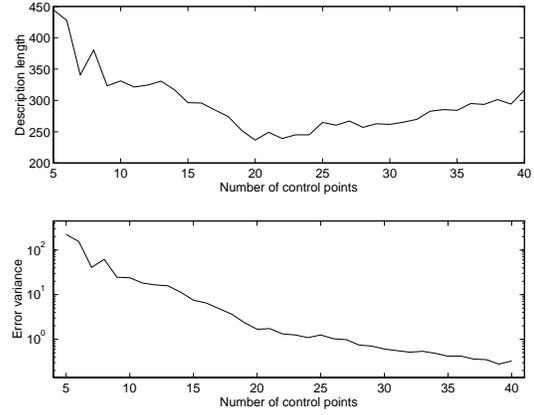


Figure 4. Description length (minimum at 20) and error variance plots (relative to Fig. 3).

## 4. B-splines for contour estimation

### 4.1. Observation model

Consider the observed image  $\mathbf{I}$  (a  $w_x \times w_y$  array of gray levels) as a random function of an object whose (closed) boundary  $\mathbf{v}$  is smooth in the sense that it is described by a discretized uniform periodic spline with some (unknown) control point vector  $\mathbf{c}_{(k)}$ , i.e.,  $\mathbf{v} = \mathbf{B}_{(k)}\mathbf{c}_{(k)}$ . Since the goal is to estimate  $k$  and  $\mathbf{c}_{(k)}$  from  $\mathbf{I}$ , full specification of the image model is needed. Although other choices are compatible with our approach, let us assume: **(a)** conditional independence, i.e., given the contour, all the pixels are independent; and **(b)** homogeneity, i.e., all the pixels in the inner (outer) region have identical distributions characterized by the same parameter vector  $\phi_{\text{in}}$  ( $\phi_{\text{out}}$ ). Then, with  $\phi \equiv [\phi_{\text{in}}, \phi_{\text{out}}]$ ,

$$p(\mathbf{I}|\mathbf{c}_{(k)}, \phi) = p(\mathbf{I}|\mathbf{v}, \phi) = \prod_{(i,j) \in \mathcal{I}(\mathbf{v})} p(I_{(i,j)}|\phi_{\text{in}}) \times \prod_{(i,j) \in \mathcal{O}(\mathbf{v})} p(I_{(i,j)}|\phi_{\text{out}})$$

with  $\mathbf{v} = \mathbf{B}_{(k)}\mathbf{c}_{(k)}$ , where  $I_{(i,j)}$  stands for 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  $\mathbf{v}$ ;  $p(I_{(i,j)}|\phi_{\text{in}})$  and  $p(I_{(i,j)}|\phi_{\text{out}})$  are the pixel-wise conditional probabilities, of the inner and outer regions, respectively.

### 4.2. Estimation criterion

Any adaptive scheme must estimate, not only  $\mathbf{c}_{(k)}$ , but also  $\phi$ , based on the likelihood function. An obvious choice would be the ML criterion; however, as in Section 2.2, it can not be used to estimate  $k$ . Then, we use the MDL criterion proposed above (recall that  $w = w_x w_y$ ):

$$\left( \widehat{\mathbf{c}}_{(k)}, \widehat{\phi} \right) = \arg \min_{\mathbf{c}_{(k)}, \phi} \left\{ -\log p(\mathbf{I}|\mathbf{c}_{(k)}, \phi) + k \log w \right\}.$$

This minimization can, as above, be rearranged into

$$\hat{k} = \arg \min_k \left\{ k \log w - \max_{\phi, \mathbf{c}_{(k)}} \{ \log p(\mathbf{I} | \mathbf{c}_{(k)}, \phi) \} \right\}; \quad (12)$$

however, unlike in the fitting problem above, the inner maximization can not be split into nested maximizations with respect to  $\mathbf{c}_{(k)}$  and  $\phi$ . By now, let us simply denote the result, which is a function of  $k$  and  $\mathbf{I}$ , as  $G(\mathbf{I}, k)$ ; then

