1
Splines in biomedical image processing

S. Jonic and C.O.S. Sorzano

1.1
Introduction

The most common definition of splines is that they are piecewise polynomials with pieces smoothly connected together. To obtain a continuous representation of a discrete signal in one or more dimensions, one commonly fits it with a spline. The fit may be exact (interpolation splines) or approximate (least-squares or smoothing splines) [1]. By increasing the spline degree, one progressively switches from the simplest continuous representations (piecewise constant and piecewise linear) to the traditional continuous representation characterized by a band-limited signal model (for the degree equal to the infinity). The traditional Shannon's sampling theory recommends the use of an ideal low-pass filter (anti-aliasing filter) when the input signal is not band-limited. In the spline sampling theory, the Shannon's anti-aliasing filter is replaced with another filter specified by the spline representation of the signal [2].

The most frequently used are B-splines because of the computational efficiency provided by their short support. It has been shown that B-spline basis functions have the minimal support for a given order of approximation [3]. Cubic B-splines offer a good trade-off between the computational cost and the interpolation quality [1]. Also, B-splines are the preferred basis function because of their simple analytical form that facilitates manipulations [1]. Other interesting properties are that they are maximally continuously differentiable and that their derivatives can be computed recursively [1]. Thanks to the separability property of B-splines, the operations on multidimensional data can be performed by a successive processing of one-dimensional (1D) data along each dimension [1]. Besides, they have multiresolution properties that make them very useful for constructing wavelet bases and for multi-scale processing [4, 1]. Because of all these properties, many image processing applications take advantage of B-splines.

In the first part of this chapter, we present the main theoretical results and theorems about splines which are of use in biomedical imaging applications (Section 1.2). Basically, splines are used as interpolants, basis functions, for the construction of smooth

curves and surfaces, and in multiscale analysis. In Section 1.2.1, we show how interpolation is related to sampling and the posterior reconstruction of the original signal from samples. When talking about spline interpolants, we can distinguish between tensor product splines and polyharmonic splines. The former is based on the tensor product of two 1D spline functions, while the latter is based on the use of the so-called radial basis functions. The interpolation theory explains the behavior of splines for the construction of smooth curves and surfaces. However, the traditional formulation of the problem in computer graphics can help gaining more insight into the spline properties. In Section 1.2.2, we thus briefly review the construction of such smooth curves and surfaces, which is very much used in biomedical imaging for visualization of volumes as well as for the calculation of trajectories and surfaces of interest within a given volume. Finally, we review the multiscale properties of splines that allow the definition of spline pyramids and spline wavelets (Section 1.2.3). Both constructions allow to address problems in a coarse-to-fine fashion. These multiscale approaches are not only faster but they are usually more robust to noise and yield smoother functionals when optimization problems are involved.

In the second part of the chapter (Section 1.3), we review the main applications of splines in biomedical image processing, such as geometric image transformations (1.3.1), rigid-body and elastic image registration (1.3.2 and 1.3.3, respectively), contour and shape modelling (1.3.4), and motion reconstruction (1.3.5).

1.2 Main theoretical results about splines

In this section, we present the main achievements of the spline theory that are used in biomedical imaging applications.

1.2.1 Splines as interpolants and basis functions

1.2.1.1 Tensor product splines

The presentation of the theoretical properties of the splines will be done in 1D space. However, the extension to two-dimensional (2D) space is readily performed using the tensor product. For instance, if $\varphi_{1D}(x)$ is a 1D spline, the 2D tensor product spline is defined as $\varphi_{2D}(x,y) = \varphi_{1D}(x)\varphi_{1D}(y)$.

The interpolation context

Given a discrete set of measurements (x_i, y_i), interpolating is the art of "filling in the gaps" with a continuous function $y = f(x)$ such that we meet the constraints imposed by our measurements, i.e., $y_i = f(x_i)$. In biomedical imaging our measurements are typically image values, $z_i = I(x_i, y_i)$, being z_i the gray value of our image, and (x_i, y_i) its location in space. For color images, we could decompose the color into

three components (red, green and blue; hue, saturation and value; etc.), each one would impose a constraint of the same kind as the one for gray values. Interpolating is important to know the value of our image between known pixel values. This is useful for rotations, translations, unwarpings, demosaicking, downsampling, upsampling, etc.

Given, N measurements, we can estimate a polynomial of degree $N-1$ (it has N coefficients and, therefore, N degrees of freedom) passing through all these measurements. In fact, this polynomial is unique and it is given by Newton's general interpolation formula:

$$f(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + ... + a_{N-1}(x-x_0)(x-x_1)...(x-x_{N-2}), \tag{1.1}$$

where a_n represents the divided differences defined as

$$\begin{aligned} a_0 &= y_0 \\ a_n &= \sum_{i=0}^{n} \frac{y_i}{\prod\limits_{\substack{j=0 \\ j\neq i}}^{n} (x_i - x_j)} \end{aligned} \tag{1.2}$$

The use of polynomials for interpolation is justified not only because they are simple to manipulate, differentiate, or integrate but it is also justified by a theorem by Weierstrass stating that any continuous function on a closed interval can be approximated uniformly to any degree of accuracy by a polynomial of a sufficient degree [5].

Regularly spaced interpolation: the generalized sampling theorem

However, polynomials are not the only functions interpolating the dataset. If the x_i points are regularly distributed ($x_i = iT$, for some integer i and a sampling rate T), then Whittaker showed that the series

$$C(x) = \sum_{i=-\infty}^{\infty} y_i \mathrm{sinc}\left(\frac{x - x_i}{T}\right) \tag{1.3}$$

also interpolates the input measurements [6] (sinc is defined in as $\mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$). $C(x)$ is called the cardinal function. Shannon [7] realized that this representation was unique for any function whose maximum frequency is smaller than $\frac{1}{2T}$ Hz. This is the famous Shannon sampling theorem which is valid for any function that is bandlimited.

The sampling theorem can be extended to a larger space of functions: the Hilbert space of L_2 functions, i.e., all functions that are square integrable in the Lebesgue sense ($\|f\|^2 = \langle f, f \rangle < \infty$, where the inner product between two real functions is defined as $\langle f, g \rangle = \int_{-\infty}^{\infty} f(x)g(x)dx$). The set of bandlimited functions is a subset of L_2. The sampling theorem is generally formulated in this space as

$$C(x) = \sum_{i=-\infty}^{\infty} c_i \varphi_i(x), \tag{1.4}$$

where c_i are some coefficients that have to be computed from the y_i input data, and $\varphi_i(x)$ is a shifted version of a basis function $\varphi(x)$ ($\varphi_i(x) = \varphi(x - x_i)$). In the particular

case of bandlimited functions, the basis function used is $\varphi(x) = \mathrm{sinc}(\frac{x}{T})$ and the set of all bandlimited functions is spanned by the family of functions $\varphi_i(x)$. In other words, any bandlimited function can be expressed by a linear combination of the infinite set of $\varphi_i(x)$ functions as the sampling theorem shows. From now on, we will drop the argument of functions as long as there cannot be confusion.

Let us consider any function φ in L_2, and the subspace V generated by its translations with step T

$$V = \left\{ f(x) = \sum_{i=-\infty}^{\infty} c_i \varphi_i(x) : c_i \in l_2 \right\}, \tag{1.5}$$

where l_2 is the space of all square summable sequences. It can be easily proved that if a set of functions φ_i in a Hilbert space is orthonormal (i.e. $\langle \varphi_i, \varphi_j \rangle = \delta_{i-j}$), then the projection of any function f in L_2 onto the subspace V is

$$\tilde{f} = P_V f = \arg\min_{g \in V} \|f - g\| = \sum_{i=-\infty}^{\infty} \langle f, \varphi_i \rangle \, \varphi_i \tag{1.6}$$

Note that

$$\langle f, \varphi_i \rangle = \int_{-\infty}^{\infty} f(x)\varphi(x - x_i)dx = \left. f(x) * \varphi(-x) \right|_{x=x_i}, \tag{1.7}$$

i.e., the inner product $\langle f, \varphi_i \rangle$ can be easily computed by sampling at $x = x_i$ the linear convolution of $f(x)$ and $\varphi(-x)$. In fact, the antialiasing filter used to effectively limit the frequency of a signal before sampling corresponds exactly to the computation of this inner product. Note that if f is already bandlimited, then the convolution of $f(x)$ with $\varphi(-x)$ (whose frequency response is just a rectangle with maximum frequency $\frac{1}{2T}$) is $f(x)$ and, thus, $c_i = y_i$.

The problem of using $\varphi(x) = \mathrm{sinc}\left(\frac{x}{T}\right)$ is that the sinc decays very slowly and the convolutions needed for the computation of the inner product are impractical. Thus, one might set off in search of new functions φ with more computationally appealing properties. First of all, one could relax the orthonormality condition and may ask only that the set $\{\varphi_i\}$ defines a Riesz basis, i.e.,

$$A \|c_i\|_{l_2}^2 \leq \left\| \sum_{i=-\infty}^{\infty} c_i \varphi_i(x) \right\|_{L_2}^2 \leq B \|c_i\|_{l_2}^2 \quad \forall c_i \in l_2 \tag{1.8}$$

for some constants A and B depending on φ. Orthonormality is a special case when $A = B = 1$. The left condition assures that the basis functions are linearly independent, while the right condition assures that the norm of f is bounded and, therefore, the subspace V is a valid subspace of L_2. An important requirement for the basis is that it is able to represent any function with any desired level of accuracy by simply diminishing T. It can be shown that this condition can be reformulated as the partition of unity [1]

$$\sum_{i=-\infty}^{\infty} \varphi_i(x) = 1 \quad \forall x, \tag{1.9}$$

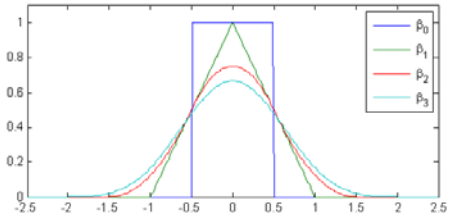

Fig. 1.1 Plot of the first four B-splines (degrees 0,1,2, and 3)

i.e. the sum of all the basis functions (shifted versions of φ) is a constant function.

