

# Butterworth wavelet transforms derived from discrete interpolatory splines: recursive implementation

Amir Z. Averbuch<sup>a,\*</sup>, Alexander B. Pevnyi<sup>b</sup>, Valery A. Zheludev<sup>a</sup>

<sup>a</sup>*School of Mathematical Science, Department of Computer Science, Tel Aviv University, Ramat Aviv, Tel Aviv 69978, Israel*

<sup>b</sup>*Department of Mathematics, Syktyvkar University, Syktyvkar, Russia*

Received 27 November 2000; received in revised form 6 April 2001

---

## Abstract

In the paper we present a new family of biorthogonal wavelet transforms and the related library of biorthogonal symmetric waveforms. For the construction we used the interpolatory discrete splines which enabled us to design a library of perfect reconstruction filter banks. These filter banks are related to Butterworth filters. The construction is performed in a “lifting” manner. The difference from the conventional lifting scheme is that the transforms of a signal are performed via recursive filtering with the use of IIR filters. These filters have linear phase property and the basic waveforms are symmetric. The filters allow fast cascade or parallel implementation. We present explicit formulas for construction of wavelets with arbitrary number of vanishing moments. In addition, these filters yield perfect frequency resolution. The proposed scheme is based on interpolation and, as such, it involves only samples of signals and it does not require any use of quadrature formulas. © 2001 Elsevier Science B.V. All rights reserved.

*Keywords:* Wavelet transform; Butterworth filters; Recursive filters; Lifting scheme

---

## 1. Introduction

Continuous polynomial splines have a rich history as a source for wavelet constructions [2–4,6,9,22,24,25]. But only few authors [11,15,17] use the discrete splines for this purpose. However, discrete splines are natural tool for processing discrete time signals.

In this work we employed the interpolatory discrete splines [16] as a tool for devising a discrete biorthogonal wavelet scheme. The proposed construction is somewhat related to Donoho’s interpolating wavelet construction [7] as it was modified later by Sweldens [19] into what is called the “lifting scheme”. The lifting scheme allows custom design and fast implementation of the transforms. Briefly, the idea of the computation is that values of the signal located at odd positions are predicted by values in the midpoints of the spline that interpolates even values of the signal. Then, the odd subarray is replaced by the difference between the current and the predicted subarrays. On smooth well-correlated fragments of the signal, these differences will be near zero whereas irregular fragments will produce

---

\* Corresponding author. Tel.: +972-3-640-2020; fax: +972-3-640-9357.

E-mail address: amir@math.tau.ac.il (A.Z. Averbuch).

significant differences. This result resembles the operation of the wavelet transform. To further extend this resemblance we should employ the new odd subarray for updating the existing even subarray. The goal of this update is to smooth the even subarray and thus reduce the aliasing which is a consequence of the decimation. Based upon the above strategy, we constructed a new family of biorthogonal wavelet and wavelet packet transforms and a related library of biorthogonal symmetric waveforms. Note that other interpolators can be used as the predicting aggregates. In [2,25] a similar approach was developed through the use of polynomial interpolatory splines. In that paper the computations were conducted in the frequency domain using FFT. In [7,19] were employed interpolatory polynomials.

Our construction resulted in perfect reconstruction filter banks that are linear phase. The corresponding wavelets are symmetric. Our investigation revealed an interesting relation between the discrete splines and the Butterworth filters [13] commonly used in signal processing. The filter banks constructed in the paper comprise filters which act as a bi-directional (forward and backward) half-band Butterworth filters. The frequency response of Butterworth filters are maximally flat and we succeeded in construction of the dual filters with similar property. We name the corresponding wavelets the Butterworth wavelets. We present explicit formulas for the construction of wavelets with arbitrary number of vanishing moments.

The one-pass Butterworth filters were used already for devising orthogonal nonsymmetric wavelets [8]. The computations there were conducted in time domain using recursive filtering. A scheme using recursive filters for the construction of biorthogonal symmetric wavelets was presented in [12,14]. Our scheme also is based on recursive filtering of the signals. The filters which we use are symmetric and allow fast cascade or parallel implementation. Another advantage of our approach is that we stay completely in the signal processing setting.

Note that IIR filters with rational transfer functions are inherent in signal processing schemes using spline functions. They allow recursive implementation. Construction and implementation of such filters were comprehensively studied in [20–22]. The paper is organized as follows: In Section 2 we outline some facts about the discrete splines and the discrete-time Butterworth filters. In Section 3, we devise a family of biorthogonal wavelet-type transforms of signals using lifting steps. The lifting scheme that we propose operates with IIR filters contrary to the conventional lifting scheme. Both the primal and dual schemes for construction are considered. We emphasize the fact that the lifting scheme together with the proposed construction yields an efficient computational algorithm. Section 4 is devoted to the description of the properties of the constructed filter banks. In Section 5 we discuss recursive implementation of the filtering process and present some examples. In Section 6 we describe basis wavelets for the first decomposition scale. The transforms that were presented in Section 3 are one-level (scale) wavelet-type transforms. They can be extended into coarser scales in two ways. One way is to use the multiscale wavelet transform when the frequency domain is split in logarithmic fashion. Another way is to use the wavelet packet transform when the partition of the frequency domain is near uniform and it is being refined in each subsequent scale of the transform. In Section 7 we describe the wavelet type transform. Throughout the paper we present a wide collection of wavelets and their spectra.

## 2. Preliminaries

### 2.1. Discrete splines

In this section, we outline briefly the properties of discrete splines which are needed for further constructions. For detailed description of the subject see [10,16].

The sequences  $\{a(k)\}_{k=-\infty}^{\infty}$ , which belong to the spaces  $l_1$ , we will call the discrete-time signals. The space of discrete-time signals we denote by  $\mathcal{S}$ . The  $z$ -transform of a signal  $\{a(k)\} \in \mathcal{S}$  is defined as

follows:

$$a(z) = \sum_{k=-\infty}^{\infty} z^{-k} a(k).$$

Throughout the paper we assume that  $z = e^{i\omega}$ . We recall the following properties of the  $z$ -transform:

$$a(k) = \sum_{l=-\infty}^{\infty} b(k-l)c(l) \Leftrightarrow a(z) = b(z)c(z), \tag{2.1}$$

$$a_e(z^2) \triangleq \sum_{k=-\infty}^{\infty} z^{-2k} a(2k) = \frac{1}{2}(a(z) + a(-z)), \tag{2.2}$$

$$a_o(z^2) \triangleq \sum_{k=-\infty}^{\infty} z^{-2k} a(2k+1) = \frac{z}{2}(a(z) - a(-z)), \tag{2.3}$$

$$a(z) = a_e(z^2) + z^{-1}a_o(z^2). \tag{2.4}$$

The discrete B-spline of the first order is defined by the following sequence:

$$B_{1,n}(j) = \begin{cases} 1 & \text{if } j=0, \dots, 2n-1, \quad n \in \mathbb{N}, \\ 0 & \text{otherwise, } \quad j \in \mathbb{Z}. \end{cases}$$

We define the higher order B-splines as the discrete convolutions by recurrence:

$B_{p,n} = B_{1,n} * B_{p-1,n}$ . Obviously, the  $z$ -transform of the B-spline of order  $p$  is

$$B_{p,n}(z) = (1 + z^{-1} + z^{-2} + \dots + z^{-2n+1})^p \quad p = 1, 2, \dots$$

In this paper we are interested only in the case when  $p = 2r$ ,  $r \in \mathbb{N}$  and  $n = 1$ . The corresponding splines are denoted as  $B_r = B_{2r,1}$ . In this case we have  $B_r(z) = (1 + z^{-1})^{2r}$ . The B-spline  $B_r(j)$  is symmetric about the point  $j = r$  where it attains its maximal value. We define the central B-spline  $Q_r(j)$  of order  $2r$  as the shift of the B-spline:

$$Q_r(j) \triangleq B_r(j+r), \quad Q_r(z) = z^r B_r(z) = z^r (1 + z^{-1})^{2r}. \tag{2.5}$$

The discrete spline of order  $2r$  is defined as a linear combination, with real-valued coefficients, of shifts of the central B-spline of order  $2r$ :

$$S_r(k) \triangleq \sum_{l=-\infty}^{\infty} c(l)Q_r(k-2l). \tag{2.6}$$

**Definition 2.1.** Let  $\{e(k)\} \in \mathcal{S}$  be a given sequence. The discrete spline  $S_r$  is called the interpolatory spline if the following relations hold:

$$S_r(2k) = e(k), \quad k \in \mathbb{Z}. \tag{2.7}$$

The points  $\{2k\}$  are called the nodes of the spline.

The following proposition shows how the interpolatory splines of any order can be constructed.

