Current Development in Theory and Applications of Wavelets
Volume 4, Number 3, 2010, Pages 289-305
Published Online: January 26, 2011
This paper is available online at http://pphmj.com/journals/cdtaw.htm

SHIFTED B-SPLINE INTERPOLATION FILTER

H. OLKKONEN and J. T. OLKKONEN

Department of Physics and Mathematics
University of Eastern Finland
B. O. Box 1627, 70211 Kuopio, Finland
e-mail: hannu.olkkonen@uef.fi

VTT Technical Research Centre of Finland
02044 VTT, Finland
e-mail: juuso.olkkonen@vtt.fi

Abstract

B-spline signal processing is an important tool in signal processing society. B-splines can be adapted in interpolation and approximation of underlying continuous signals and systems, adjustment of fractional time delays in communication devices and in numerical signal analysis. In this work, we introduce the shifted B-spline interpolation algorithm, where the B-spline kernel is shifted by a fraction Δ of the sampling period. Experimental tests show that the interpolation error is strongly dependent on Δ and attains a minimum at $\Delta = 0.25$. We describe the z-transform filter for efficient parallel implementation of the shifted B-spline interpolation algorithm. The performance of the Δ-shifted B-spine interpolation filter is significantly higher compared with the cubic convolution interpolation and the standard B-spline interpolation.

2010 Mathematics Subject Classification: 65Z05.

Keywords and phrases: B-splines, interpolation, signal approximation, fractional delay filters, numerical analysis, adaptive systems, microprocessors, VLSI.

This work was supported by the National Technology Agency of Finland (TEKES).

Received September 22, 2010

I. Introduction

Many of the underlying signal models and systems are continuous. However, microchips handle only discrete data sequences, which raises an apparent need for the fast interpolation algorithms viable in microprocessor and VLSI environments. Based on the Shannon's sampling theorem the sinc-interpolation is usually employed as an ideal interpolation filter, which has the impulse response

$$h[n] = \frac{\sin \pi(n - \tau)}{\pi(n - \tau)} = \mathrm{sinc}(n - \tau). \tag{1}$$

However, infinitely long impulse response of the sinc-interpolation requires time-consuming computations, which limits its usability in many applications. On the other hand, the truncated sinc impulse response is different for every delay τ, which requires a coefficient look-up table.

A class of interpolation methods is based on the fractional delay (FD) filters, which are important tools in many areas of modern communication systems. The FD filters produce a delay which is a fraction of the sampling period. A comprehensive tutorial review [1] summaries the methods for construction of the FD filters. The most promising approaches are based on the allpass Thiran filters, albeit many other techniques have been described [2-6].

The B-spline signal processing has recently attained a central role in signal approximation and interpolation theory. Due to the work of Unser et al. [7-9] the fast B-spline filtering algorithms form standard tools in image and signal analysis [10-13]. The B-spline approximation of the continuous signal has an asymptotic equivalence with the Shannon's sampling theorem [14]. The generalized interpolation theory with B-splines is summarized in [15] and compared with other methods.

Plonka [16, 17] has studied the effect of the shift of the interpolation nodes on the approximation error in the B-spline interpolation. According to the theoretical considerations the shift $\tau = 0$ yielded the minimal error norm. Blu et al. [18] applied the same idea to the piecewise-linear interpolation. On the contrary, they observed that interpolation error diminishes if the sampling knots are shifted by a fixed amount. The theoretical analysis of Blu et al. [18] predicted that the optimal shift of the knots is close to 1/5 of the distance between the knots.

In this work, we introduce the shifted B-spline interpolation algorithm, where the B-spline kernel is shifted by a fraction Δ of the sampling period. We describe the *z*-transform filter for efficient parallel implementation of the shifted B-spline interpolation algorithm. We show that the performance of the Δ-shifted B-spline interpolation filter is significantly higher compared with the cubic convolution interpolation [19] and the standard B-spline interpolation.

II. Theoretical Considerations

A. B-spline interpolation

Let us consider the underlying continuous-time signal $x(t)$, which is sampled at discrete time $t = nT$ $(n = 0, 1, 2, \ldots)$ intervals. For clarity the sampling interval T is normalized as $T = 1$. The B-spline interpolation of the signal $x(t)$ is based on the summation [20]

$$x(t) = \sum_{k} c[k]\beta_p(t - k), \tag{2}$$

where $c[k]$ is the scale coefficient sequence. B-splines $\beta_p(t)$ are defined as p-times convolution of a rectangular pulse $p(t)$:

$$\beta_p(t) = \underbrace{p(t) * p(t) * \cdots p(t)}_{p\text{ times}} \quad p(t) = \begin{cases} 1, & \text{for } 0 \le t \le 1, \\ 0, & \text{elsewhere.} \end{cases} \tag{3}$$

The Laplace transform of the B-spline comes from

$$L\{p(t)\} = \frac{1}{s}(1 - e^{-s}) \Rightarrow \beta_p(s) = \frac{1}{s^p}(1 - e^{-s})^p = \sum_{k=0}^{p} \binom{p}{k} (-1)^k \frac{e^{-ks}}{s^p} \tag{4}$$

and the inverse Laplace transform yields

$$\beta_p(t) = \frac{1}{(p-1)!} \sum_{k=0}^{p} \binom{p}{k} (-1)^k (t - k)^{p-1} u(t - k), \tag{5}$$

where the step function $u(x) = x$ for $x \ge 0$ and $u(x) = 0$ for $x < 0$.