The function $\varphi(x) = \operatorname{sinc}\left(\frac{x}{T}\right)$ meets all these requirements and is, by far, the most widely known function due to the sampling theorem, beside the fact that it is infinitely supported making convolutions more complicated. However, other functions also meet these requirements. The shortest function meeting them is the box function

$$\varphi(x) = \beta_0\left(\frac{x}{T}\right) = \begin{cases} 1 & |x| < \frac{T}{2} \\ 0 & |x| > \frac{T}{2} \end{cases}, \tag{1.10}$$

which is actually the cardinal B-spline of degree 0. The cardinal B-spline of degree n is simply obtained by convolving $\beta_0(x)$ with itself n times. For instance, $\beta_1\left(\frac{x}{T}\right)$ is the triangular function defined between $-T$ and T, and $\beta_2\left(\frac{x}{T}\right)$ is a parabolic function defined between $-\frac{3}{2}T$ and $\frac{3}{2}T$. In general, $\beta_n\left(\frac{x}{T}\right)$ is an evenly-symmetric, piecewise polynomial of degree n defined for $|x| < \frac{n+1}{2}T$. The following equations show the cardinal B-splines of degree 1, 2 and 3 for $T = 1$ (see also Fig. 1.1):

$$\beta_1(x) = \begin{cases} 1 - |x| & |x| \le 1 \\ 0 & |x| > 1 \end{cases}, \tag{1.11}$$

$$\beta_2(x) = \begin{cases} \frac{3}{4} - |x|^2 & |x| \le \frac{1}{2} \\ \frac{1}{2}\left(|x| - \frac{3}{2}\right)^2 & \frac{1}{2} < |x| \le \frac{3}{2} \\ 0 & |x| > \frac{3}{2} \end{cases}, \tag{1.12}$$

$$\beta_3(x) = \begin{cases} \frac{2}{3} + \frac{1}{2}|x|^2(|x| - 2) & |x| \le 1 \\ \frac{1}{6}(2 - |x|)^3 & 1 < |x| \le 2 \\ 0 & |x| > 2 \end{cases}, \tag{1.13}$$

In general, the cardinal B-spline of degree n can be expressed as

$$\beta_n(x) = \frac{1}{n!}\sum_{k=0}^{n+1}(-1)^k\binom{n+1}{k}\left(x - \left(k - \frac{n+1}{2}\right)\right)_+^n, \tag{1.14}$$

where $(x)_+^n$ is the one-sided n-th power of x, $(x)_+^n = \begin{cases} x^n & x \ge 0 \\ 0 & x < 0 \end{cases}$.

Cardinal B-splines are piecewise polynomials because they can be represented by polynomials of degree n on $|x|$ that are different for each interval $i\,T \leq |x| < (i+1)\,T$. All these functions meet the aforementioned requirements and therefore can be used as basis functions for the representation of signals. Moreover, they are well localized and compactly supported in real space (consequently, infinitely supported in Fourier space) making computations in real space affordable.

The representation of signals using cardinal B-splines is intimately related to interpolation: the use of β_0 is equivalent to nearest neighbor interpolation, the use of β_1 is equivalent to linear interpolation (bilinear interpolation when the 2D tensor product B-spline is constructed). β_3 has proved to be a good compromise between computational complexity, compact support in real space, and approximation error.

At this point, we have already shown that the sampling theorem is a particular case of an interpolation problem and how it can be generalized for L_2 functions instead of working exclusively with band-limited functions. The question is how to compute the representation coefficients c_i in Eq. (1.4). We have seen a possibility for orthonormal basis in Eq. (1.7). However, cardinal B-splines are not orthonormal in general (only cardinal B-splines of degree 0 are orthogonal). In the general case, the calculation of the c_i coefficients is similar to the orthonormal case. The only difference is that instead of using the basis φ_i itself, we have to use the dual basis $\tilde{\varphi}_i$

$$\tilde{f} = P_V f = \sum_{i=-\infty}^{\infty} \langle f, \tilde{\varphi}_i \rangle \, \varphi_i. \tag{1.15}$$

The dual basis is uniquely defined by the biorthogonality condition $\langle \tilde{\varphi}_i, \varphi_j \rangle = \delta_{i-j}$ and it also inherits the shift invariant property of the original basis function, $\tilde{\varphi}_i(x) = \tilde{\varphi}(x - x_i)$. However, we still do not have a clear way of computing the c_i's because we do not have a close-form formula for the dual basis. It can be proved [2] that the Fourier transform of $\tilde{\varphi}$ is given by

$$\widehat{\tilde{\varphi}}(j\Omega) = \frac{\widehat{\varphi}(j\Omega)}{\sum\limits_{k=-\infty}^{\infty} |\widehat{\varphi}(j(\Omega + 2\pi k))|^2}, \tag{1.16}$$

where $\widehat{f}(j\Omega)$ represents the continuous Fourier transform of the function $f(x)$. Fortunately, Unser and coworkers [8, 9] derived a very efficient way of computing these coefficients using standard digital filters. This is actually the way of producing these coefficients. These filters are derived in 1D. However, they are easily extended to nD. In the case of images, these filters are run individually on each row of the image, producing a new image of coefficients over the horizontal axis, x. Then, they are run on each column of the new image finally producing the coefficients of the 2D tensor product B-spline. Once the coefficients are produced, images are treated as if they were continuous functions, although they are stored as a discrete set of cardinal B-spline coefficients.

Now, one may wonder if we could design a basis function based on B-splines such that $c_i = y_i$. This would be an interpolating spline and we would be back to a situation similar to the interpolation scheme presented in the sampling theorem, Eq. (1.3). The

following function is such an interpolating spline

$$\varphi_{int}(x) = \sum_{i=-\infty}^{\infty} q_{int}[i]\varphi_i(x), \tag{1.17}$$

where $q_{int}[i]$ is the l_2 sequence defined as the inverse Z transform of $Q_{int}(z) = \frac{1}{\sum_{k=-\infty}^{\infty} \varphi(kT)z^{-k}}$, and $\varphi(x) = \beta_n\left(\frac{x}{T}\right)$.

Approximation error: How far are we from the truth?

An important concept related to the generalized sampling theorem explained above is how well it reproduces any function f. Let us call f_T the approximation with a given sampling rate T. This problem has been studied by the approximation theory proving that the following three statements are equivalent:

1. Let f be a sufficiently smooth function (f belongs to the Sobolev space W_2^L, i.e., its L first derivatives belong to L_2). Then, as T approaches 0, there exists a constant C independent of f such that $\|f - f_T\| \leq CT^L \left\|f^{(L)}\right\|$.

2. The first L moments of φ are constants, i.e., $\sum_{i=-\infty}^{\infty} (x - x_i)^m \varphi_i(x) = \mu_m$ for $m = 0, 1, ..., L-1$.

3. The first L monomials can be exactly represented, i.e., for each monomial x^m ($m = 0, 1, ..., L-1$), there exist constants $c_i \in l_2$ such that $x^m = \sum_{i=-\infty}^{\infty} c_i \varphi_i(x)$.

An important consequence of this result is that the approximation error of different basis functions depends mostly on its design, i.e., given two polynomials of the same degree, one of them may approach smooth functions more quickly than the other. L is called the order of approximation and entirely depends on the moments of φ or the number of monomials that can be represented. In particular, B-splines of degree n have an order of approximation of $L = n + 1$. Another important consequence is that in order to converge to function f as $T \to 0$, the basis function φ must have $L \geq 1$, or what is the same, that it fulfills the partition of unity. It is well known [10] that windowed sincs (which are commonly used as a solution to the infinite support of the sinc function) do not meet this condition. One may also try to design the φ family such that they have order of approximation L with a minimum support. This is how the MOMS (Maximal-Order interpolation of Minimal Support) set of functions is designed. It turns out that these functions are linear combinations of the cardinal B-spline of degree L and its derivatives [3].

The reader interested in the generalized sampling theorem and this approach to interpolation may expand this information in [11, 10, 12, 1, 2].

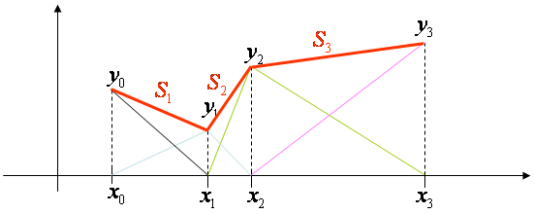

Fig. 1.2 Example of B-spline (degree 1) interpolation of irregularly spaced samples

Back to irregular interpolation problems

So far, we have already introduced cardinal B-splines, dual splines, and interpolation splines. In fact, splines is a broad family of functions of which we have only seen those used with a regular spacing. In general, a function $S_n(x)$ is a spline of degree n if: 1) it is defined by piecewise polynomials of degree at most n; 2) it is of class C^{n-1}, i.e., it has $n-1$ continuous derivatives even at the points joining the different polynomial pieces [13].

An alternative approach to splines is based on the idea of curve interpolation and knots instead of on the idea of sampling. This other approach allows for irregularly spaced samples in a much more direct way. Let us assume that we are interpolating a real function in the interval $[a,b]$ with a piecewise polynomial. We subdivide this interval in N adjacent pieces such that each piece is defined in the interval $[x_i, x_{i+1}]$. The subdivision is such that $x_0 = a$, $x_N = b$, $x_i < x_{i+1}$ and $[a,b] = \bigcup_{i=0}^{N-1} [x_i, x_{i+1}]$. The input samples (x_i, y_i) are called knots and they are fixed points through which the interpolated polynomial must pass. Note that knots need not to be equally spaced in the interval $[a,b]$. We have to be specially careful to select the interpolating polynomials such that not only the spline itself is continuous, but all its derivatives up to order $n-1$ are also continuous, even at the knots where the function at the left of the knot is defined by a certain polynomial, and at the right of the knot by a different polynomial (see Fig. 1.2).

With $N+1$ knots, the spline is split into N intervals. A spline of degree n has $n+1$ coefficients in each interval, therefore the spline has $(n+1)N$ degrees of freedom. Let us call $S_i(x)$ the polynomial of degree n in each interval ($i = 1, 2, ...N$). This spline must satisfy:

– Interpolation of the knot values: $2N$ degrees of freedom

$$\begin{aligned} &S_1(x_0) = y_0 \\ &S_i(x_i) = y_i = S_{i+1}(x_i) \quad i = 1, 2, ..., N-1 \\ &S_N(x_N) = y_N \end{aligned} \tag{1.18}$$

– Continuity of the $n-1$ derivatives: $(n-1)(N-1)$ degrees of freedom

$$\begin{array}{ll} S_i^{(1)}(x_i) = S_{i+1}^{(1)}(x_i) & i = 1, 2, ..., N-1 \\ S_i^{(2)}(x_i) = S_{i+1}^{(2)}(x_i) & i = 1, 2, ..., N-1 \\ ... & \\ S_i^{(n-1)}(x_i) = S_{i+1}^{(n-1)}(x_i) & i = 1, 2, ..., N-1 \end{array} \tag{1.19}$$

However, there are still $n-1$ unfixed degrees of freedom. The way of choosing these degrees of freedom defines different kinds of splines. For instance, a natural spline of degree $n = 3$ is one of the most common cases, in which, the second derivative at the extremes is set to 0, $S_i^{(2)}(x_0) = S_i^{(2)}(x_N) = 0$. There are efficient algorithms for the solution of the resulting equation system [14].

The expression of the polynomial of degree n in any of the intervals $[x_i, x_{i+1})$ can be recursively constructed as

$$\begin{array}{l} S_{i,0}(x) = \begin{cases} 1 & x_i < x \leq x_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ S_{i,l}(x) = \frac{x - x_i}{x_{i+l} - x_i} S_{i,l-1}(x) + \frac{x_{i+l+1} - x}{x_{i+l+1} - x_i} S_{i+1,l-1}(x), \end{array} \tag{1.20}$$

where $S_{i,l}$ is the B-spline of degree l in the interval $[x_i, x_{i+1})$. It is common use to repeat the extremes of the knots as many times as the finally desired spline degree. For example, the actual list of knots for degree 3 would be $\{x_0, x_0, x_0, x_0, x_1, x_2, ..., x_{N-1}, x_N, x_N, x_N, x_N\}$. Because of the repeated knots we admit by convention in the previous formula that $\frac{0}{0} = 0$.