**Proposition 2.1.** *The interpolatory spline which satisfies the conditions (2.7) is represented as follows:*

$$S_r(k) = \sum_{l=-\infty}^{\infty} c(l)Q_r(k-2l), \quad c(z^2) = \frac{2e(z^2)}{z^r(1+z^{-1})^{2r} + (-z)^r(1-z^{-1})^{2r}}. \tag{2.8}$$

**Proof.** Let us rewrite (2.7) using (2.6)

$$\sum_{l=-\infty}^{\infty} c(l)Q_r(2(k-l)) = e(k)$$

and apply the  $z^2$  transform on both sides of this equation. From (2.2) it follows that the  $z^2$ -transform of the decimated B-spline is

$$\begin{aligned} Q_{r,e}(z^2) &= \sum_{l=-\infty}^{\infty} z^{-2l} Q_r(2l) = \frac{1}{2}(Q_r(z) + Q_r(-z)) \\ &= \frac{1}{2}(z^r(1+z^{-1})^{2r} + (-z)^r(1-z^{-1})^{2r}). \end{aligned}$$

Hence, we have, using (2.1)

$$\frac{1}{2}(z^r(1+z^{-1})^{2r} + (-z)^r(1-z^{-1})^{2r})c(z^2) = e(z^2).$$

To make sure that  $Q_{r,e}(z^2)$  is nonzero, substitute  $z = e^{i\omega}$ . We have

$$\begin{aligned} Q_{r,e}(z^2) &= \frac{1}{2}[e^{ir\omega}(1+e^{-i\omega})^{2r} + (-1)^r e^{ir\omega}(1-e^{-i\omega})^{2r}] \\ &= \frac{1}{2}\left[\left(2\cos\frac{\omega}{2}\right)^{2r} + \left(2\sin\frac{\omega}{2}\right)^{2r}\right] > 0. \end{aligned}$$

Therefore,  $Q_{r,e}(z^2) > 0$  on the circle  $|z|=1$ . Hence (2.8) follows.  $\square$

For further development we need to know the values of the splines in the midpoints between the nodes, which we denote as  $\sigma(k) = S_r(2k+1)$ ,  $k \in \mathbb{Z}$ .

**Proposition 2.2.** *The z-transform of the interpolatory spline in the midpoints are*

$$\sigma(z^2) = zU(z)e(z^2), \quad U(z) \triangleq \frac{(1+z^{-1})^{2r} - (-1)^r(1-z^{-1})^{2r}}{(1+z^{-1})^{2r} + (-1)^r(1-z^{-1})^{2r}}. \tag{2.9}$$

In addition,  $U(-z) = -U(z)$ .

**Proof.** Similar to previous considerations, we apply the  $z^2$ -transform on both sides of the following equation:

$$\sigma(k) = \sum_{l=-\infty}^{\infty} c(l)Q_r(2k+1-l).$$

From (2.4) we derive

$$\begin{aligned} Q_{r,o}(z^2) &= \sum_{l=-\infty}^{\infty} z^{-2l} Q_r(2l+1) = \frac{z}{2}(Q_r(z) - Q_r(-z)) \\ &= \frac{z}{2}(z^r(1+z^{-1})^{2r} - (-z)^r(1-z^{-1})^{2r}). \end{aligned}$$

Hence, using (2.8) we get

$$\sigma(z^2) = \frac{z}{2}(z^r(1+z^{-1})^{2r} - (-z)^r(1-z^{-1})^{2r})c(z^2) = zU(z)e(z^2). \quad \square$$

### 2.2. Discrete-time Butterworth filters

We recall briefly the notion of Butterworth filter. For details we refer to [13]. The input  $x(n)$  and the output  $y(n)$  of a linear discrete time shift-invariant system are linked as

$$y(n) = \sum_{k=-\infty}^{\infty} f(k)x(n-k). \tag{2.10}$$

Such a processing of the signal  $x(n)$  is called digital filtering and the sequence  $f(n)$  is called the impulse response of the filter. Its  $z$ -transform  $f(z) = \sum_{n=-\infty}^{\infty} z^{-n} f(n)$  is called the transfer function of the filter. Denote by

$$X(\omega) = \sum_{n=-\infty}^{\infty} e^{-i\omega n} x(n), \quad Y(\omega) = \sum_{n=-\infty}^{\infty} e^{-i\omega n} y(n), \quad F(\omega) = \sum_{n=-\infty}^{\infty} e^{-i\omega n} f(n),$$

the discrete Fourier transforms of the sequences. Then, we have from (2.10)  $Y(\omega) = F(\omega)X(\omega)$ . The function  $F(\omega)$  is called the frequency response of the digital filter. The digital Butterworth filter is a filter with the maximally flat frequency response. The magnitude squared frequency responses  $F_l(\omega)$  and  $F_h(\omega)$  of the digital low-pass and high-pass Butterworth filters of order  $r$ , respectively, are given by the formulas

$$|F_l(\omega)|^2 = \frac{1}{1 + (\tan \omega/2 / \tan \omega_c/2)^{2r}}, \quad |F_h(\omega)|^2 = 1 - |F_l(\omega)|^2 = \frac{1}{1 + (\tan \omega_c/2 / \tan \omega/2)^{2r}},$$

where  $\omega_c$  is the so-called cutoff frequency.

We are interested in the half-band Butterworth filters that is  $\omega_c = \pi/2$ . In this case

$$|F_l(\omega)|^2 = \frac{1}{1 + (\tan \omega/2)^{2r}}, \quad |F_h(\omega)|^2 = 1 - |F_l(\omega)|^2 = \frac{1}{1 + (\cot \omega/2)^{2r}}.$$

If we put  $z = e^{i\omega}$  then we obtain the magnitude squared transfer function of the low-pass filter

$$|f_l(z)|^2 = \left( 1 + \frac{(-1)^r (1 - z^{-1})^{2r}}{(1 + z^{-1})^{2r}} \right)^{-1} = \frac{(1 + z^{-1})^{2r}}{(1 + z^{-1})^{2r} + (-1)^r (1 - z^{-1})^{2r}}. \tag{2.11}$$

Similarly, we have the magnitude squared transfer function of the high-pass filter

$$|f_h(z)|^2 = \frac{(-1)^r (1 - z^{-1})^{2r}}{(1 + z^{-1})^{2r} + (-1)^r (1 - z^{-1})^{2r}}. \tag{2.12}$$

It is readily seen that the function  $U$  defined in (2.9) is related to these transfer functions:

$$1 + U(z) = 2|f_l(z)|^2, \quad 1 - U(z) = 2|f_h(z)|^2. \tag{2.13}$$

### 3. Biorthogonal transforms

We introduce a family of biorthogonal wavelet-type transforms that operate on the signal  $\mathbf{x} = \{x(k)\}$ ,  $k \in \mathbb{Z}$ , which we construct through lifting steps. We carry out the construction in the  $z$ -domain and discuss the time-domain implementation in subsequent sections.

The lifting scheme can be implemented in a primal or dual modes. We consider both.

### 3.1. Primal mode

#### 3.1.1. Decomposition

Generally, the primal lifting scheme for decomposition of signals consists of three steps: 1. Split. 2. Predict. 3. Update or lifting. Let us construct our proposed schemes in terms of these steps.

*Split*—We split the array  $\mathbf{x}$  into an even and odd sub-arrays:

$$\mathbf{e}_1 = \{e_1(k) = x(2k)\}, \quad \mathbf{d}_1 = \{d_1(k) = x(2k + 1)\}, \quad k \in \mathbb{Z}.$$

*Predict*—We use the even array  $\mathbf{e}_1$  to predict the odd array  $\mathbf{d}_1$  and redefine the array  $\mathbf{d}_1$  as the difference between the existing array and the predicted one. To be specific, we use the spline  $S_r$  which interpolates the sequence  $\mathbf{e}_1$  and predict the function  $d_1(z^2)$  which is the  $z^2$ -transform of  $\mathbf{d}_1$ . It is predicted by the function  $\sigma(z)$  defined in (2.9). The  $z$ -transform of the new  $d$ -array is defined as follows:

$$d_1^u(z^2) = d_1(z^2) - zU(z)e_1(z^2). \quad (3.14)$$

From now on the superscript  $u$  means an *update* operation of the array.

*Lifting*—We update the even array using the new odd array:

$$e_1^u(z^2) = e_1(z^2) + \beta(z)z^{-1}d_1^u(z^2). \quad (3.15)$$

Generally, the goal of this step is to eliminate aliasing which appears while downsampling the original signal  $\mathbf{x}$  into  $\mathbf{e}_1$ . By doing so we have that  $\mathbf{e}_1$  is transformed into a low-pass filtered and downsampled replica of  $\mathbf{x}$ . In Section 4.2 we will discuss how to achieve this effect by a proper choice of the control filter  $\beta$ , but for now we only require the function  $\beta(z)$  to be real-valued and obey the condition  $\beta(-z) = -\beta(z)$ , so the product  $\beta(z)z^{-1}$  is a function of  $z^2$ .

#### 3.1.2. Reconstruction

The reconstruction of the signal  $\mathbf{x}$  from the arrays  $\mathbf{e}_1^u$  and  $\mathbf{d}_1^u$  is implemented in reverse order: 1. Undo Lifting. 2. Undo Predict. 3. Unsplit.

*Undo Lifting*—We restore the even array:

$$e_1(z^2) = e_1^u(z^2) - \beta(z)z^{-1}d_1^u(z^2). \quad (3.16)$$

*Undo Predict*—We restore the odd array:

$$d_1(z^2) = d_1^u(z^2) + zU(z)e_1(z^2). \quad (3.17)$$

*Unsplit*—The last step represents the standard restoration of the signal from its even and odd components.

In the  $z$ -domain it looks as:

$$x(z) = e_1(z^2) + z^{-1}d_1(z^2). \quad (3.18)$$

### 3.2. Dual scheme

In the primal construction, that was described above, the update step followed the prediction. In some applications it is preferable to have the update step before prediction step and to control the prediction step. In particular, such dual scheme allows adaptive nonlinear wavelet transform [5] by choosing different predictors for different fragments of the signal. We now describe the dual scheme.

#### 3.2.1. Decomposition

1. We start by averaging the even array with its prediction that was derived from the odd array:

$$e_1^u(z^2) = \frac{1}{2}(e_1(z^2) + z^{-1}U(z)d_1(z^2)). \quad (3.19)$$

Such an update results in a smoother even array.

