EFFICIENT IMPLEMENTATION OF ALL-DIGITAL INTERPOLATION*

Bojan Vrcelj and P. P. Vaidyanathan
Contact Author: P. P. Vaidyanathan, Department of Electrical Engineering 136-93,
California Institute of Technology, Pasadena, CA 91125 USA,
Phone: (626) 395-4681 E-mail: ppvnath@systems.caltech.edu

July 19, 2001

EDICS number: 2-INTR

ABSTRACT

B-splines are commonly used for continuous representation of discrete time signals. This kind of representation proves to be very useful in applications such as image interpolation, rotation and edge detection. In all these applications, the first step is to compute the B-spline coefficients of the signal, and this involves the use of an IIR noncausal filter called the direct B-spline filter. The signal reconstruction is achieved using the indirect B-spline filter, which in many applications operates at a higher rate.

In this paper we introduce a simplified implementation of the signal reconstruction part that will significantly reduce the overall complexity. We also show that the direct B-spline filter can safely be replaced by a short FIR filter, without compromising the performance of the traditional method. Numerous examples will show both visually and numerically that the differences between this method and the traditional one are indeed very small. Finally, we report the performance of these newly proposed methods in other image processing applications such as edge detection and least squares approximation.

*Work supported in part by the National Science Foundation under grant MIP 0703755, and by Microsoft research, Redmond, WA.

1 INTRODUCTION

Natural signals in digital signal processing are almost always represented in the form of uniformly spaced samples. In many applications, however, it is necessary to estimate the signal values between those samples. In order to solve for intermediate signal values, it is a common approach to assume some parameterized model for the underlying continuous signal. Polynomial spline functions have a long history of application in this context [1], since they provide a flexible representation (e.g. see [2]), and by simply choosing the order of these approximation functions it is possible to control the smoothness of the resulting continuous signal. Thanks to a classical result by Schoenberg [3], it is enough to consider a very special class of piecewise polynomials, namely B-splines, since all other polynomial splines are obtained as a linear combination of shifted B-splines. The work by Unser *et al.* ([5], [6], [7]) represents a major contribution in B-spline signal processing. They showed that both the representation of discrete signals in terms of B-splines and the reverse process, i.e. interpolative signal reconstruction from its B-spline coefficients can easily be implemented using the discrete filtering. In that implementation the direct B-spline transform is realized by noncausal IIR filtering [5]; the indirect B-spline filter is a symmetric FIR filter, but it usually operates at a higher rate in applications such as signal interpolation and least squares approximation.

In this paper we will address both these filtering operations. We will show that the complexity of the interpolative signal reconstruction (indirect B-spline filtering) can be significantly reduced by employing the polyphase identities [9]. As for the direct B-spline transform, we will show that the IIR filtering is not necessary if the only objective is the visual quality of the interpolated image. Recently, spline interpolation has been considered in the framework of biorthogononal partners [10] as well as sampling theorems for nonbandlimited signals [11]. The authors in [10] show that **exact** spline interpolation is possible using only FIR filters as long as the original signal is oversampled.

In Section II, a short overview of B-splines and their use in signal interpolation is provided. An FIR substitute for the direct B-spline filter is motivated and introduced in Section III. Also, the comparison of interpolation results obtained by the traditional method and this FIR (approximate) method is provided. In Section IV, we introduce the simplified implementation of the indirect B-spline filtering that will not only reduce the complexity of the all-FIR interpolation, but also provide flexibility in design. In Section V, two other B-spline processing applications are considered, namely signal differentiation and least squares approximation. The main conclusion is that these applications are also quite insensitive to the FIR approximations introduced in Section III.

The sets of real numbers, integers and natural numbers are denoted by $\mathbb{R}$, $\mathbb{Z}$, and $\mathbb{N}$, respectively.

The set of square-summable sequences is denoted by l_2. By $||\cdot||_2$ we denote the usual l_2 norm of sequences, defined by the inner products. Subscript c denotes that the signal is continuous-time, while no subscript as in f denotes the discretized version of f_c and is nothing but the restriction of f_c to integers, such that $f(k) = f_c(k), \forall k \in \mathbb{Z}$. Also, as a general remark, t will invariably denote a continuous argument, while k, l, m, and n will denote discrete arguments.

