POLYNOMIAL SPLINE SIGNAL PROCESSING ALGORITHMS

Michael Unser and Akram Aldroubi

Biomedical Engineering and Instrumentation Program, Bldg. 13, Room 3W13
National Institutes of Health, Bethesda, MD 20892 USA

ABSTRACT

We describe new digital filtering algorithms for the processing and representation of signals using polynomial splines. We first consider the classical polynomial spline interpolation problem and show that it can be solved efficiently by recursive digital filtering. This result also yields a simple procedure for signal differentiation. We then derive filters that efficiently solve the problem of smoothing spline approximations. This technique is a regularized version of spline interpolation and is therefore less sensitive to noise. It is applied to the design of a robust edge detection algorithm with an adjustable scale parameter. Finally, we describe a filtering/sampling algorithm for least squares spline approximation. This data reduction technique is applied to the generation of a cubic spline image pyramid that is found to compare favorably with the Gauss/Laplace pyramid.

1. INTRODUCTION

In digital signal (or image) processing applications, the signals to be manipulated are represented by a set of uniformly spaced sampled values $g(\mathbf{k})$, where $\mathbf{k}$ is a vector of integer indices. Although most processing algorithms are purely discrete, there is a variety of problems that are best formulated by considering a function $g(\mathbf{x})$ of the continuous (multidimensional) variable $\mathbf{x}$; the simplest example being signal differentiation. It is therefore of interest to develop a simple and consistent procedure for mapping discrete signals into continuous ones and vice versa.

In this paper, we will consider representation of signals using polynomial splines. These functions have a number of attractive properties (simple representation, smoothness, etc) that make them very useful in numerical analysis [1, 2]. There have been some early attempts of using splines in signal processing [3], but one of the main obstacle for their acceptance in the field has been the complexity of the standard numerical algorithms that traditionally rely on matrix algebra. It is only recently that it has been fully recognized that the operations involved, in the special case of equally spaced sample points, are translation-invariant and that the interpolation problem could be solved efficiently by digital filtering [4].

The purpose of this paper is to extend the idea of translation-invariant processing via digital filters to a larger class of spline techniques including smoothing splines and least squares approximations. The corresponding algorithms are essentially filtering techniques and are therefore well suited for signal processing. The presentation of these results is mainly expository. For a full mathematical treatment and a detailed discussion of the implementation of these techniques, we refer the reader to the forthcoming series of reports [5, 6].

2. PRELIMINARY NOTIONS

2.1 Polynomial splines

Polynomial splines of order n are C^{n-1} functions (i.e., continuous functions with continuous derivatives up to order $n-1$) that are piecewise polynomials of degree n. Here, we consider a subdivision in polynomial segments defined over the intervals $[k,k+1)$, $k\in Z$ when n is odd, and $[k-1/2,k+1/2)$, $k\in Z$ when n is even.

A classical result in approximation theory is that such polynomial splines can be represented by the expansion

$$g^n(x)=\sum_{k=-\infty}^{+\infty} c(k)\beta^n(x-k), \tag{1}$$

where $\beta^n(x)$ is Schoenberg's central B-spline of order n [7]. This representation is unique in the sense that any polynomial spline $g^n(x)$ is entirely characterized by its sequence of B-spline coefficients $\{c(k)\}_{k\in Z}$. The basis functions $\beta^n(x)$ can be generated iteratively by repeated convolution of a B-spline of order 0

$$\beta^n(x)=\beta^0 * \beta^{n-1}(x) \tag{2}$$

where $\beta^0(x)$ is the indicator function in the interval $[-\frac{1}{2},\frac{1}{2})$; i.e.,

$$\beta^0(x)=\begin{cases}1, & x\in[-\frac{1}{2},\frac{1}{2}),\\ 0, & \text{elsewhere.}\end{cases} \tag{3}$$

One of the key properties of B-splines, which follows directly from (2), is their compact support.