The polynomial in each one of the subintervals is of the general form

$$S_i(x) = \sum_{l=0}^{n} a_l x^l. \tag{1.21}$$

In this expansion we have used the fact that the set $\left\{1, x, x^2, ..., x^n\right\}$ is a basis of the polynomials of degree n. However, this is not an orthonormal basis, which causes numerical instabilities when solving for the a_l coefficients. Alternatively, we could have used any other basis of the polynomials of degree n, $\{P_0(x), P_1(x), P_2(x), ..., P_n(x)\}$

$$S_i(x) = \sum_{l=0}^{n} a_l P_l(x). \tag{1.22}$$

Employing different polynomial basis gives rise to different kinds of splines: Bernstein polynomials are used in the Bézier splines; Hermite polynomials in the Hermite splines; basis splines in the B-splines; etc. This idea is further explored in Section 1.2.2.

Generalized splines

We have already seen that splines are piecewise polynomials. They can be seen as linear combination of basis functions which, in their turn, are Green functions of a

differential operator. Let us consider splines of degree 0. These are piecewise constant functions that can be formulated as

$$S_0(x) = \sum_{i=-\infty}^{\infty} a_i u(x - x_i) \tag{1.23}$$

where $u(x)$ is the Heaviside's step function. The step function is the causal Green function of the first derivative operator $\mathcal{D}$ (i.e., $\mathcal{D}\{u\} = \delta$). This implies that the first derivative of splines of degree 0 is of the form

$$\mathcal{D}\{S_0\}(x) = \sum_{i=-\infty}^{\infty} a_i \delta(x - x_i). \tag{1.24}$$

Splines of degree 1 can be seen as the linear combination of shifted versions of the Green function of the second derivative operator ($\mathcal{D}^2$). In general, splines of degree n are a linear combination of shifted versions of the Green function of the $(n+1)$-th derivative operator ($\mathcal{D}^{n+1}$).

Nowadays, there is much interest in the variational formulation of imaging problems (denoising, impainting, regularization of functionals, etc.; for more details on the variational approach, the reader is referred to Chapter 24 in this same book). The variational formulation relies on the definition of a differential operator $\mathcal{L}$. Generalized splines are splines specially designed for a specific differential operator. In other words, $S(x)$ is a $\mathcal{L}$-spline if

$$\mathcal{L}\{S\}(x) = \sum_{i=-\infty}^{\infty} a_i \delta(x - x_i). \tag{1.25}$$

In this way, $\mathcal{L}$-splines span a space which is specially well suited for the problem at hand. In particular, they have been proved useful for the regularization of approximation problems and the estimation of fractal processes. The reader interested in this kind of splines is referred to [15, 16].

1.2.1.2 Polyharmonic splines

The extension to several dimensions can be done by the tensor product of 1D splines as seen in the previous section, or by the specific design of the so-called polyharmonic splines, among which the most famous is the thin-plate spline.

Let us assume that we have a set of input multivariate data points $(\mathbf{x}_i, y_i)$ we would like to interpolate ($\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$). The goal would be finding a hypersurface $y = f(\mathbf{x})$ such that the surface contains the input data points.

We will look for our interpolant in the Beppo-Levi space $BL^{(2)}(\mathbb{R}^d)$, i.e., the space of all L_2 functions from $\mathbb{R}^d$ to $\mathbb{R}$ such that its second derivative is also in L_2. This space is large enough to contain a suitable interpolator. In this space we can define the rotationally invariant seminorm

$$\|f\|^2 = \int_{\mathbb{R}^d} \sum_{\eta=1}^{d} \sum_{\xi=1}^{d} \frac{\partial^2 f}{\partial x_\eta \partial x_\xi} d\mathbf{x} \tag{1.26}$$

Duchon [17] showed that the interpolant minimizing the just introduced seminorm in $BL^{(2)}(\mathbb{R}^d)$ is of the form

$$f(\mathbf{x}) = p_m(\mathbf{x}) + \sum_{i=1}^{N} c_i \phi_{d,k}(\|\mathbf{x} - \mathbf{x}_i\|), \tag{1.27}$$

where $\|\mathbf{x}\|$ denotes the standard Euclidean norm in $\mathbb{R}^d$, N is the number of measurements in the input data, c_i is a set of coefficients to determine, k is any integer number such that $2k > d$ (in fact, the approximation order is $k - \frac{d}{2}$ [18], therefore, from one hand it is interesting choosing high values of k although on the other hand these result in numerical instabilities in the determination of the spline coefficients as will be seen later; a trade-off between these two goals must be achieved), $\phi_{d,k}$ is a radial basis function, and $p_m(\mathbf{x})$ is polynomial in $\mathbf{x}$ of degree at most m being $m = k - \lceil \frac{d}{2} \rceil$. The radial basis function $\phi_{d,k}$ is

$$\phi_{d,k}(r) = \begin{cases} r^{2k-d} \log(r) & \text{for } d \text{ even} \\ r^{2k-d} & \text{for } d \text{ odd} \end{cases} \tag{1.28}$$

The function $\phi_{2,2} = r^2 \log(r)$ is the so-called thin-plate spline in $\mathbb{R}^2$ and the minimization of the seminorm in $\mathbb{R}^2$ can be understood as the minimization of the bending energy of a thin sheet of metal that interpolates the input data points. In $\mathbb{R}^3$, $\phi_{3,2}$ is called the biharmonic spline, and $\phi_{3,3}$ the triharmonic spline.

The interpolation equations $f(\mathbf{x}_i) = y_i$ do not fully determine the function f (note that we have N input data points, but N coefficients c_i for the radial basis functions and $\sum_{p=0}^{m} \binom{d}{p}$ coefficients for the polynomial). In fact, the polynomial of degree m comes from the fact that it is in the kernel of the seminorm and, therefore, the addition of any polynomial of degree m is "invisible" to the seminorm minimization (for instance, for the thin-plate spline case, $m = 1$, and all polynomials of degree 1 have null second derivatives). In this way, we have to impose extra conditions which generally are

$$\sum_{i=1}^{N} c_i q(\mathbf{x}_i) = 0 \tag{1.29}$$

for all polynomials $q(\mathbf{x})$ of degree at most m (in the case of the thin-plate spline, we would have to use $q(\mathbf{x}) = 1$, $q(\mathbf{x}) = x_1$, $q(\mathbf{x}) = x_2$). Let us assume that the set of coefficients of the polynomial p are written in vector form as $\mathbf{p}$. Then, the polyharmonic interpolation can be solved by the following equation system

$$\begin{pmatrix} \mathbf{\Phi} & \mathbf{P} \\ \mathbf{P}^t & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{c} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix}, \tag{1.30}$$

where $\mathbf{c}$ and $\mathbf{y}$ are column vectors with the c_i coefficients and the y_i measurements, respectively. $\mathbf{\Phi}$ is the $N \times N$ system matrix corresponding to the measurements, i.e., $\Phi_{ij} = \phi_{d,k}(\|\mathbf{x}_i - \mathbf{x}_j\|)$ and P is a matrix related to some basis of polynomials up to degree m. Let $\{p_1, p_2, ..., p_l\}$ be such a basis, then $P_{ij} = p_j(\mathbf{x_i})$. For example, for

the thin-plate spline case we could define $p_1(\mathbf{x}) = 1$, $p_2(\mathbf{x}) = x_1$, and $p_3(\mathbf{x}) = x_2$; but any other basis of polynomials of degree at most 1 would do. Note that the size of matrix P is $N \times l$ (in the thin-plate spline, $l = 3$).

This equation system is usually ill-conditioned due to the non-local nature of the $\phi_{d,k}(r)$ functions (they are not locally supported, instead they grow to infinity with growing r). Moreover, the complexity of solving the equation system depends on the number of sample points, N. Finally, the evaluation of the polyharmonic spline involves as many operations as input data points (although fast algorithms have been developed for tackling this latter problem [19]). For solving the problem of the ill-conditioning of the equation system, a localization of the polyharmonic spline can be done. This is a process in which the non-compactly supported $\phi_{d,k}(r)$ function is substituted as a weighted sum of compactly supported functions (for instance, B-splines). For further information on this technique the reader is referred to [20, 21].

In case of noisy data we can relax the condition of passing through all data points and substitute this constraint by a least-squares minimization:

$$f^* = \arg \min_{f \in BL^{(2)}(\mathbb{R}^d)} \lambda \|f\|^2 + \frac{1}{N} \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i))^2 \tag{1.31}$$

It can be proved [22] that the coefficients $\mathbf{c}$ and $\mathbf{p}$ are the solution of the following linear equation system

$$\begin{pmatrix} \mathbf{\Phi} - 8N\lambda\pi\mathbf{I} & \mathbf{P} \\ \mathbf{P}^t & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{c} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix}, \tag{1.32}$$

One may wonder why these radial basis functions are called splines, at least, they do not seem to fit our previous definition of piecewise polynomial functions. The solution is a slight modification of our concept of spline (particularly, the requirement of being piecewise). Let us consider the triharmonic ($\phi_{3,3} = r^3$) functions. It is clear that it is a cubic polynomial in r, and its second derivative is continuous everywhere. Therefore, it is a cubic spline.

The reader interested in this topic may expand this information in [18, 23, 24]

1.2.2 Splines for smooth curves and surfaces

One of the main applications of splines is in the design of smooth curves and surfaces. For this application, one can use the already known interpolation and approximation properties of splines. However, the language and formulation used for the construction of curves and surfaces provide extra insight into the properties of splines. For this reason we will make in this chapter a quick review of the use of splines for this purpose.

Parametric curves and surfaces

Without loss of generality we can define a parametric curve in the three-dimensional (3D) space as an explicit function $\mathbf{C}(u) = (x(u), y(u), z(u))$ depending on some

parameter u in the range $[0,1]$. In the same way, we could define a parametric surface as $\mathbf{S}(u,v) = (x(u,v), y(u,v), z(u,v))$ with $(u,v) \in [0,1]^2$. Let us first consider the construction of curves and at the end we will extend the corresponding concepts to surfaces.

Functions $x(u)$, $y(u)$ and $z(u)$ are too general and do not lead to standardized computer implementations. It would be better if these functions could be expressed as a linear combination of some basis functions in such a way that

$$\mathbf{C}(u) = \sum_{i=0}^{N} \mathbf{a}_i f_i(u), \tag{1.33}$$

where $\mathbf{a}_i$ is a 3D vector of coefficients $\mathbf{a}_i = (a_{ix}, a_{iy}, a_{iz})$ and $f_i(u)$ ($i = 0, 1, ..., N$) is a set of basis functions.

