# Wavelet transforms generated by splines

Amir Z. Averbuch  Valery A. Zheludev
School of Computer Science
Tel Aviv University
Tel Aviv 69978, Israel

**Abstract**

In this paper we design a new family of biorthogonal wavelet transforms that are based on polynomial and discrete splines. The wavelet transforms are constructed from various types of interpolatory and quasi-interpolatory splines. The transforms use finite and infinite impulse response filters and are implemented in a fast lifting mode. We analyze properties of the generated wavelets. We describe successful applications of the designed transforms to still image compression.

## 1 Introduction

In this paper we describe a new generic technique for the design of biorthogonal wavelet transforms. Some of the results that are presented in this paper have already appeared in [6]–[9]. However, this paper contains a unified theory that combines a full theoretical justifications of the previous results with new facts about the spline-based wavelet transforms. The developed technique enables us to construct a wide family of transforms with various properties. It supports flexible adaptation of the transforms to the problems under consideration. In particular, the newly designed transforms prove to be efficient for distinct computational problems such as image compression, feature extraction for signal identification, to name a few. The performance of the suggested transforms for still image compression is similar to the performance of the transform with 9/7 wavelets ([4]) on most of the benchmark images. Our

approach combines custom-design capabilities which are inherent in the lifting schemes [46] with the usage of the well-developed theory of interpolatory, quasi-interpolatory, continuous and discrete splines [34, 43, 55, 56].

Polynomial splines are a common source for wavelet constructions. Until recently, two approaches governed the construction of wavelet schemes that use splines. One is based on orthogonal ([10, 26]) and semi-orthogonal wavelets in spline spaces [12, 49, 57]. This approach produces, in particular, compactly supported spline wavelets. However, their dual wavelets have infinite support. The other approach, which employs splines in wavelet analysis, was introduced by Cohen, Daubechies and Feauveau [14], who constructed symmetric compactly supported spline wavelets whose dual wavelets remain compactly supported and symmetric but do not belong to a spline space.

However, since the introduction of the lifting scheme for the design of wavelet transforms [46], a new way has been opened for the use of splines as a tool for devising wavelet transforms.

The basic lifting scheme for the wavelet transform of a discrete-time signal consists of three steps:

**Split** – The signal is split into even and odd subarrays.

**Predict** – The filtered even array is used to predict the odd array. Then, the odd array is redefined as the difference between the existing array and the predicted one. If the predictor is correctly chosen then this step decorrelates the signal and reveals its high-frequency component.

**Update** – To eliminate aliasing, which appears while downsampling the original signal, and to obtain the low-frequency component of the signal, the even array is updated using the filtered new odd array.

The newly produced even and odd subarrays are the coefficients from a single decomposition step of the wavelet transform. The inverse transform is implemented in a reverse order.

The transform generates biorthogonal wavelet bases for the signal space. The structure of the transform and its generated wavelets are determined by the choice of the predicting and updating filters. In the construction by Donoho [21], an odd sample is predicted from

a polynomial interpolation of neighboring even samples. We propose to construct a spline, which interpolates or quasi-interpolates even samples of a signal and to use values of this spline at midpoints between the (quasi-)interpolation points as predictions for odd samples of the signal. By using splines of various types and orders we obtain a variety of filters for the *predict* step. After a proper modification, these filters can be used for the *update* step in the lifting scheme. Different combinations of the prediction and update filters that are derived from splines, generate a wide family of biorthogonal symmetric wavelet transforms with diverse properties. In the following we specify how to use different types of splines for the design of filters.

**Continuous interpolatory splines:** There is a difference between using interpolatory splines of even and odd orders for prediction. A spline of order $p$ (degree $p-1$), which interpolates a polynomial of degree $p-1$, coincides identically with this polynomial. In particular, it exactly restores values of the polynomial at midpoints between the interpolation points. This property results in $p$ vanishing moments of the analysis wavelets. However, the interpolatory spline of odd order (even degree) with equidistant nodes possesses the so-called super-convergence property at the midpoints [55]. To be specific, if a spline of order $p = 2r-1$ interpolates a polynomial of degree $p = 2r-1$ on the grid $\{2kh\}$ then it predicts exactly the values of the polynomial at points $\{2kh+1\}$. Here $h$ is a step of the grid. Thus, the spline of order $p = 2r-1$ generates an analysis wavelet with the same number $2r$ of vanishing moments as a spline of order $p = 2r$. But the computational cost of the implementation of the filter derived from the spline of order $p = 2r-1$ is lower than the cost for computing with the spline of order $p = 2r$.

**Discrete interpolatory splines:** Another option is to use the discrete interpolatory splines [31]. Discrete splines are functions that are defined on $\mathbb{Z}$, which are the counterparts of polynomial splines. In this case, explicit formulas for the transforms that have any number of vanishing moments are established. Moreover, our investigation reveals an interesting relation between discrete splines and Butterworth filters, which are commonly used in signal processing [30]. The filter banks used in our scheme, comprise filters which act as bi-directional half-band Butterworth filters. The frequency response of a Butterworth filter is maximally flat and we succeed in the construction of dual

filters with a similar property.

One-directional causal Butterworth filters were used for devising orthogonal non-symmetric wavelets [25]. The computations there were conducted in time domain using recursive filtering. A scheme that used recursive filters for the construction of biorthogonal symmetric wavelets and their application to image processing was presented in [29, 33].

Unlike the construction in [21], the above transforms use pairs of causal and anticausal filters with infinite impulse response (IIR). Fortunately, the transfer functions of the employed filters are rational. Therefore, filtering can be performed recursively. Note that the application of a pair of causal and anticausal filters to a signal, however fast it may be, cannot be implemented in real-time. Therefore finite impulse response (FIR) filters are more suitable for real-time processing.

**Quasi-interpolatory splines:** There is a way to devise wavelet transforms that employ FIR filters whose properties are similar to the properties of the above mentioned interpolatory transforms. It can be done using the so-called local quasi-interpolatory polynomial splines [55]. Like the interpolatory splines, the quasi-interpolatory splines of odd order also possess the super-convergence property.

**Parametric splines:** Analysis of approximation properties of interpolatory and quasi-interpolatory splines enables us to devise parameterized sets of splines, that are used for the prediction. In particular, specific choices of the parameters result in increasing numbers of vanishing moments.

Lifting implementation of a wavelet transform of a signal is equivalent to processing the signal by a perfect reconstruction filter bank. This filter bank generates analysis and synthesis scaling functions which are solutions for the refinement equations [16]. These scaling functions are constructed via a cascade algorithm, which is closely related to subdivision schemes. We investigate convergence of the cascade algorithm and the regularity of the derived scaling functions and wavelets. For this purpose we employ methods that are developed in the theory of subdivision schemes [22, 23] for the schemes that employ FIR filters. The extension of the technique to schemes with IIR filters requires some modifications.

When the filter bank consists of FIR filters, the corresponding scaling functions are compactly supported. This is not the case for IIR filters. We prove that the scaling functions generated by filters with rational transfer functions decay exponentially as their arguments tend to infinity. Obviously this result is not surprising there are hints to this fact in [15, 25]. But the authors never saw a proof of this result. In some sense, a reciprocal fact was established in [17]. Under certain assumptions exponential decay of a refined function implies exponential decay of the refinement mask.

Note that IIR filters with rational transfer functions, which allow recursive implementation, appear in signal processing algorithms using spline functions. Construction and implementation of these filters was studied in [47, 48]. Our scheme that implements these filters is close to that of [48].

The rest of the paper is organized as follows. In the introductory Section 2 we outline the lifting scheme of wavelet transforms and discuss its relation to the conventional setting of wavelet transforms. Namely, we describe filter banks and bases of the space of signals, which originated from the prediction and update filters. In Section 3 we establish some necessary properties of polynomial splines and describe the derivation of the prediction filters from interpolatory and local quasi-interpolatory splines. We also construct parameterized sets of splines and corresponding prediction filters, which have rational and polynomial transfer functions. In Section 4 we design prediction filters using discrete splines and explain the relation of these filters to Butterworth filters. In Section 5 we indicate that slightly updated prediction filters can be employed as update filters. In Section 6 we discuss the implementation of filters with rational transfer functions and, in particular, application of recursive filters to finite-length signals. Section 7 is devoted to the analysis of the convergence of the cascade algorithm with IIR filter banks and of the properties of the corresponding scaling functions and wavelets. In particular, a theorem about the exponential decay of the scaling functions is proved. We prove that when the prediction filter originates from the polynomial interpolatory splines of even order $2r$, the corresponding synthesis scaling function coincides with the fundamental spline of order $2r$. From the superconvergence property, the scaling functions derived from splines of odd order are smoother than the splines themselves. In Section 8 we list a number of filters that were derived from splines

and, by combining these filters, we construct a number of biorthogonal wavelet transforms. We provide graphical illustrations and summarize the properties of these transforms. Then, in Section 9, we present image compression results after applying these transforms. In Appendix I (Section 10) we describe a direct 2D implementation of a transform, which uses FIR filters. The above theory was developed for the signals that belong to $l_1$. However a parallel theory can be developed for periodic signals using the discrete Fourier transform (DFT). The construction is carried out in the Fourier domain and calculations are performed via the fast Fourier transform (FFT). We outline briefly the periodic scheme in Appendix II (Section 11). An advantage of the periodic scheme lies in the fact that an increase in the order of the spline used for prediction, which leads to an increase in the number of vanishing moments, does not affect the computational complexity of the implementation. Therefore, periodic wavelets with any number of vanishing moments can be explicitly constructed.

# 2 Preliminaries: Biorthogonal wavelet transforms

In this section we outline known facts, which are needed for the construction of biorthogonal wavelet transforms.

We call the sequences $\mathbf{x} \triangleq \{x_k\},\ k \in \mathbb{Z}$, which belong to the space $l_1$, discrete-time signals. The $z$-transform of a signal $\mathbf{x}$ is defined as

$$X(z) \triangleq \sum_{k\in\mathbb{Z}} z^{-k}\, x_k.$$

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

$$x_k = \sum_{l\in\mathbb{Z}} b_{k-l} c_l \Longleftrightarrow X(z) = B(z)\, C(z) \tag{2.1}$$

$$X_e(z^2) \triangleq \sum_{k\in\mathbb{Z}} z^{-2k}\, x_{2k} = \frac{1}{2}\left(X(z) + X(-z)\right) \tag{2.2}$$

$$X_o(z^2) \triangleq \sum_{k\in\mathbb{Z}} z^{-2k}\, x_{2k+1} = \frac{z}{2}\left(X(z) - X(-z)\right) \tag{2.3}$$

$$X(z) = X_e(z^2) + z^{-1} X_o(z^2). \tag{2.4}$$

The input $x_n$ and the output $y_n$ of a linear discrete time shift-invariant system are linked

as

$$y_n = \sum_{k\in\mathbb{Z}} f_k x_{n-k}. \tag{2.5}$$

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

$$\widehat{X}(\omega) = \sum_{n\in\mathbb{Z}} e^{-i\omega n} x_n, \quad \widehat{Y}(\omega) = \sum_{n\in\mathbb{Z}} e^{-i\omega n} y_n, \quad \widehat{F}(\omega) = \sum_{n\in\mathbb{Z}} e^{-i\omega n} f_n$$

the discrete-time Fourier transforms of the sequences. The function $\widehat{F}(\omega)$ is called the frequency response of the digital filter. Then, we have from (2.5)

$$Y(z) = F(z)X(z), \text{ and } \widehat{Y}(\omega) = \widehat{F}(\omega)\widehat{X}(\omega).$$

## 2.1 Lifting scheme of the wavelet transform

We use for the construction and implementation of biorthogonal wavelet transforms the so-called lifting scheme, which was introduced by Sweldens [46]. The lifting scheme of a wavelet transform of a signal $\mathbf{x}$ can be implemented in either primal or dual mode. We outline both modes.

### 2.1.1 Primal decomposition

Generally, the primal lifting mode of the wavelet transform consists of four steps: 1. Split. 2. Predict. 3. Update or lifting. 4. Normalization.

**Split -** The array $\mathbf{x}$ is split into even and odd sub-arrays: $\mathbf{e^1} = \{e_k^1 = x_{2k}\}$, $\mathbf{o^1} = \{o_k^1 = x_{2k+1}\}$, $k \in \mathbb{Z}$.

**Predict -** The even array $\mathbf{e^1}$ is filtered by some filter $U(z)$, in order for the filtered version of $\mathbf{e^1}$ to predict the odd array $\mathbf{o^1}$. Then, the existing array $\mathbf{o^1}$ is replaced by the array $\mathbf{o_\nu^1}$, which is the difference between $\mathbf{o^1}$ and the predicted array. The filter $U(z)$ is called the *prediction* filter. In the $z$-domain the operations are described as follows:

$$O_\nu^1(z) = O^1(z) - U(z)E^1(z), \tag{2.6}$$

where $O^1_\nu(z),\ O^1(z),\ E^1(z)$ are the $z$-transforms of the signals $\mathbf{o^1},\ \mathbf{o^1_\nu},\ \mathbf{e^1}$, respectively. From now on the subscript $\nu$ designates the new array. We assume that the function $U(z)$ is regular at a certain vicinity of the unit circle $|z| = 1$ including the circle. In addition, we assume that $z^{-1}U(z^2)$ is a real-valued function as $|z| = 1$. If the filtered version of $\mathbf{e^1}$ well approximates $\mathbf{o^1}$ then, after this step, the signal is decorrelated.

**Update (lifting) -** Generally, downsampling the original signal $\mathbf{x}$ into $\mathbf{e^1}$ depletes the smoothness of the signal. To obtain a sparse signal similar to the original $\mathbf{x}$, the new odd array is filtered by an *update* filter, which we prefer to denote $V(z)/2$. The filtered array is used to increase the smoothness of the even array $\mathbf{e^1}$:

$$E^1_\nu(z) = E^1(z) + \frac{1}{2}V(z)\,O^1_\nu(z). \tag{2.7}$$

The assumption about the filter $V(z)$ is similar to the assumption about $U$: the function $V(z)$ must be regular at a certain vicinity of the unit circle $|z| = 1$ including the circle and $zV(z^2)$ must be a real-valued function as $|z| = 1$. Provided that the filter $V$ is properly chosen, the even array $\mathbf{e^1}$ is transformed into a smoothed and downsampled replica of $\mathbf{x}$.

**Normalization -** Finally, the smoothed array $\mathbf{s^1}$ and the array of details $\mathbf{d^1}$ are obtained by the following operation: $\mathbf{s^1} = \sqrt{2}\,\mathbf{e^1_\nu}, \quad \mathbf{d^1} = \mathbf{o^1_\nu}/\sqrt{2}$.

The key issue in the lifting scheme is the proper choice of the filters $U$ and $V$. We address this issue in subsequent sections.

### 2.1.2 Primal reconstruction

One of the most attractive features of lifting schemes is that the reconstruction of the signal $\mathbf{x}$ from the arrays $\mathbf{s^1}$ and $\mathbf{d^1}$ is implemented by the reverse decomposition:

**Undo Normalization -** $\mathbf{e^1_\nu} = \mathbf{s^1}/\sqrt{2} \quad \mathbf{o^1_\nu} = \sqrt{2}\,\mathbf{d^1}$.

**Undo Lifting -** The even array

$$E^1(z) = E^1_\nu(z) - \frac{1}{2}V(z)O^1_\nu(z) \tag{2.8}$$

is restored.

**Undo Predict -** The odd array

$$O^1(z) = O^1_\nu(z) + U(z)E^1(z) \tag{2.9}$$

is restored.

**Undo Split -** The last step is the standard restoration of the signal from its even and odd components. In the $z$-domain it appears as:

$$X(z) = E^1(z^2) + z^{-1}O^1(z^2). \tag{2.10}$$

### 2.1.3 Dual mode

In the above primal construction the update step followed the prediction. In some applications it is preferable to apply the update before the prediction and to control the prediction step. In particular, such a dual scheme provides an adaptive nonlinear wavelet transform [13] by choosing different predictors for different fragments of the signal. So, in the dual mode the *update* step precedes the *predict* step:

**Update**– The even array is averaged with the filtered odd array:
$E^1_\nu(z) = \left(E^1(z) + z^{-1}U(z)\,O^1(z)\right)/2.$

**Predict** – The odd array is predicted by the filtered new even array:
$O^1_\nu(z) = O^1(z) - zV(z)E^1_\nu(z).$

In the reconstruction the operations are reversed.