2 B-SPLINES IN SIGNAL REPRESENTATION

In many signal processing applications it is of interest to regard the discrete signal $f(k)$ as samples of a continuous time signal $f_c(t)$. While this kind of representation is not unique, it is especially useful if $f_c(t)$ is a smooth function in the sense that it is differentiable a certain number of times everywhere. Many researchers have proposed the use of spline functions in this context (e.g. see [6] and references therein). Splines of order n are equal to polynomials of degree n on each interval between two knots, and those polynomials are connected in such a way that the overall function is $(n-1)$ times continuously differentiable even at the knots ([2], pp. 3-7). It was shown by Schoenberg [3] that every n-th order spline with equidistant knots, $f_c^n(t)$ can be represented as

$$f_c^n(t) = \sum_{k=-\infty}^{\infty} c(k)\beta^n(t-k), \quad \forall t \in \mathbb{R}. \tag{1}$$

Here $c(k)$ is an l_2 sequence of reals and $\beta^n(t)$ is a centered B-spline of order n, which is obtained as an n-fold convolution of the centered unit pulse with itself[1]

$$\beta^n(t) = \beta^{n-1} * \beta^0(t) = \beta^0 * \beta^0 * \ldots * \beta^0(t),$$

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

Note that B-splines are compactly supported and symmetric around zero. For $n, m \in \mathbb{N}$, the discrete B-spline $b_m^n(k)$ is defined [5], [6] as a sequence of integral samples of the corresponding (n-th order) continuous B-spline, expanded by a factor of m; in other words, $b_m^n(k) = \beta^n(\frac{k}{m})$. Let us denote by $F(z)$, $B_m^n(z)$, and $C(z)$ the z-transforms of $f(k)$, $b_m^n(k)$, and $c(k)$ respectively. If in (1) we fix $f_c^n(k) = f(k), \forall k \in \mathbb{Z}$, which is often a natural choice in signal representation, we can rewrite (1) by restricting t to integers as

$$F(z) = C(z)B_1^n(z). \tag{3}$$

[1]For more detailed discussion of B-splines and their properties, reader is referred to [2] and [6].

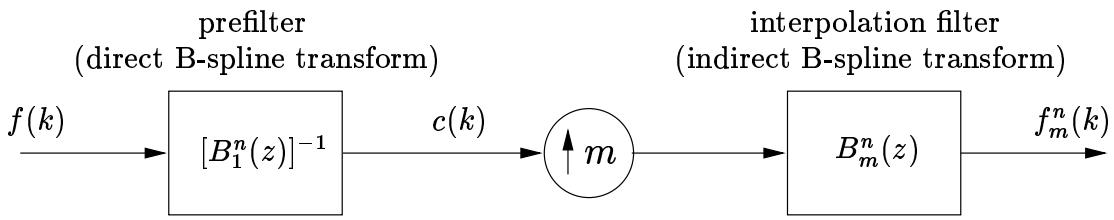

Figure 1: Block diagram of the B-spline signal interpolation.

As shown in [4], all zeros of $B_1^n(z)$ for all $n \in \mathbb{N}$ are real and negative and none of them is equal to -1. Moreover, since it is a symmetric polynomial, if $B_1^n(z)$ vanishes at $z = z_i$, then it will also vanish at $z = z_i^{-1}$. Therefore, by carefully choosing the region of convergence, we can make $[B_1^n(z)]^{-1}$ stable, but noncausal IIR. The poles of that filter for different values of n are tabulated in [7], p. 837, Table I. It is clear from (3) that given the input sequence $f(k)$, its B-spline coefficients $c(k)$ are obtained by performing the "direct B-spline transform," i.e. in the z-domain by multiplying $F(z)$ by $[B_1^n(z)]^{-1}$. Since the latter transfer function is noncausal, a two-pass recursive algorithm is necessary, as described in [5] and [7].

Signal interpolation is now one step away. Having obtained the coefficients $c(k)$ as previously described, we just need to recall that the spline interpolation by an integral factor m of the signal $f(k)$ is given by $f_c^n(\frac{k}{m})$, which we will denote by $f_m^n(k)$. From (1) we now have

$$f_m^n(k) = \sum_{l=-\infty}^{\infty} c(l) b_m^n(k - lm), \tag{4}$$

so that the overall system for interpolation is shown in Fig. 1. The operation of interpolative signal reconstruction in (4) is called the "indirect B-spline transform". The digital filter $B_m^n(z)$ is symmetric FIR. Moreover, it has a special structure as described in [6]. When m and n are not both even, it can be decomposed as

$$B_m^n(z) = \frac{z^\alpha}{m^n} \left(\frac{1 - z^{-m}}{1 - z^{-1}} \right)^{n+1} B_1^n(z) = z^\alpha m [M_m(z)]^{n+1} B_1^n(z), \tag{5}$$

where $\alpha = (m-1)(n+1)/2$, and the filter $M_m(z) = \frac{1}{m}(1 + z^{-1} + \ldots + z^{-(m-1)})$ will be referred to as the running sum. The decomposition as in (5) is beneficial from the point of view of computational complexity for $m \geq 3$. However, note that the digital filter $B_m^n(z)$ operates at a higher rate, because the input signal has m times more samples (in each dimension), and this makes the filtering operation costly.