The B-spline interpolation (2) differs from the classic interpolation in that it approximates the signal by scaled and delayed B-splines, whereas in classic interpolation the original discrete data sequence is convolved with the interpolation kernel.

Table I

p	$N_p(z)$	$\beta_p(z)$
1	$\dfrac{1}{1-z^{-1}}$	1
2	$\dfrac{z^{-1}}{(1-z^{-1})^2}$	z^{-1}
3	$\dfrac{z^{-1}+z^{-2}}{2(1-z^{-1})^3}$	$\dfrac{z^{-1}+z^{-2}}{2}$
4	$\dfrac{z^{-1}+4z^{-2}+z^{-3}}{6(1-z^{-1})^4}$	$\dfrac{z^{-1}+4z^{-2}+z^{-3}}{6}$
5	$\dfrac{z^{-1}+11z^{-2}+11z^{-3}+z^{-4}}{24(1-z^{-1})^4}$	$\dfrac{z^{-1}+11z^{-2}+11z^{-3}+z^{-4}}{24}$
6	$\dfrac{z^{-1}+26z^{-2}+66z^{-3}+26z^{-4}+z^{-5}}{120(1-z^{-1})^4}$	$\dfrac{z^{-1}+26z^{-2}+66z^{-3}+26z^{-4}+z^{-5}}{120}$

B. Discrete B-splines

The discrete B-spline $\beta_p[n]$ equals to the continuous B-spline at integer values of time. Hence, the Laplace transform (4) and the z-transform of the discrete B-spline have inverse transforms which coincide at integer values in the time domain. Using the relation

$$L^{-1}\left(\frac{1}{s^p}\right) = \frac{t^{p-1}u(t)}{(p-1)!} \tag{6}$$

we obtain the z-transform of the discrete B-spline

$$\beta_p(z) = Z\{\beta_p[n]\} = Z\left\{L^{-1}\left(\frac{1}{s^p}(1-e^{-s})^p\right)\bigg|_{t=n}\right\} = N_p(z)(1-z^{-1})^p, \tag{7}$$

where

$$N_p(z) = Z\left\{L^{-1}\left(\frac{1}{s^p}\right)\bigg|_{t=n}\right\} = \sum_{n=0}^{\infty} \frac{n^{p-1}}{(p-1)!} z^{-n}. \tag{8}$$

We have $N_1(z) = 1/(1-z^{-1})$. By differentiating in respect to z we obtain a recursion

$$N_{p+1}(z) = \frac{-z}{p}\frac{dN_p(z)}{dz} \tag{9}$$

Table I gives the $N_p(z)$ and the discrete B-spline $\beta_p(z)$ for $p = 1$ to 6.

C. Shifted B-spline interpolation filter

By introducing a fractional time-shift $\Delta \in [0, 1]$ we define the shifted B-spline interpolation as

$$x(t) = \sum_k c[\Delta, k]\beta_p(t + \Delta - k), \tag{10}$$

where $c[\Delta, k]$ is the modified scale coefficient sequence. At $t = n$ the z-transform of (10) gives

$$Z\{x[n]\} = X(z) = C(\Delta, z)Z\{\beta_p[n + \Delta]\} = C(\Delta, z)\beta_p(\Delta, z), \tag{11}$$

where $\beta_p(\Delta, z)$ denotes the Δ-shifted discrete B-spline. The interpolated signal is then obtained as

$$Z\{x[n + d]\} = C(\Delta, z)Z\{\beta_p[n + \Delta + d]\} = C(\Delta, z)\beta_p(\Delta + d, z), \tag{12}$$

where the fractional time-shift $d \in [0, 1]$. By solving the scale sequence $C(\Delta, z)$ from (11) we get

$$Z\{x[n + d]\} = \beta_p^{-1}(\Delta, z)\beta_p(\Delta + d, z)X(z). \tag{13}$$

Now we may define the interpolated signal as

$$Z\{x[n + d]\} = G(\Delta, d, z)X(z), \tag{14}$$

where $G(\Delta, d, z)$ is the shifted B-spline interpolation filter

$$G(\Delta, d, z) = \beta_p^{-1}(\Delta, z)\beta_p(\Delta + d, z). \tag{15}$$

D. Shifted B-spline filter

In the following we deduce the z-transform for the shifted B-spline. By denoting $\varepsilon = \Delta + d$ and following (7) we have

$$\beta_p(\varepsilon, z) = Z\left\{ L^{-1}\left(\frac{e^{s\varepsilon}}{s^p}(1-e^{-s})^p \right)\Bigg|_{t=n} \right\} = \sum_{n=0}^{\infty} \frac{(n+\varepsilon)^{p-1} u(n+\varepsilon)}{(p-1)!} z^{-n}(1-z^{-1})^p. \quad (16)$$