## 2.2 Filter banks

Let $\Phi(z) \triangleq (1 + z^{-1}\widehat{U}(z^2))/2$ and define the following filters

$$\tilde{G}(z) \triangleq \sqrt{2}z^{-1}\Phi(-z), \quad \tilde{H}(z) \triangleq \sqrt{2}(1 + zV(z^2)\Phi(-z)), \tag{2.11}$$

$$H(z) \triangleq \sqrt{2}\Phi(z), \quad G(z) \triangleq \sqrt{2}z^{-1}\left(1 - zV(z^2)\Phi(z)\right). \tag{2.12}$$

Here $\tilde{H}(z)$ and $\tilde{G}(z)$ are the low- and high-pass primal analysis filters, respectively, and $H(z)$ and $G(z)$ are the low- and high-pass primal synthesis filters, respectively. These four filters form a perfect reconstruction filter bank for any choice of the filters $U$ and $V$.

**Proposition 2.1** *If $z^{-1}U(z^2)$ and $zV(z^2)$ are real valued on the unit circle $z = e^{i\omega}$ then the decomposition and reconstruction equations can be represented as follows:*

$$\begin{aligned} S^1(z^2) &= \frac{1}{2}\Big(\overline{\tilde{H}(z)}\, X(z) + \overline{\tilde{H}(-z)}\, X(-z)\Big) && (2.13)\\ D^1(z^2) &= \frac{1}{2}\Big(\overline{\tilde{G}(z)}\, X(z) + \overline{\tilde{G}(-z)}\, X(-z)\Big) && (2.14)\\ X(z) &= H(z)S^1(z^2) + G(z)\, D^1(z^2), && (2.15) \end{aligned}$$

*where $S^1(z)$ and $D^1(z)$ are the z-transforms of the arrays $\mathbf{s^1}$ and $\mathbf{d^1}$, respectively.*

*In addition the perfect reconstruction property holds*

$$H(z)\,\overline{\tilde{H}(z)} + G(z)\,\overline{\tilde{G}(z)} = 2 \quad H(z)\,\overline{\tilde{H}(-z)} + G(z)\,\overline{\tilde{G}(-z)} = 0, \tag{2.16}$$

*and the transfer functions are linked to each other as follows:*

$$\tilde{G}(z) = z^{-1}H(-z); \quad G(z) = z^{-1}\,\tilde{H}(-z). \tag{2.17}$$

**Proof:** We start with the primal decomposition formula (2.14). We modify Eq. (2.6) using Eqs. (2.2) and (2.3). So, we have:

$$\begin{aligned} O^1_\nu(z^2) &= \frac{z}{2}\,\Big(X(z) - X(-z) - z^{-1}U(z^2)\Big(X(z) + X(-z)\Big)\Big)\\ &= \frac{z}{2}\,\Big(X(z)\Big(1 - z^{-1}U(z^2)\Big) - X(-z)\Big(1 + z^{-1}U(z^2)\Big)\Big)\,. \end{aligned} \tag{2.18}$$

To obtain (2.14), it is sufficient to note that the function $\tilde{G}(z) = z^{-1}(1 - z^{-1}U(z^2))/\sqrt{2}$ possesses the property $\overline{\tilde{G}(-z)} = -z\,\Big(1+z^{-1}U(z^2)\Big)/\sqrt{2}$ and $z^{-1}U(z^2)$ is a real-valued function as $|z| = 1$. Thus, we see that (2.18) is equivalent to (2.14).

To prove (2.13) we use the already proved relation (2.14). Then, the decomposition formula (2.7) can be rewritten as

$$\begin{aligned} E^1_\nu(z^2) &= \frac{1}{2}\Big(X(z) + X(-z)\Big) + \frac{V(z^2)}{4}\,\Big(\overline{\tilde{G}(z)}\, X(z) + \overline{\tilde{G}(-z)}\, X(-z)\Big)\\ &= \frac{1}{2}\,\Big(X(z)(1 + V(z^2)z\Phi(-z)) + X(-z)(1 + V(z^2)(-z)\Phi(z))\Big)\,. \end{aligned}$$

Hence, (2.13) follows.

To verify the reconstruction formula (2.15), we first rewrite Eq. (2.9) using Eq. (2.8),

$$\begin{aligned} O^1(z^2) &= O^1_\nu(z^2) + U(z^2)\left(E^1_\nu(z^2) - \frac{V(z^2)}{2}\, O^1_\nu(z^2)\right)\\ &= O^1_\nu(z^2)\left(1 - \frac{V(z^2)}{2}U(z^2)\right) + U(z^2)E^1_\nu(z^2). \end{aligned} \tag{2.19}$$

Then, we substitute (2.8) and (2.19) into (2.10).

From the definitions of the filters we immediately derive

$$H(z)\,\overline{\tilde{H}(z)} + G(z)\,\overline{\tilde{G}(z)} = 2(\Phi(z) + \Phi(-z)) = 2.$$

The second equation in (2.16) can be similarly checked. The relations (2.17) are apparent. ■

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

$$\tilde{H}^d(z) = H(z)/2,\ \tilde{G}^d(z) = G(z), \quad H^d(z) = 2\tilde{H}(z),\ G^d(z) = \tilde{G}(z). \tag{2.20}$$

## 2.3 Bases for signal space

The perfect reconstruction filter banks, described above, are associated with the biorthogonal pairs of bases in the space of discrete-time signals.

In section 2.2 we introduced a family of filters by their transfer functions $H(z)$, $G(z)$, $\tilde{H}(z)$, $\tilde{G}(z)$. Their impulse responses are $\{h_k\}$, $\{g_k\}$, $\{\tilde{h}_k\}$, $\{\tilde{g}_k\}$, $k \in \mathbb{Z}$, respectively. It means, for example, that $H(z) = \sum_{k\in\mathbb{Z}} z^{-k} h_k$. On the other hand, $\mathbf{h} \triangleq \{h_k\}$ is the signal, which emerges as a result of the application of the filter $H(z)$ to the impulse signal $\delta_n$ (the Kroneker delta). Similar relations hold for the other functions. The shifts of these impulse response signals form a biorthogonal pair of bases for the signal space.

**Proposition 2.2** *Any signal $\mathbf{x} \in l^1$ can be represented as follows:*

$$x_l = \sum_{k\in\mathbb{Z}} s_k^1\,\varphi_{l-2k}^1 + \sum_{k\in\mathbb{Z}} d_k^1\,\psi_{l-2k}^1,$$

*where $\varphi_k^1 \triangleq h_k,\ \psi_k \triangleq g_k,\ k \in \mathbb{Z}$.*

*The coordinates $s_k^1$ and $d_k^1$ are the following inner products:*

$$s_k^1 = \sum_{n\in\mathbb{Z}} \tilde{\varphi}_{n-2k}^1 x_n, \quad d_k^1 = \sum_{n\in\mathbb{Z}} \tilde{\psi}_{n-2k}^1 x_n, \tag{2.21}$$

*where $\tilde{\varphi}_k^1 \triangleq \tilde{h}_k,\ \tilde{\psi}_k^1 \triangleq \tilde{g}_k,\ k \in \mathbb{Z}$.*

**Proof:** We start with the reconstruction formula (2.15), which we rewrite as:

$$X(z) = X^h(z) + X^g(z) \text{ where } X^h(z) = H(z)S^1(z^2), \;\; X^g(z) = G(z)\, D^1(z^2). \tag{2.22}$$

We can write

$$\sum_{l\in\mathbb{Z}} z^{-l} x_l^h = \sum_{k,n\in\mathbb{Z}} z^{-n-2k} s_k^1 h_n = \sum_{k,n\in\mathbb{Z}} z^{-n-2k} s_k^1 \varphi_n^1 \Longleftrightarrow x_l^h = \sum_{k\in\mathbb{Z}} s_k^1\, \varphi_{l-2k}^1.$$

Similarly, we derive the relation

$$x_l^g = \sum_{k\in\mathbb{Z}} d_k^1\, \psi_{l-2k}^1.$$

Let us consider the decomposition formula (2.13). From Property (2.1) we conclude that $\overline{\tilde{H}(z)}\, X(z)$ is the $z$-transform of the sequence

$$a_k \triangleq \sum_{n\in\mathbb{Z}} \tilde{h}_{n-k} x_n = \sum_{n\in\mathbb{Z}} \tilde{\varphi}_{n-k}^1 x_n.$$

Now from (2.13) and (2.2) we have $s_k^1 = a_{2k} = \sum_{n\in\mathbb{Z}} \tilde{\varphi}_{n-2k}^1 x_n$, which proves the first equation in (2.21). The second equation is similarly proved. ■

Proposition 2.2 justifies the following definition.

**Definition 2.1** *The discrete time signals $\varphi^1 \triangleq \{\varphi_k^1\}$ and $\psi^1 \triangleq \{\psi_k^1\}$, $k \in \mathbb{Z}$, are called the low- and high-frequency synthesis discrete wavelets of the first scale, respectively. The signals $\tilde{\varphi}^1 \triangleq \{\tilde{\varphi}_k^1\}$ and $\tilde{\psi}^1 \triangleq \{\tilde{\psi}_k^1\}$, $k \in \mathbb{Z}$ are called the low- and high-frequency analysis discrete wavelets of the first scale, respectively.*

**Corollary 2.1** *The following biorthogonal relations hold:*

$$\sum_{n\in\mathbb{Z}} \tilde{\varphi}_{n-2k}^1\, \varphi_{n-2l}^1 = \sum_{n\in\mathbb{Z}} \psi_{n-2k}^1\, \tilde{\psi}_{n-2l}^1 = \delta_k^l, \quad \sum_{n\in\mathbb{Z}} \tilde{\varphi}_{n-2k}^1\, \psi_{n-2l}^1 = \sum_{n\in\mathbb{Z}} \tilde{\psi}_{n-2l}^1\, \varphi_{n-2k}^1 = 0,\ \forall l,\, k.$$

We say that a discrete wavelet $\psi$ has $m$ vanishing moments if the following relations hold: $\sum_{k\in\mathbb{Z}} k^s \psi_k = 0, \quad s = 0, 1, \ldots, m-1.$

**Proposition 2.3** *Let the transfer functions $U(z)$ and $V(z)$, which are used for the* predict *and* update *steps, respectively, be rational and have no poles on the unit circle $|z| = 1$. If $1 + z^{-1}U(z^2)$ comprises the factor $(z + 2 + z^{-1})^r$ then the high-frequency analysis wavelet $\tilde{\psi}^1$ has $2r$ vanishing moments. If the above condition is satisfied and, in addition, $1 + zV(z^2)$ comprises the factor $(z + 2 + z^{-1})^p$ then the high-frequency synthesis wavelet $\psi^1$ has $2s$ vanishing moments, where $s = \min(p, r)$.*

**Proof:** It follows from the conditions of the proposition that $1 - z^{-1}U(z^2)$ comprises the factor $(z - 2 + z^{-1})^r$. Let us ignore for a moment the assumption $|z| = 1$ and examine the function

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

of the complex variable $z$. We have $\tilde{q}(z) = z^{-1}(1-zU(z^{-2}))/\sqrt{2} = (1-z)^{2r}\,\widetilde{Q}(z)$, where $\widetilde{Q}(z)$ is a regular function at the vicinity of $z = 1$. It is clear that $\tilde{q}^{(s)}(1) = 0,\ s = 0, 1, \ldots, 2r-1$. On the other hand,

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

Note that any monomial $k^l$ can be represented as a linear combination of the polynomials $k^{[n]},\ n = 0, \ldots, l$. Thus, the wavelet $\tilde{\psi}^1$ has $2r$ vanishing moments. Denote $q(z) \triangleq G(z^{-1}) = \sum_{k\in\mathbb{Z}} z^k\psi^1_k$. If $1 - zV(z^2)$ comprises the factor $(z - 2 + z^{-1})^p$ then we have

$$\begin{aligned} q(z) &= z\sqrt{2}\left(1 - \frac{z^{-1}V(z^{-2})(1 + zU(z^{-2}))}{2}\right) \\ &= \frac{z}{\sqrt{2}}\Big((1 - z^{-1}V(z^{-2})) + (1 - zU(z^{-2})) + zU(z^{-2})(1 - z^{-1}V(z^{-2}))\Big) = (1-z)^{2s}\,Q(z), \end{aligned}$$

where $s = \min(p, r)$ and $Q(z)$ is a regular function in the vicinity of $z = 1$. Thus, the wavelet $\psi^1$ has $2s$ vanishing moments. ■

## 2.4 Multiscale wavelet transforms

We described in Sections 2.1 – 2.3 one step of a wavelet transform of a signal from a fine scale to a coarse one. Expansion of the transform to coarser scales is implemented in a recursive way. In this transform we store the array $\mathbf{d}^1$ and decompose the array $\mathbf{s}^1$. The transformed arrays $\mathbf{s}_2$ and $\mathbf{d}_2$ of the second scale are derived from the even and odd sub-arrays of the array $\mathbf{s}^1$ with the same lifting steps as those described in Section 2.1. As a result we get that the signal $\mathbf{x}$ is transformed into three subarrays: $\mathbf{x} \leftrightarrow \mathbf{d}^1 \bigcup \mathbf{d}_2 \bigcup \mathbf{s}_2$. The coefficients of the transform are linked as follows:

$$S^2(z^2) = \frac{1}{2}\Big(\overline{\tilde{H}(z)}\,S^1(z) + \overline{\tilde{H}(-z)}\,S^1(-z)\Big), \tag{2.23}$$

$$D^2(z^2) = \frac{1}{2}\Big(\overline{\tilde{G}(z)}\,S^1(z) + \overline{\tilde{G}(-z)}\,S^1(-z)\Big)$$

$$S^1(z) = H(z)S^2(z^2) + G(z)\,D^2(z^2). \tag{2.24}$$

Again, the transform leads to expansion of the signal with a biorthogonal pair of bases. We describe this re-expansion in more detail.

**Definition 2.2** *The discrete time signals*

$$\varphi_l^2 \triangleq \sum_{k\in\mathbb{Z}} h_k \varphi_{l-2k}^1, \quad \psi_l^2 \triangleq \sum_{k\in\mathbb{Z}} g_k \varphi_{l-2k}^1 \tag{2.25}$$

*are called the low- and high-frequency discrete synthesis wavelets of the second scale, respectively. The discrete time signals*

$$\widetilde{\varphi}_l^2 \triangleq \sum_{k\in\mathbb{Z}} \widetilde{h}_k \widetilde{\varphi}_{l-2k}^1, \quad \widetilde{\psi}_l^2 \triangleq \sum_{k\in\mathbb{Z}} \widetilde{g}_k \widetilde{\varphi}_{l-2k}^1 \tag{2.26}$$

*are called the low- and high-frequency discrete analysis wavelets of the second scale, respectively.*

**Proposition 2.4** *The signal* $\boldsymbol{x}$ *can be expanded as follows:*

$$x_l = \sum_{k\in\mathbb{Z}} s_k^2 \, \varphi_{l-4k}^2 + \sum_{k\in\mathbb{Z}} d_k^2 \, \psi_{l-4k}^2 + \sum_{k\in\mathbb{Z}} d_k^1 \, \psi_{l-2k}^1. \tag{2.27}$$

*The coordinates in (2.27) are the following inner products:*

$$s_k^2 = \sum_{n\in\mathbb{Z}} x_n \, \tilde{\varphi}_{n-4k}^2, \quad d_k^2 = \sum_{n\in\mathbb{Z}} x_n \, \tilde{\psi}_{n-4k}^2.$$

**Proof:** Using (2.24) and (2.22) we represent the $z$-transform of the signal $\mathbf{x}$ as follows:

$$\begin{aligned} &X(z) = X^{hh}(z) + X^{gh}(z) + X^{g}(z), \text{ where} \\ &X^{hh}(z) \triangleq H(z)H(z^2)S^2(z^4), \;\; X^{gh}(z) \triangleq H(z)G(z^2)\, D^2(z^4). \end{aligned}$$

We can write

$$\begin{aligned} &\sum_{l\in\mathbb{Z}} z^{-l} x_l^{hh} = \sum_{k,n,m\in\mathbb{Z}} z^{-n-2k-4m} s_m^2 h_k \varphi_n^1 = \sum_{l\in\mathbb{Z}} z^{-l} \sum_{m\in\mathbb{Z}} s_m^2 \sum_{k\in\mathbb{Z}} h_k \varphi_{l-2k-4m}^1 \\ &\Longleftrightarrow x_l^{hh} = \sum_{m\in\mathbb{Z}} s_m^2 \varphi_{l-4m}^2, \end{aligned}$$

and $\varphi_l^2$ is defined in (2.25). Similarly, we derive the relation $x_l^{gh} = \sum_{k\in\mathbb{Z}} d^2 \, \psi_{l-4k}^2$, where $\psi_l^2$ is defined in (2.25).