3 FIR TRUNCATION OF THE DIRECT B-SPLINE FILTER

It has been noticed that the poles of the direct B-spline filter lie far from the unit circle, at least for spline orders that are most commonly used in practice. This causes the impulse response of the noncausal IIR filter $[B_1^n(z)]^{-1}$ to decay very fast. The central part of the impulse response of the cubic direct B-spline filter is shown in Fig. 2. Let us denote by $\gamma_1^3(k)$ the impulse response of $[B_1^3(z)]^{-1}$ and by $h_5(k)$ the five-tap truncation of $\gamma_1^3(k)$; in other words

$$h_5(k) = \begin{cases} \gamma_1^3(k) & \text{for } -2 \le k \le 2 \\ 0 & \text{otherwise} \end{cases}$$

and similarly by $h_7(k)$ the seven-tap truncation of $\gamma_1^3(k)$. Then the relative energies in the cropped parts are given by

$$\frac{||\gamma_1^3(k) - h_5(k)||_2^2}{||\gamma_1^3(k)||_2^2} = 0.00069061,$$

$$\frac{||\gamma_1^3(k) - h_7(k)||_2^2}{||\gamma_1^3(k)||_2^2} = 0.00004958.$$

This implies that even for severe truncations, such as length five or length seven, the resulting projection coefficients $\tilde{c}(k)$ from Fig. 1 will only become slightly perturbed. It is important to note here that the resulting interpolated signal $\tilde{f}_m^n(k)$ is still an oversampled polynomial spline of order n, only the values at its knots will not correspond to the starting signal $f(k)$, but some slightly different signal $\tilde{f}(k)$. The resulting signal $f_m^n(k)$ remains in the desired space, so that all the differentiability properties of $f_m^n(t)$ are preserved. This, of course, holds true for any choice of the prefilter. In particular, it will hold if instead of plain truncation of the impulse response from

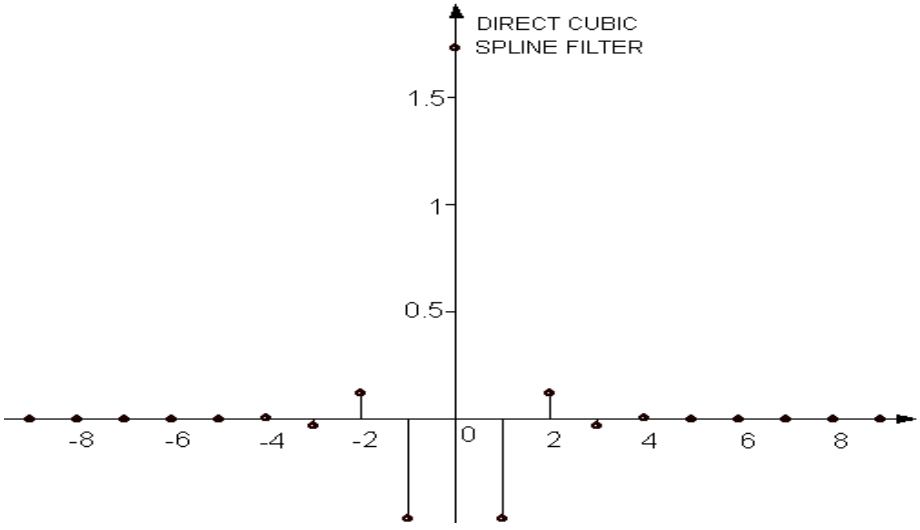

Figure 2: Impulse response $\gamma_1^3(k)$ of the direct cubic B-spline filter $[B_1^3(z)]^{-1}$ for $|k| \le 9$.

Fig. 2 (which corresponds to multiplying it with the rectangular window), some other windowing technique is used. This approach has proved to be beneficial.

In the experiments we used eleven different regions from different images. These testing examples were selected on the basis of diversity in content, edge information and contrast, so that the results would give a comprehensive picture of the comparative performance. These regions (ten of them are 150×150 pixels and one is 200×200 pixels) were the inputs[2] into the interpolation scheme from Fig. 1. Cubic B-splines ($n = 3$) were used, since they appear to be particularly popular among researchers. The interpolation was by a factor $m = 3$. We compared the performance of four different schemes:

1. the noncausal IIR implementation (**NONCIIR**), as in [5] or [7];
2. the implementation where the direct B-spline filter from (1) is replaced by its five-tap truncation (**5TFIR**);
3. the implementation where the direct B-spline filter from (1) is replaced by its seven-tap truncation (**7TFIR**), and
4. the implementation where the direct B-spline filter from (1) is replaced by a five-tap Kaiser windowed truncation (**5KFIR**).