Due to binomial series we have

$$(n+\varepsilon)^{p-1} = \sum_{k=0}^{p-1} \binom{p-1}{k} n^{p-1-k} \varepsilon^k. \quad (17)$$

By inserting (17) to (16) we get

$$\beta_p(\varepsilon, z) = \sum_{k=0}^{p-1} \frac{\varepsilon^k}{k!} \sum_{n=0}^{\infty} \frac{n^{p-k-1}}{(p-k-1)!} z^{-n}(1-z^{-1})^p. \quad (18)$$

Applying (7) and (8) we have

$$\beta_p(\varepsilon, z) = \sum_{k=0}^{p-1} \frac{\varepsilon^k}{k!}(1-z^{-1})^k \beta_{p-k}(z). \quad (19)$$

The ε-shifted B-spline filter (19) has the FIR structure. It is essential to point out that the ε-shifted B-spline filter (19) is defined only for positive shift $\varepsilon \geq 0$. This is due to the relation (6), which is valid only for $t \geq 0$. Correspondingly, the binomial series expansion (17) is valid only for $n \geq 0$ and $\varepsilon \geq 0$.

E. Implementation of the shifted B-spline interpolation filter

Due to (19) we obtain the shifted B-spline interpolation filter (15) as

$$G(\Delta, d, z) = \beta_p^{-1}(\Delta, z) \sum_{k=0}^{p-1} \frac{(\Delta+d)^k}{k!}(1-z^{-1})^k \beta_{p-k}(z). \quad (20)$$

It can be shown that for any $\Delta \in [0, 1]$ $G(\Delta, 0, z) = 1$ and $G(\Delta, 1, z) = z$. The inverse shifted B-spline filter $\beta_p^{-1}(\Delta, z)$ in (20) can be computed by the procedure described in Appendix I. The shifted B-spline interpolation filter (20) can be written in parallel realization

$$G(\Delta, d, z) = \beta_p^{-1}(\Delta, z) \sum_{k=0}^{p-1} (\Delta+d)^k F_k(z), \quad (21)$$

where $F_k(z) = (1 - z^{-1})^k \beta_{p-k}(z)/k!$ when the signal is interpolated using different values of $d \in [0, 1]$ the signal can be first prefiltered by $\beta_p^{-1}(\Delta, z)$ and then filtered by the p parallel $F_k(z)$ filters. The interpolated signal is then the weighted sum of the outputs $y_k[n]$ of the $F_k(z)$ filters. In time domain we obtain the interpolated signal as

$$x[n + d] = \sum_{k=0}^{p-1} (\Delta + d)^k y_k[n]. \tag{22}$$

The parallel realization (22) enables feasible implementation of the shifted B-spline interpolation filter in microprocessor and VLSI environments. In addition to the basic interpolation and fractional shift operations the time domain relation (22) allows elaborate numerical processing tasks such as differentiation, integration, computation of the statistical functions and estimation of fractional time delays in communication systems. This is based on the notation that the fractional time shift $d \in [0, 1]$ in (22) can be considered as continuous time variable and e.g. the derivative of signal can be computed from

$$\frac{d}{dt} x[n + d] = \sum_{k=0}^{p-1} k(\Delta + d)^{k-1} y_k[n]. \tag{23}$$

The proof is given in Appendix II.

Experimental Results

A. Interpolation error versus Δ-shift

An extensive series of tests of the effect of the Δ-shift on the interpolation error was carried out using different synthetic continuous functions with varying frequency content. The B-spline order p varied between $p = 3 - 6$. The interpolation error was defined as the absolute difference of the signal and the interpolated value: $e(t) = |x(t) - x_{\text{int}}(t)|$. When the interpolation error was computed as a function of the Δ-shift, two clear local minima were observed: $\Delta = 0$ and $\Delta = 0.25$. In every test the value $\Delta = 0.25$ yielded the minimum interpolation error. A typical test is given in Figure 1, where the interpolation error is given for $\Delta \in [0.21, 0.29]$ and

$d = 0.5$ for the sinusoidal signal $x[n] = \sin(0.1n)$ and the B-spline order $p = 4$. The interpolation error attains the minimum at $\Delta = 0.25$ (Figure 1). In the sequel we use $\Delta = 0.25$ as a default value, when we consider the properties of the B-spline interpolation filter.