Another well known result is that the pth derivative of a spline of order n can be obtained by applying the pth order central difference operator; the result of this calculation is a

U.S. Government work not protected by U.S. copyright.

spline of order n-p [7]. Specifically, the first and second derivatives of B-splines of order n are given by

$$\frac{d\beta^n(x)}{dx} = \beta^{n-1}(x+\tfrac{1}{2}) - \beta^{n-1}(x-\tfrac{1}{2}) \quad (4)$$

$$\frac{d^2\beta^n(x)}{dx^2} = \beta^{n-2}(x+1) - 2\beta^{n-2}(x) + \beta^{n-2}(x-1). \quad (5)$$

2.2 Discrete splines

The discrete B-spline kernels of order n with an expansion factor m. are obtained by sampling the continuous B-splines

$$b_m^n(k) = \beta^n(k/m) \quad \overset{z}{\longleftrightarrow} \quad B_m^n(z). \quad (6)$$

These kernels all have a finite support; they are also characterized by their z-transform $B_m^n(z)$. In this notation, the superscript n refers to the degree of the piecewise polynomials, while the subscript m represents the dilation factor. The z-transforms of the basic B-spline kernels (m=1) for n=0 to 3 are given in Table I. The third and fourth column correspond to the discrete representation of the first and second B-spline derivatives ($d_{(1)}^n(k)$ and $d_{(2)}^n(k)$) obtained by sampling (4) and (5), respectively.

TABLE I
TRANSFER FUNCTIONS OF BASIC B-SPLINE KERNELS AND SAMPLED FIRST AND SECOND DERIVATIVES

n	$B_1^n(z)$	$D_{(1)}^n(z)$	$D_{(2)}^n(z)$
0	1	-	-
1	1	$z-1$	-
2	$\frac{z+6+z^{-1}}{8}$	$\frac{z-z^{-1}}{2}$	$z-2+z^{-1}$
3	$\frac{z+4+z^{-1}}{6}$	$\frac{z-z^{-1}}{2}$	$z-2+z^{-1}$

2.3 Discrete signals and operators

To simplify the presentation, we will consider one-dimensional discrete signals $\{a(k)\}_{k\in Z}$. The operators that will be useful for our purpose are as follows.

• *Convolution* : The discrete convolution between two sequences a and b is denoted by $b*a(k)$. b can be viewed as the impulse response of a filter that is applied to the signal a. This filter is entirely characterized by its z-tranform $B(z)$ (transfer function).

• *Convolution inverse* : If $B(z)$ has no zeros on the unit circle, the inverse operator $(b)^{-1}$ exists and is uniquely defined by

$$(b)^{-1} * b(k) = \delta_0(k) \quad (7)$$

where $\delta_0(k)$ denotes the unit pulse at the origin.

•*Up-sampling by a factor of m* : This operation is required for expanding signals. It is defined by

$$[b]_{\uparrow m}(k) = \begin{cases} b(l), & k = ml \\ 0, & \text{otherwise} \end{cases} \quad \overset{z}{\longleftrightarrow} \quad B(z^m) \quad (8)$$

• *Down-sampling by a factor of m* : This operation is required for decimating signals. It is defined by

$$[b]_{\downarrow m}(k) = b(mk) \quad \overset{z}{\longleftrightarrow} \quad \frac{1}{m}\sum_{q=0}^{m-1} B\left((ze^{j2\pi q})^{1/m}\right). \quad (9)$$

2.4 Extensions for higher dimensional signals

Although all our results will be presented for the one-dimensional case, they are directly applicable to higher-dimensions through the use of tensor product polynomial splines [2]. Practically, this means that the corresponding interpolation and approximation algorithms are separable and can be implemented by successive one-dimensional processing along the various dimensions of the data.