The optimal Kaiser parameter[3] β_{opt} in 5KFIR will be the one that maximizes the PSNR of the approximation $\tilde{f}_m^n(k)$ with respect to the exact cubic spline interpolant $f_m^n(k)$ obtained by NONCIIR, after normalization and storing the result in a byte-format. This value β_{opt} will obviously depend on the signal $f(k)$. Therefore, our approach was to find the optimal values β_{opt} for each of the eleven training examples and then use the weight-averaged value as the universal parameter β_{univ}. The weights were chosen to be proportional to the corresponding PSNR value, so that the β_{opt} from the good training images are favored. The truncated impulse responses were scaled in order for the resulting system to preserve the average value of the input. The PSNR values were computed according to the formula

$$\mathrm{PSNR} = 20\log_{10}\frac{255mN}{||\tilde{f}_m^n(k) - f_m^n(k)||_2},$$

where the size of the input image is $N \times N$. In Table 1, the PSNRs of the images obtained by methods 2, 3 and 4 with respect to the image obtained by method 1 are given. The optimal values of Kaiser parameters for different training images and the corresponding PSNRs are shown in Table

[2]Note that both direct and indirect B-spline transforms are separable, so that on a two-dimensional signal we just need to perform successive one-dimensional filtering along each of the coordinates.

[3]The Kaiser windowing technique (see [12]) contains two parameters: N, which is equal to 5 in this case, and β.

Image	5TFIR	7TFIR	5KFIR
Airplane	44.2	52.2	56.4
Barbara	43.7	52.3	53.3
Boat	43.0	50.9	53.4
Feynman	44.7	52.1	56.5
France	39.6	50.2	44.0
Goldhill	44.7	52.2	52.5
Lena	42.8	50.8	54.4
Mandril	40.1	49.3	47.4
Mountain	37.2	46.3	42.0
Parrots	40.7	48.9	52.9
Peppers	45.5	52.7	53.6

Table 1: Comparing the PSNR values in dB of three different methods with respect to the traditional NONCIIR method; the universal Kaiser parameter $\beta_{univ} = 1.76$.

Image	β_{opt}	PSNR [dB]
Airplane	1.87	56.7
Barbara	1.76	53.3
Boat	1.81	53.5
Feynman	1.91	57.0
France	1.48	44.6
Goldhill	1.71	52.5
Lena	1.85	54.6
Mandril	1.65	47.5
Mountain	1.56	42.5
Parrots	1.87	53.1
Peppers	1.79	53.6

Table 2: The optimal values for the Kaiser parameter and the corresponding PSNRs in the method 5KFIR.

2. These PSNR values are so high that it is hard to see any differences. We demonstrate this point in Figs. 3-6, where the four resulting images for the input image Mountain are shown; note that this was the image with the worst overall performance according to the PSNR values, yet the differences are hardly noticeable. From Table 1, we can see that the error increases in the testing examples where the high frequency content is significant, which is expected. As mentioned in [5], the direct B-spline transform tends to enhance the high frequencies. After truncating its impulse response we are slightly degrading that performance, and hence the error is more noticeable. However, the average human visual system cannot detect the difference. The coefficients of the five-tap filter $H_5(z)$ used in method 4 are given in Table 3. Notice that this filter is symmetric FIR of length 5, which makes the filtering operation very easy to implement. Furthermore, and most importantly, fast all-FIR implementation can be used in **real-time interpolation** of one-dimensional signals, which cannot be achieved with the two-pass recursive algorithm as the one described in [7].

The idea of truncating an IIR impulse response is by no means new to the signal processing community. It has also been proposed in a similar setting [8] in which the goal was to find the inverse of an FIR filter. However, the solution proposed in [8] minimizes the mean-squared error at the output $\tilde{c}(k)$ of the FIR approximation, which, in general, will not yield the best approximation at the output $\tilde{f}_m^n(k)$ of the interpolation system of Fig. 1. Moreover, it is a simple exercise to show that the best FIR approximation in this context (minimizing the mean-squared error of the output $\tilde{f}_m^n(k)$) is signal-dependent. Therefore, the Kaiser-windowed filter from Table 3 should be

Figure 3: Cubic spline interpolation of Mountain image using the NONCIIR method.

Figure 4: Approximate cubic spline interpolation of Mountain image using the 5TFIR method.

Figure 5: Approximate cubic spline interpolation of Mountain image using the 7TFIR method.

Figure 6: Approximate cubic spline interpolation of Mountain image using the 5KFIR method, $\beta = 1.76$.

0.06049527	-0.37739071	1.63379087	-0.37739071	0.06049527

Table 3: Coefficients of the five-tap Kaiser windowed approximation $H_5(z)$ of the direct B-spline filter $[B_1^n(z)]^{-1}$.

viewed just as one possible solution that yields good results on a variety of examples. We compared the performance of this 5KFIR method to the least-squares five-tap approximation as in [8] and found the 5KFIR to be superior in all but the three images with significant high frequency content (France, Mandril and Mountain).