It is seen from (2.23) that $s_k^2 = b_{2k}$, where

$$b_k \triangleq \sum_{l\in\mathbb{Z}} \tilde{h}_{l-k} s_l^1 = \sum_{l\in\mathbb{Z}} \tilde{h}_{l-k} \sum_{n\in\mathbb{Z}} \tilde{\varphi}_{n-2l}^1 x_n$$
$$= \sum_{n\in\mathbb{Z}} x_n \sum_{l\in\mathbb{Z}} \tilde{h}_l \tilde{\varphi}_{n-2l-2k}^1 = \sum_{n\in\mathbb{Z}} x_n \tilde{\varphi}_{n-2k}^2,$$

where $\tilde{\varphi}_n^2$ is defined in (2.26). Formulas for $d_k^2$ are derived similarly. ■

The discrete wavelets related to the subsequent scales are defined iteratively via the two-scale equations

$$\varphi_l^{j+1} \triangleq \sum_{k\in\mathbb{Z}} h_k \varphi_{l-2k}^j, \quad \psi_l^{j+1} \triangleq \sum_{k\in\mathbb{Z}} g_k \varphi_{l-2k}^j$$
$$\widetilde{\varphi}_l^{j+1} \triangleq \sum_{k\in\mathbb{Z}} \widetilde{h}_k \widetilde{\varphi}_{l-2k}^j, \quad \widetilde{\psi}_l^{j+1} \triangleq \sum_{k\in\mathbb{Z}} \widetilde{g}_k \widetilde{\varphi}_{l-2k}^j.$$

# 3 Design of prediction filters using polynomial splines

We derive the prediction filters from splines in the following way: We construct a spline, which interpolates or quasi-interpolates even samples of a signal and predict odd samples as the values of the spline calculated at the midpoints between the points of (quasi-)interpolation. Analytically, this operation reduces to filtering the even array. We will show that splines of odd order are more suitable for this design due to the property of superconvergence of splines of odd order at the midpoints.

## 3.1 B-splines

We outline here some known properties of B-splines and establish some facts needed for our constructions.

The central B-spline of first order is the characteristic function of the interval $[-1/2, 1/2]$. The central B-spline of order $p$ is the convolution $M^p(x) = M^{p-1}(x) * M^1(x) \quad p \geq 2$. Note that the B-spline of order $p$ is supported on the interval $(-p/2, p/2)$. It is positive within its support and symmetric around zero. The B-spline $M^p$ consists of pieces of polynomials of degree $p-1$ that are linked to each other at the nodes such that $M^p \in C^{p-2}$. Nodes of B-splines of even order are located at points $\{k\}$ and of odd order, at points $\{k+1/2\},\ k \in \mathbb{Z}$.

The Fourier transform of the B-spline of order $p$ is

$$\widehat{M^p}(\omega) \triangleq \int_{-\infty}^{\infty} e^{-i\omega x} M^p(x)\, dx = \left(\frac{\sin \omega/2}{\omega/2}\right)^p. \tag{3.1}$$

We introduce two sequences which are important for further construction:

$$\mathbf{u}^p \triangleq \{M_k^p\}, \quad \mathbf{w}^p \triangleq \left\{M^p\left(k+\frac{1}{2}\right)\right\},\ k \in \mathbb{Z}. \tag{3.2}$$

Due to the compact support of B-splines, these sequences are finite. In Table 1 we present the sequences $\mathbf{u}^p$ and $\mathbf{w}^p$ for some values of $p$.

| $k$ | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|---|---|---|
| $u_k^2$ | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| $u_k^3 \times 8$ | 0 | 0 | 0 | 1 | 6 | 1 | 0 | 0 | 0 |
| $u_k^4 \times 6$ | 0 | 0 | 0 | 1 | 4 | 1 | 0 | 0 | 0 |
| $u_k^5 \times 384$ | 0 | 0 | 1 | 76 | 230 | 76 | 1 | 0 | 0 |
| $u_k^6 \times 120$ | 0 | 0 | 1 | 76 | 230 | 76 | 1 | 0 | 0 |
| $u_k^7 \times 46080$ | 0 | 1 | 722 | 10543 | 23548 | 10543 | 722 | 1 | 0 |
| $w_k^3 \times 2$ | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| $w_k^4 \times 48$ | 0 | 0 | 1 | 23 | 23 | 1 | 0 | 0 | 0 |
| $w_k^5 \times 24$ | 0 | 0 | 1 | 11 | 11 | 1 | 0 | 0 | 0 |
| $w_k^6 \times 3840$ | 0 | 1 | 237 | 1682 | 1682 | 237 | 1 | 0 | 0 |
| $w_k^7 \times 720$ | 0 | 1 | 57 | 302 | 302 | 57 | 1 | 0 | 0 |

Table 1: Values of the sequences $\mathbf{u}^p$ and $\mathbf{w}^p$ .

The discrete-time Fourier transforms of these sequences are

$$\begin{aligned}
\widehat{u^p}(\omega) &\triangleq \sum_{-\infty}^{\infty} e^{-i\omega k} M_k^p = P^p\left(\cos\frac{\omega}{2}\right), \\
\widehat{w^p}(\omega) &\triangleq \sum_{-\infty}^{\infty} e^{-i\omega k} M^p\left(k+\frac{1}{2}\right) = e^{\omega/2} Q^p\left(\cos\frac{\omega}{2}\right).
\end{aligned} \tag{3.3}$$

Here the functions $P^p$ and $Q^p$ are real-valued polynomials. If $p = 2r-1$ then $P^p$ is a polynomial of degree $2r-2$ and $Q^p$ is a polynomial of degree $2r-3$. If $p = 2r$ then $P^p$ is a polynomial of degree $2r-2$ and $Q^p$ is a polynomial of degree $2r-1$.

These polynomials were extensively studied in [39, 41]. In particular the following facts were established.

**Proposition 3.1 ([39])** *The polynomials $P^p$ and $Q^p$ can be derived via the recurrence relations*

$$\begin{aligned} Q^{p+1}(y) &= yP^p(y) + \frac{1-y^2}{p}\frac{dP^p(y)}{dy}, \quad P^1(y) \equiv 1, \\ P^{p+1}(y) &= yQ^p(y) + \frac{1-y^2}{p}\frac{dQ^p(y)}{dy}, \quad P^2(y) \equiv 1. \end{aligned}$$

The $z$-transforms of the sequences $\mathbf{u}^p$ and $\mathbf{w}^p$ are

$$u^p(z) = \sum_{-\infty}^{\infty} z^{-k} M_k^p = E_p^u(z), \quad w^p(z) = \sum_{-\infty}^{\infty} z^{-k} M^p(k+1/2) = E_p^w(z), \tag{3.4}$$

where $E_p^u(z)$ and $E_p^w(z)$ are the so-called Euler-Frobenius polynomials [41].

**Proposition 3.2 ([41])** *On the circle $z = e^{i\omega}$ the Laurent polynomials $E_p^u(z)$ are strictly positive. Their roots are all simple and negative. Each root $\gamma$ can be paired with a dual root $\widetilde{\gamma}$ such that $\gamma\,\widetilde{\gamma} = 1$. Thus, if $p = 2r-1$, $p = 2r$ then $u^p(z)$ can be represented as follows:*

$$u^p(z) = \prod_{n=1}^{r-1} \frac{1}{\gamma_n}(1+\gamma_n z)(1+\gamma_n z^{-1}), \;\; 0 < \gamma_1 < \gamma_1 < \ldots < \gamma_{r-1} < 1. \tag{3.5}$$

We introduce a rational function.

$$R^p(y) \triangleq \frac{Q^p(y)}{P^p(y)}. \tag{3.6}$$

**Examples:**

$$P^3(y) = (1+y^2)/2, \quad Q^3(y) = y, \; 1 - R^3(y) = \frac{(1-y)^2}{1+y^2}, \; 1 + R^3(y) = \frac{(1+y)^2}{1+y^2}.$$

$$\begin{aligned} P^4(y) &= \frac{1+2y^2}{3}, \quad Q^4(y) = \frac{5y+y^3}{6}, \\ 1 - R^4(y) &= \frac{(1-y)^2(-y+2)}{2(1+2y^2)}, \quad 1 + R^4(y) = \frac{(1+y)^2(y+2)}{2(1+2y^2)}. \end{aligned}$$

$$\begin{aligned} P^5(y) &= \frac{5+18y^2+y^4}{24}, \quad Q^5(y) = \frac{2y+y^3}{3}, \\ 1 - R^5(y) &= \frac{(1-y)^3(-y+5)}{5+18y^2+y^4}, \quad 1 + R^5(y) = \frac{(1+y)^3(y+5)}{5+18y^2+y^4}. \end{aligned}$$

**Properties of the rational functions $R^p$.** Using Eq. (3.1) we have

$$M^p(x-k) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\omega(x-k)}\left(\frac{\sin\omega/2}{\omega/2}\right)^p d\omega = \sum_{l=-\infty}^{\infty} e^{2\pi i l x}\int_0^1 e^{2\pi i v(x-k)}\frac{(\sin\pi v)^p(-1)^{lp}}{\pi(l+v))^p}\,dv$$

$$= \int_0^1 e^{-2\pi i v k} m_x^p(v)\,dv, \quad \text{where } m_x^p(v) \triangleq e^{2\pi i v x}(\sin\pi v)^p \sum_{l=-\infty}^{\infty} e^{2\pi i l x}\frac{(-1)^{lp}}{(\pi(l+v))^p}. \tag{3.7}$$

The relation (3.7) means that $M^p(x-k)$ is a Fourier coefficient of the 1-periodic function $m_x^p(v)$ and this function can be represented as a sum:

$$m_x^p(v) = \sum_{k=-\infty}^{\infty} e^{-2\pi i k v}\, M^p(x+k). \tag{3.8}$$

Equations (3.3) and (3.8) imply the following representations:

$$P^p(\cos\omega/2) = \widehat{u^p}(\omega) = m_0^p(\omega/2\pi) = (\sin\omega/2)^p \sum_{l=-\infty}^{\infty}\frac{(-1)^{lp}}{(\pi l+\omega/2)^p} \tag{3.9}$$

$$Q^p(\cos\omega/2) = e^{-i\omega/2}\widehat{w^p}(\omega) = e^{-i\omega/2} m_{\frac{1}{2}}^p(\omega/2\pi) = (\sin\omega/2)^p \sum_{l=-\infty}^{\infty}\frac{(-1)^{l(p+1)}}{(\pi l+\omega/2)^p}.$$

It is obvious from (3.9) that

$$P^p(1) = Q^p(1) = 1. \tag{3.10}$$

**Proposition 3.3** *If $p = 2r$ then*

$$1 + R^{2r}(\cos\omega/2) = \frac{2(\cos\omega/4)^{2r}P^{2r}(\cos\omega/4)^{2r}}{P^{2r}(\cos\omega/2)}.$$

**Proof:** From Eqs.(3.9), (3.10) we have

$$1 + R^{2r}(\cos\omega/2) = 1 + \frac{Q^{2r}(\cos\omega/2)}{P^{2r}(\cos\omega/2)} = \frac{2(\sin\omega/2)^{2r}\sum_{l=-\infty}^{\infty}(\pi 2l+\omega/2)^{-2r}}{P^{2r}(\cos\omega/2)}$$

$$= \frac{2(\cos\omega/4)^{2r}(\sin\omega/4)^{2r}\sum_{l=-\infty}^{\infty}(\pi l+\omega/4)^{-2r}}{P^{2r}(\cos\omega/2)} = \frac{2(\cos\omega/4)^{2r}P^{2r}(\cos\omega/4)}{P^{2r}(\cos\omega/2)}.$$

■

**Lemma 3.1** *If $p = 2r-1$ then at the vicinity of $\omega = 0$*

$$1 - R^{2r-1}(\cos\omega/2) = A_{2r-1}\left(\frac{\omega}{2}\right)^{2r} + O(\omega^{2r+2}), \quad A_{2r-1} \triangleq \frac{(4^r-1)}{r(2r-2)!}|b_{2r}|, \tag{3.11}$$

*where $b_s$ is the Bernoulli number of order $s$.*

**Proof:** From Eqs. (3.9), (3.10) we have

$$1 - R^{2r-1}(\cos\omega/2) = \frac{P^{2r-1}(\cos\omega/2) - Q^{2r-1}(\cos\omega/2)}{P^{2r-1}(\cos\omega/2)} = \frac{-2(\sin\omega/2)^{2r-1}T_{2r-1}(\omega)}{P^{2r-1}(\cos\omega/2)},$$

where $T_{2r-1}(\omega) \triangleq \sum_{l=-\infty}^{\infty} \frac{1}{(\pi(2l+1)+\omega/2)^{2r-1}}$.

The function $T_{2r-1}(\omega)$ is infinitely differentiable at the point $\omega = 0$ and its vicinity and the Taylor expansion holds

$$T_{2r-1}(\omega) = \sum_{n=0}^{\infty} \frac{T_{2r-1}^{(n)}(0)}{n!}\omega^n.$$

We can write

$$T_{2r-1}^{(n)}(0) = (-1)^n \sum_{l=-\infty}^{\infty} \frac{2^{2r-1}(2r-1)\dots(2r+n-2)}{(2\pi(2l+1))^{2r-1+n}}.$$

Hence, we see that

$$T_{2r-1}(0) = \sum_{l=-\infty}^{\infty} \frac{1}{(\pi(2l+1))^{2r-1}} = 0.$$

Similarly, $T_{2r-1}^{(2k)}(0) = 0\ \forall k \in \mathbb{N}$. This is not the case for the odd order derivatives:

$$T_{2r-1}^{(2k+1)}(0) = -\frac{(2r-1)\dots(2(r+k)-1)}{2^{2k+1}\pi^{2(r+k)}} \sum_{l=0}^{\infty} \frac{1}{(2l+1))^{2(r+k)}}.$$

Using a known formula [1]

$$\sum_{l=0}^{\infty} \frac{1}{(2l+1))^{2n}} = \frac{(2^{2n}-1)\pi^{2n}}{2(2n)!}|b_{2n}|,$$

we get

$$T_{2r-1}^{(2k+1)}(0) = -\frac{(2r-1)\dots(2(r+k)-1)}{2^{2k+1}} \frac{(2^{2(r+k)}-1)}{2(2(r+k))!}|b_{2(r+k)}|$$
$$= -\frac{(2^{2(r+k)}-1)}{2^{2(k+1)}(r+k)(2r-2)!}|b_{2(r+k)}|.$$

Finally,

$$T_{2r-1}(\omega) = \sum_{k=0}^{\infty} \frac{T_{2r-1}^{(2k+1)}(0)}{(2k+1)!}\omega^{(2k+1)} \tag{3.12}$$
$$= -\sum_{k=0}^{\infty} \frac{(2^{2(r+k)}-1)}{2^{2(k+1)}(r+k)(2r-2)!(2k+1)!}|b_{2(r+k)}|\omega^{(2k+1)}$$
$$= -\frac{(4^r-1)}{4^r(2r-2)!}|b_{2r}|\omega + O(\omega^3).$$

Note that the polynomial $P^{2r-1}(\cos\omega/2)$ of degree $2r-2$ can be represented as a polynomial of the argument $\sin^2(\omega/4)$. For $\omega = 0$, $P^{2r-1}(\cos\omega/2) = 1$, we have

$$P^{2r-1}(\cos\omega/2) = 1 + \varpi^{2r-1}\left(\sin^2\frac{\omega}{4}\right), \quad \varpi^{2r-1}\left(\sin^2\frac{\omega}{4}\right) \triangleq \sum_{n=1}^{2r-2} \alpha_n \sin^{2n}\frac{\omega}{4}. \tag{3.13}$$

Equations (3.12) and (3.13) imply (3.11).■

**Corollary 3.1** *If $p = 2r-1$ then the following factorization formulas hold*

$$1 - R^{2r-1}(\cos\omega/2) = (\sin^{2r}\omega/4)\rho_r(\omega), \quad \rho_r(\omega) = \frac{p_{r-2}(\sin^2\omega/4)}{1 + \varpi^{2r-1}(\sin^2\omega/4)}, \tag{3.14}$$

*where $p_{r-2}$ are polynomials of degree $r-2$. The rational functions $\rho_r$ can be represented as*

$$\rho_r(\omega) = 4^r A_{2r-1} + \sum_{n=1}^{\infty} \gamma_n \sin^{2n}\frac{\omega}{4}. \tag{3.15}$$

*The coefficients $\gamma_n$ in (3.15) are such that the series converges absolutely for all real $\omega$.*

This claim follows from Lemma 3.1. It follows from Proposition 3.2 that in the representation (3.13) the absolute value of the polynomial $\varpi^{2r-1}(\sin^2\omega/4)$ remains less than 1 for all real $\omega$. This implies the absolute convergence of the series in (3.15).