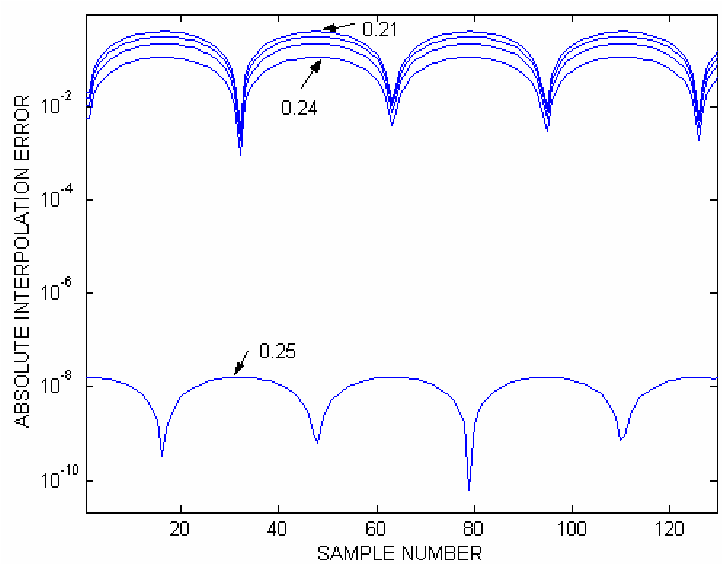

Figure 1. The interpolation error for the sinusoidal signal $x[n] = \sin(0.1n)$ and the B-spline order $p = 4$ at $\Delta = 0.21(0.29)$, $0.22(0.28)$, ..., 0.25 and $d = 0.5$.

B. Frequency response of the shifted B-spline interpolation filter

The frequency response of the shifted B-spline interpolation filter was obtained by inserting $z = \exp(j\omega)$ in (20). Figure 2 shows the magnitude and phase responses of the B-spline $(p = 4)$ interpolation filter for $d \in [0.21, 0.25]$ and $\Delta = 0.25$. For $d < 0.5$ the magnitude response behaves like a low-pass filter. At $d = 0.5$ the frequency response is constant. The phase response is highly linear in the frequency range $0 \leq \omega \leq 0.7\pi$. Figure 3 shows the magnitude and phase responses of the B-spline $(p = 4)$ interpolation filter for $d \in [0.6, 0.9]$ and $\Delta = 0.25$. For $d > 0.5$ the frequency response increases slightly from $\omega = 0.4\pi$ towards $\omega = \pi$. The magnitude and phase responses approach more ideal, when the B-spline order increases. This, however, leads to the longer filters and it is usually more effective to increase the sampling rate to obtain the required precision.

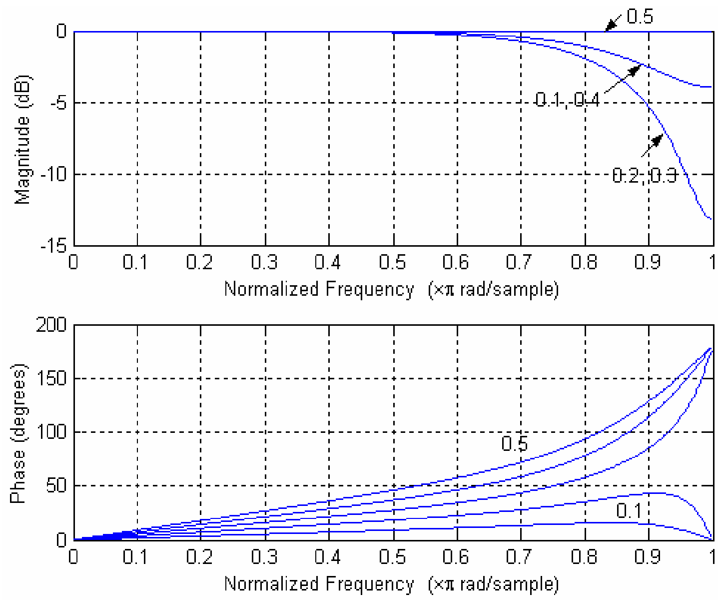

Figure 2. Magnitude and phase responses of the shifted B-spline $(p = 4)$ interpolation filter for $d \in [0.1, 0.5]$ and $\Delta = 0.25$.