4 POLYPHASE-BASED SIMPLIFICATION OF THE INDIRECT B-SPLINE FILTER

The digital filter $B_m^n(z)$ appearing at the final stage of the interpolation process showed in Fig. 1 is a symmetric FIR filter of length[4] $(n+1)m-1$. However, it operates on the signal that is dense with zeros (from the expander by m) and therefore it is to be expected that the total number of operations per input pixel could be reduced. The implementation of the indirect B-spline filter proposed in this section will achieve this as well as allow for some flexibility in design.

The filter $B_m^n(z)$ can be written in a Type 2 polyphase form [9] as

$$B_m^n(z) = \sum_{l=0}^{m-1} z^l R_l(z^m), \tag{6}$$

where $R_l(z)$, for $0 \le l \le m-1$ is the l-th polyphase component of $B_m^n(z)$. Notice that (according to our notation) for $l > 0$, the length of $R_l(z)$ is $n+1$, and that $R_0(z) = B_1^n(z)$. Now, we can move the expander from Fig. 1 past those polyphase components and get the scheme shown in Fig. 7, where the switch is in the position CUBIC. This scheme is completely analogous to the one in Fig. 1 when we substitute the direct B-spline filter by its five-tap truncation. The performance of this scheme was reported in Section III. However, the scheme of Fig. 7 has the advantage that all the polyphase component filters operate at the lower rate than the composite interpolation filter $B_m^n(z)$. Furthermore, in the ideal case when instead of $H_5(z)$ in Fig. 7, there is $[B_1^n(z)]^{-1}$, we can see that $R_0(z)$ and $[B_1^n(z)]^{-1}$ cancel out, which results in the scheme with the switch in the position REVER (reversible). The advantage of this scheme over the previous one is that the complexity is even lower and the interpolation becomes completely **reversible**, in the sense that $[\tilde{f}_m^n(k)]_{\downarrow m} = f(k)$. The only price to pay in this case is that now the signal $\tilde{f}_m^n(k)$ does not represent exactly the cubic

[4]For simplicity of presentation, in the following we assume that the order of spline n is odd. Similar analysis can be performed for n even, and all the main results continue to hold.

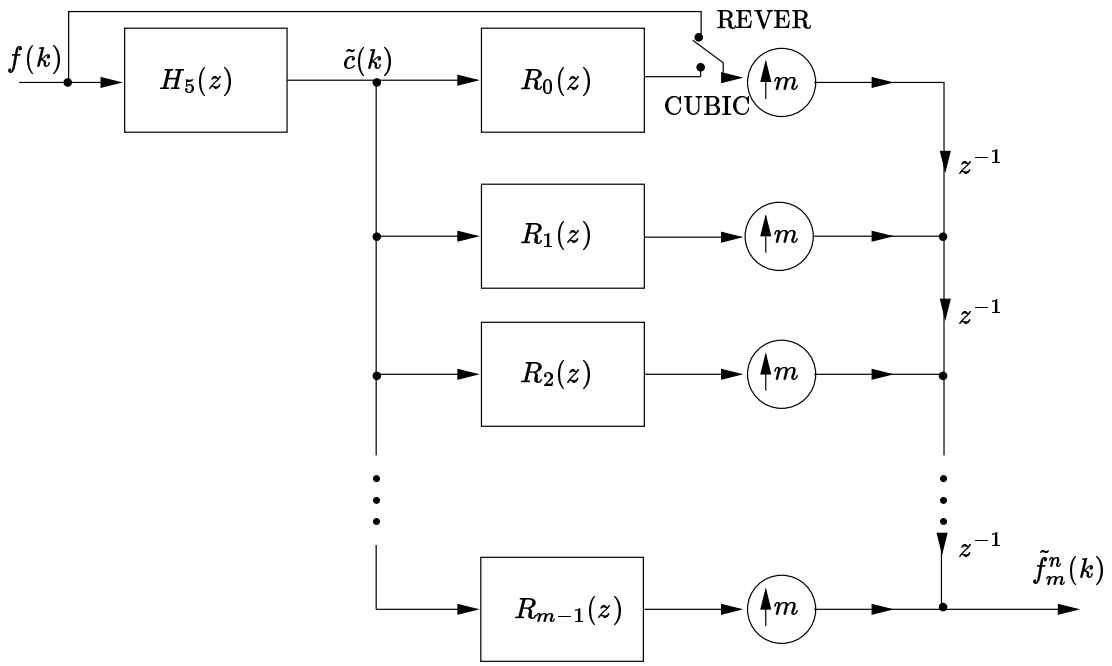

Figure 7: Modified scheme for interpolation, simplified using the polyphase decomposition.

spline interpolant of some perturbed signal $\tilde{f}(k)$, but is modified at the knots of that interpolant in order to agree with the exact signal $f(k)$.