## 3.2 Interpolatory splines

Shifts of B-splines form a basis in the space of splines of order $p$ on the grid $\{2k\}$. Namely, any spline $S^p$ has the following representation:

$$S^p(x) = \sum_l q_l\, M^p(x/2 - l). \tag{3.16}$$

Denote $\mathbf{q} \triangleq \{q_l\}$ and let $Q(z)$ be the $z$-transform of $\mathbf{q}$. We also introduce the sequences $\mathbf{s}_e^p \triangleq \{s_{e,k}^p \triangleq S^p(2k)\}$, $\mathbf{s}_o^p \triangleq \{s^{p,k} \triangleq S^p(2k+1)\}$ and $\mathbf{s}^p = \{s_k^p \triangleq S^p(k)\}$ of values of the spline on the grid points, on the midpoints and on the whole set $\{k\}$. The $z$-transform of the sequence $\mathbf{s}^p$ is

$$s^p(z) = s_e^p(z^2) + z^{-1}s_o^p(z^2). \tag{3.17}$$

We have

$$s_{e,k}^p = \sum_l q_l\, M^p(k-l), \quad s_{o,k}^p = \sum_l q_l\, M^p\left(k - l + \frac{1}{2}\right).$$

Thus, $s_e^p(z) = Q(z)u^p(z)$ and $s_o^p(z) = Q(z)w^p(z)$, where $u^p(z)$ and $w^p(z)$ are the functions defined in (3.4). From these formulas we can derive expressions for the coefficients of a spline $S_i^p$ which interpolates a given sequence $\mathbf{e} \triangleq \{e_k\} \in l^1$ at grid points:

$$\begin{aligned} &S_i^p(2k) = e_k,\ k \in \mathbb{Z}, \Longleftrightarrow Q(z)u^p(z) = e(z) \\ &\Longleftrightarrow Q(z) = \frac{e(z)}{u^p(z)} \Longleftrightarrow q_l = \sum_{n=-\infty}^{\infty} \lambda^p(l-n)e(n), \end{aligned} \tag{3.18}$$

where $\lambda^p \triangleq \{\lambda_k^p\}$ is the sequence which is defined via its $z-$transform:

$$\lambda^p(z) = \sum_{k=-\infty}^{\infty} z^{-k}\lambda_k^p = \frac{1}{u^p(z)}. \tag{3.19}$$

It follows immediately from (3.5) that the coefficients $\{\lambda_k^p\}$ decay exponentially as $|k| \to \infty$. Substitution of (3.18) into (3.16) results in an alternative representation of the interpolatory spline:

$$S_i^p(x) = \sum_{l=-\infty}^{\infty} e_l\, L^p(x/2 - l), \qquad \text{where } L^p(x) \triangleq \sum_l \lambda_l^p\, M^p(x-l). \tag{3.20}$$

The spline $L^p(x)$ defined in (3.20) is called the fundamental spline. It interpolates the Kroneker delta sequence $\delta_k$, i.e. it vanishes at all the integer points except $x = 0$, where $L^p(0) = 1$. By decaying the coefficients $\{\lambda_k^p\}$, the spline $L^p(x)$ decays exponentially as $|x| \to \infty$. Therefore, the representation (3.20) of the interpolatory spline remains valid for the sequences $\{e_k\}$, which may grow no faster than a power of $k$ [42]. The values of the fundamental spline at midpoints are

$$L^p\left(k + \frac{1}{2}\right) = \sum_l \lambda_l^p\, M^p\left(k - l + \frac{1}{2}\right). \tag{3.21}$$

Denote by $U_i^p(z)$ the $z$-transform of the sequence $\{L^p\,(k+1/2)\},\ k \in \mathbb{Z}$. Then, we obtain from Eqs. (3.19) and (3.21)

$$U_i^p(z) = \frac{w^p(z)}{u^p(z)}. \tag{3.22}$$

Hence the values of the interpolatory spline at the midpoints are

$$s_{o,k}^p = \sum_n L^p\left(k + \frac{1}{2} - n\right) e_n \Longleftrightarrow s_o^p(z) = U_i^p(z)e(z). \tag{3.23}$$

Switching into the signal processing terminology, we say that, in order to derive the values of the interpolatory spline at the midpoints $\{2k+1\}$ between the points $\{2k\}$ of interpolation, we have to filter the data $\{e_k\}$ by the filter $U_i^p$ whose impulse response $\{L^p(k+1/2)\},\ k \in \mathbb{Z}$ is infinite but decays exponentially as $|k| \to \infty$ (IIR filter). As we mentioned above, if a spline interpolates even samples of a signal the values of the spline at the midpoints are used as a prediction of the odd samples of the signal. Thus, the filters $U_i^p$ may serve as prediction filters in the lifting scheme of a wavelet transform.

**Properties of the rational functions $U_i^p(z)$.** If we substitute $z = e^{i\omega}$ then we see that $z^{-1}U_i^p(z^2) = R^p(\cos(\omega)$ (the function $R^p$ was defined in (3.6)). Thus, the properties of the functions $U_i^p(z)$ follow immediately from the properties of $R^p$ that were established in Section 3.1. We outline these properties.

**Proposition 3.4** **1.** *The function $z^{-1}U_i^p(z^2)$ is real valued on the unit circle $z = e^{i\omega}$.*

**2.** *If $p = 2r$ then*

$$1 + z^{-1}U^{2r}(z^2) = \frac{(1+z)^{2r}u_{2r}(z)}{2^{2r-1}z^r u_{2r}(z^2)}, \quad 1 - z^{-1}U^{2r}(z^2) = \frac{(z-2+z^{-1})^r u_{2r}(z)}{2^{2r-1}u_{2r}(z^2)}. \quad (3.24)$$

**3.** *If $p = 2r-1$ then the following factorization formulas hold*

$$1 - z^{-1}U^{2r-1}(z^2)(z) = (z-2+z^{-1})^r \vartheta_r(z) \quad \vartheta_{2r-1}(z) = \frac{q_{r-2}(z-2+z^{-1})}{u^{2r-1}(z^2)}, \quad (3.25)$$

*where $q_{r-2}$ are polynomials of degree $r-2$. The rational function $\vartheta_{2r-1}$ can be represented as*

$$\vartheta_{2r-1}(z) = A_{2r-1} + \sum_{n=1}^{\infty} \epsilon_n (z-2+z^{-1})^n, \ \textit{where } A_{2r-1} \triangleq \frac{(4^r-1)}{r(2r-2)!}|b_{2r}|, \quad (3.26)$$

*$b_s$ is the Bernoulli number of order $s$ and the coefficients $\{\epsilon_n\}$ in (3.26) are such that the series absolutely converges for all $z$, $|z| = 1$.*

Combining these results with Proposition 2.3, we come to the following corollary.

**Corollary 3.2** *If the filter $U_i^p$ is used as a prediction filter in the lifting scheme then, either by $p = 2r-1$ or by $p = 2r$, the high-frequency analysis wavelet $\tilde{\psi}^1$ has $2r$ vanishing moments.*

**Super-convergence property.** Proposition 3.4 implies the super-convergence property of the interpolatory splines of odd order (even degree). Recall, that in general the interpolatory spline of order $p$ (degree $p-1$), which interpolates the values of a polynomial of degree not exceeding $p-1$, coincides with this polynomial (in other words, the spline is exact on polynomials of degree not exceeding $p-1$). However, we will show that the spline $S_i^{2r-1}$ of odd order $2r-1$ (degree $2r-2$), which interpolates the values of a polynomial of degree $2r-1$ on the grid $\{2k\}$ restores the values of this polynomial at the mid-points $\{2k+1)\}$ between the grid points. Therefore, we claim that $\{2k+1)\}$ are points of super-convergence of the spline $S_i^{2r-1}$. This property results in the vanishing moments property of the wavelets constructed using the filters $U_i^{2r-1}$.

We denote by $\mathbf{D}^2$ the operator of central second difference: $\mathbf{D}^2 f_k = f_{k-1} - 2f_k + f_{k+1}$. It is applied to a signal via filtering with the transfer function $D^2(z) = z - 2 + z^{-1}$. The Laurent polynomial $D^{2r}(z) = (z-2+z^{-1})^r$ corresponds to the difference of order $2r$, which is denoted as $\mathbf{D}^{2r}$.

**Proposition 3.5** *Let a function $f(x)$, which may grow no faster than a power of $x$ as $|x| \to \infty$, be sampled on the grid $\{k\}$ and $\boldsymbol{f} \triangleq \{f_k\}, k \in \mathbb{Z}$. Let the spline $S_i^{2r-1}$ of order $2r-1$ interpolate $\boldsymbol{f}$ on the grid $\{2k\}$. Then*

$$S_i^{2r-1}(2k+1) = f(2k+1) - A_{2r-1}\boldsymbol{D}^{2r} f(2k+1) - \sum_{n=1}^{\infty} \epsilon_n \boldsymbol{D}^{2(r+n)} f(2k+1),$$

$$\text{where } A_{2r-1} \triangleq \frac{(4^r-1)}{r(2r-2)!}|b_{2r}|,$$

*$b_s$ is the Bernoulli number of order $s$ and the set $\{\epsilon_n\}$ of coefficients guarantees the absolute convergence of the series.*

**Proof:** The difference between the odd subarray $\{f(2k+1)\}$ and the array $\{s_{o,k}^{2r-1}\}$ of values of the spline at the midpoints is

$$f(2k+1) - s_{o,k}^{2r-1} = f(2k+1) - \sum_n L^p\left(k + \frac{1}{2} - n\right) f(2n) \tag{3.27}$$

$$= \sum_n a_{2k+1-n}^p f(n), \quad \text{where } a_k^p = \begin{cases} \delta_l, & \text{if } k = 2l \\ -L^p(l+1/2), & \text{if } k = 2l+1. \end{cases}$$

Equation (3.27) means that to obtain the difference $f(2k+1) - s_o^p(k)$ we must apply the filter $\mathbf{a}^p \triangleq \{a_r^p\}$ to the array $\mathbf{f}$ and take odd samples of the generated array. The $z$-transform of the filter $\mathbf{a}^{2r-1}$ is $A^{2r-1}(z) = 1 - z^{-1}U_i^{2r-1}(z^2)$. Due to (3.25) and (3.26), the function $A^{2r-1}(z)$ is factorized into $A^{2r-1}(z) = (z - 2 + z^{-1})^r \vartheta^{2r-1}(z)$, where

$$\vartheta_{2r-1}(z) = A_{2r-1} + \sum_{n=1}^{\infty} \epsilon_n (z - 2 + z^{-1})^n.$$

The series converges absolutely as $|z| = 1$. Hence, the assertion of the proposition follows. ■

**Corollary 3.3** *Let the function $f$ be a polynomial of degree $2r-1$. Then, $S_i^{2r-1}(2k+1) \equiv f(2k+1)\ \forall k \in \mathbb{Z}$. If $f(x)$ is a polynomial of degree $2r+1$ then*

$$S_i^{2r-1}(2k+1) = f(2k+1) - A_{2r-1}\boldsymbol{D}^{2r} f(2k+1) = f(2k+1) - A_{2r-1} f^{(2r)}(2k+1). \quad (3.28)$$

Originally this property was proved differently in [55].

**Remark.** It can be easily derived from (3.24) that the splines of order $2r$ restore only polynomials of degree $2r-1$ at the midpoints between the points of interpolation.

**Examples:**

**Interpolatory quadratic spline:** $p = 3$

$$U_i^3(z) = 4\frac{1+z}{z+6+z^{-1}}, \quad 1 + z^{-1}U_i^3(z^2) = \frac{(z+2+z^{-1})^2}{z^{-2}+6+z^2}. \quad (3.29)$$

The corresponding analysis wavelet has four vanishing moments.

**Interpolatory cubic polynomial spline:** $p = 4$

$$U_i^4(z) = \frac{(z+1)(z^{-1}+22+z)}{8(z+4+z^{-1})}, \quad 1 + z^{-1}U_i^4(z^2) = \frac{(z^{-1}+2+z)^2(z+4+z^{-1})}{8(z^{-2}+4+z^2)}.$$

The corresponding analysis wavelet has four vanishing moments.

**Interpolatory spline of fourth degree:** $p = 5$

$$U_i^5(z) = \frac{16(z+10+z^{-1})(1+z)}{z^2+76z+230+76z^{-1}+z^{-2}}, \quad (3.30)$$
$$1 + z^{-1}U_i^5(z^2) = \frac{(z+2+z^{-1})^3\,(z+z^{-1}-10)}{z^4+76z^2+230+76z^{-2}+z^{-4}}.$$

The corresponding analysis wavelet has six vanishing moments.

**Remark:** Due to the superconvergence property of the splines of odd order, the wavelets originating from the interpolatory splines of odd order $2r-1$ and of even order $2r$ have the same number of vanishing moments. But the structure of the filter $U_i^{2r-1}$ is simpler than the structure of $U_i^{2r}$ (compare, for example, $U_i^3(z)$ with $U_i^4(z)$). This means that the computational cost of implementation of the wavelet transform with the filter $U_i^{2r-1}$ is lower than with the filter $U_i^{2r}$. Moreover, the wavelets originating from $U_i^{2r-1}$ are better localized in time domain. Therefore, for most applications it is preferable to use filters derived from the odd-order splines.

## 3.3 Local quasi-interpolatory splines

We can see from Eq. (3.23) that in order to find the values at the midpoints of the spline that interpolates the array $\mathbf{e}$, the array has to be filtered with the filter whose transfer function is $U_i^p(z)$. This filter has infinite impulse response (IIR). Further on we discuss a fast implementation of this operation via recursive filtering, which is commonly used in signal processing. But this procedure can not be used in real-time processing. Also the recursive filtering is not appropriate when the length of a signal to be processed is large. In these situations finite impulse response (FIR) filters are better suited. In this section we describe the construction of FIR prediction filters on the base of the so-called local quasi-interpolatory (LQI) splines. Their properties are similar to the properties of filters derived from interpolatory splines. Even the property of super-convergence at midpoints remains valid for the quasi-interpolatory splines of odd order.