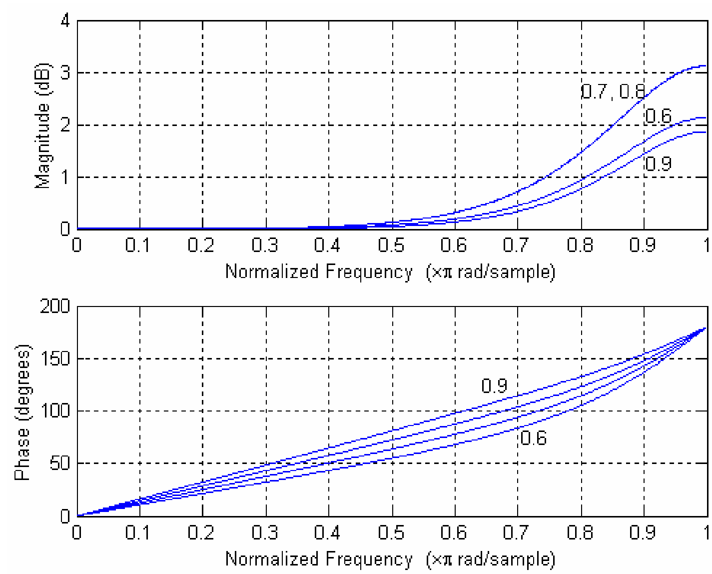

Figure 3. Magnitude and phase responses of the shifted B-spline $(p = 4)$ interpolation filter for $d \in [0.6, 0.9]$ and $\Delta = 0.25$.

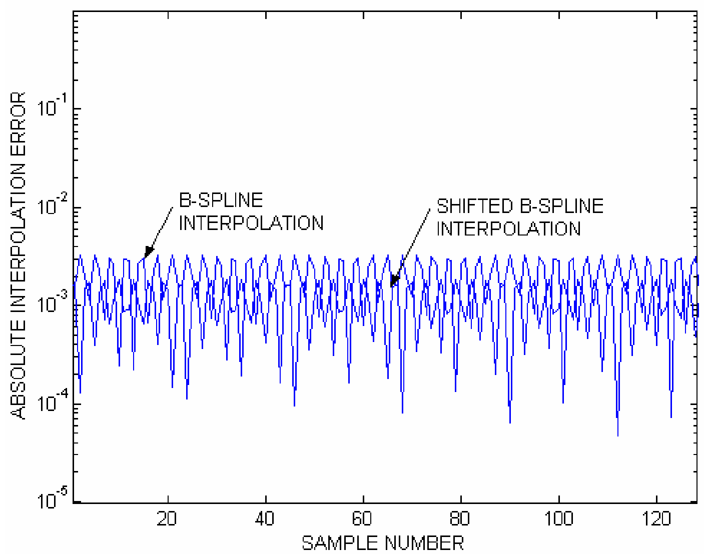

Figure 4. Comparison of the interpolation error yielded by and the shifted B-spline $(p = 4)$ interpolation filter $(\Delta = 0.25,\ d = 0.5)$ and the standard B-spline $(p = 4)$ interpolation for the signal $x[n] = \sin(n)$.

C. Comparison with the standard B-spline interpolation

The interpolation error yielded by and the shifted B-spline $(p = 4)$ interpolation filter $(\Delta = 0.25,\ d = 0.5)$ and the standard B-spline $(p = 4)$ interpolation for a sinusoidal signal $x[n] = \sin(n)$ is given in Figure 4. The interpolation error is about half of the error in standard B-spline interpolation. For the sinusoidal signal $x[n] = \sin(0.1n)$ the interpolation error is about one order smaller compared with the generalized B-spline interpolation (Figure 5).

D. Comparison with the cubic convolution interpolation

Figures 6 and 7 show the interpolation error of the shifted B-spline $(p = 4)$ interpolation filter and the cubic convolution interpolation [20] for the sinusoidal signal $x[n] = \sin(\omega n)$. At frequency $\omega = 1$ the interpolation error of the shifted B-spline is about one order lower compared with the cubic convolution method (Figure 6) and at $\omega = 0.1$ the interpolation error is about two orders lower (Figure 7).

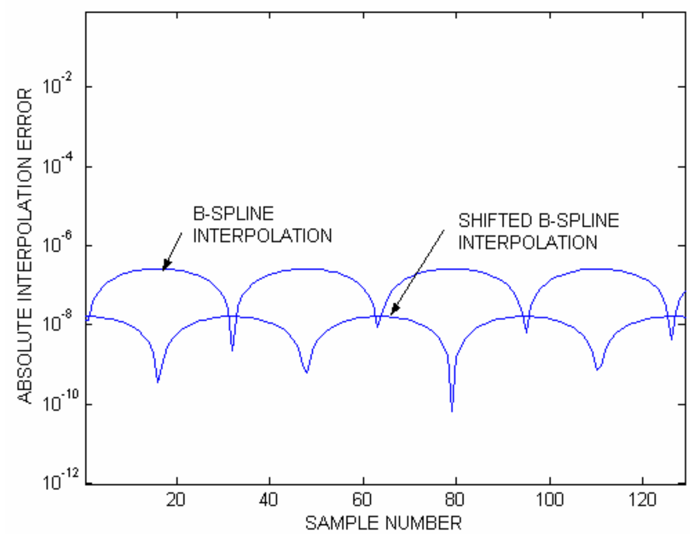

Figure 5. Comparison of the interpolation error yielded by and the shifted B-spline $(p = 4)$ interpolation filter $(\Delta = 0.25,\ d = 0.5)$ and the generalized B-spline $(p = 4)$ interpolation for the signal $x[n] = \sin(0.1n)$.

