An Exact Method for Computing the Area Moments of Wavelet and Spline Curves

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

Abstract—We present a method for the exact computation of the moments of a region bounded by a curve represented by a scaling function or wavelet basis. Using Green's Theorem, we show that the computation of the area moments is equivalent to applying a suitable multidimensional filter on the coefficients of the curve and thereafter computing a scalar product. The multidimensional filter coefficients are precomputed exactly as the solution of a two-scale relation. To demonstrate the performance improvement of the new method, we compare it with existing methods such as pixel-based approaches and approximation of the region by a polygon. We also propose an alternate scheme when the scaling function is $\mathrm{sinc}(x)$.

Index Terms—Area moments, curves, splines, wavelets, Fourier, two-scale relation, box splines, wavelet-Galerkin integrals.

1 Introduction

Moments are standard descriptors of the shape of an object [1], [2], [3]; they easily yield features that are invariant to translation and rotation [4] or, more generally, to affine transformations, which makes them useful tools for pattern recognition. In the standard formulation, they are computed as surface integrals which requires raster scanning through the image. However, there are many instances where the boundaries of objects are described by parametric curves. This is the case, for example, when the objects are detected using parametric snakes which are represented using B-spline [5], [6], [7], [8] or wavelet basis functions [9], [10]. Another simple case is when the region is described as a polygon [11].

In this paper, we address the problem of computing the area moments of objects described by such parametric curves when the basis functions are scaling functions. The popular wavelet curve descriptors also fall into this class. The originality of our approach is that the computation is exact and also more direct than the conventional pixel-based method which requires an explicit labeling of the inner region of the curve prior to computation. Moreover, the pixel-based schemes suffer a low accuracy due to the loss of subpixel details in the rasterizing process. Also, the error in the area-based computation of moments is dependent on the orientation of the shape.

Since a polygon can be represented in terms of linear splines, the computation of moments by approximating the shape as a polygon [11], [12], [13] is a particular case of our approach. While the polygon method can be made as accurate as desired by increasing the number of segments, the convergence is slow because of the low approximation order of linear splines. Moreover, it is not suitable for computing the curvature, which is an interesting shape feature as it is invariant to rotation and translations, and can be easily normalized to scale changes. This motivates us to investigate higher order schemes where the curve is represented by smoother basis functions such as B-splines and other scaling functions that appear in wavelet theory [14], [15]. These type of basis functions also occur naturally when one seeks multiresolution representation of curves which are well suited for pattern recognition and shape simplification [16], [10].

The paper is organized as follows: In Section 2, we show how Green's Theorem can be used for the computation of the area moments of a parametric curve. In Section 3, we consider the computation of the moments of such a curve represented in spline or wavelet bases. Here, we also discuss the properties of the multidimensional kernel used in the computation of moments. In Section 4, we give the implementation details of the moment computation. In the following section, we deal with the precomputation of the kernel. In Section 6, we present an alternate implementation that works for any order moments, but it is rigorously exact only when the scaling function is $\mathrm{sinc}(x)$. This is especially interesting because it makes our method applicable to the Fourier representation of curves as well. In the last section, we compare the new method with the existing schemes such as approximation using polygons and rasterizing.

2 Preliminaries

2.1 Computation of Moments Using Green's Theorem

Green's Theorem relates the volume integral of the divergence of a vector field in a closed region to the integral of the field over the surface enclosing it. In this section, we show how it can be used to compute the moments of an area enclosed by a curve.

• *The authors are with the Biomedical Imaging Group, Swiss Federal Institute of Technology, Lausanne, CH-1015 Lausanne, EPFL, Switzerland. E-mail: {mathews.jacob, thierry.blu, michael.unser}@epfl.ch.*

Manuscript received 7 June 2000; revised 6 Feb. 2001; accepted 2 Mar. 2001. Recommended for acceptance by S. Sarkar.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 112243.

0162-8828/01/$10.00 © 2001 IEEE

Consider a closed region $\mathcal{V}$, bounded by a surface $\mathcal{S}$. Green's Theorem states that, for any vector field F,

$$\int_{\mathcal{V}} (\nabla.\mathrm{F})\, dV = \int_{\mathcal{S}} \mathrm{F}.\mathrm{dS}, \tag{1}$$

where dS is the unit vector pointing out of the surface $\mathcal{S}$. Assuming the volume has a constant cross-section bounded by the curve $\mathcal{C}$ and that the variation of the field along the z-direction is zero, we can restrict the theorem to two dimensions as,

$$\int_{\mathcal{S}} \left(\frac{\partial \mathrm{F}_x}{\partial x} + \frac{\partial \mathrm{F}_y}{\partial y} \right) dxdy = \oint_{\mathcal{C}} (\mathrm{F}_y dx - \mathrm{F}_x dy). \tag{2}$$

The first integral is evaluated over the area $\mathcal{S}$ enclosed by the curve and the second one along the curve $\mathcal{C}$ in the clockwise direction. The computation of the moments involves the evaluation of the integral $\int_{\mathcal{S}} x^m.y^n.dxdy$ on the surface bounded by the curve. This, by (2), is equivalent to

$$I_{m,n} = \oint_{\mathcal{C}} \frac{x^m y^{n+1}}{n+1}\, dx, \tag{3}$$

with $\mathrm{F} = \mathrm{e_y}(\frac{x^m y^{n+1}}{n+1})$; $\mathrm{e_y}$ denotes the unit vector along the y direction. Note that the choice of F is not unique. We choose the vector field F that makes the computation simple. Another possible choice that has the same computational complexity is $\mathrm{F} = -\mathrm{e_x}(\frac{x^{m+1} y^n}{m+1})$.

2.2 Parametric Representation of a Curve

A curve in the x — y plane can be represented in terms of an arbitrary parameter t as $\mathrm{r}(t) = (x(t), y(t))$. If the curve is closed, as discussed in the paper, the functions $x(t)$ and $y(t)$ are periodic.

When the curve $\mathcal{C}$ is represented as above, $\mathrm{r}(t)$ can be approximated efficiently as linear combinations of some basis functions, which makes the representation compact and easy to handle.

In this paper, we mainly focus on the representation of the function vector $\mathrm{r}(t)$ in a scaling function basis as

$$\mathrm{r}(t) = \sum_{k=-\infty}^{\infty} \mathrm{b}_k\, \varphi(t-k). \tag{4}$$

Here, b_k denotes the sequence of vector coefficients given by $\mathrm{b}_k = (c_k, d_k)$. If the period, M, is an integer, we have $\mathrm{b}_k = \mathrm{b}_{k+M}$. This reduces the infinite summations to

$$\mathrm{r}(t) = \sum_{k=0}^{M-1} \mathrm{b}_k\, \varphi_p(t-k), \tag{5}$$

where

$$\varphi_p(t) = \sum_{k=-\infty}^{\infty} \varphi(t - k.M). \tag{6}$$

In the context of wavelets, φ is called the *scaling function*; it satisfies the two-scale difference equation