3. INTERPOLATION AND DIFFERENTIATION

3.1 Polynomial spline interpolation

Given a discrete signal $\{g(k)\}$, we would like to determine a polynomial spline $g^n(x)$ that provides an exact interpolation of this sequence. We have shown in [4] that the B-spline coefficients of $g^n(x)$ in (1) can be obtained by inverse filtering

$$c(k) = \left(b_1^n\right)^{-1} * g(k), \quad (10)$$

where $(b_1^n)^{-1}$ denotes the convolution inverse of the basic B-spline kernel b_1^n (cf. Eq. (6)). This filter can be implemented recursively [4]. The corresponding interpolation algorithm is computationally more efficient than classical approaches using matrix algorithms.

3.2 Signal expansion (zooming)

Once the B-spline coefficients have been determined, it is rather straightforward to reconstruct the initial signal at a higher sampling rate. Such a signal expansion is obtained as follows

$$g_m(k) = g^n(x)\Big|_{x=k/m} = b_m^n * [c]_{\uparrow m}(k), \quad (11)$$

where b_m^n is the finite impulse response (FIR) filter defined by (6). This filter can be implemented efficiently from a cascade of moving average filters [4, 6].

3.3 Differentiation

Similarly, we can use the continuous model (1) to differentiate the signal. Specifically, we find that the samples of the first derivative of $g^n(x)$ are given by

$$g'(k) = \frac{dg^n(x)}{dx}\bigg|_{x=k} = d_{(1)}^n * c(k) = \left(d_{(1)}^n * \left(b_1^n\right)^{-1}\right) * g(k), \quad (12)$$

where $d_{(1)}^n$ denotes the sampled derivative of a B-spline of order n (c.f. third column in Table I). The operator in large parenthesis on the right hand side of (12) represents the global impulse response of the polynomial spline differentiator.

4. SMOOTHING SPLINE APPROXIMATION

A regularized version of the previous interpolation problem that is better suited for noisy data is to find the function $\hat{g}(x)$ that minimizes an error criterion with an added smoothness term

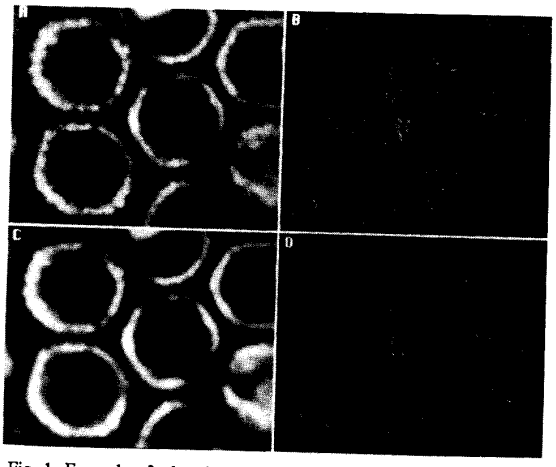

Fig. 1. Example of edge detection. (a) 226x184 region of interest of a digitized electron micrograph of Herpes Simplex Virus type 2 (negatively stained), (b) output of a standard Sobel operator, (c) result of cubic spline smoothing with λ=8 (σ_{eq}=2), (d) gradient magnitude after non-maxima deletion algorithm.

$$\varepsilon_\lambda^2 = \sum_{k=-\infty}^{+\infty} \left(g(k) - \hat{g}(k)\right)^2 + \lambda \int_{-\infty}^{+\infty} \left(\partial^r \hat{g}(x) / \partial x^r\right)^2 dx, \qquad (13)$$

where λ is an adjustable parameter. It has been shown that the solution to this problem is precisely a spline of order $2r$-1 [8]. This type of approximation is called a smoothing spline; it is fully characterized by its B-spline coefficients $c_\lambda(k)$. We have derived a filter-based algorithm that provides an efficient solution to this problem. Our main result is that the B-spline coefficients of the smoothing spline $\hat{g}_\lambda^{2r-1}$ can be obtained by convolution

$$c_\lambda(k) = s_\lambda^{2r-1} * g(k), \qquad (14)$$

where the smoothing spline filter s_λ^{2r-1} is specified in the z-transform domain:

$$s_\lambda^{2r-1}(k) \longleftrightarrow^{z} \frac{1}{B_1^{2r-1}(z) + \lambda(-z + 2 - z^{-1})^r} \qquad (15)$$

This filter is implemented recursively [6].