2. We form the details array by extracting from the odd array the new even array supplied with the control function  $\beta(z)$ :

$$d_1^u(z^2) = d_1(z^2) - 2\beta(z)ze_1^u(z^2). \tag{3.20}$$

### 3.2.2. Reconstruction

1. We restore the odd array

$$d_1(z^2) = d_1^u(z^2) + 2\beta(z)ze_1^u(z^2).$$

2. To reconstruct the even array we use  $d_1(z^2)$ :

$$e_1(z^2) = 2e_1^u(z^2) - z^{-1}U(z)d_1(z^2).$$

3. Finally,

$$x(z) = e_1(z^2) + z^{-1}d_1(z^2).$$

## 4. Filter banks

### 4.1. Relation to Butterworth filters

Lifting schemes, that were presented above, yield efficient algorithms for the implementation of the forward and backward transform of  $\mathbf{x} \leftrightarrow \mathbf{e}_1^u \cup \mathbf{d}_1^u$ . But these operations can be interpreted as transformations of the signals by a filter bank that possesses the perfect reconstruction properties.

First we define two filter transfer functions

$$\Phi_{l,r}(z) \triangleq (1 + U(z))/2, \quad \Phi_{h,r}(z) \triangleq (1 - U(z))/2. \tag{4.21}$$

From (2.13) it is clear that the linear phase filter with the transfer function  $\Phi_{l,r}(z)$  is equal to the magnitude squared transfer function of the discrete-time low-pass half-band Butterworth filter of order  $r$ . The transfer function  $\Phi_{h,r}(z)$  of a linear phase filter is equal to the magnitude squared transfer function of the high-pass half-band Butterworth filter. It means that application of these filters on a signal is equivalent to two passes applications (forward and backward) of the corresponding Butterworth filters. We call these filters the bi-directional Butterworth filters.

Define the filter functions

$$\tilde{g}(z) \triangleq 2z^{-1}\Phi_{h,r}(z), \quad \tilde{h}_\beta(z) \triangleq 1 + 2\beta(z)\Phi_{h,r}(z) = 1 + \beta(z)z\tilde{g}(z), \tag{4.22}$$

$$h(j) \triangleq 2\Phi_{l,r}(z), \quad g_\beta(z) \triangleq z^{-1}(1 - 2\beta(z)\Phi_{l,r}(z)) = z^{-1}(1 - \beta(z)h(z)). \tag{4.23}$$

**Theorem 4.1.** *The decomposition and reconstruction formulas of the primal scheme can be represented as follows:*

$$e_1^u(z^2) = \frac{1}{2}(\overline{\tilde{h}_\beta(z)}x(z) + \overline{\tilde{h}_\beta(-z)}x(-z)), \tag{4.24}$$

$$d_1^u(z^2) = \frac{1}{2}(\overline{\tilde{g}(z)}x(z) + \overline{\tilde{g}(-z)}x(-z)), \tag{4.25}$$

$$x(z) = h(z)e_1^u(z^2) + g_\beta(z)d_1^u(z^2). \tag{4.26}$$

**Proof.** We start with the primal decomposition formula (4.25). We modify Eq. (3.14) using Eqs. (2.2) and (2.3). So, we have:

$$\begin{aligned} d_1^u(z^2) &= \frac{z}{2}(x(z) - x(-z) - U(z)(x(z) + x(-z))) \\ &= \frac{z}{2}(x(z)(1 - U(z)) - x(-z)(1 + U(z))). \end{aligned} \quad (4.27)$$

To obtain (4.25), it is sufficient to note that the function  $\tilde{g}$ , defined in (4.22), possesses the property  $\tilde{g}(-z) = -z(1 + U(z))$  and  $U(z)$  is a real-valued function as  $|z| = 1$ . Thus, we see that (4.27) is equivalent to (4.25).

To prove (4.24) we use the already proved relation (4.25). Moreover, we recall that the function  $\beta$  is real-valued and  $(-z)^{-1}\beta(-z) = z^{-1}\beta(z)$ . Then the decomposition formula (3.15) can be rewritten as

$$\begin{aligned} e_1^u(z^2) &= \frac{1}{2}(x(z) + x(-z)) + \frac{\beta(z)z^{-1}}{2}(\tilde{g}(z)x(z) + \overline{\tilde{g}(-z)}x(-z)) \\ &= \frac{1}{2}(x(z)(1 + \beta(z)z^{-1}\overline{\tilde{g}(z)}) + x(-z)(1 + \beta(-z)(-z)^{-1}\overline{\tilde{g}(-z)})). \end{aligned}$$

Hence, (4.24) follows.

To verify the reconstruction formula (4.26), we first rewrite Eq. (3.17) using Eq. (3.16).

$$\begin{aligned} d_1(z^2) &= d_1^u(z^2) + zU(z)(e_1^u(z^2) - \beta(z)z^{-1}d_1^u(z^2)) \\ &= d_1^u(z^2)(1 - \beta(z)z^{-1}U(z)) + ze_1^u(z^2). \end{aligned} \quad (4.28)$$

Then we substitute (3.16) and (4.28) into (3.18).  $\square$

We call  $\tilde{h}_\beta(z)$  and  $\tilde{g}(z)$  the transfer functions of the low-pass and high-pass primal decomposition filters, respectively. We call  $h(z)$  and  $g_\beta(j)$  the transfer functions of the low-pass and high-pass primal reconstruction filters, respectively. These four filters form a perfect reconstruction filter bank [18].

**Theorem 4.2.** *If  $\beta(-z) = -\beta(z)$  then the functions  $\tilde{h}_\beta(j)$ ,  $\tilde{g}(j)$ ,  $h(j)$  and  $g_\beta(j)$  satisfy the perfect reconstruction conditions*

$$\begin{aligned} h(z)\overline{\tilde{h}_\beta(z)} + g_\beta(z)\overline{\tilde{g}(z)} &= 2 \\ h(z)\overline{\tilde{h}_\beta(-z)} + g_\beta(z)\overline{\tilde{g}(-z)} &= 0. \end{aligned} \quad (4.29)$$

**Proof.** From the definitions (4.22)–(4.23) we immediately derive

$$\begin{aligned} h(z)\overline{\tilde{h}_\beta(z)} + g_\beta(z)\overline{\tilde{g}(z)} \\ = (1 + U(z))(1 + \beta_1(z)(1 - U(z))) + (1 - U(z))(1 - \beta_1(z)(1 + U(z))) = 2. \end{aligned}$$

The equations (4.29) can be similarly checked.  $\square$

Similar facts hold for the dual transforms. Let us denote by  $\tilde{H}$  and  $\tilde{G}_\beta$  the transfer functions of the dual decomposition filters and by  $H_\beta$  and  $G$  the transfer functions of the dual reconstruction filters. The dual decomposition filters coincide (up to constant factors) with the primal reconstruction filters and vice versa, i.e.

$$\tilde{H}(z) = h(z)/2, \quad \tilde{G}_\beta(z) = g_\beta(z), \quad H_\beta(z) = 2\tilde{h}_\beta(z), \quad G(z) = \tilde{g}(z).$$

The following is an obvious observation.



**Proposition 4.1.** *The filter functions are linked to each other in the following way:*

$$\tilde{g}(z) = z^{-1}h(-z); \quad g_{\beta}(z) = z^{-1}\tilde{h}_{\beta}(-z).$$

**Remark.** We notice that the dual decomposition filter  $\tilde{H}(z)$  and the primal reconstruction filter  $h(z)$  are equal (up to constant factors) to the transfer function of the bi-directional low-pass half-band Butterworth filter of order  $r$ . The primal decomposition filter  $\tilde{g}(z)$  and the dual reconstruction filter  $G(z)$  multiplied by  $z$  are equal to the bi-directional high-pass half-band Butterworth filter.

#### 4.2. Choosing the control filter

So far we did not specify how to choose the filter  $\beta$ , which occurs during the construction of the primal filters  $g_{\beta}$  and  $\tilde{h}_{\beta}$  and the dual ones  $\tilde{G}_{\beta}$  and  $H_{\beta}$ . The only imposed requirements were that the function  $\beta(z)$  be real-valued and  $\beta(-z) = -\beta(z)$ . Therefore, we are free to use the function  $\beta_1$  for custom design of these filters and the corresponding wavelets. We consider here only one approach how to choose the control filter  $\beta(j)$  which results in retaining the maximal flatness of the filters.

As was mentioned above, the dual decomposition filter  $\tilde{H}$  (the primal reconstruction filter  $h$ ) and the primal decomposition filter  $\tilde{g}$  (the dual reconstruction filter  $G$ ) are equal to the bi-directional low- and high-pass half-band Butterworth filters of order  $r$ ,  $\Phi_{l,r}$  and  $\Phi_{h,r}$ , respectively. These filters are linear phase and maximally flat in their pass- and stop-bands due to the factors  $(1 + z^{-1})^{2r}$  for the low-pass filters and  $(1 - z^{-1})^{2r}$  for the high-pass filters (see (2.11), (2.12)). We retain similar properties for filters which depend on  $\beta$ . An easy way to achieve it is to choose

$$\beta(z) = U(z)/2 = \frac{1(1 + z^{-1})^{2r} - (-1)^r(1 - z^{-1})^{2r}}{2(1 + z^{-1})^{2r} + (-1)^r(1 - z^{-1})^{2r}}. \tag{4.30}$$

Then

$$\begin{aligned} \tilde{h}_{\beta}(z) &= 1 + \frac{1}{2}U(z)(1 - U(z)) = \frac{1}{2}[(1 + U(z)) + (1 + U(z))(1 - U(z))] \\ &= \Phi_{l,r}(z) + 2\Phi_{l,r}(z)\Phi_{h,r}(z) = \Phi_{l,r}(z)(1 + 2\Phi_{h,r}(z)). \end{aligned} \tag{4.31}$$

Similarly

$$\begin{aligned} g_{\beta}(z) &= z^{-1}(1 - \frac{1}{2}U(z)(1 + U(z))) \\ &= z^{-1}[\Phi_{h,r}(z) + 2\Phi_{l,r}(z)\Phi_{h,r}(z)] = z^{-1}[\Phi_{h,r}(z)(1 + 2\Phi_{l,r}(z))]. \end{aligned} \tag{4.32}$$

We conclude from (4.31) and (4.32) that, as in the case of the filters  $h$  and  $\tilde{g}$ , the filters  $\tilde{h}_{\beta}$  and  $g_{\beta}$  are also mirrored replicas of each other. They differ from bi-directional Butterworth filters of order  $r$  by the term  $2\Phi_{l,r}(z)\Phi_{h,r}(z)$  which affects only the central part of the frequency domain.