$$\varphi(t) = \sum_k h(k) \varphi(2t-k), \tag{7}$$

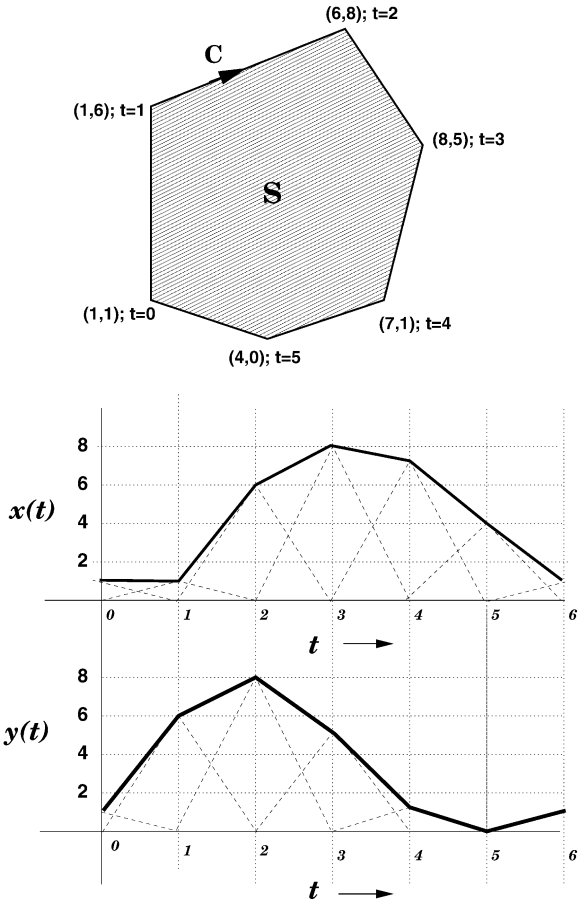


Fig. 1. Example: A polygon and its parametric representation.

where $h(k)$ is the mask of the corresponding refinement filter [14]. The scaling function representation enables us to have local control of the contour, which is desirable in many applications. It also permits a multiresolution representation of the curve [9], [17]. Moreover, the scaling function representation is affine-invariant; an affine transformation of the curve is achieved simply by transforming the coefficient vector $\mathrm{b}_k,\ k = 0, 1, \ldots, M-1$. This is because of the linearity of the representation and the partition of unity condition:

$$\sum_{k=-\infty}^{\infty} \varphi(t-k) = 1, \tag{8}$$

which is satisfied by all valid scaling functions in wavelet theory. Among the scaling functions, a case of special interest is $\varphi = \beta^n$, where β^n is the causal B-spline of degree n [18] defined by its Fourier transform

$$\hat{\beta}^s(\omega) = \left(\frac{1 - e^{-j\omega}}{j\omega} \right)^{s+1}. \tag{9}$$

This yields spline curves which are frequently used in computer graphics [19] and computer vision [6], [20], [21].

We now consider a simple example to illustrate this kind of curve representation. Given in Fig. 1 is a polygon and its parametric representation. The dotted lines show the linear

B-spline basis functions, $\beta^1(t-k)$ (tent function), multiplied by the corresponding coefficients.

The description of $\mathcal{C}$ in the scaling function basis is equivalent to a periodized wavelet representation [9]. This implies that, if we have a wavelet description of the curve, the scaling function coefficients at any scale can be obtained from the wavelet coefficients using the fast reconstruction equation described in [22]. Hence, the theory is sufficiently general to include the wavelet curve descriptors as well.

The representation of the curves in a sinc basis also falls in this class, as sinc is a valid scaling function. The description of the curve in the sinc basis as (5) is not efficient, as sinc has an infinite mask unlike most of the widely used scaling functions. It is well known (c.f. [23]) that the sinc interpolation of a periodic signal can be formulated into a numerically stable and efficient expression as

$$\mathrm{r}(t) = \sum_{k=-L}^{L} \mathrm{b}_k \exp\left(\frac{j2\pi kt}{M}\right), \tag{10}$$

where $2L+1 = M$, assuming M to be odd. A similar expression is obtained for even M as well. Here, b_k is the discrete Fourier transform of the vector sequence $\mathrm{r}(k)$. Note that (10) provides the Fourier series description of the curve, which is frequently used for the representation of closed curves [24], [25].

2.3 Differentiation of Scaling Functions

We will use the property that the kth derivative of a scaling function φ can be expressed as [26]

$$\varphi^{(k)}(x) = \Delta^k \varphi^{\{k\}}(x), \tag{11}$$

where $\varphi^{\{k\}}(x)$ denotes the scaling function whose mask is given by

$$H^{\{k\}}(z) = \left(\frac{2}{1+z^{-1}}\right)^k H(z);$$

$H(z)$ is the mask of φ. Δ denotes the backward difference operator, defined as $\Delta\,\eta(x) = \eta(x) - \eta(x-1)$.

The relation (11) follows from the fact that any mth order scaling function can be written as

$$\hat{\varphi}(\omega) = \underbrace{\left(\frac{1-e^{j\omega}}{j\omega}\right)^m}_{\hat{\beta}^{m-1}(\omega)} \hat{\gamma}(\omega),$$

where γ is a refinable distribution which does not satisfy the partition of unity. The mask of φ is $H(z) = \left(\frac{1+z^{-1}}{2}\right)^m H_\gamma(z)$. Note that $\left(\frac{1+z^{-1}}{2}\right)^m$ is the mask of β^{m-1} and H_γ the mask of γ. Differentiating φ with respect to x, k number of times ($k \le m$) yields

$$\begin{aligned}\varphi^{(k)}(x) \;&\xrightarrow{\mathcal{F}}\; (j\omega)^k \hat{\varphi}(\omega) \\ &= \big(1-e^{j\omega}\big)^k \underbrace{\left(\frac{1-e^{j\omega}}{j\omega}\right)^{m-k} \hat{\gamma}(\omega)}_{\hat{\varphi}^{\{k\}}(\omega)} \;\xrightarrow{\mathcal{F}^{-1}}\; \Delta^k \varphi^{\{k\}}(x).\end{aligned} \tag{12}$$

Thus, the mask of $\varphi^{\{k\}}(x)$ is

$$H^{\{k\}}(z) = \left(\frac{1+z^{-1}}{2}\right)^{m-k} H_\gamma(z) = \left(\frac{2}{1+z^{-1}}\right)^k H(z).$$

3 Computation of the Moments of an Area Bounded by a Parametrized Curve

To facilitate the understanding of our method, we first give a detailed derivation of the formula for the area of the region bounded by the curve. We then extend our formulation to the general case.

3.1 Computation of the Area

For the parametric representation of the curve, the area of the region is given by

$$I_{0,0} = \int_0^M y(t)\frac{dx(t)}{dt}\,dt. \tag{13}$$

When the curve is described in a scaling function basis as in (5), we have

$$I_{0,0} = \sum_{i,j=0}^{M-1} d_i c_j \int_0^M \varphi_p(t-i)\varphi_p'(t-j)dt, \tag{14}$$

where

$$\varphi_p'(t) = \frac{d\,\varphi_p(t)}{dt}.$$

Substituting for $\varphi_p'(t)$ from (6), we get

$$I_{0,0} = \sum_{i,j=0}^{M-1} d_i c_j \int_{-\infty}^{\infty} \varphi_p(t-i)\varphi'(t-j)dt, \tag{15}$$

which is equivalent to

$$I_{0,0} = \sum_{i,j=0}^{M-1} d_i c_j \underbrace{\int_{-\infty}^{\infty} \varphi_p(t-i+j)\varphi'(t)dt}_{g_0^p(i-j)}. \tag{16}$$

Again, substituting for φ_p from (6), we get the kernel $g_0^p(l)$ as the M periodized version of

$$g_0(l) = \int_{-\infty}^{\infty} \varphi'(t)\varphi(t-l)dt \tag{17}$$

as $g_0^p(l) = \sum_{k=-\infty}^{\infty} g_0(l+k.M)$. With the simplification (11), the above equation becomes

$$g_0(l) = \Delta f_0(l), \tag{18}$$

and

$$f_0(x) = \int_{-\infty}^{\infty} \varphi^{\{1\}}(t)\varphi(t-x)dt. \tag{19}$$

Note that, if $\varphi(t) = \varphi(\tau - t)$, then $f_0(x)$ can be written as the convolution $\big(\varphi^{\{1\}} * \varphi\big)(\tau + x)$. We prefer to represent the kernel g^p in terms of f due to its nice properties, discussed later.