4.1 Smoothing spline edge detector

An interesting application of this technique is edge detection. Our approach uses the following computational steps:

• *Smoothing spline approximation* : The first step is to determine the B-spline coefficients of a smoothing cubic spline that approximates the digital image. This is achieved by successive filtering along the rows (index k) and columns (index l) according to (14) with n=3. The parameter λ is a scale factor that determines the equivalent window size of the smoothing kernel; i.e., $\sigma_{eq}^2 \cong \sqrt{2\lambda}$. The fast recursive implementation of this filter is discussed in [6].

• *Gradient computation* : The x and y components of the gradient are computed by convolution with the two-dimensional separable kernels $b_1^3(k)d_{(1)}^3(l)$ and $d_{(1)}^3(k)b_1^3(l)$, respectively (cf. Table I). These quantities are used to determine the gradient magnitude and orientation.

• *Non-maxima suppression* : The gradient magnitude map is simplified by retaining only those values that are maxima along a short line segment in direction of the local gradient.

An example of edge detection for a noisy high resolution electron micrograph is shown in Fig. 1. The present algorithm can be shown to be functionally equivalent to Canny's edge detector [9]. However, it is computationally much more efficient.

5. LEAST SQUARES SPLINE APPROXIMATION

An alternative procedure that also provides data compression is to approximate our sequence of data point by a polynomial spline with less coefficients. For this purpose, we expand the B-spline functions and increase their spacing by a factor of m. Specifically, we want to find a polynomial spline of the form

$$g_m^n(x) = \sum_{k=-\infty}^{+\infty} c_m(k)\beta^n(x/m - k) \qquad (16)$$

that provides the best approximation of our signal g(k). The error criterion to be minimized is the residual sum of squares

$$\varepsilon_m^2 = \sum_{k\in Z} \left(g(k) - b_m^n * [c_m]_{\uparrow_m}(k)\right)^2. \qquad (17)$$

By using the fact that the error should be orthogonal to the lower resolution approximation space [10], we find that the optimal coefficients are solution of the equation

$$\left[b_m^n * b_m^n *\right]_{\downarrow_m}(k) = \left[b_m^n * g(k)\right]_{\downarrow_m}(k). \qquad (18)$$

Hence, the coefficients of the least squares approximation $g_m^n(x)$ can obtained by filtering and decimation

$$c_m(k) = \left(\left[b_m^n * b_m^n\right]_{\downarrow_m}\right)^{-1} * \left[b_m^n * g\right]_{\downarrow_m}(k) = \left[\overset{\circ}{b}_m^n * g\right]_{\downarrow_m}(k) \qquad (19)$$

where $[.]_{\downarrow m}$ is the decimation operator defined by (9). This procedure leads to a data reduction by a factor of m.

An application of these technique is the generation of image pyramids which are data structures that have been used for multiresolution image processing and data compression [11].

5.1 Polynomial spline pyramid

A pyramid structure is a sequence of fine-to-coarse signal representations $g_{(0)}, g_{(1)}, g_{(2)}, \ldots, g_{(I)}$ with a size that is reduced by a factor of two from one level to the next; $g_{(0)}$ is the initial signal. We can construct such a pyramid by taking the samples of a sequence of least squares polynomial splines with knot size 2^i, i=1,...I. The decomposition algorithm is implemented by recursive application of a REDUCE operation:

Fig. 2. Error images between consecutive pyramid levels.
(A) : Gaussian/Laplacian pyramid. (B) : Least squares cubic spline pyramid.