Therefore, the scheme in Fig. 7 allows for a choice between two desirable properties of signal interpolation. With the switch in the position CUBIC the resulting signal is an oversampled piecewise polynomial with all the desirable smoothness properties, but the interpolation is not reversible. However, from the PSNR values presented in Tables 1 and 2, one can easily conclude that the errors introduced in the original signal positions are quite small. On the other hand, when the switch is in the position REVER the interpolation becomes reversible, but the output samples are not obtained by oversampling a spline. Considering the specific application, the designer can choose between these two implementations. It is important to note that the average value of the input is preserved by both implementations. This first order approximation is achieved by scaling the truncated impulse response of the direct spline filter $H_5(z)$ as mentioned in Section III.

Repeating the same experiments as shown in Tables 1 and 2, but with the scheme from Fig. 7, switch in position REVER, we found that the PSNR values improved by 1 to 2 dB, so that for example the worst PSNR for the five-tap Kaiser windowed truncation, using $\beta_{univ} = 1.77$ was 44.0 dB (Mountain image).

Now we give a quantitative estimate of the reduction in complexity of the indirect B-spline

transform alone. Consider the case of interpolating a square image (dimensions $N \times N$) by a factor m, using the n-th order spline. We will compare the complexity of three different methods:

1. just FIR filtering with $B_m^n(z)$ (we call this complexity $\mathbf{C_{FIR}}$);
2. decomposing $B_m^n(z)$ as in (5): $B_m^n(z) = z^{\alpha} m[M_m(z)]^{n+1} B_1^n(z)$ (complexity $\mathbf{C_{decom}}$), and
3. polyphase representation as in Fig. 7, with the switch in position REVER (complexity $\mathbf{C_{poly}}$).

The filter $B_m^n(z)$ is symmetric and of length $(n+1)m-1$, which implies $m(n+1)/2$ multiplications and $m(n+1)-2$ additions per input sample, per dimension. The size of the input image is $(Nm)^2$, so that in terms of only multiplications and additions:

$$\mathrm{C_{FIR}} = N^2 m^3 (n+1)\ [MUL] + 2N^2 m^2 [m(n+1)-2]\ [ADD].$$

If we use the decomposition given by method 2, we should also notice that $mM_m(z)$ applied to the upsampled image corresponds just to repeating the pixel values, and therefore, we should count the complexity of $B_1^n(z)$ plus that of n running sums only. The complexity of $B_1^n(z)$ is $(n-1)$ additions and $(n+1)/2$ multiplications per input sample, per dimension. As for the running sums $M_m(z)$, each of those demands two additions and one multiplication per input sample, per dimension. Thus the complexity of this filtering becomes:

$$\mathrm{C_{decom}} = N^2 m^2 (3n+1)\ [MUL] + N^2 m^2 (6n-2)\ [ADD],$$

which is more efficient than $\mathrm{C_{FIR}}$ for larger values of m. In the polyphase realization from Fig. 7, we have $(m-1)$ FIR filters $R_l(z)$, $1 \le l \le m-1$, each of length $(n+1)$. These require $(n+1)$ multiplications and n additions per input sample in each of the two stages (one stage per dimension). The size of the input signal to the first stage is N^2 and to the second stage is $N^2 m$. However, in the second stage, the polyphase component $R_0(z)$ needs to operate on a fraction $(m-1)/m$ of the input pixels, so that the complexity of the implementation 3 becomes:

$$\mathrm{C_{poly}} = N^2 (m^2+m-2)(n+1)\ [MUL] + N^2 (m^2+m-2) n\ [ADD],$$

and this is clearly superior to the two previous methods. Moreover, note that many operations in the third method are performed in parallel, so that the filters can operate at the lower rate. Computational complexities of the three methods are compared in Table 4. In this example, $n = 3$, $m = 4$, and the size of the input image is $N \times N$.

Operation	Method 1	Method 2	Method 3
MUL	$256N^2$	$160N^2$	$72N^2$
ADD	$448N^2$	$256N^2$	$54N^2$

Table 4: Computational complexities of the indirect B-spline transform for $n = 3$ and $m = 4$, implemented using three different methods.

n	N_{tap}	β_{univ}	average PSNR [dB]
3	5	0	42.4
5	7	2.93	44.8
7	9	3.10	43.4

Table 5: The Kaiser parameters β_{univ} and the corresponding PSNR values achievable for different spline orders n and number of taps N_{tap} in the FIR approximation of $[B_1^n(z)]^{-1}$.