$$\hat{k} = \arg \min_k \{ k \log w - G(\mathbf{I}, k) \}. \quad (13)$$

## 5. Algorithms

### 5.1. Solving for $\mathbf{c}_{(k)}$ , with fixed $k$ and $\phi$

Our first building block is the implementation of the inner maximization in Eq. (12) for fixed  $k$  and  $\phi$ . The maximization w.r.t.  $\mathbf{c}_{(k)}$  can be rewritten as a constrained maximization with respect to  $\mathbf{v}$ ,

$$\max_{\mathbf{c}_{(k)}} \{ \log p(\mathbf{I} | \mathbf{c}_{(k)}, \phi) \} = \max_{\mathbf{v} \in \mathcal{R}(\mathbf{B}_{(k)})} \{ \log p(\mathbf{I} | \mathbf{v}, \phi) \},$$

where  $\mathcal{R}(\mathbf{B}_{(k)})$  is the range space of  $\mathbf{B}_{(k)}$ , which means that the search is constrained to contours that can be written as  $\mathbf{v} = \mathbf{B}_{(k)} \mathbf{c}_{(k)}$ , for some  $\mathbf{c}_{(k)}$ . To solve it, we use the *gradient projection* method [16]. Of course, being an ascent algorithm, it may be stuck in local maxima; however, the experimental results show that this is seldom a problem with the adopted region-based model. Formally:

#### Algorithm 1

**Inputs:**  $k$ ,  $\phi$ , and an initial valid contour  $\hat{\mathbf{v}}^{(0)} \in \mathcal{R}(\mathbf{B}_{(k)})$ .

**Output:** a contour estimate  $\hat{\mathbf{v}} \in \mathcal{R}(\mathbf{B}_{(k)})$ .

**Initialization:** Build  $\mathbf{B}_{(k)}$ , compute  $\mathbf{B}_{(k)}^\perp$ , let  $n = 0$ .

**Step 1:** Compute a small step in the direction of the gradient with respect to the contour

$$\delta \mathbf{v} = \varepsilon \mathbf{sgn} \left( \nabla (\log p(\mathbf{I} | \mathbf{v}, \phi)) \Big|_{\mathbf{v} = \hat{\mathbf{v}}^{(n)}} \right); \quad (14)$$

$\mathbf{sgn}$  is a coordinate-wise vector sign function.

**Step 2:** Project  $\delta \mathbf{v}$  onto  $\mathcal{R}(\mathbf{B}_{(k)})$ , and update the contour estimate as

$$\hat{\mathbf{v}}^{(n+1)} = \hat{\mathbf{v}}^{(n)} + \mathbf{B}_{(k)}^\perp \delta \mathbf{v}. \quad (15)$$

**Step 3:** If  $\|\hat{\mathbf{v}}^{(n+1)} - \hat{\mathbf{v}}^{(n)}\| < \alpha$ , stop, output  $\hat{\mathbf{v}} = \hat{\mathbf{v}}^{(n+1)}$ ; if not, increment  $n$ , go back to Step 1.

Since the coordinates are (integer) pixel locations, the gradient in Step 1 is approximated by discrete differences. It is possible to show that this gradient is always normal to the contour [26]. Parameter  $\varepsilon$  should be kept small to avoid instabilities near the minima (in the examples ahead,  $\varepsilon = 1$ , although variable step methods can be devised).

### 5.2. Solving for $\phi$ and $\mathbf{c}_{(k)}$ , with fixed $k$

Here, we use an iterative estimation/maximization scheme having Algorithm 1 as one of its steps. Although conceptually related to the *expectation-maximization* (EM) algorithm [6], it is not an EM algorithm.

#### Algorithm 2

**Inputs:**  $k$ , and an initial valid contour  $\hat{\mathbf{v}}^{(0)} \in \mathcal{R}(\mathbf{B}_{(k)})$ .

**Outputs:** estimates  $\hat{\phi}$  and  $\hat{\mathbf{v}}$

**Initialization:** Let  $q = 0$ .

**Step 1:** Given  $\hat{\mathbf{v}}^{(q)}$ , compute the ML estimate  $\hat{\phi}^{(q)} \equiv [\hat{\phi}_{\text{in}}^{(q)} \ \hat{\phi}_{\text{out}}^{(q)}]$  according to

$$\hat{\phi}_{\text{in}}^{(q)} = \arg \max_{\phi_{\text{in}}} \left\{ \prod_{(i,j) \in \mathcal{I}(\hat{\mathbf{v}}^{(q)})} p(I_{(i,j)} | \phi_{\text{in}}) \right\}; \quad (16)$$

an equivalent expression yields  $\hat{\phi}_{\text{out}}^{(q)}$ .