### 5. Recursive implementation of the transforms

From (2.9) one can see that the function  $zU(z)$  depends actually on  $z^2$  and we denote it as

$$F_r(z^2) \triangleq zU(z) = z \frac{(1 + z^{-1})^{2r} - (-1)^r(1 - z^{-1})^{2r}}{(1 + z^{-1})^{2r} + (-1)^r(1 - z^{-1})^{2r}} = z \frac{(1 + z)^{2r} - (-1)^r(z - 1)^{2r}}{(1 + z)^{2r} + (-1)^r(z - 1)^{2r}}. \tag{5.33}$$

We denote the numerator and the denominator of the function  $F_r(z^2)$  as follows:

$$\begin{aligned} N_r(z) &\triangleq z((1 + z)^{2r} - (-1)^r(z - 1)^{2r}), \\ D_r(z) &\triangleq (z + 1)^{2r} + (-1)^r(z - 1)^{2r}. \end{aligned} \tag{5.34}$$

**Proposition 5.1.** *If the order of the filter is  $r = 2p + 1$  then the following representation holds:*

$$D_r(z) = 4r(\alpha_1^r \alpha_2^r \dots \alpha_p^r)^{-1} z^r \prod_{k=1}^p (1 + \alpha_k^r z^{-2})(1 + \alpha_k^r z^2), \tag{5.35}$$

where

$$\alpha_k^r = \cot^2 \frac{(p+k)\pi}{2r} < 1, \quad k = 1, \dots, p.$$

If  $r = 2p$  then

$$D_r(z) = 2(\alpha_1^r \alpha_2^r \dots \alpha_p^r)^{-1} z^r \prod_{k=1}^p (1 + \alpha_k^r z^{-2})(1 + \alpha_k^r z^2), \tag{5.36}$$

where

$$\alpha_k^r = \cot^2 \frac{(2p+2k-1)\pi}{4r} < 1, \quad k = 1, \dots, p.$$

**Proof.** Suppose that  $r = 2p + 1$ . The equation  $D_r(z) = 0$  is equivalent to  $(z + 1)^{2r} = (z - 1)^{2r}$ . Hence, the roots of  $D_r(z) = 0$  can be found from the relation  $z_k + 1 = e^{2\pi i k / 2r} (z_k - 1)$ ,  $k = 1, 2, \dots, 2r - 1$ . We have

$$z_k = \frac{e^{2\pi i k / 2r} + 1}{e^{2\pi i k / 2r} - 1} = -i \cot \frac{k\pi}{2r}, \quad k = 1, 2, \dots, 2r - 1. \tag{5.37}$$

The points  $x_k = \cot k\pi/2r$  are symmetric around zero and  $x_{2p+1-k} = x_{r-k} = 1/x_k$ , therefore we can write

$$D_r(z) = 4rz \prod_{k=1}^{2p} (z^2 + x_k^2) = 4rz \prod_{k=1}^p (z^2 + \alpha_k^r)(z^2 + (\alpha_k^r)^{-1}),$$

where  $\alpha_k^r = x_{p+k}^2$ . Hence, (5.35) follows.

When  $r = 2p$  the roots are derived from the equation  $z_k + 1 = e^{2\pi i (k-1/2) / 2r} (z_k - 1)$ . So we have

$$z_k = -i \cot \frac{(2k+1)r\pi}{4r}, \quad k = 0, 1, \dots, 2r - 1. \tag{5.38}$$

Hence, (5.36) is derived.  $\square$

**Remark.** It is clear from (2.11) and (2.12) that the function  $D_r(z)$  coincides with the denominator of the magnitude squared transfer functions  $|f_l(z)|^2$  and  $|f_h(z)|^2$  of the half-band Butterworth filters of order  $r$ . In [8] the formulas for the poles of the functions  $|f_l(z)|^2$  are given without a proof. Those formulas are equivalent to (5.37) and (5.38).

**Proposition 5.2.** *If the order is  $r = 2p$  then the following representation holds:*

$$N_r(z) = 4r(\gamma_1^r \gamma_2^r \dots \gamma_{p-1}^r)^{-1} z^r (1 + z^2) \prod_{k=1}^{p-1} (1 + \gamma_k^r z^{-2})(1 + \gamma_k^r z^2), \tag{5.39}$$

where

$$\gamma_k^r = \cot^2 \frac{(p+k)\pi}{2r} < 1, \quad k = 1, \dots, p - 1.$$

If  $r = 2p + 1$  then

$$N_r(z) = 2(\gamma_1^r \gamma_2^r \dots \gamma_p^r)^{-1} z^r (1 + z^2) \prod_{k=1}^p (1 + \gamma_k^r z^{-2})(1 + \gamma_k^r z^2), \tag{5.40}$$

where

$$\gamma_k^r = \cot^2 \frac{(2p + 2k + 1)\pi}{4r} < 1, \quad k = 1, \dots, p.$$

**Proof.** Suppose that  $r = 2p$ . Then the equation  $N_r(z) = 0$  is equivalent to  $(z + 1)^{2r} = (z - 1)^{2r}$ . As in Proposition 5.1 we have

$$z_k = -i \cot \frac{k\pi}{2r} = -i \cot \frac{k\pi}{4p}, \quad k = 1, 2, \dots, 2r - 1,$$

but, unlike the former case, we have here two roots whose modules are equal to one:  $z_p = -i$ ,  $z_{3p} = i$ . The root  $z_{2p} = 0$ . We have

$$N_r(z) = 4rz^2(1 + z^2) \prod_{k=1}^{p-1} (z^2 + \gamma_k^r)(z^2 + 1/\gamma_k^r).$$

Hence, (5.39) follows. Now let  $r = 2p + 1$ . In this case we have

$$z_k = -i \cot \frac{(1 + 2k)\pi}{4r} = -i \cot \frac{(1 + 2k)\pi}{8p + 4}, \quad k = 0, 1, \dots, 2r - 1.$$

Again  $z_p = -i$ ,  $z_{3p+1} = i$ . So we obtain

$$N_r(z) = z(1 + z^2) \prod_{k=1}^p (z^2 + \gamma_k^r)(z^2 + 1/\gamma_k^r).$$

Hence (5.40) follows.

**Corollary 5.1.** *If  $r = 2p + 1$  then the function  $F_r(z)$  is represented as follows:*

$$\begin{aligned} F_r(z) &= \frac{\alpha_1^r \alpha_2^r \dots \alpha_p^r (1 + z) \prod_{k=1}^p (1 + \gamma_k^r z^{-1})(1 + \gamma_k^r z)}{2r \gamma_1^r \gamma_2^r \dots \gamma_p^r \prod_{k=1}^p (1 + \alpha_k^r z^{-1})(1 + \alpha_k^r z)} \\ &= A_r (1 + z) \prod_{k=1}^p R_r(z, k) \prod_{k=1}^p R_r(z^{-1}, k), \end{aligned} \tag{5.41}$$

where

$$A_r \triangleq \frac{\alpha_1^r \alpha_2^r \dots \alpha_p^r}{2r \gamma_1^r \gamma_2^r \dots \gamma_p^r}, \quad R_r(z, k) \triangleq \frac{1 + \gamma_k^r z}{1 + \alpha_k^r z}.$$

If  $r = 2p$  then

$$\begin{aligned} F_r(z) &= \frac{2r \alpha_1^r \alpha_2^r \dots \alpha_p^r (1 + z) \prod_{k=1}^{p-1} (1 + \gamma_k^r z^{-1})(1 + \gamma_k^r z)}{\gamma_1^r \gamma_2^r \dots \gamma_{p-1}^r \prod_{k=1}^p (1 + \alpha_k^r z^{-1})(1 + \alpha_k^r z)} \\ &= A_r (1 + z) \prod_{k=1}^p R_r(z, k) \prod_{k=1}^p R_r(z^{-1}, k), \end{aligned} \tag{5.42}$$

where

$$A_r \triangleq \frac{2r \alpha_1^r \alpha_2^r \dots \alpha_p^r}{\gamma_1^r \gamma_2^r \dots \gamma_{p-1}^r}, \quad R_r(z, p) \triangleq \frac{1}{1 + \alpha_p^r z}, \quad R_r(z, k) \triangleq \frac{1 + \gamma_k^r z}{1 + \alpha_k^r z}, \quad k = 1, \dots, p - 1.$$

To illustrate the implementation we consider the primal decomposition procedure. Since  $zU(z) = F_r(z^2)$  then the predicting formula (3.14) is equivalent to

$$d_1^u(z) = d_1(z) - F_r(z)e_1(z). \tag{5.43}$$

Eq. (5.43) means that in order to obtain the detail array  $\mathbf{d}_1^u$ , we must process the even array  $\mathbf{e}_1$  by the filter  $F_r$  with the transfer function  $F_r(z)$  and extract the filtered array from the odd array  $\mathbf{d}_1$ . But Corollary 5.1 implies that both filters  $F_{2p}$  and  $F_{2p+1}$  can be split into a cascade of  $p$  elementary causal recursive filters, denoted by  $\overrightarrow{R_r(k)}$ , with the transfer function  $R_r(z^{-1}, k)$ ,  $p$  elementary anticausal recursive filters, denoted by  $\overleftarrow{R_r(k)}$ , with the transfer function  $R_r(z, k)$  and the FIR filter  $Q$  with the transfer function  $1 + z$ . On the other hand, the filters can be decomposed into sums of elementary recursive filters. Such a decomposition also allows parallel implementation of the transform.