A natural choice for the f_i functions would seem to be the power basis $(1, u, u^2, ..., u^N)$. These would naturally give the Taylor expansion of the functions $x(u)$, $y(u)$ and $z(u)$.

$$\mathbf{C}(u) = \begin{pmatrix} \sum_{i=0}^{N} a_{ix} u^i \\ \sum_{i=0}^{N} a_{iy} u^i \\ \sum_{i=0}^{N} a_{iz} u^i \end{pmatrix} \tag{1.34}$$

However, this choice has a number of disadvantages: first, the $\mathbf{a}_i$ coefficients convey very little insight about the curve itself, they are simply suitable coefficients for the expansion; second, the evaluation of the power basis is prone to round-off errors, specially for high order basis. To partially avoid these problems, one could use Bézier's method. Instead of using the power basis, he used the Bernstein polynomials

$$\mathbf{C}(u) = \sum_{i=0}^{N} \mathbf{p}_i B_{i,N}(u). \tag{1.35}$$

Now the $\mathbf{p}_i$ coefficients have a geometrical meaning, they are points in space "attracting" the curve toward themselves (Fig. 1.3). They are called control points. Moreover $\mathbf{p}_0$ is the first point of the curve ($\mathbf{C}(0) = \mathbf{p}_0$) and $\mathbf{p}_N$ is the last one ($\mathbf{C}(1) = \mathbf{p}_N$). Bernstein polynomials span the same space as the power basis and, therefore, are mathematically equivalent to them. The N-th degree Bernstein polynomial $B_{i,N}$ is defined as

$$B_{i,N}(u) = \frac{N!}{i!(N-i)!} u^i (1-u)^{N-i} \tag{1.36}$$

Curves defined by a single polynomial in the whole interval $[0,1]$, as Bézier curves, still have a number of shortcomings: first, when the number of control points N is high, the order of the polynomial is also high (which we already saw that yields numerically unstable algorithms); second, although control points mostly influence the curve in its vicinity, they participate in the whole curve and control exercised through control points is not sufficiently local. As a solution to this problem, cardinal B-splines have been proposed (they were already introduced for irregular interpolation (1.20)).

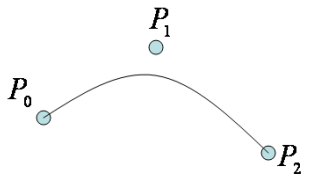

Fig. 1.3 Example of Bézier curve with three control points

Cardinal B-splines span the space of polynomials of some fixed degree n. They are compactly supported and, thus, they avoid numerical instabilities and the local control problem previously described. Segments between control points define the interpolation intervals of the B-spline. The n-th degree B-spline curve is

$$\mathbf{C}(u) = \sum_{i=0}^{N} \mathbf{p}_i S_{i,n}(u). \tag{1.37}$$

We could combine two interesting ideas of Bézier and B-spline curves. Bézier introduces the idea that control points are "attractors" (although they cannot directly control the attraction strength) while B-splines provide numerically stable and local functions (and control the attraction strength through the B-spline degree). Non-uniform rational B-splines (NURBS) take the best of both worlds

$$\mathbf{C}(u) = \sum_{i=0}^{N} \mathbf{p}_i R_{i,n}(u). \tag{1.38}$$

Each control point attracts the curve with a weight w_i and these weights give the NURBS function (which is not a polynomial):

$$R_{i,n}(u) = \frac{w_i S_{i,n}(u)}{\sum\limits_{j=0}^{N} w_j S_{j,n}(u)}, \tag{1.39}$$

because of the B-spline in the numerator, NURBS are compactly supported. By increasing w_i, we can decrease the distance between the curve and $\mathbf{p}_i$.

Note that, in general, none of these parametric representations of curves (Bézier, B-spline, NURBS) are interpolating, i.e., control points $\mathbf{p}_i$ do not belong to the curve (except $\mathbf{p}_0$ and $\mathbf{p}_N$). How close the curve is from the control points can be controlled in B-spline curves by increasing the degree of the B-spline (p). In NURBS, it is controlled by both n and w_i.

B-spline surfaces of degree n are usually defined through the tensor product of two n-th degree B-spline curves (Fig. 1.4)

$$\mathbf{S}(u,v) = \sum_{i=0}^{N} \sum_{j=0}^{M} \mathbf{p}_{ij} S_{i,n}(u) S_{j,n}(v). \tag{1.40}$$

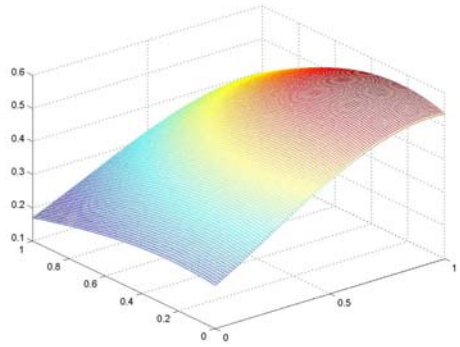

Fig. 1.4 Example of B-spline surface

An interesting property of the tensor product surfaces is that if we fix any of the two variables, say $v = v_0$, then the remaining function is a B-spline curve

$$\mathbf{S}(u, v_0) = \sum_{i=0}^{N} \left(\sum_{j=0}^{M} \mathbf{p}_{ij} S_{j,n}(v_0) \right) S_{i,n}(u). \tag{1.41}$$

Construction and properties of curves and surfaces using splines

The construction of a B-spline curve needs the definition of the knots u_i delimiting the interpolation intervals (each knot is associated to a given control point). Note that $u_0 = 0$ and $u_N = 1$. As already indicated it is customary to repeat the first and last knots $n + 1$ times so that the set of knots become:

$$U = \{\underbrace{u_0, ..., u_0}_{n+1 \text{ times}}, u_1, u_2, ..., u_{N-1}, \underbrace{u_N, ..., u_N}_{n+1 \text{ times}}\} \tag{1.42}$$

The first and last knots are not the only ones that can be repeated, any u_i in the sequence can be repeated. However, increasing the multiplicity of a control point decreases the derivability of the B-spline curve at that point. B-spline curves are of class C_∞ (continuous and all its derivatives continuous) everywhere except at the knots. At the knots, they are of class C_{n-k} (i.e., they are continuous and have $n - k$ continuous derivatives), where n is the B-spline degree and k is the multiplicity of the knot. By repeating a point n times, the curve is C_0 at that knot and interpolates the corresponding control point (if all points are repeated n times, then we are simply joining the control points with straight lines).

Simply by using B-splines as basis functions, B-spline curves and surfaces enjoy a number of useful properties:

– **Locality**: Outside the interval $[u_i, u_{i+n+1})$, $S_{i,n}(u) = 0$. This fact guarantees numerical stability in the evaluation of the curve values. Conversely, in a given

interval $[u_i, u_{i+1})$, there are at most $n+1$ B-splines different from 0, which establishes the basis for computationally efficient algorithms to evaluate the curve value. Another way of considering this property is that as one moves along the curve, at each knot, one of the past B-splines switches off (the basis function n knots ago no longer affects the curve shape) and a new B-spline switches on.

- **Partition of unity**: For any interval $[u_i, u_{i+1})$, $\sum_{j=i-n}^{i} S_{j,n}(u) = 1$. This means that we can approximate any C_2 curve as closely as desired as long as we provide sufficient control points.

- **Spatial bounds**: B-spline curves and surfaces are contained within the convex hull of the control points.

1.2.3 Splines for multiscale analysis

The multiscale capabilities of splines come in two different flavors: spline pyramids and spline wavelets. Each one of these approaches exploit a different feature of splines that make them suitable for multiresolution analysis. The reader interested in this topic may get more details in [1].

1.2.3.1 Spline pyramids

Let us assume that we know the representation of a certain function $f(x)$ with B-splines of odd degree n and a sampling rate T

$$f(x) \quad = \quad \sum_{i=-\infty}^{\infty} c_i \beta_n\left(\tfrac{x}{T} - i\right) = \left(\sum_{i=-\infty}^{\infty} c_i \delta\left(x - i\right)\right) * \beta_n\left(\tfrac{x}{T}\right) = c_{\mathrm{II}}(x) * \beta_n\left(\tfrac{x}{T}\right). \tag{1.43}$$

One may wonder how a finer representation of f would be. For this we consider the relationship between an odd degree B-spline and its contraction by M to a final sampling rate $\frac{T}{M}$. It can be shown [1] that

$$\beta_n\left(\frac{x}{T}\right) = \sum_{k=-\infty}^{\infty} h_k \beta_n\left(\frac{x}{\frac{T}{M}} - k\right) = h_{\mathrm{II}}(x) * \beta_n\left(\frac{x}{\frac{T}{M}}\right) \tag{1.44}$$

i.e., we can decompose a wide B-spline of degree n as the weighted sum of thinner B-splines of the same degree. For any M and n, the Z transform of the weight sequence h_k is

$$H(z) = z^{\frac{(M-1)(n+1)}{2}} \frac{1}{M^n} \left(\sum_{m=0}^{M-1} z^{-m}\right)^{n+1} \tag{1.45}$$

The case $M = 2$ plays an important role in the design of wavelets, and the corresponding property is called the 2-scale relationship. In case of B-splines of degree 1, the two

scale relationship is simply

$$\beta_1\left(\frac{x}{T}\right) = \frac{1}{2}\beta_1\left(\frac{x}{\frac{T}{2}}+1\right) + \beta_1\left(\frac{x}{\frac{T}{2}}\right) + \frac{1}{2}\beta_1\left(\frac{x}{\frac{T}{2}}-1\right) \tag{1.46}$$

and, in general, for splines of odd degree n and $M = 2$ we have

$$h_k = \begin{cases} 2^{-n}\binom{n+1}{k+\frac{n+1}{2}} & |k| \leq \frac{n+1}{2} \\ 0 & |k| > \frac{n+1}{2} \end{cases} \tag{1.47}$$

Substituting the expression of the wide B-splines by the weighted sum of fine B-splines, we obtain

$$f(x) = \left(\left(\uparrow_M \{c_{\mathrm{II}}(x)\} * h\right)_{\mathrm{II}}(x)\right) * \beta_n\left(\frac{x}{\frac{T}{M}}\right) \tag{1.48}$$

That is, to obtain a finer representation of a signal we simply have to upsample its B-spline coefficients and convolve them with a finite weight sequence depending on the scaling factor M and the spline degree. Note that the function represented with splines at the finer resolution is exactly the same as the original one. No interpolation has been performed on the way.

Creating a coarser representation of the function $f(x)$ is a little bit more involved since we cannot have exactly the same function but an approximation to it

$$\tilde{f}(x) = \tilde{c}_{\mathrm{II}}(x) * \beta_n\left(\frac{x}{MT}\right), \tag{1.49}$$

and we have to devise a way of estimating the coefficients $\tilde{c}$ from the c coefficients. The easiest way is to look for the $\tilde{c}$ that minimize the L_2 norm of the error $\left\|f - \tilde{f}\right\|$. It can be proved [25] that the solution to this least-squares problem is

$$\tilde{c} = \frac{1}{2}\left(\left(b_k^{2n+1}\right)^{-1} * \downarrow_M \left\{h_k * b_k^{2n+1} * c_k\right\}\right), \tag{1.50}$$

being $\downarrow_M \{\cdot\}$ the downsampling operator, c_k the B-spline coefficients of the function f with sampling rate T, h_k the sequence described in Eq. (1.47), and b_k^{2n+1} the sequence $b_k^{2n+1} = \beta_{2n+1}(k)$. Note that $\left(b_k^{2n+1}\right)^{-1}$ is the inverse of this sequence. It can be easily understood by inverting the Z transform of the sequence b_k^{2n+1} and then performing an inverse Z transform. While b_k^{2n+1} is compactly supported, $\left(b_k^{2n+1}\right)^{-1}$ is not and convolution with this sequence requires the design of an IIR (Infinite Impulse Response) filter [9, 26].