For the example given in Fig. 1, we have

$$g_0(l) = \Delta\big(\beta^0 * \beta^1\big)(l+2) = \Delta(\beta^2)(l+2)$$
$$g_0 : (0.5, 0, -0.5); \quad l \in \{-1, 0, 1\},$$

where β^n is the causal B-spline function of degree n. Now, for the polygon, $c(k) : (1, 1, 6, 8, 7, 4)$ and $d(k) : (1, 6, 8, 5, 1, 0)$. Hence, by (16), we have

$$I_{0,0} = \frac{1}{2}\langle(6, 8, 5, 1, 0, 1), (5, 7, 1, -4, -6, -3)\rangle$$
$$= 42 \text{ units}.$$

Here, $\langle x_1, x_2\rangle$ stands for the ℓ_2 inner product given by $\sum_k x_1(k)x_2(k)$.

3.2 General Formula

Having shown how to compute the area, we proceed on to the general case. The formula for the computation of the general moments are given by the following theorem:

Theorem 1. *Let $\mathcal{C}$ be a closed curve in the x-y plane represented in the parametric form in a periodized scaling function basis as (4). Then, the (m, n)th order area moment of the region $\mathcal{S}$, bounded by the curve $\mathcal{C}$, given by*

$$I_{m,n} = \int_{\mathcal{S}} x^m y^n \, dxdy \quad \text{for } m, n \geq 0 \tag{20}$$

can be computed as

$$I_{m,n} = \frac{1}{n+1} \sum_{k\in\mathcal{R}} \sum_{\substack{\mathrm{i}\in\mathcal{R}^{m+1} \\ \mathrm{j}\in\mathcal{R}^n}} c_k \, \mathrm{c_i}^{[m]} \mathrm{d_j}^{[n+1]} g^p_{m+n}(\mathrm{i}-k, \mathrm{j}-k), \tag{21}$$

where $\mathcal{R}$ is the integer range $[0 \ldots M-1]$. The kernel g^p_{m+n} in (21) is

$$g^p_{m+n}(\mathrm{k}) = \int_{-\infty}^{\infty} \varphi'(t)\, \varphi_p(t-k_1) \ldots \varphi_p(t-k_{m+n+1})\, dt. \tag{22}$$

Here, $\mathrm{c}^{[m]}$ stands for the m-times tensor product[1] $c \otimes c \ldots \otimes c$ and $\mathrm{i}-k$ denotes the sequence

$$(i_1 - k, i_2 - k, \ldots i_{m+1} - k).$$

Proof. For a parametric curve, the evaluation of the (m, n)th order moment given by (20) can be reduced to

$$I_{m,n} = \frac{1}{n+1} \int_0^M x^m(t) y^{n+1}(t) \frac{dx(t)}{dt} dt \tag{23}$$

by (3). When the curve is described in a scaling function basis, we have

$$I_{m,n} = \frac{1}{n+1} \sum_{k\in\mathcal{R}} \sum_{\substack{\mathrm{i}\in\mathcal{R}^{m+1} \\ \mathrm{j}\in\mathcal{R}^n}} c_k \, \mathrm{c_i}^{[m]} \mathrm{d_j}^{[n+1]} \int_0^M \varphi_p(t-\mathrm{i}_1) \ldots$$
$$\varphi_p(t-\mathrm{i}_m)\varphi_p(t-\mathrm{j}_1) \ldots \varphi_p(t-\mathrm{j}_{n+1})\varphi'_p(t-k)dt. \tag{24}$$

Substituting for $\varphi'_p(t)$ from (6), we get

$$I_{m,n} = \frac{1}{n+1} \sum_{k\in\mathcal{R}} \sum_{\substack{\mathrm{i}\in\mathcal{R}^{m+1} \\ \mathrm{j}\in\mathcal{R}^n}} c_k \, \mathrm{c_i}^{[m]} \mathrm{d_j}^{[n+1]} \int_{-\infty}^{\infty} \varphi_p(t-\mathrm{i}_1) \ldots$$
$$\varphi_p(t-\mathrm{i}_m)\varphi_p(t-\mathrm{j}_1) \ldots \varphi_p(t-\mathrm{j}_{n+1})\varphi'(t-k)dt. \tag{25}$$

The integral in the above equation is equivalent to

$$\underbrace{\int_{-\infty}^{\infty} \varphi'(t)\varphi_p(t+k-\mathrm{i}_1)\ldots\varphi_p(t+k-\mathrm{i}_m)\varphi_p(t+k-\mathrm{j}_1)\ldots\varphi_p(t+k-\mathrm{j}_{n+1})dt}_{g^p_{m+n}(\mathrm{i}-k,\mathrm{j}-k)}. \tag{26}$$

Hence, the (m, n)th order moment is

$$I_{m,n} = \frac{1}{n+1} \sum_{k\in\mathcal{R}} \sum_{\substack{\mathrm{i}\in\mathcal{R}^{m+1} \\ \mathrm{j}\in\mathcal{R}^n}} c_k \mathrm{c_i}^{[m]} \mathrm{d_j}^{[n+1]} \; g^p_{m+n}(\mathrm{i}-k, \mathrm{j}-k). \tag{27}$$

□

As in the case of the area, the kernel g_p is obtained by the M-periodization of

$$g_{m+n}(\mathrm{k}) = \int_{-\infty}^{\infty} \varphi'(t)\varphi(t-k_1)..\varphi(t-k_{m+n+1}).dt, \tag{28}$$

where $\mathrm{k} \in \mathbb{Z}^{m+n+1}$. Expressing φ' in terms of $\varphi^{\{1\}}$, we get

$$g_{m+n}(\mathrm{k}) = f_{m+n}(\mathrm{k}) - f_{m+n}(\mathrm{k}-1), \tag{29}$$

where

$$f_{m+n}(\mathrm{x}) = \int_{-\infty}^{\infty} \varphi^{\{1\}}(t)\varphi(t-x_1)..\varphi(t-x_{m+n+1})dt, \tag{30}$$

where $\mathrm{x} = (x_1, x_2 \ldots, x_{m+n+1}) \in \mathbb{R}^{m+n+1}$. The kernel f has many interesting properties, which are discussed next.

3.3 Properties of the Kernel—f

1. **Finite Support**. As the kernel is an integral of products of the translates of finitely supported functions, it has a finite support as well. If the scaling function is continuous and has a support $[0, N]$, then the kernel will be supported on the integer points in the interval

$$I = [-N+1, N-2] \times \ldots$$
$$[-N+1, N-2] \times [-N+1, N-2]. \tag{31}$$

2. **Symmetry**. The fact that the kernel is obtained from the integration of similar translated scaling functions introduces a lot of symmetry. As (30) is symmetric with respect to the parameters $k_1, k_2, ..$, interchanging them will not affect the value of the kernel. This implies

$$f(\mathrm{k}) = f(\boldsymbol{\sigma}_i(\mathrm{k})), \tag{32}$$

where $\boldsymbol{\sigma}_i$ indicates all possible $(m+n+1)!$ permutation operators. In addition, if the scaling functions are symmetric as in the case of splines, we have

$$f(\mathrm{k}) = f(-\mathrm{k}). \tag{33}$$

Both these properties together imply $2((m+n+1)!)$ relations, which are used to accelerate the computation of the kernel as well as the moments.