5 OTHER APPLICATIONS AND REMARKS

5.1 Higher Order Spline Interpolation

In Section III we showed that noncausal IIR filters $[B_1^n(z)]^{-1}$ can in most applications be replaced by relatively short FIR filters. We demonstrated our point on the example of cubic spline interpolation. Sometimes it is desirable to use higher order splines. In the method proposed by Unser *et al.* [7], the complexity of the direct B-spline filtering grows linearly with the spline order, since they obtain the higher order solution by cascading the noncausal IIR blocks. However, the poles of the higher order B-spline filters are still far from the unit circle. For example, the 7-th order direct B-spline filter has three poles inside the unit circle (the other three are reciprocals of these) and all of them have magnitudes less than 0.54. That is why the same method of IIR truncation still yields good results, and even at **sublinear growth** with respect to the filter length. We performed experiments on the same set of eleven images mentioned earlier.[5] Our goal was to find the length of the Kaiser window (with some universally selected parameter β_{univ}), such that the resulting FIR approximation of the filter $[B_1^n(z)]^{-1}$ would produce PSNRs similar to those obtained by the 5TFIR method, for $n = 3$ (which still produced visually good results). The conclusion is that for $n = 5$ it is enough to use a seven-tap FIR, and for $n = 7$ it suffices to use a 9-tap FIR. The values of the universal Kaiser parameters and the obtained average PSNRs on the set of testing images are shown in Table 5. The first row in that table corresponds to the 5TFIR method, and thus $\beta_{univ} = 0$ in this case.

[5] All the images used for testing, as well as the Matlab code are available on [15].

5.2 Differentiation

Besides signal interpolation, B-splines find their application in other signal processing techniques, such as edge detection and noise reduction (e.g. see [6]). The reason for this is that many operations performed in the digital domain are nothing but the discretized continuous domain operations. Differentiation is one such example. While well-defined for continuous signals, it is an ill-posed problem for digital signals. That is why some discrete approximations, such as the finite difference operator, are used instead. However, by utilizing the spline model for the continuous underlying signal, it is possible to find a very elegant solution for signal differentiation (as shown by Unser *et al.* [6]), and implement it completely in the digital domain. The first step in this implementation is once more the direct B-spline filtering. The same approach of Kaiser windowing the direct B-spline impulse response (Kaiser window of length 5 with $\beta_{univ} = 1.76$ for $n = 3$) was tested in this application, and the result is satisfactory. The PSNR values with respect to the exact gradient-based method using IIR filter $[B_1^n(z)]^{-1}$ (cf. [7], Sec. IV-A) were ranging from 33.7 dB (Airplane image) to 22.6 dB (Mountain image). Since the output of the edge detector is a binary image, a fair way to express those results would be to specify the percentage of the pixels in error. The edge information in the Airplane image differed in 0.04 percent of the pixels, whereas in the Mountain image it differed in 0.55 percent of the pixels. In Fig. 8 and Fig. 9, the visual comparison between the exact edge detector and this all-FIR approximation is provided for the Lena image. Both images are displayed at the lower resolution for easier comparison. The differences are hardly noticeable (and the PSNR between the two images is 27.0 dB).

Figure 8: Edge detection using the first derivative and cubic spline model; exact implementation based on NONCIIR method.

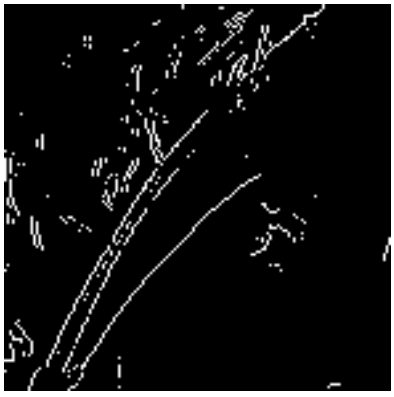

Figure 9: Edge detection using the first derivative and cubic spline model; approximation based on 5KFIR method, $\beta = 1.76$.

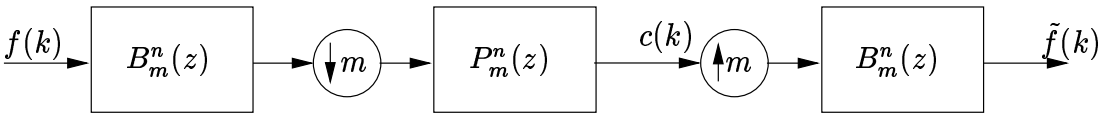

Figure 10: Block diagram of the least squares spline filtering.