1.2.3.2 Spline wavelets

A quick review on wavelets

Wavelets are basis functions to expand our function $f(x)$

$$f(x) = \sum_{s=-\infty}^{\infty}\sum_{i=-\infty}^{\infty} c_{si}\psi_{si}(x). \tag{1.51}$$

i is our standard location parameter (our basis function ψ_{si} will be located at samples x_i), while s is a new parameter controlling the sampling rate and the size of the basis function

$$\psi_{si}(x) = \frac{1}{\sqrt{T_s}}\psi\left(\frac{x - x_i}{T_s}\right). \tag{1.52}$$

Function ψ is called the mother wavelet, while the basis functions ψ_{si} are called child wavelets. Positive s values result in child wavelets finer than the mother wavelet, while negative s values result in child wavelets wider than the mother wavelet. Typically, $T_s = 2^s$ and $x_i = iT_s$. In this way, we see that our function is decomposed as a weighted combination of functions at the same scale as the mother wavelet ($s = 0$) and located at the integer samples $x_i = i$, plus a weighted combination of functions twice wider than the mother wavelet ($s = 1$) and located at samples $x_i = 2i$, plus a weighted combination of functions four times wider than the mother wavelet ($s = 2$) and located at samples $x_i = 2^2 i$, and so on. On the other side, we have a weighted sum of functions twice thinner than the mother wavelet ($s = -1$) and located at samples $x = \frac{i}{2}$, and so on.

The question now is which are the requirements on the mother wavelet to be able to approximate functions in $L^2(\mathbb{R})$. The answer is that a mother wavelet must fulfill the admissibility and regularity conditions. The admissibility condition is

$$\int_{-\infty}^{\infty} \frac{|\widehat{\psi}(j\Omega)|^2}{|\Omega|} d\Omega < \infty, \tag{1.53}$$

where $\widehat{\psi}(j\Omega)$ is the continuous Fourier transform of the mother wavelet. This condition implies that $\widehat{\psi}(0) = 0$ and, therefore, the mother wavelet behaves as a band-pass like filter and has zero average over x, i.e., the mother wavelet must oscillate in real space and this is where the name wavelet is coming from, it behaves as a wave. The regularity condition is related to the approximation order, i.e., how well can a wavelet decomposition represent a given signal. We saw in Eq. (1.15) that the projection on the space spanned by the wavelets ψ_{si} is performed through the inner product between the dual wavelet and the function to be represented. Let us expand f in its Taylor series (for simplicity of notation around 0, although it could have been done around any point and the conclusions would be the same)

$$\begin{aligned}
f = \sum_{s=-\infty}^{\infty}\sum_{i=-\infty}^{\infty} & \left\langle f(0) + f^{(1)}(0)x + ... + \tfrac{f^{(n)}(0)}{n!}x^n + O(x^{n+1}), \tilde{\psi}_{si}\right\rangle \psi_{si} \\
= \sum_{s=-\infty}^{\infty}\sum_{i=-\infty}^{\infty} & \left(f(0)\left\langle 1, \tilde{\psi}_{si}\right\rangle + f^{(1)}(0)\left\langle x, \tilde{\psi}_{si}\right\rangle + ... \right. \\
& \left. + \tfrac{f^{(n)}(0)}{n!}\left\langle x^n, \tilde{\psi}_{si}\right\rangle + \left\langle O(x^{n+1}), \tilde{\psi}_{si}\right\rangle\right)\psi_{si} \\
= \sum_{s=-\infty}^{\infty}\sum_{i=-\infty}^{\infty} & \left(f(0)\left\langle 1, \tilde{\psi}_{0i}\right\rangle T_s + f^{(1)}(0)\left\langle x, \tilde{\psi}_{0i}\right\rangle T_s^2 + ... \right. \\
& \left. + \tfrac{f^{(n)}(0)}{n!}\left\langle x^n, \tilde{\psi}_{0i}\right\rangle T_s^{n+1} + \left\langle O(x^{n+1}), \tilde{\psi}_{0i}\right\rangle T_s^{n+2}\right)\psi_{si}
\end{aligned} \tag{1.54}$$

Because of the admissibility criterion and the relationship between a wavelet and its dual (1.16), the average of the dual is also zero ($\left\langle 1, \tilde{\psi}_{si}\right\rangle = 0$). If we design the wavelet

so that its dual has n vanishing moments ($\left\langle x, \tilde{\psi}_{si} \right\rangle = ... = \left\langle x^n, \tilde{\psi}_{si} \right\rangle = 0$), then the wavelet coefficients c_{si} decrease as T_s^{n+2}. This is related to the approximation order [27] (the error between the reconstructed function, Eq. (1.51), and the desired function is as small as possible) and the number of null moments of the wavelet designed is called the number of vanishing moments.

Designing good wavelets involve having good approximation properties, and yielding efficient algorithms for the analysis (computation of the c_{si} coefficients) and synthesis (reconstruction of the signal through Eq. (1.51)). Splines are involved in wavelet analysis in two ways: first, we can explicitly design mother wavelets that incorporate some spline; second, it can be shown that all other mother wavelets can be decomposed as the convolution of a spline (responsible for a good number of appealing properties of wavelets) and a distribution. Before getting into these two topics let us further describe the wavelet decomposition by the introduction of the scaling function.

Let us rewrite Eq. (1.51) as

$$f(x) = \sum_{s=-\infty}^{\infty} d_s(x), \tag{1.55}$$

where $d_s(x)$ represents the inner sum. Mother wavelets must have zero average over x, this means that they are high pass filters and, therefore, $d_s(x)$ can be understood as the details at level s. In this way, Eq. (1.51) can be reinterpreted as the superposition of details at different levels s. We could split the wavelet transform, Eq. (1.51), in two sums

$$f(x) = \sum_{s=-\infty}^{S-1} d_s(x) + \sum_{s=S}^{\infty} d_s(x) = \sum_{s=-\infty}^{S-1} d_s(x) + f_S(x) \tag{1.56}$$

The details at a given scale S cover a certain bandpass region in frequency space. The details at scale $S+1$ use wider wavelet basis and, therefore, cover another bandpass region of lower frequency than at scale S. Adding the details at scales $S+2$, $S+3$, ..., ∞ really covers the whole low frequency spectrum up to scale S. Thus, we can interpret this equation as the decomposition of $f(x)$ as a lowpass approximation $f_S(x)$ plus the superposition of all details finer than those at scale S. We could define a function φ_S such that it implements the lowpass filter needed to produce f_S such that

$$f_S = \sum_{i=-\infty}^{\infty} \langle f, \tilde{\varphi}_{Si} \rangle \, \varphi_{Si}. \tag{1.57}$$

For $S = 1$, φ_1 (or simply φ) is called the scaling function and plays a central role in wavelet theory since it gives rise to a very efficient implementation of the wavelet decomposition based on filter banks [28]. It can be shown [27] that the admissibility criterion of the mother wavelet can be translated into three criteria that the scaling factor must meet:

– Riesz basis: the scaling function must be a Riesz basis, i.e., there must be two real constants $0 < A \leq B < \infty$ such that

$$A \leq \sum_{k=-\infty}^{\infty} \left| \widehat{\varphi}(j(\Omega + 2\pi k)) \right|^2 \leq B. \tag{1.58}$$

– Two-scale relationship: there must exist a sequence $h_i \in l_2$ allowing to express dilations of the scaling function

$$\varphi\left(\frac{x}{2}\right) = 2h_{\mathrm{II}}(x) \star \varphi(x). \tag{1.59}$$

This property is key to have multiresolution analysis and its equation is sometimes called the dilation equation. Scaling functions must be solutions of this equation.

– Partition of unity: the sum of the scaling function at the integer points must add up to 1

$$\sum_{i=-\infty}^{\infty} \varphi(x-i) = 1 \tag{1.60}$$

This property is key to have a dense representation in L_2 as finer and finer details are added, i.e., to reduce the representation error as much as desired when $s \to -\infty$.

An important relationship between mother wavelets and the scaling function is that the mother wavelet can be uniquely determined by a sequence $g_i \in l_2$ such that [27]

$$\psi\left(\frac{x}{2}\right) = 2g_{\mathrm{II}}(x) \star \varphi(x). \tag{1.61}$$

This equation is called the wavelet equation and it is important because it allows to define wavelets by simply choosing a scaling function (how to compute the g_i sequence will be shown later when talking about filter banks). Therefore, from the knowledge of the scaling function we can construct automatically the corresponding mother wavelet (used for synthesis) and the two corresponding dual functions for analysis (the dual scaling function and the dual mother wavelet). In the orthogonal case, the dual scaling function is equal to the scaling function, and the dual mother wavelet is equal to the mother wavelet.

The design of wavelets is intimately related to the concept of perfect reconstruction in filter banks. In fact in the subsequent sections we will make use of the notation usually used in filter banks for showing the relationship between splines and wavelets. Let us first quickly review the main results of filter banks related to our purpose.

Consider the filter bank shown in Fig. 1.5. Usually, $H_0(z)$ is a discrete lowpass filter, and $H_1(z)$ is a discrete highpass filter. They are called the analysis filters and may overlap in frequency, causing thus some aliasing after downsampling. This aliasing has to be compensated by the synthesis filters $G_0(z)$ and $G_1(z)$. The perfect reconstruction condition is achieved if after the decomposition and reconstruction of the signal we have $\tilde{x}[n] = x[n-n_0]$ where n_0 is some delay (when processing images, it is usually preferred to set $n_0 = 0$). It can be proved [28][Chap.4] that to achieve this goal, it is a necessary and sufficient condition to have

$$\begin{array}{rcl} H_0(z)G_0(z) + H_1(z)G_1(z) & = & 2 \\ H_0(-z)G_0(z) + H_1(-z)G_1(z) & = & 0 \end{array} \tag{1.62}$$

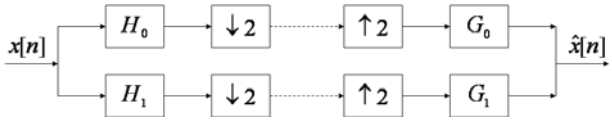

Fig. 1.5 Filter bank example with a downsampling of 2 (as it is commonly used in wavelet theory). The dashed lines represents where the signal is transmitted, compressed, etc. For perfect reconstruction we will assume that there is no loss in the transmission of the signal.