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

**Step 3:** If  $\|\hat{\mathbf{v}}^{(q+1)} - \hat{\mathbf{v}}^{(q)}\| < \beta$ , stop and output  $\hat{\phi} = \hat{\phi}^{(q+1)}$  and  $\hat{\mathbf{v}} = \hat{\mathbf{v}}^{(q+1)}$ ; else, increment  $q$ , return to Step 1.

The exact form of Eq. (16) depends on the image model. In the experiments reported below, we use Gaussian distributions which, together with the independence assumption, lead to simple and computationally attractive estimates.

Finally, from Eqs. (12) and (13),  $G(\mathbf{I}, k) = \log p(\mathbf{I} | \hat{\mathbf{v}}, \hat{\phi})$ , with the  $\hat{\phi}$  and  $\hat{\mathbf{v}}$  provided by Algorithm 2. Its particular form depends on the image model; in the Gaussian case,

$$-G(\mathbf{I}, k) = \frac{N_{\text{in}}}{2} \log \hat{\sigma}_{\text{in}}^2(k, \mathbf{I}) + \frac{N_{\text{out}}}{2} \log \hat{\sigma}_{\text{out}}^2(k, \mathbf{I})$$

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

### 5.3. Solving with respect to $k$

The complete scheme simply proceeds as follows. For each  $k$  in a given set  $\{k_{\text{min}}, \dots, k_{\text{max}}\}$ , run Algorithm 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. From these, compute  $(k \log w - G(\mathbf{I}, k))$  for each  $k$ , find the minimum, and select the corresponding contour and parameter estimates.

## 6. Experiments

The first two examples (Figs. 5–8) use synthetic images. In Fig. 5 the regions have equal variance ( $\sigma_{\text{in}} = \sigma_{\text{out}} = 60$ ), but different mean ( $\mu_{\text{in}} = 80, \mu_{\text{out}} = 180$ ), while in Fig. 7 they have equal mean ( $\mu_{\text{in}} = \mu_{\text{out}} = 120$ ) and different variance ( $\sigma_{\text{in}} = 40, \mu_{\text{out}} = 120$ ). The description lengths (minima for  $k = 12$  and  $k = 10$ , respectively) are plotted in Fig. 8. The parameter estimates are: for Fig. 5,  $\hat{\mu}_{\text{in}} = 80.89, \hat{\mu}_{\text{out}} = 179.49, \hat{\sigma}_{\text{in}} = 61.05, \hat{\sigma}_{\text{out}} = 60.44$ ; for Fig. 7,  $\hat{\mu}_{\text{in}} = 120.35, \hat{\mu}_{\text{out}} = 119.72, \hat{\sigma}_{\text{in}} = 41.23, \hat{\sigma}_{\text{out}} = 119.21$ .

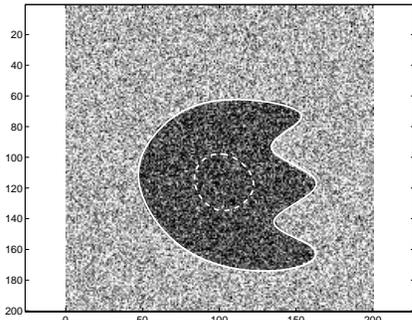


Figure 5. Synthetic image and estimated contour (initialization is shown by broken curve).