**Definition 3.1** *Let a function $f$ be sampled on the grid $\{k\}$, $\mathbf{f} \stackrel{\Delta}{=} \{f_k\}$, and $F(z)$ be the $z$-transform of $\mathbf{f}$. A spline $S^p$ of order $p$ represented in the form (3.16) is called local if the array $\mathbf{q}$ of its coefficients is derived by FIR filtering of the array $\mathbf{f}$:*

$$Q(z) = \Gamma(z)F(z), \tag{3.31}$$

*where $\Gamma(z)$ is a Laurent polynomial. The local spline of order $p$ is called quasi-interpolatory (LQI) if it is exact on polynomials of degree not exceeding $p-1$.*

It means that if $f$ is a polynomial of degree $p-1$, then the spline $S^p(x) \equiv f(x)$.

To predict the odd samples we use the values at the midpoints of the splines, which quasi-interpolate the even samples. If $\mathbf{w}^p$ is the sequence defined in (3.4) then the FIR prediction filter, which stems from an LQI spline of order $p$, is

$$U_q^p(z) \triangleq \Gamma(z) w^p(z). \tag{3.32}$$

Explicit formulas for the construction of quasi-interpolatory splines were given in [39]. Properties of quasi-interpolatory splines and, in particular, their approximation accuracy were studied in [55]. In this work we are interested in splines of odd order $p = 2r - 1$. There are many FIR filters, which generate quasi-interpolatory splines, but there is only one filter whose impulse response has minimal length $2r - 1$ for each order $p = 2r - 1$.

**Proposition 3.6 ([39])** *An LQI spline of order $p = 2r - 1$ can be produced by filtering (3.31) with filters $\Gamma$ of length no less than $2r - 1$. There exists a unique filter $\Gamma_m$ whose impulse response is of length $2r - 1$, which produces the minimal quasi-interpolatory spline $\tilde{S}_m^{2r-1}(x)$. Its transfer function is:*

$$\Gamma_m(z) = 1 + \sum_{k=1}^{r-1} \beta_k^{2r-1} \, (z^{-1} - 2 + z)^k, \tag{3.33}$$

*where the coefficients $\beta_k^{2r-1}$ are derived from the generating function*

$$\left( \frac{2 \arcsin t/2}{t} \right)^p = \sum_{k=0}^{\infty} (-1)^k \beta_k^p t^{2k}.$$

**Proposition 3.7 ([55])** *If $f$ is a polynomial of degree $2r$ and $\mathbf{f} \triangleq \{f(2k)\}$ then the following relation holds for the minimal quasi-interpolatory spline $S_m^{2r+1}$ of order $2r-1$ (degree $2r-2$) for $t = (2k + 1 + \tau)$, $\tau \in [0, 1]$:*

$$S_m^{2r-1}(t) = f(t) - 2^{2r-1} f^{(2r-1)}(t) \frac{b_{2r-1}(\tau)}{(2r-1)!} + 2^{2r} f^{(2r)} \cdot \left( \frac{(2r-1) b_{2r}(\tau)}{(2r)!} - \beta_r^{2r-1} \right),$$

*where $b_s(\tau)$ is the Bernoulli polynomial of degree $s$.*

We recall that the values $b_s \triangleq b_s(0)$ are called the Bernoulli numbers [1]. If $s > 1$ is odd then $b_s = 0$. Hence, we obtain the super-convergence property, which is similar to the property of interpolatory splines of odd order.

**Corollary 3.4** *If $f$ is a polynomial of degree $2r-1$ , $\mathbf{f} \triangleq \{f(2k)\}$ and $S_m^{2r-1}$ is a minimal LQI spline of order $2r-1$ (degree $2r-2$ ) then $S_m^{2r-1}(2k+1) \equiv f(2k+1),\ k \in \mathbb{Z}$. If $f$ is a polynomial of degree $2r$ then*

$$S_m^{2r-1}(2k+1) = f(2k+1) + 2^{2r} f^{(2r)} B_{2r-1}, \quad B_{2r-1} \triangleq \frac{(2r-1)b_{2r}}{(2r)!} - \beta_r^{2r-1}. \tag{3.34}$$

## 3.4 Parametric splines

### 3.4.1 Upgrade of LQI splines

The representation (3.34) provides tools for custom design of predicting splines that retain or even enhance the approximation accuracy of the minimal LQI spline at the midpoints. This is achieved via an upgrade of the filter $\Gamma_m$.

**Proposition 3.8** *Let $f$ be a polynomial of degree $2r-1$ and $\mathbf{f} \triangleq \{f(2k)\}$. If the coefficients of a spline $S_\rho^{2r-1}$ of order $2r-1$ are derived using Eq. (3.31) and the filter $\Gamma_\rho$ of length $2r+1$ with the transfer function*

$$\Gamma_\rho(z) = \Gamma_m(z) + \rho(z^{-1} - 2 + z)^r \tag{3.35}$$

*then $S_\rho^{2r-1}(2k+1) \equiv f(2k+1),\ k \in \mathbb{Z}$ by any real $\rho$. However, if $\rho = -B_{2r-1}$ then the identity $S_\rho^{2r-1}(2k+1) \equiv f(2k+1)$ remains valid when $f$ is a polynomial of degree $2r$.*

**Proof:** The spline $S_\rho^{2r-1}$, which is constructed using the filter $\Gamma_\rho$, can be represented by the sum:

$$S_\rho^{2r-1} = S_m^{2r-1} + \rho S_d^{2r-1}, \quad S_d^{2r-1}(t) \triangleq \sum_{l \in \mathbb{Z}} \left(\mathbf{D}^{2r} f(2l)\right) M^{2r-1}(t/2 - l). \tag{3.36}$$

We evaluate the spline $S_d^{2r-1}$ using the well-known asymptotic relation for a function $\phi \in C^3$ ([40]):

$$\sum_l \phi(hl)\, M^p(t/h - l) = \phi(t) + \frac{ph^2\phi''(t)}{24} + O(h^3\phi^{(3)}). \tag{3.37}$$

Hence, if $f$ is a polynomial of degree $2r$ then we have $S_d^{2r-1}(t) = \mathbf{D}^{2r} f = 2^{2r} f^{(2r)}$ and

$$S_\rho^{2r-1}(t) = f(t) - 2^{2r-1} f^{(2r-1)}(t) \frac{b_{2r-1}(\tau)}{(2r-1)!} + 2^{2r} f^{(2r)} \left( \frac{(2r-1)b_{2r}(\tau)}{(2r)!} - \beta_r^{2r-1} + \rho \right),$$

At midpoints we have

$$S_m^{2r-1}(2k+1) = f(2k+1) + 2^{2r} f^{(2r)} \cdot (B_{2r-1} + \rho). \tag{3.38}$$

The equation (3.38) proves the proposition. ■

If the parameter $\rho$ is chosen such that $\rho = (-1)^{r-1}|\rho|$ then the spline $S_\rho^{2r-1}$ possesses the smoothing property [56]. In the case when $\rho = -B^r$ we call the spline the extended LQI spline and denote it as $S_e^{2r-1}$.

We recall that, given a filter $\Gamma(z)$, the prediction filter is derived from Eq. (3.32).

**Examples:**

**Minimal LQI quadratic spline:** The filters are

$$\begin{aligned} &\Gamma_m(z) = 1 - \frac{z^{-1} - 2 + z}{8},\ U_m^3(z) = \frac{-z^{-1} + 9 + 9z - z^2}{16}, \\ &1 - z^{-1} U_m^3(z^2) = \frac{(z^{-1} - 2 + z)^2 (z^{-1} + 4 + z)}{16}. \end{aligned} \tag{3.39}$$

The corresponding analysis wavelet has four vanishing moments.

**Extended LQI quadratic spline:**

$$\begin{aligned} &\Gamma_e(z) = \Gamma_m(z) + \frac{3}{128}(z^{-1} - 2 + z)^2, \\ &U_e^3(z) = \frac{3z^{-2} - 25z^{-1} + 150 + 150z - 25z^2 + 3z^3}{256}, \\ &1 + z^{-1} U_e^3(z^2) = \frac{(z^{-1} + 2 + z)^3 (3z^{-2} + 18z^{-1} + 38 + 18z + 3z^2)}{256}. \end{aligned} \tag{3.40}$$

The corresponding analysis wavelet has six vanishing moments.

**Remark:** Donoho [21] presented a scheme where an odd sample is predicted by the value at the central point of a polynomial of odd degree which interpolates adjacent even samples. One can observe that our filter $U_m^3$ (3.39) coincides with the filter derived by Donoho's scheme using a cubic interpolatory polynomial. The filter $U_e^3$ (3.40) coincides with the filter derived using an interpolatory polynomial of fifth degree. Note that in Donoho's construction the *update* step does not exist. On the other hand,

the filters $1+z^{-1}U_m^3(z^2)$ and $1+z^{-1}U_e^3(z^2)$ are the autocorrelations of 4-tap and 6-tap filters by Daubechies, respectively [37].

**Minimal LQI spline of fifth order (fourth degree):**

$$\begin{aligned}
&\Gamma_m(z) = 1 - \frac{5}{24}(z^{-1}-2+z) + \frac{47}{1152}(z^{-1}-2+z)^2,\\
&U_m^5(z) = \frac{47(z^{-3}+z^4)+89(z^{-2}+z^3)-2277(z^{-1}+z^2)+15965(1+z)}{27648}\\
&1-z^{-1}U_m^5(z^2) = \frac{-(z^{-1}-2+z)^3}{27648}\\
&\times\Big(47(z^{-4}+z^4)+282(z^{-3}+z^3)+1076(z^{-2}+z^2)+3166(z^{-1}+z)+5414\Big).
\end{aligned} \tag{3.41}$$

Like the quadratic extended spline the minimal spline of fourth degree produces an analysis wavelet with six vanishing moments. But the computational cost of the implementation of the transform with the filter $U_m^5$ is higher than that with the filter $U_e^3$.

### 3.4.2 Upgrade of interpolatory splines

We can use a similar approach for increasing the approximation accuracy of interpolatory splines of odd order at midpoints. It can be done in two ways. Let $S_i^{2r-1}$ be a spline of order $2r-1$, which interpolates a sequence $\{e_k\}$ on the grid $\{2k\}$.

**Upgrade of the numerator of the transfer function:** We introduce a new spline $S_\rho^{2r-1}$ as follows:

$$\widetilde{S}_\rho^{2r-1} = S_i^{2r-1} + \rho S_d^{2r-1}, \quad S_d^{2r-1} \triangleq \sum_l (\mathbf{D}^{2r}e_l)\, M^{2r-1}(t/2-l). \tag{3.42}$$

Then an assertion similar to Proposition 3.8 holds.

**Proposition 3.9** *Let $f$ be a polynomial of degree $2r-1$ and $e_k \triangleq \{f(2k)\},\ k\in\mathbb{Z}$. Then for any real $\rho$ the spline*

$$\widetilde{S}_\rho^{2r-1}(2k+1) \equiv f(2k+1), \quad \widetilde{S}_\rho^{2r-1}(2k) \equiv f(2k),\ k\in\mathbb{Z}.$$

*However, if $f$ is a polynomial of degree $2r+1$ and $\rho = 4^{-r}A_{2r-1}$, where $A_{2r-1}$ is defined in (3.11), then the spline restores values of $f$ at the midpoints $\{2k+1\},\ k\in\mathbb{Z}$.*

We denote the spline with $\rho = 4^{-r}A_{2r-1}$ by $\widetilde{S}_A^{2r-1}$. As before, we use values of the spline at midpoints as prediction of the odd subarray of the processed signal. Using (3.22) and (3.4), we get that the corresponding prediction filter is

$$\begin{aligned}\widetilde{U}_A^{2r-1}(z) &= U_i^{2r-1}(z) + 4^{-r}A_{2r-1}w^{2r-1}(z)(z-2+z^{-1})^r \\ &= \frac{w^{2r-1}(z)(1+4^{-r}A_{2r-1}\,u^{2r-1}(z)\,(z-2+z^{-1})^r)}{u^{2r-1}(z)}.\end{aligned}$$

By this means we update the numerator of the rational transfer function of the filter $U_i^{2r-1}(z)$. In other words, we change the FIR component of the filter.

**Upgrade of the denominator of the transfer function:** Another way consists of modifying the denominator rather than the numerator of the transfer function. As is seen from (3.5), the Laurent polynomial $u^{2r-1}(z)$ can be represented as the sum $u^{2r-1}(z) = 1 + \sum_{k=1}^{r-1}\alpha_k(z-2+z^{-1})^k$.

We introduce a new Laurent polynomial

$$u_\rho^{2r-1}(z) \triangleq 1+\sum_{k=1}^{r-1}\alpha_k(z-2+z^{-1})^k+\rho(z-2+z^{-1})^r = u^{2r-1}(z)+\rho(z-2+z^{-1})^r \quad (3.43)$$

and the rational function

$$\check{U}_\rho^{2r-1}(z) \triangleq \frac{w^{2r-1}(z)}{u_\rho^{2r-1}(z)}. \quad (3.44)$$

Similarly to the construction of the interpolatory splines we define the quasi-fundamental spline

$$L_\rho^{2r-1}(x) \triangleq \sum_l \lambda_{\rho,l}^{2r-1}\,M^{2r-1}(x-l), \quad \sum_{k=-\infty}^{\infty} z^{-k}\lambda_{\rho,k}^{2r-1} = \frac{1}{u_\rho^{2r-1}(z)}. \quad (3.45)$$

Given a sequence $\{e_k\}$, we construct the spline

$$\check{S}_\rho^{2r-1}(t) = \sum_{l=-\infty}^{\infty} e_l\,L_\rho^{2r-1}(t/2-l). \quad (3.46)$$

**Proposition 3.10** *Let $f$ be a polynomial of degree $2r-1$ and $\mathbf{f} \triangleq \{f(2k)\}$. Then for any real real $\rho$ such that $u_\rho^{2r-1}(e^{i\omega}) \neq 0\ \forall\omega \in \mathbb{R}$, the spline*

$$\check{S}_\rho^{2r-1}(2k+1) \equiv f(2k+1), \quad \widetilde{S}_\rho^{2r-1}(2k) \equiv f(2k),\ k \in \mathbb{Z}.$$

*However, if $f(t)$ is a polynomial of degree $2r+1$ and $\rho = -A^r/4^r$, where $A_{2r-1}$ is defined in (3.11), then the spline restores its values at the midpoints $\{2k+1\}$, $k \in \mathbb{Z}$.*

**Proof:** Similarly to (3.27) we can represent the difference as

$$f(2k+1) - \check{S}_\rho^{2r-1}(2k+1) = \sum_{n\in\mathbb{Z}} a^p_{\rho,2k+1-n} f(n),$$

where the $z$-transform of the sequence $\{a^p_{\rho,k}\}$ is $A^p_\rho(z) = 1 - z^{-1}\check{U}^{2r-1}_\rho(z^2)$. Then, using (3.25) we have

$$1 - z^{-1}\check{U}^{2r-1}_\rho(z^2) = (z-2+z^{-1})^r \frac{q_{r-2}(z-2+z^{-1}) + \rho(z+2+z^{-1})^r}{u^{2r-1}(z^2) + \rho(z^2-2+z^{-2})^r}$$
$$= (z-2+z^{-1})^r \frac{A_r + \rho\, 4^r + \sum_{k=1}^r \nu_k (z-2+z^{-1})^k}{u^{2r-1}(z^2) + \rho(z^2-2+z^{-2})^r}.$$

Hence, the assertion of the proposition concerning the midpoints $\{2k+1\}$ follows.