The first equation takes care of the distortion through the filterbank, while the second equation takes care of alias cancellation. An easy way to fulfill the second equation is by setting $G_0(z) = H_1(-z)$ and $G_1(z) = -H_0(-z)$. In this way, we only have to design two filters (G_0 and H_0, for instance) instead of four (H_0, H_1, G_0 and G_1). With this constraint for the reconstruction filters, and defining $P(z) = H_0(z)G_0(z)$, the non-distortion condition can be rewritten as [28][Chap.4]

$$P(z) + P(-z) = 2. \tag{1.63}$$

The connection between wavelets and filter banks is that the sequences h_i and g_i needed for the dilation and wavelet equations are precisely the impulse responses of the digital filters $G_0(z)$ and $G_1(z)$, respectively. G_0 is called the refinement filter. The impulse responses of H_0 and H_1 yield the corresponding sequences for the dual scaling function and the dual mother wavelet. Knowing G_0 permits us to establish the equation the scaling function must fulfill, Eq. (1.59), however we still have to solve it. It can be proved [28][Chap.6] that the Fourier transform of the scaling function is

$$\widehat{\varphi}(j\Omega) = \prod_{k=1}^{\infty} H\left(e^{j\frac{\Omega}{2^k}}\right) \tag{1.64}$$

where $H(e^{j\omega})$ is the Fourier transform of the discrete sequence h_i in Eq. (1.59).

An important result from wavelet theory is that if the filter bank is to be used for a wavelet decomposition, then $P(z)$ can be factorized as

$$P(z) = \left(\frac{1+z^{-1}}{2}\right)^N Q(z), \tag{1.65}$$

where $Q(z)$ is a distribution and $\left(\frac{1+z^{-1}}{2}\right)^N$ is the Z transform of the sequence of samples at integer values of the causal B-spline of degree $N-1$. In fact, many of the interesting wavelet properties, as will be seen later, come from the fact that they have B-splines in their design.

Summarizing, the design of a wavelet is as follows: choose a $Q(z)$ such that $P(z)$ fulfills Eq. (1.63); then split the factors of $P(z)$ (see Eq. (1.65)) into $H_0(z)$ and $G_0(z)$; finally, use the alias cancellation constraint to find $H_1(z)$ and $G_1(z)$. If these filters are designed to be linear phase, the corresponding scaling functions and mother wavelets are symmetric and antisymmetric, respectively.

So far we have discussed the so-called biorthogonal filter banks. This name is related to the use of a basis function for synthesis and its dual for analysis. Orthogonal filter banks (and orthogonal) basis functions are a particular case of the biorthogonal ones in which the analysis and synthesis functions (filters in the case of filter banks) are the same (note that in the case of filter banks they can only be the same filter if $P(z)$ is centered around $n = 0$). Orthonormal filter banks are a special case of orthogonal filter banks in which the impulse response of the filters involved have unit energy (conversely, the norm of the corresponding orthonormal basis functions is 1).

Orthonormality of the wavelets can also be expressed in Fourier space and doing this helps to build orthonormal wavelets. Let a_i be the inner products among shifted versions of the scaling function, i.e., $a_i = \langle \varphi(x), \varphi(x-i) \rangle$. Obviously, the orthonormality condition is $a_i = \delta_i$ (i.e., zero for all i except $i = 0$ for which it is 1). Let $A(z)$ be the Z transform of the sequence a_i and $A(e^{j\omega})$ its Fourier transform. The orthonormality condition is then $A(z) = 1$ and $A(e^{j\omega}) = 1$. It can be proved [28][Chap.6] that if φ is not an orthonormal scaling function, then the following scaling function is orthonormal

$$\widehat{\varphi}_{\mathrm{orth}}(j\Omega) = \frac{\widehat{\varphi}(j\Omega)}{\sqrt{A(e^{j\Omega})}}. \tag{1.66}$$

Interesting wavelet properties can be obtained through a correct design of $P(z)$ [29][Chap.35]. For instance: if $p[n]$ (the inverse Z transform of $P(z)$) is obtained through the autocorrelation of a sequence $r[n]$, then the corresponding wavelet is orthogonal; if $P(z)$ is a linear phase system, and it is decomposed into two linear phase systems H_0 and G_0, then the scaling and wavelet functions are symmetric and antisymmetric, respectively. If $P(z)$ is FIR (Finite Impulse Response) filter, and H_0 and G_0 are also FIR filters, then the corresponding scaling and wavelet functions are compactly supported.

Wavelets based on splines

There are a number of wavelets explicitly based on splines. We will categorize them according to their orthogonality properties:

- **Biorthogonal spline wavelets**: The simplest approach to wavelet design with splines is to consider the scaling function to be a B-spline of degree N. Then, $G_0(z) = \left(\frac{1+z^{-1}}{2}\right)^N$. H_0 must be of the form $H_0(z) = \left(\frac{1+z^{-1}}{2}\right)^{N'} Q(z)$. A natural choice is to select $Q(z)$ to be a polynomial of degree $q > N - 2$ (this is needed to have at least a zero at $z = -1$ [28][Chap.7]). The coefficients of the $Q(z)$ polynomial are determined through the non-distortion equation, Eq. (1.63). This design has a number of drawbacks. Note that the scaling function (a B-spline) is not orthogonal to integer shifted versions of itself. The scaling function is not orthogonal, neither, to its wavelet function which complicates, in part, the mathematics of the decomposition. Due to this biorthogonality we have different filters for analysis (dual filters) and synthesis. Finally, the space spanned by the scaling function (all B-splines of degree at most N) is not the

same as the space spanned by the dual scaling function (which is not even defined in terms of splines). On the other hand, it has an important advantage: the analysis and synthesis filters are FIR what makes them very convenient.

– **Semiorthogonal spline wavelets**: Assume that G_0 is defined as in the biorthogonal case. If we want the scaling function and its mother wavelet to be orthogonal (which simplifies the expansion since we can consider the space of functions to be a direct sum of the spaces spanned by the scaling function, and the spaces spanned by the wavelets at different scales), then we can construct the highpass filter G_1 as $G_1(z) = -z^{1-2N}G_0(z)A(-z^{-1})$ (H_0 and H_1 are computed through the perfect reconstruction equations). Doing this we achieve orthogonality between the scaling function and its mother wavelet. Wavelets at different scales are also orthogonal (however, wavelets at a given scale are not orthogonal with shifted versions of themselves). The scaling function and its dual scaling function are neither orthogonal nor the same, but the space spanned by both functions is the same [28][Chap.7]. The price paid for these advantages is the appearance of IIR filters on the analysis side.

– **Orthogonal spline wavelets**: If we orthogonalize the scaling function using Eq. (1.66), then we obtain orthogonal scaling and wavelet functions. Now we have the same analysis and synthesis filters and spaces at the cost of having IIR filters in both operations. These orthogonalized spline wavelets are also known as Battle-Lemarie wavelets.

A notable property of these families of wavelets is that all of them, for a given order N, seem to have the best approximation properties among all wavelet families if the expansion is truncated at a given scale [30].

The previous wavelets were based on a scaling function with a factor of the form $\left(\frac{1+z^{-1}}{2}\right)^N$ which we attributed to the Z transform of the sequence $b_k = \beta_+^{N-1}(k)$, i.e., samples at the integer values of the causal B-spline of degree $N-1$. In this construction, N must be an integer number and the Fourier transform of the B-spline (as continuous function) is

$$\widehat{\beta}_+^{N-1}(j\Omega) = \left(\frac{1-e^{-j\Omega}}{j\Omega}\right)^N. \tag{1.67}$$

Causal fractional B-splines are defined in Fourier space as [30]

$$\widehat{\beta}_+^{\alpha-1}(j\Omega) = \left(\frac{1-e^{-j\Omega}}{j\Omega}\right)^\alpha, \tag{1.68}$$

where α is any real number (not necessarily integer) larger than -1. For non-integer values of α the B-spline is no longer compactly supported. Fractional B-splines generate valid scaling functions as long as $\alpha > -\frac{1}{2}$ and have a fractional approximation order α. Causal fractional B-splines can also be symmetrized yielding a generalization of the centered B-splines of integer degree [30]. It is well known that B-splines of degree N can reproduce any polynomial of degree N. A remarkable property of fractional B-splines of degree α ($N-1 < \alpha < N$) is that they can also reproduce polynomials of degree N, i.e., they can reproduce polynomials of a degree slightly larger than α.

Properties of spline wavelets

Wavelets are useful for representing functions because they are dense in $L_2(\mathbb{R})$, i.e., they can approximate any arbitrary function in $L_2(\mathbb{R})$ by adding finer and finer details. Moreover, the number of coefficients significantly different from zero can be made small by increasing the approximation order. It is known that the approximation order of the wavelet is related to the number of vanishing moments [27]. In this section we will show that many of these properties of wavelets are due to the degree of the B-spline present in its scaling function. The reader interested in this topic is referred to [27].

It can be proved [27] that a function φ is an admissible scaling function if and only if it can be decomposed as a causal fractional B-spline ($\alpha > -\frac{1}{2}$) and a stable tempered distribution:

$$\varphi = \beta_+^{\alpha-1} * \varphi_0. \tag{1.69}$$

A tempered distribution is a generalization of functions to "generalized functions" such as Dirac's delta, Heaviside's step function, and any linear combination of them. A tempered distribution φ_0 is stable if $|\widehat{\varphi}_0(j\Omega)|$ is bounded (for instance, the Fourier transform of Dirac's delta is bounded but the Fourier transform of Heaviside's step function is not). Alternatively, one could also show [27] that the refinement filter $G_0(z)$ associated to the scaling function can be decomposed as

$$G_0(z) = \left(\frac{1+z^{-1}}{2}\right)^{\alpha} G_0'(z). \tag{1.70}$$

For integer α, the corresponding B-spline is compactly supported and gives raise to a FIR filter. If α is not integer, then the corresponding B-spline is not compactly supported and its associated filter is IIR.

In the previous section we already discussed that fractional B-splines of degree α ($N-1 < \alpha < N$) can reproduce polynomials of degree N. This property is preserved through convolution [27] so that the scaling function can also reproduce polynomials of degree N.

Under the previous decomposition of the scaling function, it can also be proved [27] that the dual mother wavelet (the one used for analysis) has $N+1$ vanishing moments, i.e., the fact that the wavelet coefficients decay very fast over scales is purely due to the B-spline contained in its scaling function. The approximation order of the wavelet decomposition is α. The same decomposition could be done on the dual scaling function $\tilde{\varphi} = \beta_+^{\tilde{\alpha}-1} * \tilde{\varphi}_0$ meaning that the synthesis wavelet has $\tilde{N}+1$ ($\tilde{N}-1 < \tilde{\alpha} < \tilde{N}$) vanishing moments.

It can also be shown [27] that the analysis wavelet practically behaves as a α-th order differentiator, i.e., $\widehat{\phi}(j\Omega) \propto (j\Omega)^{\alpha}$ as $\Omega \to 0$. More specifically,

$$\left\langle f(x), \tilde{\psi}(x - x_i) \right\rangle = \partial^{\alpha}\{\phi * f\}(x_i), \tag{1.71}$$

i.e., the wavelet coefficients are the derivative of order α of a smoothed version of f, evaluated at the point x_i. The smoothing kernel is defined by its Fourier transform $\widehat{\phi}(j\Omega) = \frac{\widehat{\tilde{\psi}}^*(j\Omega)}{(j\Omega)^{\alpha}}$. Note that because of the smoothing, the differentiation process is

more robust with respect to noise. Moreover, the α-th order derivative of the scaling function is in $L_2(\mathbb{R})$ (i.e., the scaling function belongs to the Sobolev space of order α).