The causal, anticausal and the FIR filters operate as follows:

$$\mathbf{y} = \overrightarrow{R_r(k)}\mathbf{x} \Leftrightarrow y(l) = x(l) + \gamma_k^r x(l-1) - \alpha_k^r y(l-1), \tag{5.44}$$

$$\mathbf{y} = \overleftarrow{R_r(k)}\mathbf{x} \Leftrightarrow y(l) = x(l) + \gamma_k^r x(l+1) - \alpha_k^r y(l+1), \tag{5.45}$$

$$\mathbf{y} = Q\mathbf{x} \Leftrightarrow y(l) = x(l) + x(l+1). \tag{5.46}$$

If the control filter is chosen as  $\beta(z) = U(z)/2$  then the implementation of the update step of the decomposition (see (3.15)) is just similar to the prediction step. To be specific, to obtain the smoothed array  $\mathbf{e}_1^u$ , we must process the detail array  $\mathbf{d}_1^u$  with the filter  $\Phi_r$  that has the transfer function  $\Phi_r(z) = z^{-1}F_r(z)/2$  and add the filtered array to the even array  $\mathbf{e}_1$ . But the filter  $\Phi_r$  differs from  $F_r/2$  only by one-sample delay and is acting just similarly.

Since the reconstruction in the lifting scheme differs from the decomposition only by the order of operations, its implementation is completely explained above.

### 5.1. Examples of recursive filters

Now we present a few particular cases with filters of various orders.

$r = 1$ . This is the simplest case. We have  $F_1(z) = (1 + z)/2$ . The filter  $F_1$  is reduced to the FIR filter  $Q/2$ .

$r = 2$ . In this case  $\alpha_1^2 = 3 - 2\sqrt{2} \approx 0.172$  and

$$F_2(z) = 4\alpha_1^2 \frac{1 + z}{(1 + \alpha_1^2 z)(1 + \alpha_1^2 z^{-1})}.$$

The filter can be implemented with the following cascade:

$$x_1(k) = 4\alpha_1^2 x(k) - \alpha_1^2 x_1(k-1), \quad y(k) = x_1(k) + x_1(k+1) - \alpha_1^2 y(k+1). \tag{5.47}$$

Another option stems from the following decomposition of the function  $F_2(z)$ :

$$F_2(z) = \frac{4\alpha_1^2}{1 + \alpha_1^2} \left( \frac{1}{1 + \alpha_1^2 z^{-1}} + \frac{z}{1 + \alpha_1^2 z} \right).$$

Then the filter is implemented in parallel mode:

$$y_1(k) = x(k) - \alpha_1^2 y_1(k-1),$$

$$y_2(k) = x(k+1) - \alpha_1^2 y_2(k+1), \quad y = \frac{4\alpha_1^2}{1 + \alpha_1^2} (y_1 + y_2).$$

We note that elementary filters which produce  $y_1$  and  $y_2$  are operating in opposite directions.  
 $r = 3$ . In this case  $\gamma_1^3 = 7 - 4\sqrt{3} \approx 0.0718$ ,  $\alpha_1^3 = 1/3$  and

$$F_3(z) = \frac{1}{18\gamma_1^3} \frac{1 + \gamma_1^3 z}{1 + z/3} \frac{1 + \gamma_1^3 z^{-1}}{1 + z^{-1}/3} (1 + z).$$

This formula leads to a cascade implementation of the transform. For the parallel implementation we use the following decomposition of the transfer function:

$$F_3(z) = \frac{1}{6} \left[ -\frac{8}{1 + z/3} - \frac{8}{9} \frac{z^{-1}}{1 + z^{-1}/3} + z + \frac{35}{3} \right].$$

$r = 4$ . Now  $\gamma_1^4 = 3 - 2\sqrt{2} \approx 0.1716$ ,  $\alpha_1^4 = 7 - 4\sqrt{2} - 2\sqrt{20 - 14\sqrt{2}} \approx 0.4465$ ,  $\alpha_2^4 = 7 + 4\sqrt{2} - 2\sqrt{20 - 14\sqrt{2}} \approx 0.0396$ . So, we have the cascade representation:

$$F_4(z) = \frac{8\alpha_1^4\alpha_2^4}{\gamma_1^4} \frac{1 + \gamma_1^4 z}{1 + \alpha_1^4 z} \frac{1 + \gamma_1^4 z^{-1}}{1 + \alpha_1^4 z^{-1}} \frac{1}{1 + \alpha_2^4 z} \frac{1}{1 + \alpha_2^4 z^{-1}} (1 + z).$$

Denote  $r_1 = -20.751762964438$ ,  $r_2 = -0.226770684430$ ,  $r_3 = -0.020180988365$ ,  $r_4 = -0.001285362767$ . Then the transfer function is represented as follows:

$$F_4(z) = 8 \left[ \frac{r_1\alpha_2^4}{1 + \alpha_2^4 z} + \frac{r_2\alpha_1^4}{1 + \alpha_1^4 z} + \frac{r_3 z^{-1}}{1 + \alpha_1^4 z^{-1}} + \frac{r_4 z^{-1}}{1 + \alpha_2^4 z^{-1}} + 1 \right].$$

### 5.2. Remarks on implementation

*Finite signals*—The application of recursive filters on finite-duration signals requires initialization of the filter. Namely, if the input samples  $\{x(k)\}$  are available only for  $k = 1, \dots, K$  then, to start causal processing with the causal filter  $b/(1 + az^{-1})$  as  $y(k) = bx(k) - ay(k - 1)$ , we need to know the value  $y(0)$ . While performing the anticausal filter  $b/(1 + az)$  as  $y(k) = bx(k) - ay(k + 1)$ , the value  $y(K + 1)$  is required. This initialization problem is discussed in [21] and our scheme of initialization of filters is similar to the scheme suggested in [21].

*Long signals*—Another problem arises in real time processing of long signals such as, for example speech signals. In this case the anticausal part of the filters cannot be implemented. One way to overcome this obstacle is to approximate the constructed IIR filters by FIR filters. For example, the transfer function can be written as

$$\begin{aligned} T(z) &= \frac{1 + z}{(1 + az)(1 + az^{-1})} \\ &= -a^3 z^4 + (-a^3 + a^2)z^3 + (a^2 - a^2 - a^3)z^2 - (a - 1 + (a^2 - a)a)z \\ &\quad + 1 + (a - 1)a - a^3 - (a + (a^2 - a)a)z^{-1} + (a^2 + a^3)/z^{-2} - a^3/z^{-3} + O(a^4). \end{aligned}$$

Thus, the transfer function  $T(z)$  is approximated by the 8-tap FIR filter which can be implemented as a sliding convolution.

Another way is to construct special FIR filters with properties similar to the constructed IIR filters. This can be done using local quasi interpolatory (see [23]) rather than using interpolatory splines as the predicting aggregate in the lifting scheme. We will pursue this way in our next paper.

### 6. Bases for the signal space

The perfect reconstruction filter banks, that were constructed above, are associated with the biorthogonal pairs of bases in the space  $\mathcal{S}$  of discrete-time signals.

In Section 4 we introduced a family of filters by their transfer functions  $h(z)$ ,  $g_\beta(z)$ ,  $\tilde{h}_\beta(z)$ ,  $\tilde{g}(z)$ ,  $\tilde{H}(z)$ ,  $\tilde{G}_\beta(z)$ ,  $H_\beta(z)$ ,  $G(z)$ . We denote by  $\varphi^1(k)$ ,  $\psi_\beta^1(k)$ ,  $\tilde{\varphi}_\beta^1(k)$ ,  $\tilde{\psi}^1(k)$ ,  $\tilde{\Phi}^1(k)$ ,  $\tilde{\Psi}_\beta^1(k)$ ,  $\Phi_\beta^1(k)$ ,  $\Psi^1(k)$  the impulse response functions of the corresponding filters. It means that, for example

$$h(z) = \sum_{k \in \mathbb{Z}} z^{-k} \varphi^1(k)$$

and similarly for the other functions.