As for the points $\{2k\}$, it is seen from (3.45) that the difference is

$$f(2k) - \check{S}_\rho^{2r-1}(2k) = \sum_n b^p_{\rho,k-n} f(2n),$$

where the $z$-transform of the sequence $\{b^p_{\rho,k}\}$ is

$$B^p_\rho(z) = 1 - \frac{u^{2r-1}(z)}{u^{2r-1}_\rho(z)} = \frac{\rho(z-2+z^{-1})^r}{u^{2r-1}_\rho(z)}.$$

Therefore, if $f(t)$ is a polynomial of degree $2r-1$ then the spline interpolates it at the points $\{2k\}$. ■

**Examples:**

**Quadratic spline with upgraded numerator:**

$$\tilde{U}^{2r-1}_A(z) = \frac{(1+z)(1024 + (z^{-1}+6+z)(z^{-1}-2+z)^2)}{256(z^{-1}+6+z)}$$
$$1 - z^{-1}\tilde{U}^{2r-1}_A(z^2) = \frac{(z-2+z^{-1})^3}{256\,(z^2+6+z^{-2})} \times$$
$$\times \left(z^4 + 6\,z^3 + 24\,z^2 + 74\,z + 174 + 74\,z^{-1} + 24\,z^{-2} + 6\,z^{-3} + z^{-4}\right).$$

**Quadratic spline with upgraded denominator:**

$$\check{U}^{2r-1}_A(z) = \frac{64(1+z)}{16(z^{-1}+6+z) - (z^{-1}-2+z)^2}$$
$$1 - z^{-1}\check{U}^{2r-1}_A(z^2) = \frac{-(z-2+z^{-1}))^3(z+6z+1/z)}{16(z^{-2}+6+z^2) - (z^{-2}-2+z^2)^2}.$$

# 4 Design of prediction filters using discrete splines

In this section we derive prediction filters for lifting wavelet transforms using the so-called discrete splines. We will show that these filters are intimately related to the Butterworth filters which are commonly used in signal processing [30].

## 4.1 Discrete splines

We outline briefly the properties of discrete splines, which will be needed for further constructions. For a detailed description of the subject, see [31, 34]. The discrete splines are defined on the grid $\{k\}$ and present a counterpart to the continuous polynomial splines.

The discrete B-spline of first order is defined by the following sequence for $k \in \mathbb{Z}$:

$$B_k^{1,n} = \begin{cases} 1 & \text{if } k = 0, \ldots, 2n-1,\ n \in \mathbb{N}, \\ 0, & \text{otherwise.} \end{cases} \tag{4.1}$$

We define by recurrence the higher order B-splines via the discrete convolutions: $B^{p,n} = B^{1,n} * B^{p-1,n}$. Obviously, the $z$-transform of the B-spline of order $p$ is

$$B^{p,n}(z) = (1 + z^{-1} + z^{-2} + \ldots + z^{-2n+1})^p \quad p = 1, 2, \ldots.$$

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

$$Q_k^{2r} \triangleq B^{2r,1}(k + r), \quad Q^{2r}(z) = z^r B^{2r,1}(z) = z^r (1 + z^{-1})^{2r}. \tag{4.2}$$

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

$$S_k^{2r} \triangleq \sum_{l=-\infty}^{\infty} c_l Q_{k-2l}^{2r} \tag{4.3}$$

of shifts of the central B-spline of order $2r$. Our scheme to design prediction filters using the discrete splines remains the same as the above scheme, that is based on polynomial interpolatory splines. Namely, we construct the discrete spline, which interpolates even samples $\{e_k = f_{2k}\}$ of a signal $\mathbf{f} \triangleq \{f_k\},\ k \in \mathbb{Z}$, and use the values $S_{2k+1}^{2r}$ for the prediction

of even samples $\{o_k = f_{2k+1}\}$. As in Section 3.2 we denote by $\mathbf{s}^e \triangleq \{s^e_k \triangleq S^{2r}_{2k}\}$, $\mathbf{s}^o \triangleq \{s_{o,k} \triangleq S^{2r}_{2k+1}\}$ and $\mathbf{s} = \{s_k \triangleq S^{2r}_k\}$ the values of the spline at grid points, at midpoints and in the whole set $\{k\}$, respectively.

We construct the spline $S^{2r}_d$ such that

$$S^{2r}_{2k} = (s^e)_k = e_k, \ k \in \mathbb{Z}. \tag{4.4}$$

Let $[x]$ denote rounding the number $x$ toward zero and $E(z)$ be the $z$-transform of the sequence $\{e_k\}$.

**Proposition 4.1** *The $z-$transform of the sequence $\mathbf{s}^o$ of values of the discrete spline at midpoints is*

$$s^o(z) = U^{2r}_d(z)E(z), \ \textit{where } U^{2r}_d(z) \triangleq \frac{\theta^{2r}(z)}{\upsilon^{2r}(z)}, \tag{4.5}$$

$$1 - z^{-1}U^{2r}_d(z^2) = \frac{2(-1)^r\left(1-z^{-1}\right)^{2r}}{\left(1+z^{-1}\right)^{2r} + (-1)^r\left(1-z^{-1}\right)^{2r}}, \tag{4.6}$$

$$\textit{and } \upsilon^{2r}(z) \triangleq \sum_{k=-[r/2]}^{[r/2]} \binom{2r}{r-2k} z^k, \quad \theta^{2r}(z) \triangleq \sum_{k=[(-r+1)/2]}^{[(r+1)/2]} \binom{2r}{r-2k+1} z^k. \tag{4.7}$$

**Proof:** Applying the $z$-transform to Eq. (4.3), we get:

$$s(z) = C(z^2)Q^{2r}_e(z^2) + z^{-1}C(z^2)Q^{2r}_o(z^2) = s_e(z^2) + z^{-1}s_o(z^2),$$

where $C(z)$ is the $z$-transform of the sequence of coefficients $\{c_l\}$ and

$$Q^{2r}_e(z^2) = \sum_{k\in\mathbb{Z}} z^{-2k}Q^{2r}_{2k} = \frac{1}{2}\left(z^r\left(1+z^{-1}\right)^{2r} + (-z)^r\left(1-z^{-1}\right)^{2r}\right) = \upsilon^{2r}(z^2), \tag{4.8}$$

$$Q^{2r}_o(z^2) = \sum_{k\in\mathbb{Z}} z^{-2k}Q^{2r}_{2k+1} = \frac{z}{2}\left(z^r\left(1+z^{-1}\right)^{2r} - (-z)^r\left(1-z^{-1}\right)^{2r}\right) = \theta^{2r}(z^2). \tag{4.9}$$

Equations (4.4), (4.8) and (4.9) imply that

$$C(z) = E(z)/\upsilon^{2r}(z), \quad s^o(z) = C(z)\theta^{2r}(z) = E(z)\frac{\theta^{2r}(z)}{\upsilon^{2r}(z)}.$$

To make sure that the ratio has no poles on the unit circle, we substitute $z = e^{i\omega}$ into $\upsilon^{2r}(z^2)$. We have

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

■

Thus the filters $U_d^{2r}(z)$ can be used as the prediction filters in the lifting scheme. They are closely related to the so-called discrete-time Butterworth filters.

## 4.2 Discrete-time Butterworth filters

We recall briefly the notion of Butterworth filter. For details we refer to [30]. The magnitude squared frequency responses $\widehat{F}_l^r(\omega)$ and $\widehat{F}_h^r(\omega)$ of the low- and high-pass digital Butterworth filters of order $r$, respectively, are given by the formulas

$$|\widehat{F}_l^r(\omega)|^2 = \frac{1}{1+(\tan\frac{\omega}{2}/\tan\frac{\omega_c}{2})^{2r}}, \quad |\widehat{F}_h^r(\omega)|^2 = 1-|\widehat{F}_l^r(\omega)|^2 = \frac{1}{1+(\tan\frac{\omega_c}{2}/\tan\frac{\omega}{2})^{2r}}$$

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

$$|\widehat{F}_l^r(\omega)|^2 = \frac{1}{1+(\tan\frac{\omega}{2})^{2r}}, \quad |\widehat{F}_h^r(\omega)|^2 = 1-|\widehat{F}_l^r(\omega)|^2 = \frac{1}{1+(\cot\frac{\omega}{2})^{2r}}.$$

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

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

Similarly, the magnitude squared transfer function of the high-pass filter is:

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

It is readily seen that the function $U_d^{2r}$, defined in (4.5), is related to these transfer functions:

$$\frac{1}{2}\left(1+z^{-1}U_d^{2r}(z^2)\right) = |F_l^r(z)|^2, \quad \frac{1}{2}\left(1-z^{-1}U_d^{2r}(z^2)\right) = |F_h^r(z)|^2. \tag{4.10}$$

We will show that the structure of the filters $U_d^{2r}$ is similar to the structure of the filters $U_i^{2r-1}$ derived from polynomial interpolatory splines. For this purpose we analyze the denominator of the rational function $U_d^{2r}(z)$.

**Proposition 4.2** *If $r = 2p+1$ then:*

$$v^{2r}(z) = 2r\prod_{k=1}^{p}\frac{1}{\gamma_k^r}(1+\gamma_k^r z^{-1})(1+\gamma_k^r z), \tag{4.11}$$

*where*

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

*If* $r = 2p$ *then*

$$\upsilon^{2r}(z) = \prod_{k=1}^{p} \frac{1}{\gamma_k^r}(1 + \gamma_k^r z^{-1})(1 + \gamma_k^r z) \tag{4.12}$$

*where*

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

**Proof:** Denote

$$D_r(z) \triangleq 2z^r \upsilon^{2r}(z^{-2}) = (z+1)^{2r} + (-1)^r (z-1)^{2r}. \tag{4.13}$$

Suppose that $r = 2p + 1$. The equation $D_r(z) = 0$ is equivalent to $(z+1)^{2r} = (z-1)^{2r}$. Hence, $z_0 = 0$ and nonzero roots of $D_r(z) = 0$ can be found from the relation $z_k + 1 = e^{2\pi i k/2r}(z_k - 1),\ k = 1, 2, \ldots, 2r-1$. Hence, we have

$$z_k = \frac{e^{2\pi i k/2r} + 1}{e^{2\pi i k/2r} - 1} = -i \cot \frac{k\pi}{2r}, \quad k = 1, 2, \ldots, 2r-1. \tag{4.14}$$

The points $x_k = \cot \frac{k\pi}{2r}$ are symmetric about zero and $x_{2p+1-k} = x_{r-k} = 1/x_k$. Therefore we can write

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

where $\gamma_k^r = x_{p+k}^2$. Hence, (4.11) 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

$$z_k = -i \cot \frac{(2k+1)r\pi}{4r},\ k = 0, 1, \ldots, 2r-1. \tag{4.15}$$

Hence, (4.12) is derived. ■

**Remark.** We recall that the function $\upsilon^{2r}(z^2)$ 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 [25] the formulas for the poles of the functions $|f_l(z)|^2$ are given without a proof. These formulas are equivalent to Eqs. (4.14) and (4.15).

**Examples:**

**The simplest case, $r = 1$:** We have

$$U_d^2(z) = \frac{1+z}{2},\ 1 - z^{-1}U_d^2(z^2) = -\frac{z^{-1} - 2 + z}{2}. \tag{4.16}$$

The filter $U_d^2$ is FIR. From Proposition 2.3, the high-frequency analysis wavelet $\tilde{\psi}^1$ has two vanishing moments.

**Cubic discrete spline, $r = 2$:**

$$U_d^4(z) = 4\frac{1+z}{z+6+z^{-1}}, \quad 1 - z^{-1}U_i^3(z^2) = \frac{(z-2+z^{-1})^2}{z^{-2}+6+z^2}. \tag{4.17}$$

It is readily seen that the filter $U_d^4$ coincides with the filter $U_i^3$ derived from the quadratic polynomial spline (see (3.29). The high-frequency analysis wavelet $\tilde{\psi}^1$ has four vanishing moments.

**Discrete spline of sixth order, $r = 3$:** We have

$$U_d^6(z) = \frac{(z+14+z^{-1})(1+z)}{6z^{-1}+20+6z)} \quad 1 - z^{-1}U_d^6(z^2) = -\frac{(z^{-1}-2+z)^3}{6z^2+20+6z^{-2}}. \tag{4.18}$$

The high-frequency analysis wavelet $\tilde{\psi}^1$ has six vanishing moments.

**Discrete spline of eighth order, $r = 4$:**

$$\begin{aligned} U_d^8(z) &= \frac{8(1+z)(z^{-1}+6+z)}{z^{-2}+28z^{-1}+70+28z+z^2} \\ 1 - z^{-1}U_d^6(z^2) &= \frac{(z^{-1}-2+z)^4}{z^{-4}+28z^{-2}+70+28z^2+z^4}. \end{aligned} \tag{4.19}$$

The high-frequency analysis wavelet $\tilde{\psi}^1$ has eight vanishing moments.

# 5 Filters for the *update* step

In Sections 3 and 4 we presented a family of filters $U$ for the *predict* step, which was derived from splines of various types. To complete the construction of the transform we need to define the filter $V$ for the *update* part. The fact that any choice of these filters retains the perfect reconstruction property of the transform is a great advantage of the lifting scheme.

Proposition 2.3 indicates that, in order to produce synthesis and analysis filters with similar properties, it is advisable to choose $V(z) = \breve{U}(z)/z$, where $\breve{U}$ is one of the filters $U$ presented above. In particular, the filter $\breve{U}$ may coincide with the filter $U$, which is used for the prediction. In this case the numbers of vanishing moments of the high-frequency analysis and synthesis wavelets are equal to each other. On the other hand, by combining various pairs of the prediction $U$ and the update $V(z) = \breve{U}(z)/z$ filters, we obtain a wide family of biorthogonal wavelet transforms with diverse properties.

# 6 Implementation of filters with rational transfer functions

## 6.1 Recursive filtering

Most of the presented prediction and update filters have infinite impulse response. However, the rational structure of their transfer functions enables us to implement transforms via fast recursive filtering, which is commonly used in signal processing. Recursive filtering can be carried out in either cascade or parallel mode. We illustrate these procedures on the filter $U_i^3$ as an example. This filter is based on the quadratic interpolatory spline (see (3.29)). The rational function $U_i^3(z)$ is represented as:

$$U_i^3(z) = \overleftarrow{R}(z)\overrightarrow{R}(z)P(z) = \frac{1}{1+\gamma}(\overrightarrow{R}(z) + z\overleftarrow{R}(z)), \tag{6.1}$$

where $\gamma = 3 - 2\sqrt{2} \approx 0.172$ and

$$\overrightarrow{R}(z) \triangleq \frac{1}{1+\gamma z^{-1}}, \quad \overleftarrow{R}(z) \triangleq \frac{1}{1+\gamma z}, \quad P(z) \triangleq 4\gamma(1+z).$$

Thus, filtering $Y(z) = U_i^3(z)X(z)$ can be implemented via the following cascade:

$$\begin{aligned} &Y^1(z) = P(z)X(z), \longrightarrow Y^2(z) = \overrightarrow{R}(z)Y^1(z) \longrightarrow Y(z) = \overleftarrow{R}(z)Y^2(z) \Longleftrightarrow \\ &y_k^1 = 4\gamma(x_k + x_{k+1}, \longrightarrow y_k^2 = y_k^1 - \gamma y_{k+1}^2, \longrightarrow y_k = y_k^2 - \gamma y_{k+1}, \end{aligned} \tag{6.2}$$

or in parallel mode:

$$\left.\begin{aligned} Y^1(z) &= \overrightarrow{R}(z)X(z) \\ Y^2(z) &= z\overleftarrow{R}(z)X(z) \end{aligned}\right\} \longrightarrow Y(z) = \frac{1}{1+\gamma}(Y^1(z) + Y^2(z)) \Longleftrightarrow \tag{6.3}$$

$$\left.\begin{aligned} y_k^1 &= x_k - \gamma y_{k-1}^1 \\ y_k^2 &= x_{k+1} - \gamma y_{k+1}^2 \end{aligned}\right\} \longrightarrow y_k = \frac{1}{1+\gamma}(y_k^1 + y_k^2).$$

Due to Propositions 4.2 and 3.2, all the presented rational transfer functions can be expanded into products or sums of the elementary recursive blocks of type (6.1). Therefore this example is instructive.

## 6.2 Finite-length signals

Application of wavelet transforms to finite-length signals and, in particular, to images, requires an extension of the signals beyond their boundaries [11]. The extension is even more important in our scheme since we implicitly assumed in our construction that the signals are defined on infinite intervals. When the filter banks are symmetric, the **HH** extension in the terminology of [11] is most efficient. It means that the signal $\mathbf{x} = \{x_k\},\ k = 1, \dots N$, is symmetrically extended with the repetition of boundary samples through both ends of the interval. Namely, $x_0 \triangleq x_1$, $x_{-1} \triangleq x_2, \dots x_{-k} \triangleq x_{k+1}$ and $x_{N+1} \triangleq x_N$, $x_{N+2} \triangleq x_{N-1}, \dots x_{N+k} \triangleq x_{N-k+}$. This results in periodization of the signal with period $2N$. This extended signal is denoted by $\tilde{\mathbf{x}}$ and its $z$ transform is $\tilde{X}(z)$.