One may wonder what is, then, the role of the distributional part of the scaling function, $G'_0(z)$. It is mainly responsible of the orthogonalization and localization properties of the wavelet decomposition. A careful design of the distributional part of the scaling functions allows to have FIR filters in the analysis side, the synthesis side or both. Furthermore, fractional B-splines are not orthogonal, the distributional part is the one in charge of providing the necessary orthogonality properties of the scaling function and its corresponding wavelets.

1.3 Main applications of splines for biomedical imaging

1.3.1 Geometric image/volume transformations

Geometric image transformations such as resizing, rotation and translation are common operations in image processing. In medical imaging, there are three typical applications of image resizing: re-slicing to compensate for the fact that the resolution of the 3D data is often anisotropic (i.e., the within-slice resolution of CT or MR data is typically finer than the across-slice resolution), image zooming for focusing on details for diagnostic purposes, and image pyramids for multi-scale processing (e.g., many iterative image processing algorithms can be applied in a coarse-to-fine fashion, which reduces the computation time and also tends to improve robustness [4]). The so-called rigid-body transformation is described by a combination of rotation and translation. It is commonly used in the so-called rigid-body intensity-based image registration techniques (Section 1.3.2). A general affine geometric transformation covers any combination of translation, rotation around any center, skewing, shearing, mirroring, and scaling. Its typical application is in elastic image registration (Section 1.3.3).

When no image/volume size reduction is required, the geometric transformations are often implemented by standard spline interpolation. The simplest methods are nearest-neighbour and bilinear interpolation. The piecewise constant model generates noticeable blocking artefacts, while the bilinear one tends to lose details through image blurring. Better interpolation performance is achieved by switching to higher order spline interpolation, such as cubic B-spline interpolation, which offers a good trade-off between the computational cost and the interpolation quality [4]. For image/volume size reductions, interpolation is not entirely suitable because of potential aliasing problems [31]. The standard remedy is to apply some kind of low-pass filtering prior to re-sampling. An alternative is to obtain the best solution in the least-squares sense to reduce aliasing artefacts [31]. A method that allows resizing images/volumes with arbitrary scaling factors (not necessarily powers of two) by computing oblique or orthog-

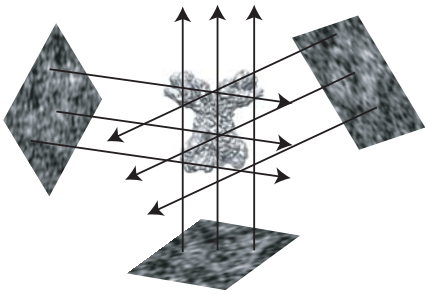

Fig. 1.6 3D electron microscopy. Experimental images corresponding to three arbitrarily chosen projection directions and a 3D model of the Phosphorylase Kinase [37]. The arrows represent the directions of back-projection in 3D space.

onal projections (least-squares approximations) for splines of any degree was proposed in [32].

1.3.2 Rigid-body image/volume registration

The intensity-based registration can be viewed in an optimization framework in which the registration problem consists in searching for a geometric transformation of a source image/volume that gives the image/volume that best matches a target image/volume, under a chosen similarity measure. The restriction of the motion to rigid-body motion means that the distance between the points of the object is the same in the registered source and target images/volumes. Two typical biomedical applications of the rigid-body image/volume registration are computer-assisted orthopaedic surgery [33] and 3D electron microscopy [34]. A problem of a particular interest for both applications is the registration (alignment) of a 3D model to a set of 2D projection images, which requires a rigid-body transformation of the volume for the computation of projections [35].

In computer-assisted orthopaedic surgery, the need arises to visualize intra-operatively the position of surgical tools on preoperative 3D CT patient data [33]. A solution to the problem of integration of the CT into the intra-operative procedure is to align the CT with a set of intra-operatively acquired cone-beam X-ray images of the same patient [36].

In 3D electron microscopy, the structure of a macromolecular complex can be computed in three dimensions from a set of parallel-beam projection images of the same complex acquired in a microscope [34]. For the so-called single-particle analysis, images of the sample containing thousands of copies of the same complex are collected. Ideally, the copies have the same structure. Their orientation in the 3D space is random and unknown, and the position of the center of each complex is unknown. These parameters have to be estimated before applying a method for 3D reconstruction. Given

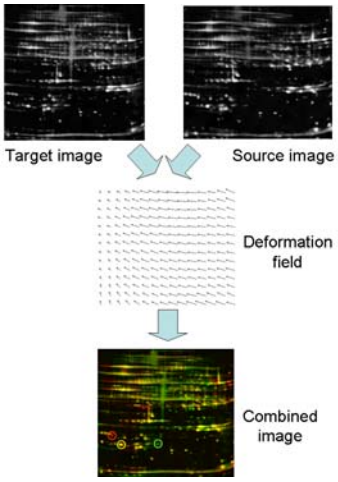

Fig. 1.7 Elastic registration of two 2D protein gels (source and target) using the method [39]. The deformation field represents the continuous deformation required to convert the source into the target. The combined image shows the target image in the red channel and the warped source in the green channel. Note that there are red spots (proteins expressed in the target image and not expressed in the source image), green spots (just the opposite), and yellow spots (proteins equally expressed in both images).

a first guess for the 3D model, one estimates the unknown parameters by aligning the images with the 3D model (reference model) [38] (Fig. 1.6). A new reconstruction is then computed using the images and the estimated orientations and positions. The new model can be used as the reference model to resume the alignment in the next iteration of the iterative refinement of the estimated parameters [34]. It has been shown that such procedures can yield 3D models of sub-nanometer resolution [37].

1.3.3
Elastic image/volume registration

Elastic registration is frequently employed in medical image analysis to combine data which describe anatomy, both because biological tissues are in general not rigid and because anatomy varies between individuals. Various techniques have been developed for intensity-based registration, as well as for matching surfaces, curves, or landmarks in two images or two volumes [40, 41]. In the intensity-based techniques, the solution of the registration problem is the deformation field that warps the source image/volume so that the resulting image/volume best matches the target one. The registration is achieved by minimizing a cost function, which represents a combination of the cost associated with the image/volume similarity and the cost associated with the smoothness of the transformation (regularization term).

Many authors have proposed to use linear combinations of B-splines placed on a regular grid to model the deformation field [42, 43, 44]. The available techniques differ in the form of employed regularization term, as well as in the employed image/volume similarity metrics and the optimization method. The approaches that model the deformation field with a linear combination of cubic B-splines produce good results but have a high computational cost. The computation can be accelerated using multi-scale image/volume processing, and spline pyramids provide a convenient tool for this. Moreover, spline model can be used for all computation aspects of the registration: image pyramid, geometric transform, and the gradient of the optimization criterion [4, 36].

An illustration of elastic image registration is given in Fig. 1.7. One of the major difficulties in the analysis of electrophoresis 2D gels is that the gels are affected by spatial distortions due to run-time differences and dye-front deformations, which results in images that differ significantly in the content and in the geometry. The method proposed in [39] models the deformation field using B-splines, whose advantage is that the model can be adapted to any continuous deformation field simply by changing the spacing between splines [39]. The method computes quasi-invertible deformation fields so that the source image can be mapped onto the target image and vice versa, which helps the optimizer to reduce the chance of getting trapped in a local minimum and allows the simultaneous registration of any number of images [39].

1.3.4 Contour and shape modelling

Shape analysis is required in many biomedical imaging applications. B-splines are well suited to the contour and surface modelling. For example, NURBS geometries, based on B-spline basis functions, are ideal for modelling anatomical shapes that cannot easily be reduced to analytical forms such as straight lines, ellipses, or spheres, and have long been the industry standard for computer-aided modelling of free-form geometries [45]. NURBS are thus used in computational musculoskeletal models, which are 3D graphics-based models that accurately simulate the anatomical architecture and/or the biomechanical behaviour of muscles, tendons, ligaments, cartilage and bones, with applications ranging from pathological diagnosis to surgical planning [45].

Another issue is also the determination of location, orientation and size of filamentous or spherical bright structures in images. These problems arise in segmentation and characterization of biological cell images, analysis of vessel distributions in medical images, and detection of DNA filaments in electron micrograph images. The evaluation of low order moments is a standard method for shape analysis. Since moments are integral-based features, they are robust against noise. Furthermore, low order moments have a direct geometrical interpretation. The method described in [46] uses local weighted geometric moments that are computed from an image within a sliding window, at multiple scales and with B-splines as window functions. B-splines are well-suited window functions because they are positive, symmetric, separable, and nearly isotropic, in addition to being refinable [46]. Besides feature extraction, these

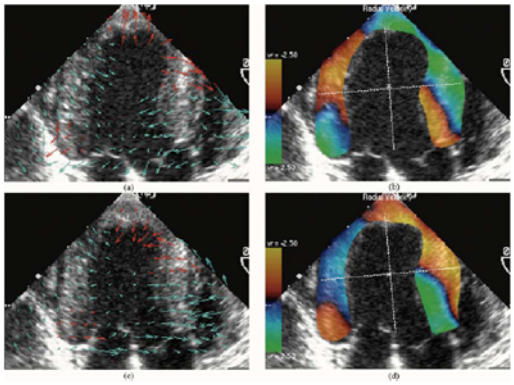

Fig. 1.8 Echocardiograms with superimposed motion information after infarction. (a) Estimated velocity field during systole. (b) Color-coded radial velocity during systole. (c) Velocity field during diastole. (d) Color-coded radial velocity during diastole. Regions with abnormal wall motion (dyskinesia) are highlighted by red arrows. The principal axes of inertia of the ROI are indicated by dotted lines. Reprinted, with permission, from [50], © 2005 IEEE.

local moments can be used in many algorithms of sliding-window type, such as noise reduction and optical-flow estimation [46].

Let us also mention "snakes" that are standard tools for extracting deformable contours in images. A snake is an energy minimization spline segment with external and internal forces [47]. It simulates an elastic material that can dynamically conform to local image features. The internal forces constrain the rigidity of the curve and, thus, have regularization function. In the case of B-spline snakes, the smoothness of the curve can be alternatively controlled by adapting the scale of the basis functions [48]. An example of using spline snakes is the estimation of the cell boundary in the multiparameter fluorescence imaging for automatic phenotyping of multiple cell types [49].

1.3.5
Motion reconstruction

Non-rigid motion analysis is an important issue in medical image analysis. The basic problem involves the estimation/reconstruction of a dense motion of the elastic material from sparse measurements of that motion, using some form of elastic splines (e.g., thin-plate spline, snakes). For instance, several approaches have been proposed for automated computer-based analysis to quantify heart motion from 2D echocardiographic sequences. One approach consists of segmenting and tracking myocardial borders using active contours or active appearance models [47, 51]. However, motion information is only obtained for myocardial borders which are often poorly defined. A different approach is to estimate motion for the entire image content using so-called optical flow methods [52]. A popular optical flow algorithm is the Lucas-Kanade method, which

estimates the motion locally, assuming that the velocity field is constant within a window [53].