1. $c^{[0]}$ is defined as the neutral element $c^{[0]} \otimes c^{[m]} = c^{[m]}$.

3. **Two-Scale Relation**. We now show that the kernel satisfies a two-scale relation, which is the key to our computational approach. This property follows from the fact that the scaling functions $\varphi(t)$ and $\varphi^{\{1\}}(t)$, from which the kernel is derived, satisfy two-scale relations. If we consider (30) and rewrite the φ and $\varphi^{\{1\}}$ in terms of the corresponding two-scale relations (cf. (7)), we get

$$f_{m+n}(\mathrm{k}) = \sum_{\mathrm{l}\in\mathbb{Z}^{m+1}} H_{m+n}(\mathrm{l}).f_{m+n}(2\mathrm{k}-\mathrm{l}), \tag{34}$$

where $\mathrm{k}\in\mathbb{Z}^{m+1}$. The mask H in the above equation is

$$H_m(l_1,l_2,..,l_m) = \frac{1}{2}\sum_k h_1(k).h(k-l_1)..h(k-l_m). \tag{35}$$

The z-transform of the mask is given by

$$H_m(z_1,z_2,..,z_m) = \frac{1}{2}H_1(\textstyle\prod_{k=1}^m z_k)\displaystyle\prod_{k=1}^m H(z_k^{-1}). \tag{36}$$

It is this property that enables us to compute the kernels exactly, by solving a linear system of equations. This technique, which is discussed later, is analogous to the computation of the integer (or dyadic rational) samples of a scaling function from the transition operator [14].

Note that a scaling relation similar to (34) was also considered by mathematicians in the context of the wavelet-Galerkin method for the computation of integrals involving products of scaling functions and their derivatives [27], [28]. The work of Dahmen and Micchelli is essentially theoritical; Restrepo and Leaf concentrated on numerical issues and proposed a solution which is equivalent to the computation of our kernel g_m instead of f_m. This slightly complicates the approach and also increases the dimensionality of the problem; this issue is discussed further in Section 5.1.

The above mentioned properties imply that the kernel can be computed exactly for any finitely supported scaling function, as discussed in Section 5. In the next section, we will give some examples for the kernels when the scaling functions are B-splines.

3.4 Examples with Splines

Splines possess nice approximation properties. The B-splines have the maximum approximation order among the class of functions that satisfy a two-scale relation with a given support. Hence, they give better local control of the contour. Moreover, they are symmetric, which facilitates the computation of the kernel and moments as discussed before. So, it is worthwhile to analyze the properties of the kernels for a spline representation of the curve. For the results used in this section, refer to [18].

We consider causal B-splines, as they satisfy a two-scale relation for all orders. The refinement filter for a B-spline of degree n is the binomial filter

$$h(k) = \frac{1}{2^n}\binom{n+1}{k}. \tag{37}$$

If we choose β^s, a B-spline of degree s, as φ, then $\varphi^{\{1\}} = \beta^{s-1}$; that is a spline of degree $s-1$. Hence, the kernel f as given by (30) is a **box spline** [29] sampled at the integers. In particular,

$$f_0(k) = \beta^{2s}(k+s+1). \tag{38}$$

The spline functions have a closed-form representation in the Fourier domain, which the kernels also inherit. By taking the continuous Fourier transform of (30), when the scaling function is a B-spline, we get

$$\hat{f}_n^s(\omega);\ \ \omega\in\mathbb{Z}^n = \hat{\beta}^{s-1}(|\omega|)\prod_{i=1}^n \hat{\beta}^s(\omega_i), \tag{39}$$

where $|\omega|$ stands for $\sum_{j=1}^n \omega_j$. By using Poisson's formula

$$\sum_k \hat{f}_n^s(\omega+2k\pi) = \frac{1}{2\pi}\sum_k f_n^s(k)e^{-2j\pi\omega k}, \tag{40}$$

we get the discrete Fourier transform of the kernel as the 2π-periodized version of (39).

We give some examples of kernels for the computation of the first three moments when we have a linear spline representation. For linear splines, the kernel $f_{m-1}(k_1,k_2,\ldots,k_m)$ is supported in the interval $[-1,0]\times[-1,0]\ldots[-1,0]$. The kernels are

$$f_0(k_1); k_1\in\{-1,0\} : \frac{1}{2}\cdot[1\quad 1], \tag{41}$$

$$f_1(k_1,k_2); k_1,k_2\in\{-1,0\} : \frac{1}{6}\cdot\begin{bmatrix}1 & 2\\ 2 & 1\end{bmatrix}, \tag{42}$$

$$\begin{cases} f_2(-1,k_2,k_3); k_2,k_3\in\{-1,0\} & : \frac{1}{12}\cdot\begin{bmatrix}1 & 1\\ 3 & 1\end{bmatrix}\\ f_2(0,k_2,k_3); k_2,k_3\in\{-1,0\} & : \frac{1}{12}\cdot\begin{bmatrix}1 & 3\\ 1 & 1\end{bmatrix}\end{cases}. \tag{43}$$

It is interesting to see that the computation of the moments using the linear spline kernel is the same as when the polygon is triangulated in a specified way and the moments of individual triangles added up as in [11].

We also give the kernel f_0 for the cubic spline representation.

$$f_0(k_1); k_1 = -3,\ldots 2 : \frac{1}{720}\cdot[1,\ \ 57,\ \ 302,\ \ 302,\ \ 57,\ \ 1]. \tag{44}$$

The higher order kernels are omitted due to space constraints. They can be downloaded from http://bigwww.epfl.ch/jacob.

4 Implementation

In this section, we analyze equation (21) and simplify it for faster computation. We start with the simplest case: the area of the region.

The area bounded by the curve (cf. (15)) is computed as

$$I_{0,0} = \sum_{k=0}^{M-1} c_k^p \sum_{l=-N+1}^{N-2} d_{k+l}^p g_0(l), \tag{45}$$

where g_0 is given by (17). The sequences c_k^p and d_k^p are M-periodized versions of the coefficients c_k and d_k with respect to the period M. This is simply because convolving a nonperiodized sequence with a periodized kernel is equivalent to convolving a periodized sequence with a nonperiodized kernel. We have also reduced the range of summation of the inner sum to $-N+1$ to $N-2$, which is typically much less than the range 0 to $M-1$. Similarly, for the higher order moments all the summations, except the outer one, are in the range $-N+1$ to $N-2$.

From (45), we see that the computation of the area involves just a filtering operation by $g(-l) = g^T(l)$, followed by an inner product. This can be written as,

$$I_{0,0} = \langle c^p, g_0^T * d^p \rangle, \tag{46}$$

where $\langle .,. \rangle$ stands for the inner product $\langle c, d \rangle = \sum_{k=0}^{M-1} c(k)d(k)$. With a similar notation, the computation of the other moments are given as

$$I_{m,n} = \frac{\langle c^p, g_{m+n}^T * (c^{p[m]} \otimes d^{p[n+1]}) \rangle}{n+1} \tag{47}$$

$$= -\frac{\langle d^p, g_{m+n}^T * (c^{p[m+1]} \otimes d^{p[n]}) \rangle}{m+1}. \tag{48}$$

As the $(m+n+1)$-D sequence is separable, the filtering operation is much simpler than the usual $(m+n+1)$-dimensional filtering.