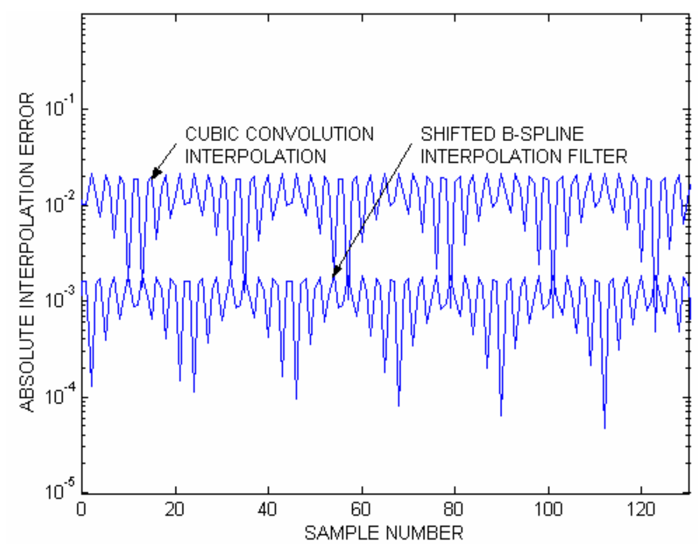

Figure 6. Comparison of the interpolation error yielded by and the shifted B-spline $(p = 4)$ interpolation filter $(\Delta = 0.25,\ d = 0.5)$ and the standard cubic convolution interpolation for the signal $x[n] = \sin(n)$.

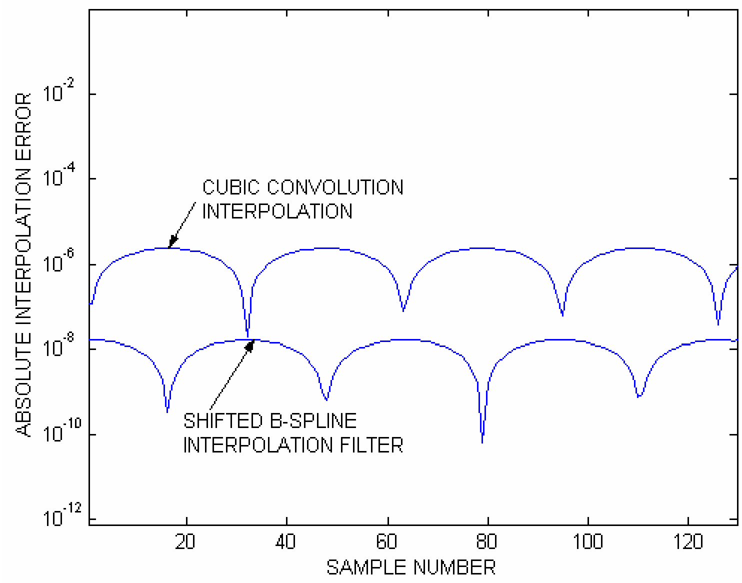

Figure 7. Comparison of the interpolation error yielded by and the shifted B-spline $(p = 4)$ interpolation filter $(\Delta = 0.25,\ d = 0.5)$ and the standard cubic convolution interpolation for the signal $x[n] = \sin(0.1n)$.

Discussion

In this work, we introduced the shifted B-spline interpolation filter, whose performance overrides the standard B-spline interpolation. A key idea in the deducing the z-transform of the shifted discrete B-spline is the use of the binomial series expansion (17), which we invented in constructing the fractional time-shift B-spline filter [22]. In parallel realization of the shifted B-spline interpolation filter (21) only Table I is needed for computation of the coefficients of the $F_k(z)$ subfilters. A surprising observation was the strong dependence of the interpolation error on the Δ-shift. The minimum interpolation error was attained at $\Delta = 0.25$. The optimum shift was not related to the B-spline order p. Analogously, this means that the pulse function for construction of the B-splines via (4) can be defined as

$$p(t) = \begin{cases} 1, & \text{for } -\Delta \le t \le 1 - \Delta, \\ 0, & \text{elsewhere.} \end{cases} \tag{24}$$

The symmetric B-splines are yielded at $\Delta = 0.5$ and the one-sided B-splines at $\Delta = 0$. The optimum value $\Delta = 0.25$ is in the middle of the two values. According to the present study the interpolation error has two local minima: $\Delta = 0$ and $\Delta = 0.25$. Plonka [16, 17] has previously studied the effect of the shift τ of the interpolation knots on the B-spline interpolation error. The theoretical analysis predicted the minimum error norm at $\tau = 0$. The present work supports this observation, since we found out a local minimum at $\Delta = 0$. However, our results indicate a deeper local minimum, when the B-spline has been shifted by a fraction $\Delta = 0.25$ of the sampling period.