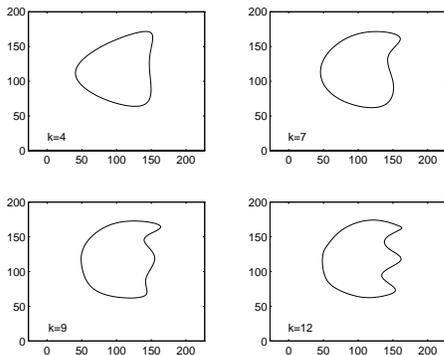


Figure 6. Evolution of the estimated shape (Fig. 5), for  $k = 4, k = 7, k = 9$ , and  $k = \hat{k} = 12$ .

The final examples consider three medical images: two MRI's (Figs. 9 and 10) and an intracoronary echographic image (Fig. 11). Note how the estimates are unaffected by the fact that the initial contours are near high-gradient areas.

We stress that the only user intervention is contour initialization which, as the examples show, is not critical; of course, for particular applications, *ad hoc* initialization methods have to be devised. Finally, we mention that the results were obtained with a MATLAB implementation; running times are from 1 to 5 min. (on a Sun SPARC 20).

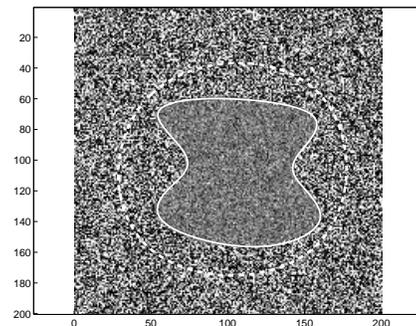


Figure 7. Synthetic image and estimated contour (initialization is shown by broken curve).

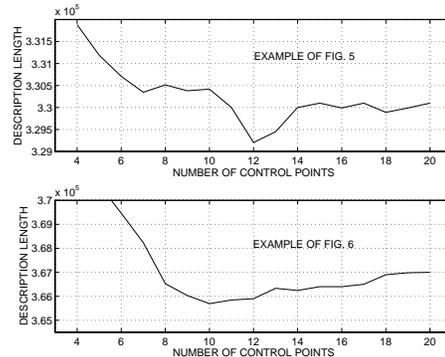


Figure 8. Evolutions of the description lengths for the examples of Figs. 5 and 7.

## 7. Concluding remarks

We have introduced a new approach to unsupervised smooth contour estimation based on a new adaptive B-spline representation. All the parameters governing the involved models (contour and image) are considered unknown and estimated from the data. Examples were presented, using synthetic and medical ultrasound images, showing the ability of the proposed method to estimate contours in an unsupervised manner, i.e. adapting to unknown shapes and observation parameters. In the case of the synthetic images, the good match between the estimated and the known true

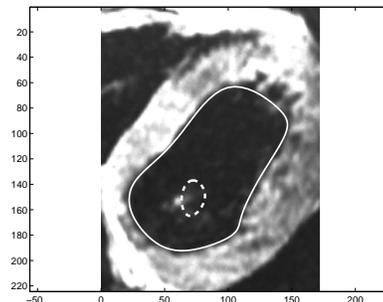


Figure 9. Heart MR image ( $\hat{k} = 10$ ).

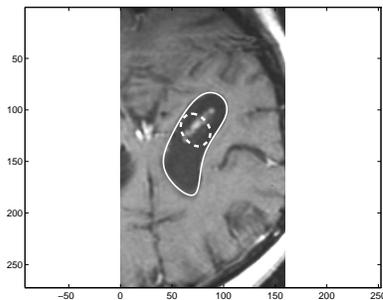


Figure 10. Brain MR image ( $\hat{k} = 9$ ).

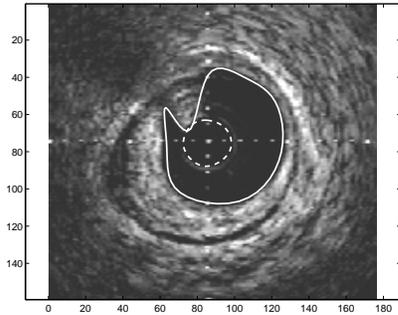


Figure 11. Intracoronary echo image ( $\hat{k} = 13$ ).

parameters testifies to the good performance of the approach.

Future work will consider more elaborate region models (e.g., different textures); the proposed formulation supports any image model with, of course, the corresponding impact on Eqs. (14) and (16). Adaptive non-uniform knot placement is also a goal which we are currently pursuing.