The complexity in the computation of the moment $I_{m,n}$ is $M.(2N-2)^{(m+n+2)}$, without taking the symmetries into account. Thus, for basis functions with small support and reasonable m and n, the complexity is quite managable.

5 Computation of the Kernel

In this section, we propose two schemes for computing the kernel. An exact space domain scheme and an approximate one in the Fourier domain.

5.1 Exact Method

In this scheme, we compute the kernels in space domain making use of the properties of kernels discussed before. We start with the computation of f_0 and later extend it to the general case. Making use of the finite support property, the two-scale relation (34) can be rewritten in the matrix form as,

$$\mathbf{A_0}.\mathrm{f}_0 = \mathrm{f}_0, \tag{49}$$

where $\mathbf{A_0}$ is the square matrix with coefficients $[\mathbf{A_0}]_{k,l} = H_0(2k-l)$ and f_0 is the vector whose elements are $f_0(n)$. As the support of f_0 is $[-N+1, N-2]$, the indices of $\mathbf{A_0}$ run from $-N+1$ to $N-2$.

It can be seen from (49) that f_0 is an eigen-vector of the matrix $\mathbf{A_0}$, with eigen-value 1. Solving for f_0 is equivalent to solving for a vector which falls in the nullspace of $(\mathbf{A_0} - \mathbf{I})$, where $\mathbf{I}$ is the identity matrix. Since $\mathrm{f}_0 \neq 0$, $\mathbf{A_0}$ must have the eigen-value 1, which is in general single. This provides f_0 up to a constant which is further set by the normalization identity

$$\sum_k f_0(k) = 1, \tag{50}$$

which can be seen from (19). This is because the function $\varphi(x)$ has at least an approximation order of one [14], which implies $\sum_k \varphi(x+k) = 1$. One of the equations in $(\mathbf{A_0} - \mathbf{I}).\mathrm{f}_0 = 0$ can be substituted for by the (50) to yield the system of equations given by

$$\mathbf{B}.\mathrm{f}_0 = \mathrm{y}; \tag{51}$$

$\mathbf{B}$ is the matrix obtained by substituting one of the rows of $(\mathbf{A_0} - \mathbf{I})$ with the row vector $[1, 1, \ldots, 1]$ and y is given by $[0, 0, 0 \ldots, 0, 0, 1]^T$ c.f [30]. Now, $\mathbf{B}$ is a full rank matrix and, hence, the eigen-vector f_0 can be solved by matrix inversion.

To represent the two-scale relations of the higher order kernels in the matrix form, we introduce a one-to-one function $\rho : [-N+1, N-2]^m \mapsto [0, (2N-2)^m - 1]$. Using this function, (34) can be rewritten as

$$f_m(\rho^{-1}(k)) = \sum_{l=0}^{(2N-2)^{m+1}-1} H_m\big(2\rho^{-1}(k) - \rho^{-1}(l)\big) f_m\big(\rho^{-1}(l)\big),$$

which is a linear system of equations. This can be written in the matrix form as

$$\mathbf{A_m}\mathrm{f}_m = \mathrm{f}_m, \tag{52}$$

where $[\mathbf{A_m}]_{i,j} = H_m(2\rho^{-1}(i) - \rho^{-1}(j))$ and $\mathrm{f}_m(i) = f(\rho^{-1}(i))$. This equation is of the same form as (49) and can be solved in the same way, with the normalization constraint $\sum_i \mathrm{f}_m(i) = 1$.

Let us now compare our computational solution with the method developed for computing g_m in the context of wavelet-Galerkin approach [27]. For a scaling function of support N, the kernel g_m is zero outside the interval

$$\begin{aligned} I' = &[-N+1, N-1] \times \ldots \\ &[-N+1, N-1] \times [-N+1, N-1]. \end{aligned} \tag{53}$$

as compared to f_m whose support is given by (31). Thus, the direct computation of g_m involves a linear system with $(2N-1)^m$ variables as compared to $(2N-2)^m$ for f_m in our case. For 3-dimensional kernels involving cubic splines, we achieve a 40 percent reduction in the number of equations. As the computational complexity in inverting a linear system is proportional to the third power of the number of equations, this implies a performance improvement of around 5 times. The approach becomes even more rewarding for higher order kernels. Moreover, the normalization constraint (50) that we use to make the system full rank is much more straightforward than the corresponding relation for the derivative functions.

Note that this simplification is covered by Dahmen and Michelli's general theory for integrals of multidimensional scaling functions [28]. This is because the mask of any mth order $1-D$ scaling function can be always factored as proposed in [28, Corollary 3.3]. In the case of wavelet-Galerikin integrals, the performance improvement can even more substantial depending on the number of derivatives.

5.2 Approximate Method for Splines

Because the spline kernel has a closed-form expression in the frequency domain, the kernel can be obtained by taking the inverse DFT of the above mentioned Fourier transform (40) sampled at an appropriate rate; we make use of the finite support property of the kernel. As sinc is a decaying function, the periodization of the Fourier transform may be approximated with an appropriately truncated sum to achieve any desired accuracy. This is because we can have an upper bound for the error that is a decreasing function of the summation range. Moreover, the symmetries of the kernel discussed before may be used for the efficient computation of the box spline kernels as in [31].

However, this technique, besides being approximate, can be used only for scaling functions that have a closed form expression in the frequency domain, i.e., splines in practice. This scheme may be useful to precompute the spline kernels for very high order moments, where the exact scheme can be computationally expensive.

6 Computation of the Area Moments Using Riemann Sums

An alternate approach to compute the moments is to approximate the integral (3) by a Riemann sum:

$$I_{m,n} = \frac{1}{(n+1)P} \cdot \sum_{l=0}^{MP-1} [x_{\text{int}}(l/P)]^m . [y_{\text{int}}(l/P)]^{n+1} . [x'_{\text{int}}(l/P)], \tag{54}$$

where P is an appropriate oversampling factor. We show in this section that this quadrature formula is exact when the curves are described in a sinc basis. For other representations, it can be used for the approximate computation of higher order moments.

6.1 Sinc Representation of the Curve

A curve represented in a sinc basis also falls into the framework of Theorem 1 because $\text{sinc}(x)$ is a valid scaling function. However, computing the moments as described in Section 4 is expensive as the mask of the sinc function is not finitely supported. We remind the reader that the representation of a periodic signal in the sinc basis is equivalent to the Fourier representation as seen in (10).

In this particular case, the moments can be computed exactly and more efficiently using (54), where the oversampling factor, P, is any integer greater than $\frac{m+n+2}{2}$.

Proposition 1. *The quadrature formula (54) is exact for the sinc representation provided that* $P \geq \lceil \frac{m+n+2}{2} \rceil$.

The continuously defined functions $x_{\text{int}}(t)$ and $y_{\text{int}}(t)$ are obtained by interpolating the sample values of the curve at the integers, using the periodized sinc function. The computation is exact because we implicitly assume that the functions $x(t)$ and $y(t)$ are bandlimited functions, with bandwidth $B = 2\pi$.