Recursive filtering of finite-length signals requires additional treatment of the boundaries. We describe the application of the filter $U_i^3(z)$ to a finite-length signal $\mathbf{x}$. We begin with the parallel mode. Note that, in principle, the filters $\overrightarrow{R}(z)X(z)$ and $z\overleftarrow{R}(z)$ can be applied to the signal $\tilde{\mathbf{x}}$ in a non-recursive mode:

$$Y^1(z) = \frac{1}{1+\gamma z^{-1}}\tilde{X}(z) = \sum_{n=0}^{\infty}(-\gamma)^n z^{-n}\tilde{X}(z),\ Y^2(z) = \frac{z}{1+\gamma z}\tilde{X}(z) = \sum_{n=0}^{\infty}(-\gamma)^n z^{n+1}\tilde{X}(z). \tag{6.4}$$

Then, the signals $y_k^1$ and $y_k^2$ can be computed in time domain in either recursive or non-recursive modes:

$$y_k^1 = \tilde{x}_k - \gamma y_{k-1}^1, \quad y_k^2 = \tilde{x}_{k+1} - \gamma y_{k+1}^2 \Longleftrightarrow \tag{6.5}$$

$$y_k^1 = \tilde{x}_k + \sum_{n=1}^{\infty}(-\gamma)^n \tilde{x}_{k-n},\ y_k^2 = \tilde{x}_{k+1} + \sum_{n=1}^{\infty}(-\gamma)^n \tilde{x}_{k+n+1}. \tag{6.6}$$

We can use (6.5) for the computation of $y_k^1$ and $y_k^2$ on the interval $k = 1, 2, \dots, N$ provided we know $y_0^1$ and $y^2(N+1)$, respectively. To evaluate these samples, we employ the non-recursive

equation (6.6). We have

$$y_0^1 = \tilde{x}_0 + \sum_{n=1}^{\infty}(-\gamma)^n \tilde{x}_{-n} \approx x_1 + \sum_{n=2}^{d}(-\gamma)^n x_n, \tag{6.7}$$

$$y_{N+1}^2 = \tilde{x}_{N+2} + \sum_{n=1}^{\infty}(-\gamma)^n \tilde{x}_{N+n+2} \approx x_{N-1}) + \sum_{n=2}^{d}(-\gamma)^n x_{N-n} \tag{6.8}$$

where $d < N$ is the prescribed depth of the initialization. The whole operation is called the initialization of the filter.

Recursive parallel filtering of the finite-length signal $\mathbf{x}$ by the filter $U_i^3$ is implemented as follows:

1. Evaluate $y_0^1$ from (6.7) and $y^2(N+1)$ from (6.8).

2. Calculate $y_k^1 = x_k - \gamma y_{k-1}^1, \quad k = 1, \dots N$ and $y_k^2 = x_{k+1} - \gamma y_{k+1}^2, \quad k = N, \dots 1.$

3. The result of filtering is $y_k = (y_k^1 + y_k^2)/(1+\gamma), \quad k = 1 \dots N.$

Equations (6.6) and (6.8) imply that $y_N^2 = y_N^1$. Hence, it follows that

$$y_N = \frac{y_N^1 + y_N^2}{1+\gamma} = \frac{2y_N^1}{1+\gamma}. \tag{6.9}$$

The cascade algorithm has the following form:

1. Evaluate $y_0^1$ from (6.7).

2. Calculate $y_k^1 = x_k - \gamma y_{k-1}^1, \quad k = 2, \dots N.$

3. Evaluate $y_N$ from (6.9).

4. Calculate $y_k = y_k^1 + y_{k+1}^1 - \gamma y_{k+1}, \quad k = N-1, \dots 1.$

Note that the depth of the initialization does not affect the perfect reconstruction property of the transforms since in lifting schemes the reconstruction steps are the reversed decomposition steps. But the results of lossy compression deteriorate if this depth is insufficient.

# 7 Scaling functions generated by rational filters

## 7.1 Cascade algorithm and subdivision schemes

We presented in previous sections a set of filters $U$ and $V$, which are used as prediction and update filters in lifting schemes. They produce a family of perfect reconstruction filter banks $H,\ G,\ \widetilde{H}$ and $\widetilde{G}$ (see Section 2.2). These filter banks generate analysis and synthesis scaling functions $\widetilde{\varphi}$, which are solutions for the refinement equations [16]:

$$\widetilde{\varphi}(t) = \sum_{k\in\mathbb{Z}} \widetilde{h}_k \widetilde{\varphi}(2t-k), \quad \varphi(t) = \sum_{k\in\mathbb{Z}} h_k \varphi(2t-k). \tag{7.1}$$

The cascade algorithm for the construction of scaling functions, which is described in [16], consists of infinite iterations of the subdivision scheme whose mask is either the low-pass synthesis filter $H$ or the analysis filter $\widetilde{H}$ of the wavelet transform. This scheme is applied to the initial data, which is the Kroneker delta $\delta_n$. If this process converges then the limit function is the scaling function generated by either the filter $H$ or $\widetilde{H}$. Therefore, methods of analysis of convergence and of regularity of limit functions, which have been developed in the theory of subdivision schemes, can be applied to the analysis of scaling functions. Our analysis is based on a technique developed in [22, 23] for the schemes that employ FIR filters. The extension of the technique to schemes with IIR filters requires some modifications. Further we provide necessary definitions and preliminary results.

A univariate stationary uniform subdivision scheme (SS) $S_a$ that is based on a filter $a(z) = \sum_{k\in\mathbb{Z}} z^{-k} a_k$, consists of the following:

Given the initial data $\mathbf{f^0} = \{f_k^0\},\ k \in \mathbb{Z}$, one refinement step is an extension of the function $f^j,\ j = 0, 1, \ldots,$ defined on the grid $\mathbf{G^j} = \{k/2^j\}_{k\in\mathbb{Z}}$: $f^j(k/2^j) = f_k^j$, onto the grid $\mathbf{G^{j+1}}$ by multirate filtering the array $\{f_k^j\}$:

$$f_k^{j+1} = \sum_{l\in\mathbb{Z}} a_{k-2l} f_l^j. \tag{7.2}$$

The insertion rule (7.2) splits into two separate rules:

$$f_{2k}^{j+1} = \sum_{l\in\mathbb{Z}} a_{k-l}^e f_l^j, \quad f_{2k+1}^{j+1} = \sum_{l\in\mathbb{Z}} a_{k-l}^o f_l^j,$$

where $a_k^e \triangleq a_{2k}$ and $a_k^o \triangleq a_{2k+1}$. The impulse response $\mathbf{a} = \{a_k\}$ of the filter $a(z)$ is called the refinement mask of the SS, $S_a$. If $a_0 = 1,\ a_{2k} = 0\ \forall k \neq 0$ then the SS is interpolatory

(ISS). In this case $f_{2k}^{j+1} = f_k^j$. The transfer function of the filter, $a(z)$, is called the symbol of the SS, $S_a$. Provided $f^j$ and $f^{j+1}$ belong to the space $l^1$, Eq. (7.2) is equivalent to the following relation in the $z$-domain:

$$f^{j+1}(z) = a(z)f^j(z^2). \tag{7.3}$$

Comparing (7.3) with (2.15), we see that this refinement step is completely identical to the reconstruction of the signal $\mathbf{x}$ from the array $\mathbf{s^1}$, provided $a(z) = H(z)$. It is apparent that all the filters presented in Sections 3 and 4 are associated with the interpolatory subdivision schemes.

**Definition 7.1** *Let the initial data be* $\mathbf{f^0} = \{f_k^0\},\ k \in \mathbb{Z}$ *and* $f^j(t)$ *be the sequence of continuous functions that interpolates the data that was generated by* $S_a$ *at the corresponding refinement level:* $\{f^j(2^{-j}k) = f_k^j = (S_a^j f^0)_k\},\ k \in \mathbb{Z}$ *. If* $\{f^j(t)\}$ *converges uniformly at any finite interval to a continuous function* $f^\infty(t)$ *as* $j \to \infty$ *then we say that the subdivision scheme* $S_a$ *converges on the initial data* $\mathbf{f^0}$ *and* $f^\infty(t)$ *is called its limit function.*

**Remark** Usually polygonal lines (second order splines) are employed as the interpolating continuous functions $f^j(t)$. However, in Section 7.5 we will use splines of arbitrary even order for this purpose.

**Definition 7.2** *If the scheme converges on the initial data* $\{f_k^0 = \delta_k\},\ k \in \mathbb{Z}$*, where* $\delta_k$ *is the Kroneker delta, then the limit function, which we denote as* $\phi_a$*, is called the basic limit function (BLF) of the scheme* $S$*.*

**Proposition 7.1 ([16])** *The BLF of the scheme* $S_a$ *is the scaling function of the wavelet transform whose low-pass filter is* $a(z)$*.*

All the above designed filters generate subdivision schemes $S_a$, whose symbols $a(z) = T(z)/P(z)$ are rational functions and possess the following properties:

**P1:** The Laurent polynomials $P(z)$ and $T(z)$ are invariant under the inversion $P(z^{-1}) = P(z)$, $T(z^{-1}) = T(z)$ and thus are real on the unit circle $|z| = 1$.

**P2:** Roots of the denominator $P(z)$ are real, simple and do not lie on the unit circle $|z| = 1$.

**P3:** $P(z)$ can be represented as follows:

$$P(z) = \prod_{n=1}^{r} \frac{1}{\gamma_n}(1+\gamma_n z)(1+\gamma_n z^{-1}), \quad 0 < |\gamma_1| < \ldots |\gamma_r| = e^{-g} < 1,\ g > 0, \qquad (7.4)$$

and, for all $k$ $\mathrm{Im}\gamma_k = 0$. If **P1** and **P2** hold then they imply **P3**.

**P4:** The symbol $a(z)$ can be factorized as follows:

$$a(z) = (1+z)q(z), \quad q(1) = 1. \qquad (7.5)$$

**Definition 7.3** *We say that a rational function $a(z)$ belongs to Class **P** if it possesses all the properties* **P1**– **P4**. *If the symbol $a(z)$ of a scheme $S_a$ belongs to Class **P** then we call $S_a$ the scheme of Class **P**.*

The above properties imply in particular that the coefficients $a_k$ of the mask of a Class **P** scheme $S_a$ are symmetric about zero.

**Proposition 7.2** *If $a(z) = T(z)/P(z)$ is the symbol of a subdivision scheme $S_a$ and Eq. (7.4) holds then the mask*

$$|a_k| \le A\, e^{-g|k|},$$

*where $A$ is a positive constant.*

**Proof:** If Eq. (7.4) holds then the symbol can be represented as follows:

$$a(z) = \sum_{n=1}^{r}\left(\frac{A_n^+}{1+\gamma_n z} + \frac{A_n^-}{1+\gamma_n z^{-1}}\right) = \sum_{n=1}^{r}\left(A_n^+ \sum_{k=0}^{\infty}(-\gamma_n)^k z^k + A_n^- \sum_{k=0}^{\infty}(-\gamma_n)^k z^{-k}\right)$$
$$= \sum_{k=0}^{\infty}\left(a_k^+ z^k + a_k^- z^{-k}\right),\ a_k^+ = \sum_{n=1}^{r} A_n^+(-\gamma_n)^k,\ a_k^- = \sum_{n=1}^{r} A_n^-(-\gamma_n)^k,$$
$$|a_k^+| \le |\gamma_r|^k \sum_{n=1}^{r}|A_n^+| \le Ae^{-gk},\ |a_k^-| \le |\gamma_r|^k \sum_{n=1}^{r}|A_n^-| \le Ae^{-gk}.$$

■

**Lemma 7.1** *If $a(z) = T(z)/P(z)$ is the symbol of a subdivision scheme $S_a$ and Eq. (7.4) holds then for any initial data $\mathbf{f^0} \in l_1$ the following inequalities are true:*

$$|f_k^j| \le A_j\, e^{-gk2^{-j+1}}, \quad j = 1, 2, \ldots. \qquad (7.6)$$

**Proof:** The mask of the scheme $S_a$ decays exponentially: $|a_k| \le Ae^{-gk}$. Due to (7.3)

$$f^1(z) = a(z)f^0(z^2) = \frac{T^1(z)}{P^1(z)},$$

where $T^1(z) \stackrel{\Delta}{=} T(z)f^0(z^2)$ and $P^1(z) = P(z)$. Hence, the roots of $P_1(z)$ are: $\rho_n^1 = -\gamma_n$, $1 \le n \le r$ and, therefore, $|f_k^1| \le A^1\, e^{-gk}$. The next refinement step produces the following $z$-transform:

$$f^2(z) = a(z)f^1(z^2) = \frac{T_2(z)}{P_2(z)}, \quad P_2(z) = P(z)P(z^2).$$

The roots of $P_2(z)$ satisfy the inequality $|\rho_n^2| \le \sqrt{|\gamma_r|} = e^{-g/2}$. Hence, $|f_k^2| \le A_2\, e^{-gk/2}$. Then (7.6) is derived by induction. ■

Let $S_a$ be the subdivision scheme of Class **P** and $S_q$ be the scheme with the symbol $q(z)$, which is defined in (7.5). Since the denominator of the symbol $q(z)$ is the same as the denominator of $a(z)$, the mask $\{q_k\}$ of the scheme $S_q$ satisfies the inequality

$$|q_k| \le Q\, e^{-g|k|}. \tag{7.7}$$

Denote by $\Delta$ the difference operator: $\Delta\, f_k = f_{k+1} - f_k$.

**Proposition 7.3 ([23])** *If the scheme $S_a$ is of Class **P** then*

$$\Delta(S_a\mathbf{f}) = S_q\mathbf{\Delta f}$$

*for any data set* $\mathbf{f}$ *that belongs to* $l^1$.

**Proof:** Obviously, $(\Delta f)(z) = (z-1)f(z)$ and, using (7.3), we have

$$\begin{aligned}(\Delta S_a f)(z) &= (z-1)(S_a f)(z) = (z-1)\, a(z) f(z^2)\\ &= q(z)(z^2-1)\, f(z^2) = q(z)\, (\Delta f)(z^2) \Leftrightarrow \Delta(S_a\mathbf{f}) = S_q\mathbf{\Delta f}.\end{aligned}$$

■

Denote $\|\mathbf{f^j}\|_\infty \stackrel{\Delta}{=} \max_{k\in\mathbb{Z}} |f_k^j|$. Equation (7.2) implies that

$$f_{2k}^{j+1} = \sum_{l\in\mathbb{Z}} a_{2k-2l} f_l^j, \quad f_{2k+1}^{j+1} = \sum_{l\in\mathbb{Z}} a_{2k+1-2l} f_l^j.$$

Hence, it follows that

$$\|\mathbf{f^{j+1}}\|_\infty \le \|S_a\|\,\|\mathbf{f^j}\|_\infty, \text{ where } \|S_a\| \triangleq \max\left\{\sum_{k\in\mathbb{Z}} a_{2k},\ \sum_{k\in\mathbb{Z}} a_{2k+1}\right\}.$$

Similarly, after $L$ refinement steps we have

$$\|\mathbf{f^{j+L}}\|_\infty \le \|S_a^L\|\,\|\mathbf{f^j}\|_\infty, \text{ where } \|S_a^L\| \triangleq \max_i\left\{\sum_k |a^{[L]}_{i+2^L k}| : 0 \le i \le 2^L-1\right\}$$

and $\left\{a_k^{[L]}\right\}$ is the mask of the operator $S_a^L$.

**Definition 7.4** *If for some $L \in \mathbb{N}$ the following inequality holds*

$$\|S_a^L\| \triangleq \max_i\left\{\sum_k |a^{[L]}_{i+2^L k}| : 0 \le i \le 2^L-1\right\} = \mu < 1 \tag{7.8}$$

*then the scheme $S_a$ is called contractive.*

## 7.2 Convergence of the cascade algorithm and regularity of scaling functions

Let $S_a$ be a subdivision scheme, whose mask $\mathbf{a} = \{a_k,\ k \in \mathbb{Z}\}$ and symbol is $a(z)$. To establish the existence of the continuous scaling function associated with the filter $a(z)$, it is sufficient to prove the convergence of the subdivision scheme $S_a$ on the initial data $\mathbf{f^0} = \{\delta_k,\ k \in \mathbb{Z}\}$. However, we prove a more general proposition, using a slightly modified version of the proof of a related proposition in [23].