The interpolation quality of the Δ-shifted B-spline interpolation exceeds the standard B-spline interpolation and the cubic convolution interpolation, which are standard tools in image processing applications. Blu et al. [18] suggested that the performance of the linear interpolation can be improved by shifting the interpolation knots by a fixed amount. The theoretical calculations suggested a value $\tau = 0.21$ for the optimum shift. The shifting of the interpolation knots was carried out with a discrete-time prefilter. The shifted data values and the shifted time coordinates were then interpolated by the existing linear interpolation hardware. The shifted B-spline interpolation filter (15) can be also interpreted in the similar way. Instead of shifting the interpolation knots the data is prefiltered by $\beta_p^{-1}(\Delta, z)$ (Appendix I) and then fed to the existing B-spline interpolation hardware, where the interpolation kernel $\beta_p(\Delta + d, z)$ is time shifted by Δ. In microprocessor and VLSI circuits the B-spline kernel is usually implemented via time domain convolution using (5). Hence, the hardware changes are minimal, when adapting the shifted B-spline interpolation. In systems, which are based on z-domain filters, the shifted B-spline interpolation filter (15) can be represented by the IIR filter, where only the nominator coefficients depend on d.

The parallel realization of the shifted B-spline interpolation filter allows the fast computation of the interpolated values as a weighted sum (22), which enables rapid and feasible signal processing tasks, such as zooming of images, differentiation, integration, computation of correlation functions, statistical signal analysis and adjustment of fractional delays in communication systems. A typical example is the multiplexed analog-to-digital converter, where the time-shifts between the channels can be corrected. In multirate signal processing the shifted B-spline interpolation filter suits readily for generation of the haft sample delays to obtain the shift invariant biorthogonal wavelet transform [22-24]. Previous methods based on the all-pass Thiran filters suffer from nonlinear phase distortion effects [25].

The present work serves as an augmented framework of the shifted B-spline interpolation filter. The z-transform of the shifted B-spline (19) itself serves as a new extension of the discrete B-spline filters. For example, we obtain the z-transform of the odd p symmetric B-spline filter by starting from one-sided B-spline and shifting it by $\Delta = 0.5$ via (19). We do not know previous z-transform solutions for the odd p symmetric B-splines.

Appendix I

Computation of the inverse discrete B-spline filter.

The inverse discrete B-spline filter can be written as a cascade realization

$$\beta_p^{-1}(\Delta, z) = c\prod_{i=1}^{n} \frac{1}{1 - b_i z^{-1}} \prod_{j=1}^{m} \frac{1}{1 - b_j z^{-1}} = c\prod_{i=1}^{n} S_i(z) \prod_{j=1}^{m} R_j(z), \tag{25}$$

where c is a constant and the roots $|\, b_i \,| \le 1$ and $|\, b_j \,| > 1$. The $S_i(z)$ filters can directly be implemented. The $R_j(z)$ filters can be implemented by the following recursive filtering procedure. First we replace z by z^{-1}

$$R_j(z) = \frac{1}{1 - b_j z^{-1}} = \frac{Y(z)}{U(z)} \Rightarrow R_j(z^{-1}) = \frac{-b_j^{-1} z^{-1}}{1 - b_j^{-1} z^{-1}} = \frac{Y(z^{-1})}{U(z^{-1})}, \tag{26}$$

where $U(z)$ and $Y(z)$ denote z-transforms of the input $u[n]$ and output $y[n]$ signals $(n = 0, 1, 2, ..., N)$. The $U(z^{-1})$ and $Y(z^{-1})$ are the z-transforms of the time reversed input $u[N - n]$ and output $y[N - n]$. The $R_j(z^{-1})$ filter is stable having a root b_j^{-1} inside the unit circle. The following Matlab program **sfilter.m** demonstrates the computation procedure:

```
function y = sfilter(u, b)
u = u(end: -1 : 1);
y = filter([0 -1/b], [1 -1/b], u);
y = y(end: -1 : 1).
```

Appendix II

The derivative of the shifted B-spline filter.

The Laplace transform of the derivative of the shifted B-spline is

$$L\left\{\frac{d}{dt}\beta_p(t+\varepsilon)\right\} = \frac{e^{s\varepsilon}}{s^{p-1}}(1-e^{-s})^p, \tag{27}$$

where $\varepsilon = \Delta + d$. Applying (16) and (17) the derivative of the signal $x[n+d]$ can be computed as

$$Z\left\{\frac{d}{dt}x[n+d]\right\} = D(\Delta, d, z)X(z), \tag{28}$$

where the derivative filter is

$$D(\Delta, d, z) = \beta_p^{-1}(\Delta, z)\sum_{k=0}^{p-2}\frac{(\Delta+d)^k}{k!}(1-z^{-1})^{k+1}\beta_{p-k-1}(z). \tag{29}$$

It is interesting to observe that the same result is obtained in the time domain if we differentiate equation (22) with respect to d

$$\frac{d}{dt}x[n+d] = \sum_{k=1}^{p-1}k(\Delta+d)^{k-1}y_k[n]. \tag{30}$$

Proof. If we take z-transform from (30) by inserting $Z\{y_k[n]\} = \beta_p^{-1}(\Delta, z)F_k(z)$ we obtain (28).