Proof. The integral (3) can be considered as an $L_{2\,(0,M)}$ inner product of two functions, which are $\lceil \frac{m+n+2}{2} \rceil$ and $\lfloor \frac{m+n+2}{2} \rfloor$ fold[2] products of the corresponding band-limited functions. Hence, they are bandlimited by $B' = B\lceil \frac{m+n+2}{2} \rceil$ and $B'' = B\lfloor \frac{m+n+2}{2} \rfloor$, respectively. So, these functions are exactly represented in the basis $\{\text{sinc}(Px - k), \forall k \in \mathbb{Z}\}$, where $2\pi P \geq B'$. Because the sinc basis is orthogonal, the $L_{2\,(0,M)}$ inner product is equivalent to the $\ell_{2\,(0,MP-1)}$ inner product. Hence, it is sufficient to compute the discrete summation instead of the integral. Finally, the sinc function is interpolating, so that the coefficients of the basis functions are the resampled curve values and, hence, the result (54). □

2. $\lfloor x \rfloor$ and $\lceil x \rceil$ denote the floor and the ceil operators, operating on a fraction x to yield the lower and upper integers that bound x.

Using the equivalence of the sinc and the Fourier representations, we can compute the interpolated samples efficiently with a MP point inverse FFT of the Fourier coefficients c_k and d_k.

We will compare the sinc moment estimator with the scaling-function-based moment estimator in the next section. One disadvantage of the Fourier(sinc) representation of curves is the loss of local control property that we were having with the finitely supported scaling functions.

The complexity in the computation of the moments in this scheme is $MP(3\log(MP) + (m+n+2))$. Here, $3MP\log(MP)$ is the cost of the inverse FFT of the sequences c_k, d_k, and $k.c_k$, and $(m+n+2)MP$ corresponds to the multiplications.

6.2 Spline Representation of the Curve

The quadrature formula (54) is also applicable to the spline representation, provided that the functions $x_{\text{int}}(t)$ and $y_{\text{int}}(t)$ are obtained by interpolating the integer sample values, using the corresponding B-spline functions. This scheme is no longer exact, but it may be a viable alternative for computing the higher order moments. The necessary condition for the computation to be reliable is that the Fourier transform of the B-spline function is essentially bandlimited to $2\pi P$, where P is the oversampling factor. The error in the moments computed with the approximate method is, thus, proportional to the residual energy of the B-spline function in the corresponding outband. As the Fourier transform of the B-spline is a decaying function of the frequency, the error will be a decaying function of P as well. Thus, any desirable accuracy may be achieved by choosing P sufficiently large.

The complexity of the spline quadrature formula is $\mathcal{O}(M(m+n+2)(m+n+2+3N)P)$, where $M(m+n+2)P$ is the total number of resampled points. The evaluation of the spline representation requires N multiplications to obtain one resampled point from the corresponding B-spline representation. Then, the computation of the discrete sum costs $m+n+2$ multiplications per resampled point. Interestingly, the approximate scheme will give better results for higher order splines as these functions will become bandlimited as the order tends to infinity [32].

7 Experiments and Results

In this section, we compare the new technique with the existing ones: approximation using polygons and rasterizing. We first consider the exact scheme proposed in Section 4. We try to estimate the parameters of a known

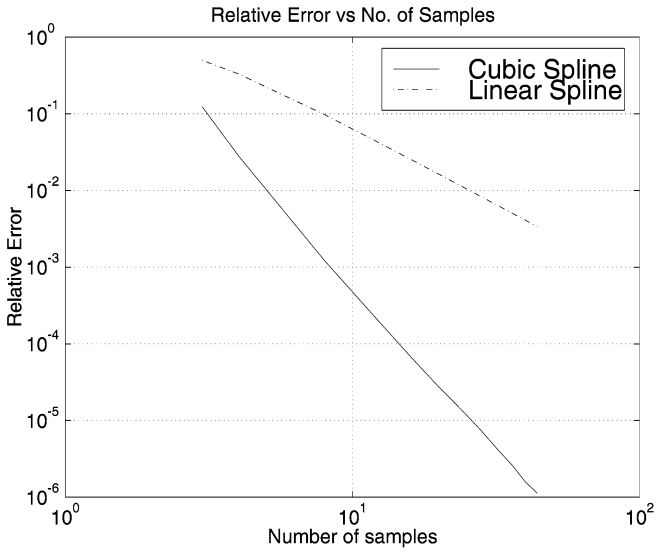


Fig. 2. Comparison of moment estimators.

ellipse and choose the relative error in the parameters as the criterion of comparison.

Our preferred choice is to represent the curve in a cubic B-spline basis due to its nice approximation properties and minimum curvature properties. To compare it with the approximation of the region as a polygon, the ellipse is sampled uniformly and the samples are interpolated using the two techniques (linear and cubic splines). The average relative error in the three centered second order moments versus the number or samples are plotted in Fig. 2. It can be seen that the relative error is much smaller for the cubic spline interpolation even at low sampling rates and that it exhibits a faster decay.

In the traditional scanning approach, the ellipse is scanned along the x and y axes with a step size Δ and the monomials are computed at the grid points assigned to the interior of the curve. Fig. 3 shows the decay of the average relative error for an ellipse versus $\frac{\sqrt{\text{Area}}}{\Delta}$ for three different

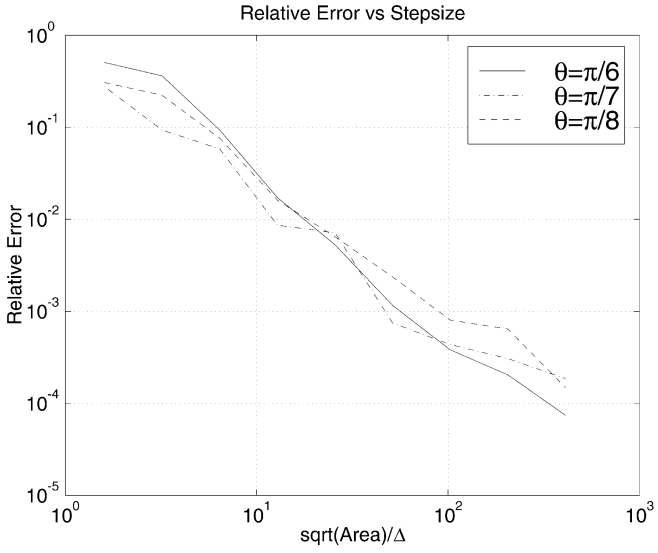


Fig. 3. Variation of error versus $\frac{1}{\Delta}$ in a raster scan moment estimator.

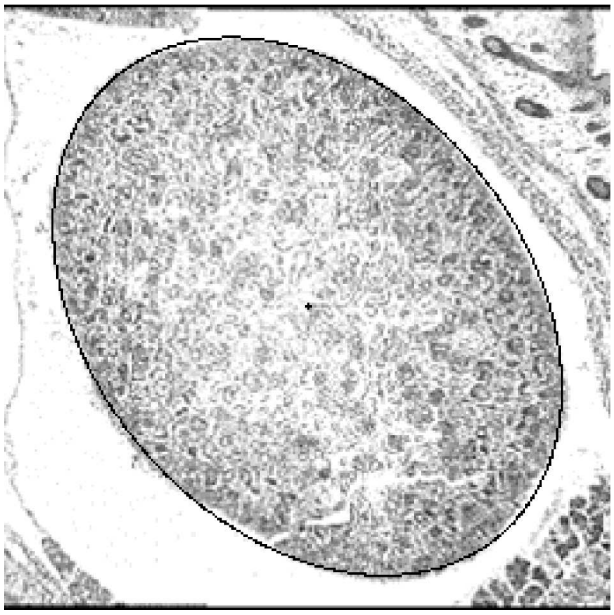


Fig. 4. Estimated ellipse for a real image.

orientations. The plot clearly shows the dependence of the accuracy on the orientation of the ellipse.