The parameters of the local motion model can be estimated in the weighted least-squares sense inside a sliding spatio-temporal B-spline window [50]. To account for typical heart motions, such as contraction/expansion and shear, the method analyzes the images locally by using a local affine model for the velocity in space and a linear model in time (Fig. 1.8). Robustness and spatial adaptability is achieved by estimating the model parameters at multiple scales within a coarse-to-fine multiresolution framework. To increase computational efficiency, a wavelet-like algorithm is used for computing B-spline-weighted inner products and moments at dyadic scales [50]. Since the estimated velocity data itself is not of direct use, it has to be processed and visualized properly to facilitate the diagnosis. In order to focus on the relevant regions of the heart, the motion information should only be displayed inside a user-defined region of interest (ROI) that typically corresponds to the myocardium. In order to follow the movement of the myocardium, the ROI contour should be automatically tracked in time using the estimated velocity field. In the method [50], robustness of the tracking process is achieved using a spline representation of the ROI contours that is fitted in the least-squares sense to the estimated motion field.

1.4 Conclusions

In this chapter, we presented the main theoretical achievements concerning splines and their main applications in biomedical image and volume processing. We reviewed spline interpolation and approximation theory by presenting two spline families: tensor product splines and polyharmonic splines. We dicussed the traditional computer-graphics curve/surface formulation and the advantages of using it in image processing. Also, we presented the multiscale properties of splines that allow the definition of spline pyramids and spline wavelets. Finally, we showed applications of splines in geometric transformations, alignment, contour and shape modelling, and motion estimation.

References

[1] Unser, M., "Splines: A perfect fit for signal and image processing," *IEEE Signal Processing Magazine*, pp. 22–38, 1999.

[2] Unser, M., "Sampling – 50 years after Shannon," *Proc. IEEE*, 88, pp. 569–587, 2000.

[3] Blu, T., Thevenaz, P. and Unser, M., "MOMS: Maximal-Order interpolation of Minimal Support," *IEEE Trans. Image Processing*, 10, pp. 1069–1080, 2001.

[4] Thevenaz, P., Ruttimann, U. and Unser, M., "A pyramid approach to subpixel registration based on intensity," *IEEE Trans Med Imaging*, 7, pp. 27–41, 1998.

[5] Weierstrass, K., "Über die analytische Darstellbarkeit sogenanter willkürlicher Functionen einer reellen Veränderlichen," *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften zu Berlin*, pp. 633–789, 1884.

[6] Whittaker, E. T., "On the functions which are represented by expansion of the interpolation theory," *Proc. Roy. Soc. Edinburgh*, A35, pp. 181–194, 1914.

[7] Shannon, C. E., "Communication in the presence of noise," *Proc. Institute Radio Engineers*, 37, pp. 10–21, 1949.

[8] Unser, M., Aldroubi, A. and Eden, M., "B-Spline signal processing: Part I – Theory," *IEEE Trans. Signal Processing*, 41, pp. 821–832, 1993.

[9] Unser, M., Aldroubi, A. and Eden, M., "B-Spline signal processing: Part II – Efficient design and applications," *IEEE Trans. Signal Processing*, 41, pp. 834–848, 1993.

[10] Meijering, E., "A chronology of interpolation: from ancient astronomy to modern signal and image processing," *Proceedings of the IEEE*, 90, pp. 319–342, 2002, doi: 10.1109/5.993400.

[11] Eldar, Y. and Michaeli, T., "Beyond bandlimited sampling," *IEEE Signal Processing Magazine*, 26, pp. 48–68, 2009, doi:10.1109/MSP.2009.932125.

[12] Thevenaz, P., Blu, T. and Unser, M., "Interpolation revisited," *IEEE Trans. Medical Imaging*, 19, pp. 739–758, 2000.

[13] Schoenberg, I. J., "Contributions to the problem of approximation of equidistant data by analytic functions," *Quart. Appl. Math.*, 4, pp. 45–99,112–141, 1946.

[14] Piegl, L. and Tiller, W., *The NURBS book*, Springer, 1997.

[15] Unser, M. and Blu, T., "Self-similarity: Part I – Splines and operators," *IEEE Trans. Signal Processing*, 55, pp. 1353–1363, 2007.

Please enter `\copyrightinfo{Title, Edition}{Author/Editor}{ISBN number}` at the beginning of your document.

[16] Blu, T. and Unser, M., "Self-similarity: Part II – Optimal estimation of fractal processes," *IEEE Trans. Signal Processing*, 55, pp. 1364–1378, 2007.

[17] Duchon, J., "Splines minimizing rotation-invariant semi-norms in Sobolev spaces," in W. Schempp and K. Zeller (eds.), "Constructive Theory of Functions of Several Variables. Lecture Notes in Mathematics.", volume 571, pp. 85–100, Springer-Verlag, 1977.

[18] Iske, A., "On the aproximation order and numerical stabilty of local Lagrange interpolation by polyharmonic splines," in "Proc. 5th Intl. Conf. Modern Developments in Multivariate Approximation," 2003.

[19] Beatson, R. K., Powell, M. J. D. and Tan, A. M., "Fast evaluation of polyharmonic splines in three dimensions," *IMA Journal of Numerical Analysis*, 27, pp. 427–450, 2007.

[20] Kybic, J., *Elastic Image Registration Using Parametric Deformation Models*, Ph.D. thesis, EPFL, 2001.

[21] Van de Ville, D., Blu, T. and Unser, M., "Isotropic polyharmonic B-splines: scaling functions and wavelets," *IEEE Trans. Image Processing*, 14, pp. 1798–1813, 2005.

[22] Wahba, G., *Spline Models for Observational Data*, SIAM, Philadelphia, PA, 1990.

[23] Carr, J. et al., "Reconstruction and representation of 3D objects with radial basis functions," in "Proc. of SIGGRAPH," pp. 67–76, 2001.

[24] Madych, W. R. and Nelson, S. A., "Polyharmonic cardinal splines," *Journal of Approximation Theory*, 60, pp. 141–156, 1990.

[25] Thevenaz, P. and Millet, P., "Multiresolution imaging of in vivo ligand-receptor interactions," in "Proc. of the SPIE International Symposium on Medical Imaging," San Diego, 2001.

[26] Unser, M., Aldroubi, A. and Eden, M., "The L_2 polynomial spline pyramid," *IEEE Trans. Pattern Analysis & Machine Intelligence*, 15, pp. 364–378, 1993.

[27] Unser, M. and Blu, T., "Wavelet theory demystified," *IEEE Trans. Signal Processing*, 51, pp. 470–483, 2003.

[28] Strang, G. and Nguyen, T., *Wavelets and Filter Banks*, Wellesley-Cambridge Press, 1997.

[29] Madisetti, V. J. and Williams, D., *Digital Signal Processing Handbook*, CRC Press, 1999.

[30] Unser, M. and Blu, T., "Fractional splines and wavelets," *SIAM Review*, 42, pp. 43–67, 2000.

[31] Unser, M., Aldroubi, A. and Eden, M., "Enlargement or reduction of digital images with minimum loss of information," *IEEE Trans Image Process*, 4, pp. 247–58, 1995.

[32] Munoz, A., Blu, T. and Unser, M., "Least-squares image resizing using finite differences," *IEEE Trans Image Process*, 10, pp. 1365–78, 2001.

[33] Nolte, L. P. and Ganz, R. (eds.), *Computer Assisted Orthopedic Surgery (CAOS)*, Hogrefe & Huber Publishers, Seattle-Toronto-Bern-Göttingen, 1999.

[34] Frank, J., *Three-Dimensional Electron Microscopy of Macromolecular Assemblies*, Academic Press, 1996.

[35] Jonic, S. et al., "Spline-based 3D-to-2D image registration for image-guided surgery and 3D electron microscopy," in M. Faupel, P. Smigielski, A. Brandenburg and J. Fontaine (eds.), "Biophotonics for Life Sciences and Medicine," pp. 255–273, Fontis Media, Lausanne VD, Switzerland, 2006.

[36] Jonic, S. et al., “An optimized spline-based registration of a 3D CT to a set of C-arm images,” *International Journal of Biomedical Imaging*, 2006, 2006, article ID 47197, 12 pages, doi:10.1155/IJBI/2006/47197.

[37] Venien-Bryan, C. et al., “The structure of Phosphorylase Kinase holoenzyme at 9.9 angstroms resolution and location of the catalytic subunit and the substrate Glycogen Phosphorylase,” *Structure*, 17, pp. 117–27, 2009.

[38] Jonic, S. et al., “Spline-based image-to-volume registration for three-dimensional electron microscopy,” *Ultramicroscopy*, 103, pp. 303–17, 2005.

[39] Sorzano, C. et al., “Elastic image registration of 2-D gels for differential and repeatability studies,” *Proteomics*, 8, pp. 62–5, 2008.

[40] Bookstein, F., *Morphometric Tools for Landmark Data: Geometry and Biology*, New York: Cambridge Univ. Press, 1997.

[41] Toga, A. (ed.), *Brain Warping*, New York: Academic, 1999.

[42] Rueckert, D. et al., “Nonrigid registration using free-form deformations: application to breast MR images,” *IEEE Trans Med Imaging*, 18, pp. 712–21, 1999.

[43] Studholme, C., Constable, R. and Duncan, J., “Accurate alignment of functional EPI data to anatomical MRI using a physics-based distortion model,” *IEEE Trans Med Imaging*, 19, pp. 1115–27, 2000.

[44] Kybic, J. et al., “Unwarping of unidirectionally distorted EPI images,” *IEEE Trans Med Imaging*, 19, pp. 80–93, 2000.

[45] Wu, F. et al., “Computational representation of the aponeuroses as NURBS surfaces in 3D musculoskeletal models,” *Comput Methods Programs Biomed*, 88, pp. 112–22, 2007.

[46] Suhling, M. et al., “Multiresolution moment filters: theory and applications,” *IEEE Trans Image Process*, 13, pp. 484–95, 2004.

[47] Kass, M., Witkin, A. and Terzopoulos, D., “Snakes: Active contour models,” *Int J Comput Vision*, 1, pp. 321–331, 1988.

[48] Brigger, P., Hoeg, J. and Unser, M., “B-spline snakes: a flexible tool for parametric contour detection,” *IEEE Trans Image Process*, 9, pp. 1484–96, 2000.

[49] Dow, A. et al., “Automatic multiparameter fluorescence imaging for determining lymphocyte phenotype and activation status in melanoma tissue sections,” *Cytometry*, 25, pp. 71–81, 1996.

[50] Suhling, M. et al., “Myocardial motion analysis from B-mode echocardiograms,” *IEEE Trans Image Process*, 14, pp. 525–36, 2005.

[51] Bosch, J. et al., “Automatic segmentation of echocardiographic sequences by active appearance motion models,” *IEEE Trans Med Imaging*, 21, pp. 1374–83, 2002.

[52] Baraldi, P. et al., “Evaluation of differential optical flow techniques on synthesized echo images,” *IEEE Trans Biomed Eng*, 43, pp. 259–72, 1996.

[53] Lucas, B. and Kanade, T., “An iterative image registration technique with an application to stereo vision,” in “Proc. Imaging understanding workshop,” pp. 121–130, 1981.