5.3 Least Squares Spline Approximation

Least squares splines are sometimes used for noise reduction. The idea is to model the unknown non-corrupted signal as a spline with fewer degrees of freedom, which actually becomes an oversampled spline. The oversampling parameter m determines the degree of smoothness. Although this model is not accurate for most natural signals, the expectation is that having fewer degrees of freedom will impose some smoothness constraints on the resulting signal, and thus eliminate the noise. The reconstruction part is, therefore, identical to the one shown in Fig. 1, and is presented in the right half of Fig. 10. The projection coefficients $c(k)$ are obtained by **projection prefiltering**. This prefiltering part is such that the resulting signal $\tilde{f}(k)$, which belongs to the space of n-th order splines oversampled by m is closest to the noisy signal $f(k)$ in the l_2 norm. The prefiltering operation is unique [10] and the complete projection scheme is demonstrated in Fig. 10. Here, the filter $P_m^n(z)$ is given by:

$$P_m^n(z) = ([(B_m^n(z))^2]_{\downarrow m})^{-1}. \tag{7}$$

This filter is again a noncausal IIR filter. Its properties are briefly mentioned in [7] and the main result is that as m grows larger, the poles of $P_m^n(z)$ get closer to those of $[B_1^{2n+1}(z)]^{-1}$, and even for $m = 3$ the corresponding poles are quite close (the difference is within 0.5 percent). As a result, this IIR filter can safely be replaced by the Kaiser windowed truncation of the corresponding impulse response, which is again symmetric. This assertion was verified by the experiment on the same set of images, with $m = 3$ and $n = 3$, so that the poles of $P_3^3(z)$ were quite close to those of $[B_1^7(z)]^{-1}$, and therefore, we chose the same value for the Kaiser parameter: $\beta_{univ} = 3.10$ and the window of length 9, as suggested by Table 5. The average PSNR with respect to the IIR method was 41.2 dB, which still qualifies as the good approximation with hardly noticeable differences. It is worth mentioning that the complexity of this method can further be reduced by deploying the same polyphase decomposition trick on both $B_m^n(z)$ filters (as described in Section IV), and moving the decimator/expander past those filters, thus lowering the operating rate.

6 CONCLUSION

There are several existing interpolation techniques well-known in computer graphics, that do not require prefiltering and that can be performed using only FIR-type operations. The method of quasi-interpolation [13] reproduces polynomials up to degree n at the expense of changing the original signal values (being nonreversible). As opposed to that, various subdivision schemes [14] (e.g. cubic Lagrange subdivision) achieve reversible interpolation, but sacrifice the approximation order.

In this work we propose a simplified, all-FIR implementation of signal interpolation based on piecewise polynomial models, the motivation being that the original cardinal spline interpolation method [5] provides good approximation order while being reversible. However, in the original method this is achieved with noncausal IIR filtering. In our approximate method, the interpolative signal reconstruction (indirect B-spline filtering) is realized efficiently using the polyphase decomposition. The proposed structure allows for an additional flexibility: the designer can choose between a nonreversible spline interpolation and a lower complexity reversible interpolation. As for the direct B-spline transform, it is shown that noncausal IIR filtering can be replaced by short FIR filtering without compromising the visual quality of the interpolated image. These findings are extended to two other B-spline processing applications, namely signal differentiation and least squares approximation.

References

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

[2] L. L. Schumaker, *Spline Functions: Basic Theory*. New York: Wiley, 1981.

[3] I. J. Schoenberg, *Cardinal Spline Interpolation*. SIAM, 1973.

[4] I. J. Schoenberg, "Cardinal interpolation and spline functions," *J. Approximation Theory*, vol. 2, pp. 167-206, 1969.

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

[6] M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing: Part I - Theory," *IEEE Trans. Signal Processing*, vol. 41, pp. 821-833, Feb. 1993.

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

[8] M. Unser and M. Eden, "FIR approximations of inverse filters and perfect reconstruction filter banks," *Signal Processing*, vol. 36, pp. 163-174, 1994.

[9] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Prentice-Hall, Englewood Cliffs, NJ, 1995.

[10] P. P. Vaidyanathan and B. Vrcelj, "Biorthogonal partners and applications," *IEEE Trans. Signal Processing*, vol. 49(5), pp. 1013-1027, May 2001.

[11] P. P. Vaidyanathan, "Generalizations of the sampling theorem: Seven decades after Nyquist," to appear in *IEEE Trans. Circuits and Systems*.

[12] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1989.

[13] M. Unser and I. Daubechies, "On the approximation power of convolution-based least squares versus interpolation," *IEEE Trans. Signal Processing*, vol. 45(7), pp. 1697-1711, Jul. 1997.

[14] I. Daubechies, I. Guskov, P. Schröder and W. Sweldens, "Wavelets on irregular point sets," *Phil. Trans. Royal Soc. Lond. A*, no 357(1760), pp. 2397-2413, Sep. 1999.

[15] `http://www.systems.caltech.edu/bojan/splines/mar00.html`