It can be seen that to achieve a relative error of 0.1 percent the interior of the ellipse has to be sampled at about 3,600 points, whereas to achieve the same error using the cubic spline interpolation we need only around nine points on the curve. In comparison, the polygon method (linear spline) requires more than 40 samples to have a similar error. More interesting is the case when the interior of the ellipse has to be sampled at about 2.5×10^5 points to achieve an error of 0.002 percent while the cubic splines require only 25 samples to achieve the same accuracy. In Fig. 4, we show the ellipse corresponding to the second order moments of the central structure in the image. The contour of the object was estimated using a snake where the curve was represented parametrically in terms of cubic B-splines; the moments are computed using our algorithm. Note that the fit is astonishingly good.

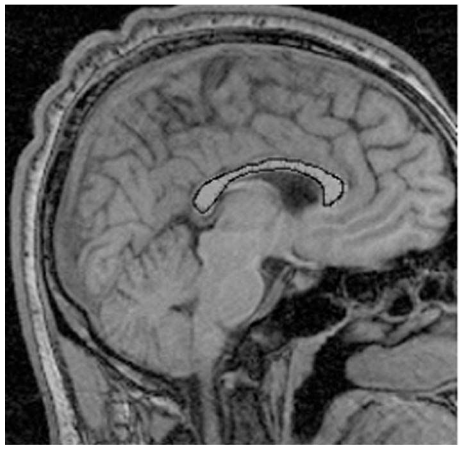


Fig. 5. Shape of corpus callosum represented using a cubic B-spline curve with 20 knot points.

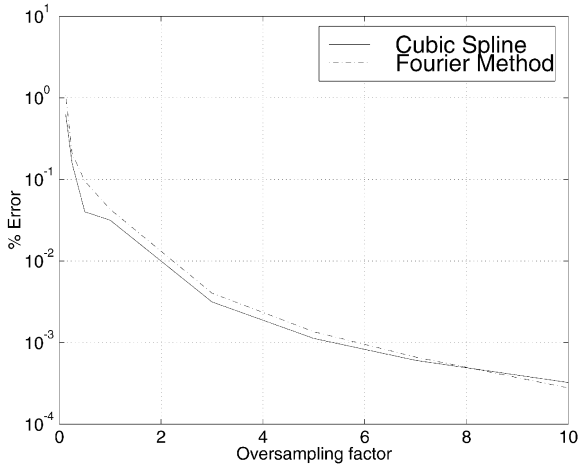


Fig. 6. Comparison of Fourier Estimator with cubic spline estimator.

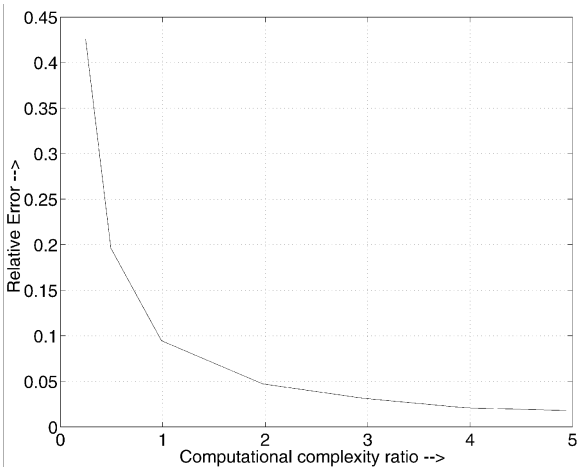


Fig. 7. Relative error versus relative computational complexity.

Having observed that the cubic spline estimator performs better than the polygon method, we now compare it with the Fourier (sinc) technique proposed in Section 6. It is not fair to use the ellipse as we did before because it can be represented exactly in a Fourier series representation with $L = 2$. So, we choose the real shape of corpus callosum shown in Fig. 5, represented in a linear spline basis with 39 knot points as the reference shape. This shape was resampled at different rates and these points were interpolated using cubic B-spline and Fourier representations, respectively. The moments of the corresponding curves were calculated using the respective algorithms discussed before. Fig. 6 shows the decay of the relative error with the resampling rate for both representations. We observe that the spline estimator is better than the Fourier estimator for small sampling rates, while the Fourier estimator performs better at very high sampling rates (typically more than eight times the number of points used for the description of $\mathcal{C}$). In the example considered, the Fourier method performs better when the shape of corpus callosum is represented with around 312 samples.

To evaluate the performance of the approximate scheme introduced in Section 6.2, we now consider the case where the corpus callosum is represented by a cubic B-spline curve with 20 knot points. The relative error in the computation of the second order moments by the quadrature formula as a function of its relative computational complexity (proportional to P) is shown in Fig. 7; here, the reference method is the kernel-based computation, which is exact. Our results indicate that, for the second order moments, the error of the quadrature formula is quite substantial (e.g., 9.4 percent). Thus, it is not advantageous for computing the lower order moments. However, the quadrature formula will eventually start to pay off for higher order moments, because its cost increases only quadratically with the degree as compared to exponentially for the kernel-based method.

8 Conclusion

In this paper, we have presented a new approach for the computation of the moments of a curve described in a wavelet or scaling function basis. It is especially useful for objects detected using parametric snakes. The main advantages of the proposed scheme over the conventional methods are:

- the exactness of the computation,
- its independence of the orientation of the shape, and
- the consistency with the snake model and the fact that it is the most direct method available.

In addition, the method is reasonably fast and easy to implement.

We recommend using our exact kernel-based approach for computing the lower order moments (typically $m + n \leq 2$) for which the kernels are available. For higher order moments, we have proposed a quadrature formula that approximates the continuous integrals with Riemann sums. The latter method is exact for the sinc basis functions; otherwise, it can be made as accurate as desirable by resampling the model at a finer rate (P sufficiently large).

Acknowledgments

The authors would like to thank the anonymous reviewer for pointing us to references [28] and [27] on the wavelet-Galerkin method which deal with the computation of wavelet related integrals. This work was supported by the Swiss National Science Foundation under grant 2100-053540.

References

[1] S. Rad, K.C. Smith, and B. Benhabib, "Application of Moment and Fourier Descriptors to the Accurate Estimation of Elliptical Shape Parameters," *Proc. IEEE Int'l Conf. Acoustics, Speech, Signal Process*, vol. 4, pp. 2465–2468, 1991.

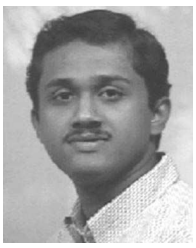
[2] R. Desai, R. Cheng, and H.D. Cheng, "Pattern Recognition by Local Radial Moments," *Proc. 12th IAPR Int'l Conf. Pattern Recognition*, 1994.

[3] K. Tsirikolias and B.G. Mertzios, "Statistical Pattern Recognition Using Efficient Two-Dimensional Moments with Applications to Character Recognition," *Pattern-Recognition*, vol. vol. 26, pp. 877–882, 1993.

[4] L.X. Shen and Y.L. Sheng, "Noncentral Image Moments for Invariant Pattern Recognition," *Optical Eng.*, vol. 34, no. 11, pp. 3181–3186, 1995.

[5] P. Brigger, J. Hoeg, and M. Unser, "B-Spline Snakes: A Flexible Tool for Parametric Contour Detection," *IEEE Trans. Image Process*, vol. 9, pp. 1484–1496, Sept. 2000.