**Theorem 6.1.** *The shifts of the functions  $\varphi^1(k)$ ,  $\psi_\beta^1(k)$ ,  $\tilde{\varphi}_\beta^1(k)$  and  $\tilde{\psi}^1(k)$  form a biorthogonal pairs of bases for the space  $\mathcal{S}$ . This means that any signal  $\mathbf{x} \in \mathcal{S}$  can be represented as:*

$$x(l) = \sum_{k \in \mathbb{Z}} e_1^u(k) \varphi^1(l - 2k) + \sum_{k \in \mathbb{Z}} d_1^u \psi_\beta^1(l - 2k).$$

The coordinates  $e_1^u(k)$  and  $d_1^u(k)$  can be represented as inner products:

$$e_1^u(k) = \langle \mathbf{x}, \tilde{\varphi}_{\beta,k}^1 \rangle, \quad \text{where } \tilde{\varphi}_{\beta,k}^1(l) = \tilde{\varphi}_\beta^1(l - 2k), \tag{6.48}$$

$$d_1^u(k) = \langle \mathbf{x}, \tilde{\psi}_k^1 \rangle, \quad \text{where } \tilde{\psi}_k^1(l) = \tilde{\psi}^1(l - 2k). \tag{6.49}$$

**Proof.** We start with the reconstruction formula (4.26) which we rewrite as

$$x(z) = x_h(z) + x_g(z) \quad \text{where } x_h(z) = h(z)e_1^u(z^2), \quad x_g(z) = g_\beta(z)d_1^u(z^2).$$

We can write

$$\sum_{l \in \mathbb{Z}} z^{-l} x_h(l) = \sum_{k,n \in \mathbb{Z}} z^{-n-2k} e_1^u(k) \varphi^1(n) \Leftrightarrow x_h(l) = \sum_{k \in \mathbb{Z}} e_1^u(k) \varphi^1(l - 2k).$$

Similarly, we derive the relation

$$x_g(l) = \sum_{k \in \mathbb{Z}} d_1^u \psi_\beta^1(l - 2k).$$

Let us consider the decomposition formula (4.24). From the property (2.1) we conclude that  $\overline{\tilde{h}_\beta(z)}x(z)$  is the  $z$ -transform of the sequence

$$a(k) \triangleq \sum_{l \in \mathbb{Z}} \tilde{\varphi}_\beta^1(l - k)x(l).$$

Now from (4.24) and (2.2) we have

$$e_1^u(k) = a(2k) = \sum_{l \in \mathbb{Z}} \tilde{\varphi}_\beta^1(l - 2k)x(l) = \langle \mathbf{x}, \tilde{\varphi}_{\beta,k}^1 \rangle$$

which proves Eq. (6.48). Similarly, Eq. (6.49) is proved.  $\square$

The proved theorem justifies the following definition.

**Definition 6.1.** The functions  $\varphi^1$  and  $\psi_\beta^1$ , which belong to the space  $\mathcal{S}$ , are called the primal low-frequency and high-frequency reconstruction wavelets of the first scale, respectively. The functions  $\tilde{\varphi}_\beta^1$  and  $\tilde{\psi}^1$ , which belong to the space  $\mathcal{S}$ , are called the primal low-frequency and high-frequency decomposition wavelets of the first scale, respectively.

**Corollary 6.1.** *The following biorthogonal relations hold:*

$$\langle \tilde{\varphi}_{\beta,k}^1, \varphi_l^1 \rangle = \langle \psi_{\beta,k}^1, \tilde{\psi}_l^1 \rangle = \delta_k^l, \quad \langle \tilde{\varphi}_{\beta,k}^1, \psi_{\beta,l}^1 \rangle = \langle \tilde{\psi}_l^1, \varphi_k^1 \rangle = 0, \quad \forall l, k.$$

**Remarks.**

1. The decomposition wavelets of the dual scheme are the reconstruction wavelets for the primal scheme and vice versa.
2. To build the primal reconstruction wavelet  $\varphi^1$ , we can apply the primal lifting reconstruction procedure to the arrays  $e_1^u(k) = \delta_0^k, d_1^u(k) = 0 \forall k$ . To obtain the wavelet  $\psi_\beta^1$ , we apply the procedure on the arrays  $d_1^u(k) = \delta_0^k, e_1^u(k) = 0 \forall k$ . To build the primal decomposition wavelets  $\tilde{\varphi}_\beta^1, \tilde{\psi}^1$  we have to apply the dual lifting reconstruction procedure on these arrays.

**Definition 6.2.** We say that a wavelet  $\psi$  has  $m$  vanishing moments if the following relations hold

$$\sum_{k \in \mathbb{Z}} k^s \psi(k) = 0, \quad s = 0, 1, \dots, m - 1.$$

**Proposition 6.1.** *The high-frequency decomposition and reconstruction wavelets of order  $r$  have  $2r$  vanishing moments.*

**Proof.** We consider first the primal decomposition wavelet  $\tilde{\psi}^1$ . Let us ignore for a moment the assumption  $|z| = 1$ . Then we examine the function

$$q(z) \triangleq \sum_{k \in \mathbb{Z}} z^k \tilde{\psi}^1(k) = \tilde{g}(z^{-1})$$

of the complex variable  $z$ . From (4.21), (2.12) and (2.12) we have

$$q(z) = z(1 - U(z^{-1})) = \frac{2(-1)^r(1 - z)^{2r}}{(1 + z)^{2r} + (-1)^r(1 - z)^{2r}} = \frac{2(-1)^r(1 - z)^{2r}}{D_r(z)}.$$

Recall that the function  $D_r(z)$  was defined in (5.34). As it follows from Proposition 5.1, the denominator  $D_r(z)$  does not vanish in some vicinity of  $z = 1$ . Therefore, in this vicinity the function  $q(z)$  is analytic and allows differentiation. It is clear that  $q^{(s)}(1) = 0, s = 0, 1, \dots, 2r - 1$ . On the other hand,

$$q^{(s)}(1) = \sum_{k \in \mathbb{Z}} k^{[s]} \tilde{\psi}^1(k), \quad \text{where } k^{[s]} \triangleq k(k - 1)(k - 2) \cdots (k - s + 1).$$

Note that any monomial  $k^l$  can be represented as a linear combination of the polynomials  $k^{[n]}, n = 0, \dots, l$ . Thus, the wavelet  $\tilde{\psi}^1$  has  $2r$  vanishing moments. Obviously, the same is true for the dual reconstruction wavelet  $\Psi^1$ .

Now we turn to the primal reconstruction wavelet  $\psi_\beta^1$ . From (4.32) we have that the function

$$p(z) \triangleq \sum_{k \in \mathbb{Z}} z^k \psi_\beta^1(k) = \frac{z}{2}(1 - U(z^{-1}))(2 + U(z^{-1})) = (1 - z)^{2r} Q(z),$$

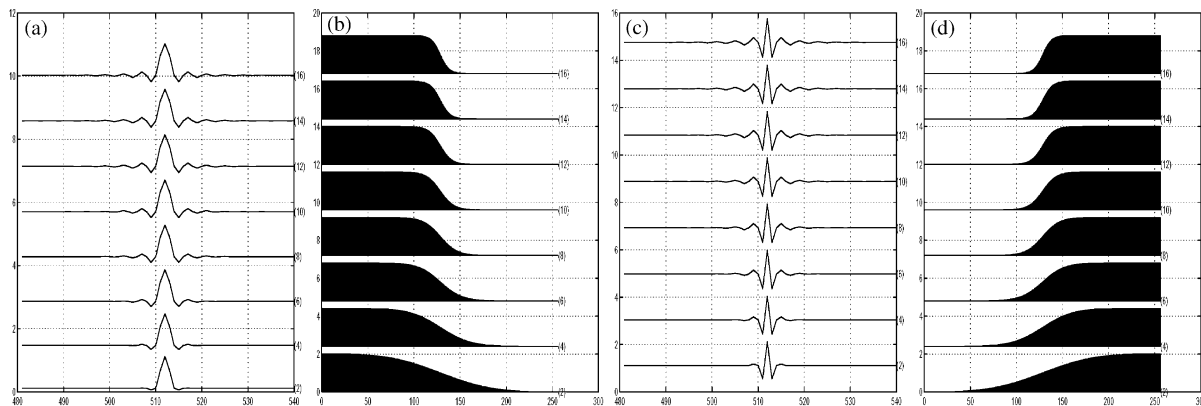


Fig. 1. (a) Low-frequency reconstruction wavelets  $\varphi^1$  of order  $r=1$  at the bottom to order  $r=8$  at the top. (b) The Fourier transforms of (a). (c) The high-frequency decomposition wavelets  $\tilde{\psi}^1$ . (d) Their Fourier transforms.

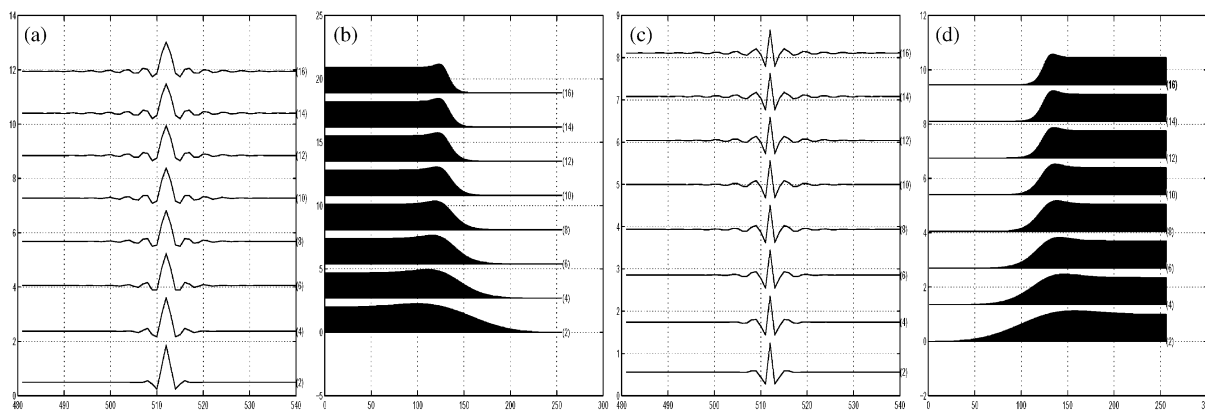


Fig. 2. (a) Low-frequency decomposition wavelets  $\tilde{\varphi}_\beta^1$  of order  $r=1$  at the bottom to order  $r=8$  at the top. (b) The Fourier transforms of (a). (c) The high-frequency reconstruction wavelets  $\psi_\beta^1$ . (d) Their Fourier transforms.

where  $Q$  is a function which is analytic in some vicinity of  $z=1$ . Hence, we conclude that the wavelet  $\psi_\beta^1$  has  $2r$  vanishing moments. The same is true for the dual reconstruction wavelet  $\tilde{\Psi}^1$ .  $\square$

In Fig. 1 we display the wavelets  $\varphi^1$  and  $\tilde{\psi}^1$  which are the impulse responses of the bi-directional discrete-time low- and high-pass half-band Butterworth filters, respectively. To illustrate the frequency response, we display the discrete Fourier transform of the periodized versions of these wavelets. In Fig. 2 we display the wavelets  $\tilde{\varphi}_\beta^1$  and  $\tilde{\psi}_\beta^1$  which depend on the filter  $\beta$ . We observe that the wavelets and their spectra are similar to the previous ones but the flatness of the filters is disturbed by the bumps (“over shooting”) near the cutoff. These bumps appear due to the term  $2\Phi_{l,r}(z)\Phi_{h,r}(z)$ . For higher  $r$  we get sharper cutoff and the “overshooting” becomes more visible.