## References

- [1] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *Proc. IEEE Int. Conf. Comp. Vision - ICCV'95*, pp. 694-699, Cambridge, 1995.
- [2] A. Chakraborty, L. Staib, and J. Duncan. Deformable boundary finding influenced by region homogeneity. *Proc. IEEE Conf. Comp. Vision and Patt. Recog. - CVPR'94*, pp. 624-627, Seattle, 1994.
- [3] F. Cohen and J. Wang. Modeling image curves using invariant 3-D object curve models, *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 16, pp. 1-12, 1994.
- [4] L. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2D and 3D images. *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 15, pp. 1131-1147, 1993.
- [5] C. deBoor. *A Practical Guide to Splines*. Springer Verlag, N. York, 1978.
- [6] A. Dempster, N. Laird, D. Rubin. Maximum likelihood estimation from incomplete data via the EM algorithm. *Jour. Royal Stat. Soc. (B)*, v. 39, pp. 1-38, 1977.
- [7] G. Farin. *Curves and Surfaces for Computer Aided Geometrical Design*. Academic Press, Boston, 1990.
- [8] M. Figueiredo and J. Leitão. Bayesian estimation of ventricular contours in angiographic images. *IEEE Trans. on Med. Imag.*, vol. 11, pp. 416-429, 1992.
- [9] M. Figueiredo and J. Leitão. Unsupervised image restoration and edge location using GMRFs and the MDL principle. *IEEE Trans. on Image Proc.*, vol. 6, no. 5, May 1997.
- [10] M. Flickner, J. Hafner, E. Rodriguez, and J. Sanz. Periodic quasi-orthogonal spline basis and applications to least squares curve fitting of digital images. *IEEE Trans. on Image Proc.*, vol. 5, pp. 71-88, 1996.
- [11] U. Grenander. *General Pattern Theory: A Mathematical Study of Regular Structures*. Oxford University Press, Oxford, 1993.
- [12] U. Grenander, M. Miller. Representation of knowledge in complex systems. *Jour. Royal Stat. Soc. (B)*, vol. 56, pp. 1-33, 1994.
- [13] A. Jain, Y. Zhong, and S. Lakshmanan. Object matching using deformable templates. *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 18, pp. 267-277, 1996.
- [14] M. Jolly, S. Lakshmanan, and A. Jain. Vehicle segmentation and classification using deformable templates. *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 18, pp. 293-308, 1996.
- [15] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. Jour. of Comp. Vision*, vol. 1, pp. 259-268, 1987.
- [16] D. Luenberger. *Linear and Nonlinear Programming*. Addison Wesley, Reading (MA), 1984.
- [17] R. Malladi, J. Sethian, and B. Vemuri. Shape modeling with front propagation. *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 17, pp. 158-175, 1995.
- [18] T. McInerney and D. Terzopoulos. Topologically adaptable snakes. *Proc. IEEE Int. Conf. Comp. Vision - ICCV'95*, pp. 840-845, Cambridge, 1995.
- [19] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*. World Scientific, Singapore, 1989.
- [20] J. Rissanen. Fisher information and stochastic complexity. *IEEE Trans. Inf. Th.*, vol. 42, pp. 40-47, 1996.
- [21] R. Ronfard. Region-based strategies for active contour models. *Int. Jour. of Comp. Vision*, vol. 13, pp. 229-251, 1994.
- [22] P. Saint-Marc, H. Rom, and G. Medioni. B - spline contour representation and symmetry detection. *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 15, pp. 1191-1197, 1993.
- [23] L. Staib and J. Duncan. Boundary finding with parametrically deformable models. *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 14, pp. 1061-1075, 1992.
- [24] G. Storvik. A Bayesian approach to dynamic contours through stochastic sampling and simulated annealing. *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 16, pp. 976-986, 1994.
- [25] A. Yuille and P. Hallinan. Deformable templates. In A. Blake and A. Yuille (Eds), *Active Vision*, pp. 21-38. MIT Press, 1992.
- [26] S. Zhu and A. Yuille. Region competition. *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 18, pp. 884-900, 1996.