[6] J.Y. Wang and F.S. Cohen, "3D Object Recognition and Shape Estimation from Image Contours Using B-Splines, Unwarping Techniques and Neural Network," *Proc. IEEE Int'l Joint Conf. Neural Networks,* 1991.
[7] M. Flickner, H. Sawhney, D. Pryor, and J. Lotspeich, "Intelligent Interactive Image Outlining Using Spline Snakes," *Proc. 28th Asilomar Conf. Signals, Systems, and Computers,* 1994.
[8] M. Figueiredo, J. Leitao, and A.K. Jain, "Unsupervised Contour Representation and Estimation Using B-Splines and a Minimum Description Length Criterion," *IEEE Trans. Image Process,* vol. 9, pp. 1075–1087, 2000.
[9] G.C.H. Chuang and J. Kuo, "Wavelet Descriptor of Planar Curves: Theory and Applications," *IEEE Trans. Image Process,* vol. 5, pp. 56–70, 1996.
[10] Y. Wang, S.L. Lee, and K. Toraichi, "Multiscale Curvature Based Shape Representation Using Bspline Wavelets," *IEEE Trans. Image Process,* vol. 8, pp. 1586–1592, 1999.
[11] M. Singer, "A General Approach to Moment Calculation for Polygons and Line Segments," *Pattern Recognition,* vol. 26, pp. 1019–1028, Jan. 1993.
[12] S.F. Bockman, "Generalising the Formula for Areas of Polygons to Moments," *Am. Math. Monthly,* vol. 96, pp. 131–133, Feb. 1989.
[13] N.J.C. Strachan, P. Nesvadba, and A.R. Allan, "A Method for Working Out the Moments of a Polygon Using an Integration Technique," *Pattern Recognition Letters,* vol. 11, pp. 351–354, May 1990.
[14] G. Strang and T.Q. Nguyen, *Wavelets and Filter Banks.* Wellesley-Cambridge Press, 1996.
[15] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding.* Prentice Hall, 1995.
[16] C. Fermuller and W. Kropatsch, "Hierarchical Curve Representation," *Proc. 11th IAPR Int'l Conf. Pattern Recognition,* 1992.
[17] J.P. Antoine, D. Barache, R.M. Cesar Jr., and L.da. Fontoura Costa, "Shape Characterisation with the Wavelet Transform," *Signal Processing,* vol. 62, pp. 265–290, 1997.
[18] M. Unser, "Splines: A Perfect Fit for Signal and Image Processing," *IEEE Signal Processing Magazine,* vol. 16, pp. 22–38, 1999.
[19] R.H. Bartels, J.C. Beatty, and B.A. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling.* Morgan Kauffmann, 1987.
[20] Z. Huang and F.S. Cohen, "Affine-Invariant B-Spline Moments for Curve Matching," *IEEE Trans. Image Process,* vol. 5, pp. 1473–1480, 1996.
[21] F.S. Cohen and J.Y. Wang, "Modeling Image Curves Using Invariant 3-D Object Curve Models, a Path to 3-D Recognition and Shape Estimation from Image Contours, Part 1," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 16, pp. 1–12, 1994.
[22] S. Mallat, "A Theory for Multiresolution Signal Decomposition: the Wavelet Representation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 11, pp. 674–693, 1989.
[23] F. Candocia and J.C. Prince, "Comments in Sinc Interpolation of Discrete Periodic Signals," *IEEE Trans. Signal Process,* vol. 46, pp. 2044–2047, 1998.
[24] A. Chakraborthy, L.H. Staib, and J.S. Duncan, "Deformable Boundary Finding in Medical Images by Integrating Gradient and Region Information," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, pp. 859–870, 1996.
[25] L.H. Staib and J.S. Duncan, "Boundary Finding with Parametrically Deformable Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 14, pp. 1061–1075, 1992.
[26] T. Blu, "Iterated Filter Banks with Rational Rate Changes Connection with Discrete Wavelet Transforms," *IEEE Trans. Signal Process,* vol. 41, pp. 3232–3244, Dec. 1993.
[27] J.M. Restrepo and G.K. Leaf, "Inner Product Computations Using Periodized Daubechies Wavelets," *Int'l J. Numerical Method in Eng.,* vol. 40, pp. 3557–3578, 1997.
[28] W. Dahmen and C.A. Micchelli, "Using the Refinement Equation for Evaluating Integrals of Wavelets," *SIAM J. Numerical Analysis,* vol. 30, pp. 507–537, 1993.
[29] C. de Boor, K. Hollog, and S. Riemenschneider, *Box Splines.* Springer-Verlag, 1998.
[30] T. Blu and M. Unser, "Quantitative Fourier Analysis of Approximation Techniques: Part II—Wavelets," *IEEE Trans. Signal Processing,* vol. 47, pp. 2796–2806, Oct. 1999.
[31] M.D. McCool, "Optimised Evaluation of Box Splines via the Inverse fft," *Graphics Interface,* 1995.
[32] A. Aldroubi, M. Unser, and M. Eden, "Cardinal Spline Filters: Stability and Convergence to the Ideal Sinc Interpolator," *Signal Processing,* vol. 28, pp. 127–138, 1992.



Mathews Jacob received the ME degree in signal processing from the Indian Institute of Science, Bangalore in 1999. Currently, he is a research assistant at the Biomedical Imaging Group at EPFL (Swiss Federal Institute of Technology), Lausanne, Switzerland. His research interests include image processing, active contour models, sampling theory, etc. He is a student member of the IEEE.

Thierry Blu graduated from Ecole Polytechnique, France, in 1986 and from Telecom Paris (ENST), France, in 1988. In 1996, he received the PhD degree in electrical engineering from ENST for a study on iterated rational filter banks applied to wide band audio coding. He is currently with the Biomedical Imaging Group at EPFL (Swiss Federal Institute of Technology), Lausanne, Switzerland, on leave from France Telecom CNET (National Center for Telecommunications Studies), Issy-les-Moulineaux, France. His research interests include (multi-) wavelets, multiresolution analysis, multirate filter banks, approximation, and sampling theory, psychoacoustics. He is a member of the IEEE.

Michael Unser (M'89-SM'94-F'99) received the MS (summa cum laude) and PhD degrees in electrical engineering in 1981 and 1984, respectively, from the Swiss Federal Institute of Technology in Lausanne, Switzerland. From 1985 to 1997, he was with the Biomedical Engineering and Instrumentation Program, National Institutes of Health, Bethesda, where he was heading the Image Processing Group. He is now a professor and head of the Biomedical Imaging Group at the Swiss Federal Institute of Technology in Lausanne, Switzerland. His main research area is biomedical image processing. He has a strong interest in sampling theories, multiresolution algorithms, wavelets, and the use of splines for image processing. He is the author of more than 80 published journal papers in these areas. Dr. Unser is an associate editor for the *IEEE Transactions on Medical Imaging*; he is on the editorial boards of *Signal Processing*, the *Journal of Visual Communication and Image Representation*, and *Pattern Recognition*. He was a former associate editor for the *IEEE Transactions on Image Processing* (1992-1995), the *IEEE Signal Processing Letters* (1994-1998), and was a member of the IMDSP Committee of the IEEE Signal Processing Society (1993-1999). He serves as regular chair for the SPIE conference on wavelet applications in signal and image processing, which has been held annually since 1993. He received the Dommer prize for excellence from the Swiss Federal Institute of Technology in 1981, the research prize of the Brown-Boveri Corporation (Switzerland) for his thesis in 1984, and the IEEE Signal Processing Society's 1995 best paper award. In January, 1999, he was elected fellow of the IEEE with the citation: "for contributions to the theory and practice of splines in signal processing."

▷ **For further information on this or any computing topic, please visit our Digital Library at** http://computer.org/publications/dlib.