All the filters which are used in our wavelet transform are combination of the bi-directional Butterworth filters. Therefore, it is appropriate to name the corresponding wavelets the *Butterworth wavelets*.



### 7. Multiscale wavelet transforms

Repeated applications of the transform can be achieved in an iterative way. It can be implemented by either a linear invertible transform of a wavelet type or by a wavelet packet type transform which results in an overcomplete representation of the signal. In this section we explain one multiscale advance of the wavelet transform.

In this transform we store the array  $\mathbf{d}_1^u$  and decompose the array  $\mathbf{e}_1^u$ . The transformed arrays  $\mathbf{e}_2^u$  and  $\mathbf{d}_2^u$  of the second decomposition scale are derived from the even and odd sub-arrays of the array  $\mathbf{e}_1^u$  by the same lifting steps as those described in Section 3. The transforms can be conducted in either the primal or the dual mode independently of the mode in which it was used for the transform to the first scale. The transform is implemented using the recursive filters presented in Section 5. As a result we get that the signal  $\mathbf{x}$  is transformed into three subarrays:  $\mathbf{x} \leftrightarrow \mathbf{d}_1^u \cup \mathbf{d}_2^u \cup \mathbf{e}_2^u$ . The reconstruction is performed in the reverse order.

Again, the transform leads to expansions of the signal with biorthogonal pairs of bases. Consider, for example, the case when both the first and second scale transforms are primal. As in Section 6, we present the  $z$ -transform of the signal as follows:

$$x(z) = x_h(z) + x_g(z) \quad \text{where } x_h(z) = h(z)e_1^u(z^2), \quad x_g(z) = g_\beta(z)d_1^u(z^2).$$

But, in turn,

$$e_1^u(z) = e_{1,h}^u(z) + e_{1,g}^u(z) \quad \text{where } e_{1,h}^u(z) = h(z)e_2^u(z^2), \quad e_{1,g}^u(z) = g_\beta(z)d_2^u(z^2).$$

Substituting the latter relation into the former one, we get

$$x(z) = x_{hh}(z) + x_{gh}(z) + x_g(z) \quad \text{where } x_{hh}(z) = h(z)h(z^2)e_2^u(z^4), \quad x_{gh}(z) = h(z)g_\beta(z^2)d_2^u(z^4).$$

Hence, the signal is expanded as follows

$$x(l) = \sum_{k \in \mathbb{Z}} e_2^u(k)\varphi^2(l - 4k) + \sum_{k \in \mathbb{Z}} d_2^u(k)\psi_\beta^2(l - 4k) + \sum_{k \in \mathbb{Z}} d_1^u(k)\psi_\beta^1(l - 2k), \tag{7.50}$$

where low- and high-frequency reconstruction wavelets of the second scale are defined as

$$\varphi^2(l) = \sum_{k \in \mathbb{Z}} \varphi^1(k)\varphi^1(l - 2k), \quad \psi_\beta^2(l) = \sum_{k \in \mathbb{Z}} \psi_\beta^1(k)\varphi^1(l - 2k).$$

The coordinates in (7.50) are inner products with 4-sample shifts of the decomposition wavelets of the second scale:

$$\tilde{\varphi}_\beta^2(l) = \sum_{k \in \mathbb{Z}} \tilde{\varphi}_\beta^1(k)\tilde{\varphi}_\beta^1(l - 2k), \quad \tilde{\psi}_\beta^2(l) = \sum_{k \in \mathbb{Z}} \tilde{\psi}_\beta^1(k)\tilde{\varphi}_\beta^1(l - 2k).$$

Namely,

$$e_2^u(k) = \langle \mathbf{x}, \tilde{\varphi}_{\beta,k}^2 \rangle, \quad \text{where } \tilde{\varphi}_{\beta,k}^2(l) = \tilde{\varphi}_\beta^2(l - 4k)$$

$$d_2^u(k) = \langle \mathbf{x}, \tilde{\psi}_{\beta,k}^2 \rangle, \quad \text{where } \tilde{\psi}_{\beta,k}^2(l) = \tilde{\psi}_\beta^2(l - 4k).$$

The cases with dual transforms are treated similarly.

In Fig. 3 we display the Butterworth wavelets of order 1 up to the fourth level and their spectra. Note that these wavelets are compactly supported and have 2 vanishing moments.

In Fig. 4 we display wavelets of order 10 up to the fourth level and their spectra. These wavelets, which are outperformed by the wavelets of lower orders in spatial localization, win in frequency localization, and smoothness. They have 20 vanishing moments.

Unlike the mechanism in the wavelet transform, in the wavelet packet transform both sub-arrays  $\mathbf{e}_1^u$  and  $\mathbf{d}_1^u$  of the first scale are subject to decomposition that produces four second-scale sub-arrays. In turn,

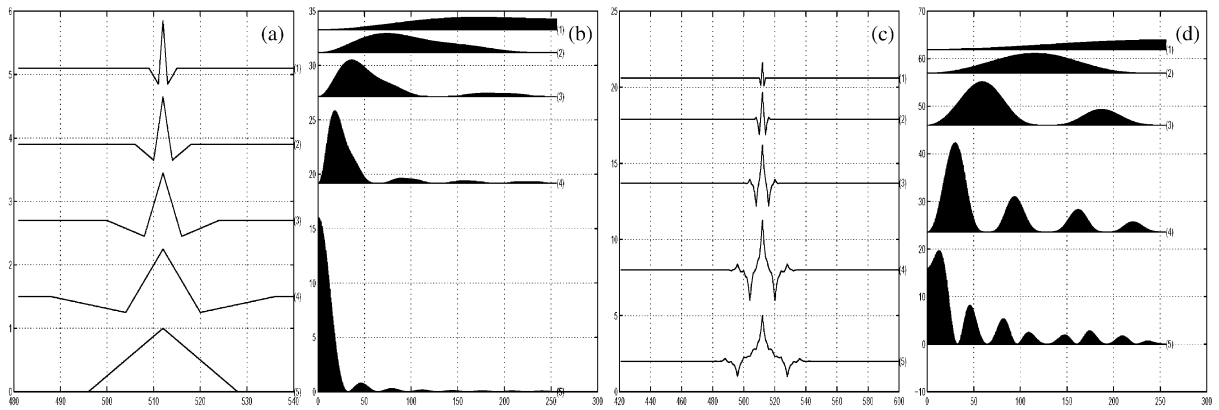


Fig. 3. (a) Reconstruction wavelets  $\psi_{\beta}^l$ ,  $l = 1 \dots 4$  of order 1 (lines 1–4) and  $\varphi^l$ ,  $l = 4$  (line 5). (b) Their spectra. (c) Decomposition wavelets  $\tilde{\psi}_{\beta}^l$ ,  $l = 1 \dots 4$  of order 1 (lines 1–4) and  $\tilde{\varphi}_{\beta}^l$ ,  $l = 4$  (line 5). (d) Their spectra.

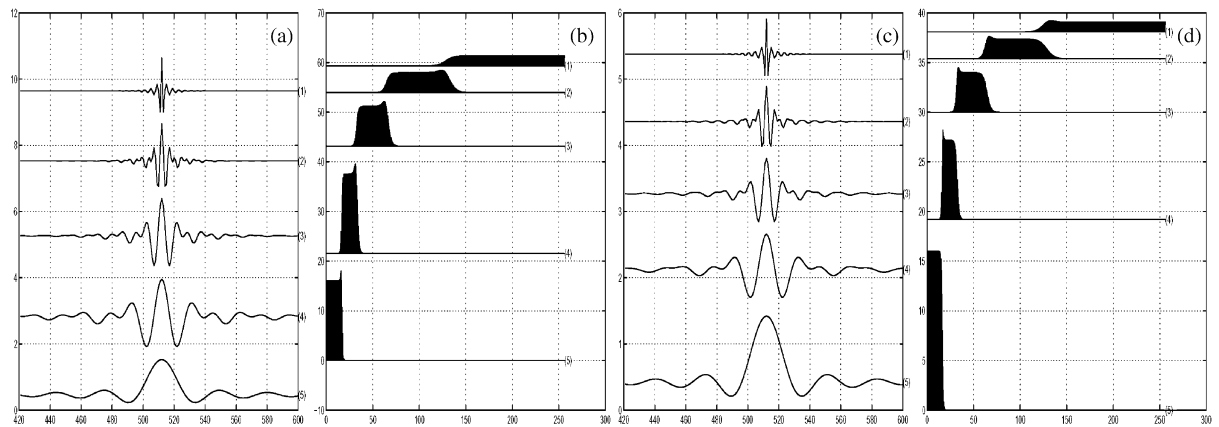


Fig. 4. (a) Reconstruction wavelets  $\psi_{\beta}^l$ ,  $l = 1 \dots 4$  of order 10 (lines 1–4) and  $\varphi^l$ ,  $l = 4$  (line 5). (b) Their spectra. (c) Decomposition wavelets  $\tilde{\psi}_{\beta}^l$ ,  $l = 1 \dots 4$  of order 1 (lines 1–4) and  $\tilde{\varphi}_{\beta}^l$ ,  $l = 4$  (line 5). (d) Their spectra.

these four arrays produce eight sub-arrays for the third scale, and so on. All sub-arrays which are related to a certain scale are stored. Without going into details, we display in Fig. 5 the wavelet packets of the third scale of first and fifteenth orders and their spectra. The wavelet packets are derived using the primal transforms.