$$g_{(i+1)}(k) = \left[\mathring{h}_2^n * g_{(i)} \right]_{\downarrow 2}(k). \tag{20}$$

The dual EXPAND operation expands a coarser image to the next finer level:

$$\hat{g}_{(i)}(k) = h_2^n * \left[g_{(i+1)} \right]_{\uparrow 2}(k) \tag{21}$$

This operation is precisely a polynomial spline interpolation of order n with an expansion factor of two. The general expression for the pre- and post-filters is obtained from (19) and (11) with n=3 and m=2:

$$\begin{cases} \mathring{h}_2^n = \left[b_1^n * \left(\left[b_2^n * b_2^n \right]_{\downarrow 2} \right)^{-1} \right]_{\uparrow 2} * b_2^n, \\ h_2^n = \left[\left(b_1^n \right)^{-1} \right]_{\uparrow 2} * b_2^n. \end{cases} \tag{22}$$

In practice, these operators are either implemented recursively or approximated by FIR filters.

This technique can be readily applied to the generation of a cubic spline image pyramid by successive processing along the rows and columns. The corresponding residual images $g_{(i)}$-$\hat{g}_{(i)}$ for the standard Lena picture and those obtained with the Gaussian/Laplacian pyramid (LP) [11] are shown Fig. 2. Identical intensity scaling factors were applied to all images to facilitate the comparison. These images display the loss of information that occurs through the application of the REDUCE function. For the initial LP, the amount of information lost at each level is quite significant; the initial subject is still readily recognizable. In the case of the cubic spline pyramid, the energy of the difference is reduced drastically. In fact, there is a 8dB improvement for the first resolution level.

Based on those results, it appears that the present technique could be used to improve the performance of the coding scheme described in [11].

CONCLUSION

In this paper, we have considered the use of continuous polynomial spline representations for signal processing applications such as interpolation, differentiation, filtering, noise reduction, and data compression. Polynomial splines can be useful in a variety of problems that are best formulated in a continuous rather than a discrete framework. In this respect, it appears that computational tasks such as differentiation, integration, or the search for extrema are especially simple to perform in the transformed B-spline domain. Some of the most obvious applications are the problem of estimating higher order derivatives from a noisy signal and edge detection in image processing.

Finally, we have shown that the method of least squares spline approximation is well suited to the generation of scale-space or multiresolution signal representations. This concept has been illustrated with the design of a cubic spline pyramid which stands as an interesting alternative to the widely used Gaussian/Laplacian pyramid. Such least squares techniques could be useful in a variety of multi-resolution image processing algorithms.

References

[1] C. de Boor, *A practical guide to splines*. New York: Springer-Verlag, 1978.

[2] P.M. Prenter, *Splines and variational methods*. New York: Wiley, 1975.

[3] H.S. Hou and H.C. Andrews, "Cubic splines for image interpolation and digital filtering", *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. ASSP-26, pp. 508-517, 1978.

[4] M. Unser, A. Aldroubi and M. Eden, "Fast B-spline transforms for continuous image representation and interpolation", *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 277-285, March 1991.

[5] M. Unser, A. Aldroubi and M. Eden, "B-spline signal processing. Part I : theory", *IEEE Trans. Signal Processing*, to appear.

[6] M. Unser, A. Aldroubi and M. Eden, "B-spline signal processing. Part II : efficient design and applications", *IEEE Trans. Signal Processing*, to appear.

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

[8] I.J. Schoenberg, "Spline functions and the problem of graduation", *Proc. Nat. Acad. Sci.*, vol. 52, pp. 947-950, 1964.

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

[10] A. Aldroubi, M. Unser and M. Eden, "Asymptotic properties of least squares spline filters and application to multi-scale decomposition of signals", in Proc. *International Symposium on Information Theory and its Applications*, Waikiki, Hawaii, November 27-30, 1990, pp. 271-274.

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