Acknowledgement

We are indebted to the reviewers' comments, which have significantly improved the paper.

References

[1] T. Laakso, V. Valimaki, M. Karjalainen and U. K. Laine, Splitting the unit delay, Tools for fractional delay filter design, IEEE Signal Processing Magazine (1996), 30-80.

[2] S. C. Pei and C. C. Tseng, An efficient design of a variable fractional delay filter using a first-order differentiator, IEEE Signal Processing Lett. 10(10) (2003), 307-310.

[3] S. C. Pei and P. H. Wang, Closed-form design of all-pass fractional delay, IEEE Signal Processing Lett. 11(10) (2004), 788-791.

[4] H. Johansson and P. Lowenborg, Reconstruction of nonuniformy sampled bandlimited signals by means of digital fractional delay filters, IEEE Trans. Signal Process. 50(11) (2002), 2757-2767.

[5] C. C. Tseng, Digital integrator design using Simpson rule and fractional delay filter, IEE Proc. Vision, Image and Signal Process. 153(1) (2006), 79-85.

[6] C. C. Tseng, Digital differentiator design using fractional delay filter and limit computation, IEEE Trans. Circuits and Systems I 52(10) (2005), 2248-2259.

[7] M. Unser, A. Aldroubi and M. Eden, B-spline signal processing, I. Theory, IEEE Trans. Signal Process. 41(2) (1993), 821-833.

[8] M. Unser, A. Aldroubi and M. Eden, B-spline signal processing, II. Efficiency design and applications, IEEE Trans. Signal Process. 41(2) (1993), 834-848.

[9] A. Aldroubi, M. Eden and M. Unser, Discrete B-spline filters for multiresolutions and wavelets of ℓ_2, SIAM J. Math. Anal. 25 (1994), 1412-1432.

[10] T. J. Lim and M. D. Macleod, On-line interpolation using spline functions, IEEE Signal Processing Lett. 3(5) (1996), 144-146.

[11] S. Samadi, M. O. Ahmad and M. N. S. Swamy, Characterization of B-spline digital filters, IEEE Trans. on Circuits and Systems-I: Regular Papers 51(4) (2004), 808-816.

[12] S. Cucchi, F. Desinan, G. Parladori and G. Sicuranza, DSP implementation of arbitrary sampling frequency conversion for high quality sound application, Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. 5 (1991), pp. 3609-3612.

[13] K. Egiazarian, T. Saramaki, H. Chugurian and J. Astola, Modified B-spline interpolators and filters: synthesis and efficient implementation, Proc. Int. Conf. Acoust., Speech, Signal Process. 3 (1996), pp. 1743-1746.

[14] M. Unser, A. Aldroubi and M. Eden, Polynomial spline approximations: Filter design and asymptotic equivalence with Shannon's sampling theorem, IEEE Trans. Information Theory 38(1) (1992), 95-103.

[15] P. Thevenaz, T. Blu and M. Unser, Interpolation revisited, IEEE Trans. Med. Imag. 19 (2000), 739-758.

[16] G. Plonka, Periodic spline interpolation with shifted nodes, J. Approx. Theory 76(1) (1994), 1-20.

[17] G. Plonka, Optimal shift parameters for periodic spline interpolation, Numer. Algorithms 6 (1994), 297-316.

[18] T. Blu, P. Thevenaz and M. Unser, Linear interpolation revitalized, IEEE Trans. Image Process. 13(5) (2004), 710-719.

[19] R. G. Keys, Cubic convolution interpolation for digital image processing, IEEE Trans. Acoust., Speech, Signal Process. 29(6) (1981), 1153-1160.

[20] I. J. Schoenberg, Contribution to the problem of approximation of equidistant data by analytic functions, Quart. Appl. Math. 4 (1946), 45-99.

[21] J. T. Olkkonen and H. Olkkonen, Fractional time-shift B-spline filter, IEEE Signal Process. Lett. 14(10) (2007), 688-691.

[22] H. Olkkonen, J. T. Olkkonen and P. Pesola, Efficient lifting wavelet transform for microprocessor and VLSI applications, IEEE Signal Processing Lett. 12(2) (2005), 120-122.

[23] H. Olkkonen and J. T. Olkkonen, Simplified biorthogonal discrete wavelet transform for VLSI architecture design, Signal, Image and Video Processing 2 (2008), 101-105.

[24] H. Olkkonen, J. T. Olkkonen and P. Pesola, FFT based computation of shift invariant analytic wavelet transform, IEEE Signal Process. Lett. 14(3) (2007), 177-180.

[25] I. W. Selesnick, The design of approximate Hilbert transform pairs of wavelet bases, IEEE Trans. Signal Process. 50(5) (2002), 1144-1152.