### 8. Conclusions

We presented a new family of biorthogonal wavelet transforms and the related library of biorthogonal symmetric waveforms. For the construction we used the interpolatory discrete splines which enabled us to design a library of perfect reconstruction filter banks. These filter banks are intimately related to Butterworth filters.

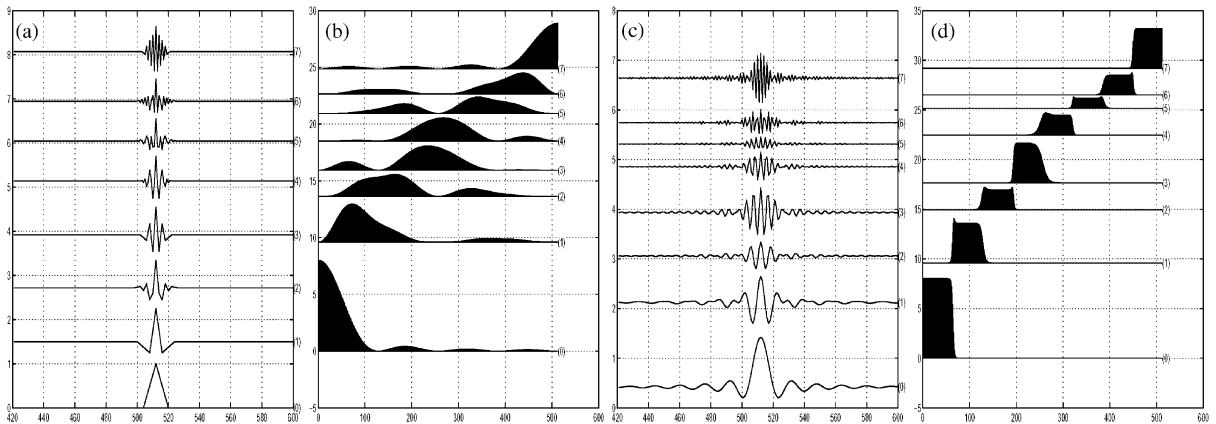


Fig. 5. (a) Wavelet packets of third scale of order 1 (compactly supported). (b) Their spectra. (c) Wavelet packets of third scale of order 15. (d) Their spectra.

The construction is performed in a “lifting” manner that allows more efficient implementation and provides tools for custom design of the filters and wavelets. As it is common in lifting schemes, the computations can be carried out “in place” and the inverse transform is performed in a reverse order. The difference with the conventional lifting scheme [19] is that all the transforms are implemented using recursive IIR filters related to digital Butterworth filters. The filters are symmetric and allow fast cascade or parallel implementation.

We established explicit formulas for the construction of wavelets with arbitrary number of vanishing moments. We also presented examples of filters needed for the construction of wavelets with the number of vanishing moments up to 8.

The computational complexity of the application of the wavelet transform on a signal of length  $N$  is  $O(N)$ . It is remarkably lower than the complexity of the transforms with the compactly supported wavelets possessing the corresponding number of vanishing moments. But it should be pointed out that for the transforms with large number of vanishing moments, the algorithm similar to the algorithm developed in [25], which is based on fast Fourier transform (FFT), is more efficient. When an FFT based algorithm is used, an increase of the order does not affect the cost of the implementation.

We should particularly emphasize that our scheme is based on interpolation and, as such, it involves only samples of signals and it does not require any use of quadrature formulas. This property is valuable for digital signal and image processing.

Also of great importance to these applications is the fact that these filters have linear phase property and the basic waveforms are symmetric. In addition, these filters yield perfect frequency resolution.

We plan to use the presented library of wavelet transforms for some applications, such as compression of multimedia images, seismic data, multiscale classification and identification of quasi-periodic signals, numerical solutions of PDEs, to name a few. Our initial results in compression of images and seismic data using wavelets with four and six vanishing moments are encouraging and will be presented in our next paper. The wavelet packets of higher orders proved to be very efficient when used within our algorithm for detection of moving vehicles [1].

## References

- [1] A.Z. Averbuch, E. Hulata, V.A. Zheludev, I. Kozlov, A wavelet packet algorithm for classification and detection of moving vehicles, *Multidimensional Systems Signal Process.* 12 (1) (2001) 9–31.
- [2] A. Averbuch, V. Zheludev, Construction of biorthogonal discrete wavelet transforms using interpolatory splines, *Appl. Comput. Harmonic Anal.*, submitted. [www.math.tau.ac.il/~amir,zhel](http://www.math.tau.ac.il/~amir,zhel)
- [3] G. Battle, A block spin construction of ondelettes. Part I. Lemarié functions, *Comm. Math. Phys.* 110 (1987) 601–615.
- [4] C.K. Chui, J.Z. Wang, On compactly supported spline wavelets and a duality principle, *Trans. Amer. Math. Soc.* 330 (1992) 903–915.
- [5] R.L. Claypoole Jr., J.M. Davis, W. Sweldens, R. Baraniuk, Nonlinear wavelet transforms for image coding via lifting, *IEEE Trans. Signal Proc.*, submitted.
- [6] A. Cohen, I. Daubechies, J.-C. Feauveau, Biorthogonal bases of compactly supported wavelets, *Comm. Pure Appl. Math.* 45 (1992) 485–560.
- [7] D.L. Donoho, Interpolating wavelet transform, Preprint 408, Department of Statistics, Stanford University, 1992.
- [8] C. Herley, M. Vetterli, Wavelets and recursive filter banks, *IEEE Trans. Signal Proc.* 41 (12) (1993) 2536–2556.
- [9] P.G. Lemarié, Ondelettes à localisation exponentielle, *J. Math. Pures Appl.* 67 (1988) 227–236.
- [10] V.N. Malozemov, A.B. Pevnyi, Discrete periodic splines and their numerical application, *Comput. Math. Math. Phys.* 38 (1998) 1181–1192.
- [11] V.N. Malozemov, A.B. Pevnyi, A.A. Tretyakov, A fast wavelet transform for discrete periodic signals and images, *Problems Inform. Transmission* 34 (2) (1998) 161–168.
- [12] D. Marpe, G. Heising, A. P. Petukhov, H. L. Cycon, Video coding using a bilinear image warping motion model and wavelet-based residual coding, *Proceedings of the SPIE Conference on Wavelet Applications in Signal and Image Processing VII*, Denver, CO, July 1999, SPIE, Vol. 3813, pp. 401–408.
- [13] A.V. Oppenheim, R.W. Shafer, *Discrete-Time Signal Processing*, Englewood Cliffs, New York, Prentice-Hall, 1989.
- [14] A.P. Petukhov, Biorthogonal wavelet bases with rational masks and their applications, *Proceedings of St. Petersburg Mathematics Society*, Vol. 7, 1999, pp. 168–193 (in Russian).
- [15] A.B. Pevnyi, V.A. Zheludev, On wavelet analysis in the discrete splines space, *Proceedings Second International Conference on Tools for Math. Modeling99*, SPTU, St. Petersburg; June 1999, Vol. 4, 1999, pp. 181–195.
- [16] A.B. Pevnyi, V.A. Zheludev, On the interpolation by discrete splines with equidistant nodes, *J. Approx. Theory* 102 (2000) 286–301.
- [17] A.B. Pevnyi, V.A. Zheludev, Construction of wavelet analysis in the space of discrete splines using Zak transform, *J. Fourier Anal. Appl.*, to appear.
- [18] G. Strang, T. Nguen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Boston, 1996.
- [19] W. Sweldens, The lifting scheme: a custom design construction of biorthogonal wavelets, *Appl. Comput. Harm. Anal.* 3 (2) (1996) 186–200.
- [20] M. Unser, A. Aldroubi, M. Eden, B-spline signal processing: Part I—theory, *IEEE Trans. Signal Process.* 41 (2) (February 1993) 821–832.
- [21] M. Unser, A. Aldroubi, M. Eden, B-spline signal processing: Part II—efficient design and applications, *IEEE Trans. Signal Process.* 41 (2) (February 1993) 834–848.
- [22] M. Unser, A. Aldroubi, M. Eden, A family of polynomial spline wavelet transforms, *Signal Processing* 30 (1993) 141–162.
- [23] V.A. Zheludev, Local quasi-interpolating splines and Fourier transforms, *Soviet Math. Dokl.* 31 (1985) 573–577.
- [24] V.A. Zheludev, Periodic splines, harmonic analysis, and wavelets, in: *Signal and Image Representation in Combined Spaces*, Edited by Y.Y. Zeevi and R. Coifman (Eds), *Wavelet Anal. Appl.*, Vol. 7, Academic Press, San Diego, CA, 1998, pp. 477–509.
- [25] V.A. Zheludev, A. Z. Averbuch, A biorthogonal wavelet scheme based on interpolatory splines, *Proceedings of the Second International Conference on Tools for Math. Modeling99*, SPTU, St. Petersburg, June 1999, Vol. 4, pp. 214–231.