**Proposition 7.4** *Let $S_a$ be a subdivision scheme of Class $\boldsymbol{P}$ and $S_q$ be the scheme, whose symbol is $q(z)$ and mask is $\{q_k\}$, $k \in \mathbb{Z}$. If the scheme $S_q$ is contractive then the scheme $S_a$ converges on any initial data $\mathbf{f^0} \in l^1$.*

**Proof:** We recall that due to Lemma 7.1, for all $j \in \mathbb{N}$ the sequences $\mathbf{f^j}$ belong to $l_1$. Let $\{f^j(t),\ j \in \mathbb{N}\}$ be the sequence of the second order splines, which interpolate the subsequently refined data, that is, $f^j(2^{-j}k) = f_k^j,\ k \in \mathbb{Z}$. We have to show that $\{f^j(t)\}$ converges to a continuous function $f^\infty(t)$ as $j \to \infty$, provided $\{f_k^0\} \in l_1$. Denote

$$D^{j+1}(t) \triangleq f^{j+1}(t) - f^j(t). \tag{7.9}$$

The maximum absolute value of this piecewise function is reached at its breakpoints. Therefore, if $t = 2^{-j}(k+\tau),\ 0 \le \tau \le 1$, then

$$|D^{j+1}(t)| \le \left| f_{2k+1}^{j+1} - \frac{f_k^j + f_{k+1}^j}{2} \right|. \tag{7.10}$$

Let $m_{2k}^{j+1} = f_k^j, \quad m_{2k+1}^{j+1} = (f_k^j + f_{k+1}^j)/2,\ k \in \mathbb{Z}$. Then the $z$-transform $m^{j+1}(z)$ is equal to

$$m^{j+1}(z) = l(z) f^j(z^2), \quad l(z) \triangleq \frac{1}{2}\left(z^{-1} + 2 + z\right) = \frac{z^{-1}}{2}(1+z)^2$$

and we obtain

$$\sup_{t\in\mathbb{R}} |D^{j+1}(t)| = \|\mathbf{f^{j+1}} - \mathbf{m^{j+1}}\|_\infty. \tag{7.11}$$

Equation (7.5) implies

$$\begin{aligned} f^{j+1}(z) - m^{j+1}(z) &= ((1+z)q(z) - l(z))\, f^j(z^2) \\ &= (1+z)\,(q(z) - (1+z)/(2z))\, f^j(z^2) \\ &= (1+z)(1-z)r(z) f^j(z^2) = r(z) h^j(z^2), \end{aligned} \tag{7.12}$$

where $h^j(z) = (\Delta(f^j))(z)$ and $r(z) = \sum_{k\in\mathbb{Z}} r_k z^{-k}$ is a rational function with the same denominator $P(z)$ as the symbol $a(z)$. Hence

$$|r_k| \le R\, e^{-g|k|}. \tag{7.13}$$

In the equality (7.12) we used the fact that $q(z) - (1+z)/(2z)$ is divisible by $z-1$ since it vanishes at $z = 1$.

Combining (7.11) and (7.12), we derive

$$\begin{aligned} &\sup_{t\in\mathbb{R}} |f^{j+1}(t) - f^j(t)| = \|\mathbf{f^{j+1}} - \mathbf{m^{j+1}}\|_\infty \le \rho \max_k |f_{k+1}^j - f_k^j| \\ &= \rho \|\Delta(\mathbf{f^j})\|_\infty \le \rho \|S_q^j(\mathbf{\Delta f^0})\|_\infty, \end{aligned}$$

where $\rho = \sum_{k\in\mathbb{Z}} |r_k|$. If (7.8) holds then

$$\sup_{t\in\mathbb{R}} |f^{j+1}(t) - f^j(t)| \le \rho \mu^{\left[\frac{j}{L}\right]} \max_{0\le n\le L} \|(\mathbf{\Delta f})^n\|_\infty, \le C\theta^j \quad \theta \triangleq \mu^{\frac{1}{L}} < 1. \tag{7.14}$$

Equation (7.14) implies that the sequence of the second order splines $\{f^j(t)\}$ converges uniformly to a continuous function $f^\infty(t)$. ■

**Proposition 7.5 ([22])** *Let $S_a$ be a subdivision scheme of Class $\boldsymbol{P}$ and in addition let the symbol be factorized as follows:*

$$a(z) = \frac{(1+z)^m}{2^m} b(z).$$

*If for some initial data $\mathbf{f^0} \in l_1$ the subdivision scheme $S_b$, whose symbol is $b(z)$, converges on the data $\mathbf{g^0} \triangleq \mathbf{\Delta^m f^0}$ to a continuous function $g^\infty(t)$ then the scheme $S_a$ converges on the data $f^0$ to a function $f^\infty(t)$, $t \in \mathbb{R}$, which has $m$ continuous derivatives and*

$$\frac{d^m}{dt^m} f^\infty(t) = g^\infty(t).$$

## 7.3 Exponential decay of BLF's

In this section we prove that, provided a subdivision scheme with the rational symbol converges, its basic limit function (scaling function) decays exponentially as its argument tends to infinity.

**Theorem 7.1** *Let $S_a$ be a subdivision scheme of Class $\boldsymbol{P}$ and $S_q$ be the scheme, whose symbol is $q(z)$ and mask is $\{q_k\}$, $k \in \mathbb{Z}$. If the scheme $S_q$ is contractive then there exists a continuous BLF $\phi_a(t)$ of the scheme $S_a$, which decays exponentially as $|t| \to \infty$. Namely, if (7.4) holds then for any $\varepsilon > 0$ there exists a constant $\Phi_\varepsilon > 0$ such that the following inequality*

$$|\phi_a(t)| \le \Phi_\varepsilon \, e^{-(g-\varepsilon)t}.$$

*is true.*

**Proof:** To simplify the calculations we assume that in (7.8) $L = 1$ (the case $L > 1$ is treated similarly). Thus,

$$\|S_q\| = \max\left\{\sum_k |q_k^e|, \ \sum_k |q_k^o|\right\} = \mu < 1, \tag{7.15}$$

where $q_k^e \triangleq q_{2k}$ and $q_k^o \triangleq q_{2k+1}$

Note that, due to Lemma 7.1, each second order spline, which interpolates the refined data $f_k^j$, $k \in \mathbb{Z}$, $j > 0$ decays exponentially as $|t| \to \infty$. Equation (3.46) implies that if $t = 2^{-j}(k+\tau)$, $0 \le \tau \le 1$ then

$$|f^j(t)| \le \max\{|f_k^J|, \ |f_{k+1}^J|\} \le A_J \, e^{-gk2^{-J+1}} \le B_J \, e^{-gt}, \quad B_J \triangleq A_J \, e^{g2^{-J+1}}.$$

The completion of the proof depends on establishing the exponential decay of the difference $d^J(t) = \phi_a(t) - f^J(t)$ for a fixed index $J$. For this purpose we analyze the local behavior of the function $D^{j+1}(t)$ defined in (7.9). Due to (7.10) it reduces to evaluation of the odd subsequence of the sequence $\mathbf{y^{j+1}}$:

$$y_l^{j+1} \triangleq \begin{cases} f_{2k+1}^{j+1} - \left(f_k^j + f_{k+1}^j\right)/2, & l = 2k+1; \\ 0, & l = 2k, \end{cases} \qquad k \in \mathbb{Z}.$$

Equation (7.12) implies that

$$y^{j+1}(z) = r(z)h^j(z^2), \quad h^j(z) = (S_q^j(\mathbf{\Delta f^0}))(z). \tag{7.16}$$

We denote

$$\mathbf{h^j} = \{h_k^j\}_{k\in\mathbb{Z}} = S_q^j\,\mathbf{h^0}, \quad h_k^0 = \delta_{k+1} - \delta_k \text{ and } H_J \triangleq \max_k |h_k^J|.$$

The rest of the proof is split into four major steps.

**Analysis of the sequence $\mathbf{h^j}$:** By Lemma 7.1 and inequality (7.15) we have

$$|h_k^J| \le B_J e^{-g|k|2^{-J+1}}, \quad H_J \le \|S_q^J\|\|\mathbf{\Delta h^0}\|_\infty \le \mu^J, \tag{7.17}$$

where $B_J$ is a positive constant. Let $k \in \mathbb{Z}_+$. Then

$$h_k^{J+1} = \sum_{l=-\infty}^{\infty} q_{k-2l} h_l^J \Longleftrightarrow h_{2k}^{J+1} = \sum_{l=-\infty}^{\infty} q_l^e h_{k-l}^J, \quad h_{2k+1}^{J+1} = \sum_{l=-\infty}^{\infty} q_l^o h_{k-l}^J. \tag{7.18}$$

We split the even subsequence into two sums: $h_{2k}^{J+1} = \chi^1(s) + \chi_2(s)$, where

$$\chi^1(s) \triangleq \sum_{l=-s}^{s} q_l^e h_{k-l}^J, \text{ and } \chi_2(s) \triangleq \sum_{l=-\infty}^{-s-1} q_l^e h_{k-l}^J + \sum_{l=s+1}^{\infty} q_l^e h_{k-l}^J.$$

It follows from (7.17) and (7.7) that

$$|\chi^1(s)| \le B_J e^{-g(k-s)2^{-J+1}} \sum_{l=-\infty}^{\infty} |q_l^e| \le B_J\,\mu e^{-g(k-s)\,2^{-J+1}},$$
$$|\chi_2(s)| \le 2H_J \sum_{l=s+1}^{\infty} |q_l^e| \le 2\mu^J Q \frac{e^{-2gs}}{1-e^{-g}}.$$

Let $s = k2^{-J}$. Then we have

$$|\chi_2(s)| \le \theta_J \mu e^{-gk2^{-J+1}}, \text{ where } \eta_J \triangleq \frac{2\mu^{J-1}Q}{1-e^{-g}},$$
$$|\chi^1(s)| \le B_J\,\mu e^{-g(1-2^{-J})k2^{-J+1}} = B_J\,\mu\, e^{gk2^{-2J+1}}\, e^{-gk2^{-J+1}}.$$

Combining the estimates, we obtain

$$|h_{2k}^{J+1}| \le \mu \left(B_J e^{gk2^{-2J+1}} + \eta_J\right) e^{-gk2^{-J+1}}.$$

The same estimate is true for the odd subsequence. So, finally we have:

$$|h_k^{J+1}| \le \mu\, B_J \beta_J\, e^{-gk2^{-J}}, \text{ where } \beta_J \triangleq \left(e^{gk2^{-2J}} + \eta_J\right).$$

Similarly, we derive the inequality

$$|h_k^{J+2}| \le \mu^2\, B_J \beta_J \beta_{J+1}\, e^{-gk2^{-J-1}},$$

and by iterating, we get

$$|h_k^{J+j}| \le \mu^j\, B_J\, e^{-gk2^{-J-j+1}} \prod_{l=0}^{j-1} \beta_{J+l}.$$

**Evaluation of the sequence $\mathbf{y^{j+1}}$:** As follows from Eq. (7.16), the odd terms of the sequence $\mathbf{y^{J+j+1}}$ are:

$$y_{2k+1}^{J+j+1} = \sum_{l\in\mathbb{Z}} r_l^o\, h_{k-l}^{J+j}. \tag{7.19}$$

Recall that the filter $\mathbf{r} = \{r_r\}$ satisfies the inequality (7.13). Thus it is obvious that equation (7.19) is similar to (7.18) and by similar means we obtain the estimate

$$|y_{2k+1}^{J+j+1}| \le \rho\mu^j\, B_J\, e^{-gk2^{-J-j+1}}\, \epsilon_j \prod_{l=0}^{j-1} \beta_{J+l}, \tag{7.20}$$

where

$$\rho = \sum_{k\in\mathbb{Z}} |r_k|, \quad \epsilon_j \triangleq \left(e^{gk2^{-2(J+j)+1}} + \frac{2\mu^{J+j-1}R}{1-e^{-g}}\right) \le Ce^{gk2^{-2(J+j)+1}}.$$

**Estimation of the difference $D^{J+j}(t) = f^{J+j}(t) - f^{J+j-1}(t)$ for $t \in [2^{-J}k, 2^{-J}(k+1)]$:**

$$\begin{aligned} &|D^{J+1}(t)| \le |y_{2k+1}^{J+1}| \le \rho B_J \epsilon^1 e^{-gk2^{-J}} \le C\rho B_J e^{gk2^{-2J}} e^{-gk2^{-J}} \\ &\le C\rho B_J e^{g2^{-J}} e^{gt2^{-J}} e^{-gt}. \end{aligned} \tag{7.21}$$

At the half-interval $t = 2^{-J-1}(2k+\tau^1),\ 0 \le \tau^1 \le 1$, we have

$$\begin{aligned} &|D^{J+2}(t)| \le |y_{2(2k)+1}^{J+2}| \le \mu\rho B_J \beta_J \epsilon_2 e^{-g2k2^{-J-1}} \le C\mu\rho B_J e^{g2k2^{-2(J+1)}} e^{-g2k2^{-(J+1)}} \\ &\le C\mu\rho B_J (1+\eta_J) e^{2g(t+1)2^{-J-1}} e^{-gt}. \end{aligned} \tag{7.22}$$

Employing $y^{J+2}_{2(2k+1)+1}$ instead of $y^{J+2}_{2(2k)+1}$, we obtain a similar estimate for the second half-interval $t = 2^{-J-1}(2k+1+\tau_2),\ 0 \le \tau_2 \le 1$. So, inequality (7.22) is true at the whole interval $[k/2^J, (k+1)/2^J]$. Denote the converging infinite product by

$$N_J(\mu) \triangleq \prod_{j=0}^{\infty}(1+\eta_{J+j}) = \prod_{j=0}^{\infty}\left(1+\frac{2\mu^{J-1+j}Q}{1-e^{-g}}\right) > 1$$

and note that

$$\prod_{j=0}^{\infty} e^{g(t+1)2^{-J-j}} = \exp\left(g(t+1)\sum_{j=0}^{\infty}2^{-J-j}\right) = e^{g(t+1)2^{-J+1}}.$$

Then, the estimates (7.21) and (7.22) can be combined as follows

$$|D^{J+1+j}(t)| \le C\mu^j N_J \rho B_J e^{g(t+1)3\cdot 2^{-J}} e^{-gt} \quad j = 0, 1. \tag{7.23}$$

One can observe that by (7.20) the latter inequality is true for any $j \in \mathbb{N}$.

**Completion of the proof:** Inequality (7.23) enables us to evaluate the difference $d^J(t) = \phi_a(t) - f^J(t)$:

$$|d^J(t)| \le \sum_{j=0}^{\infty}|D^{J+1+j}(t)| \le N_J\rho B_J e^{g(t+1)3\cdot 2^{-J}} e^{-gt}\sum_{j=0}^{\infty} C\mu^j = \frac{C_J}{1-\mu}e^{-g(1-2^{-J})t}. \tag{7.24}$$

Hence, we derive that the BLF

$$|\phi_a(t)| \le |f^J(t)| + |d^J(t)| \le B_J e^{-gt} + \frac{C_J}{1-\mu}e^{-g(1-2^{-J})t} \le \Phi_J e^{-g(1-2^{-J})t}.$$

For any $\varepsilon > 0$ we can choose $J(\varepsilon) \in \mathbb{N}$ such that $g2^{-J} < \varepsilon$. Then, we have

$$|\phi_a(t)| \le \Phi_\varepsilon\, e^{-(g-\varepsilon)t}.$$

■

## 7.4 Scaling functions and wavelets

The above analysis of subdivision schemes in Sections 7.2 and 7.3 provides the following results concerning the scaling functions.

**Corollary 7.1** *If a low-pass filter $a(z)$ is a function of Class* $\boldsymbol{P}$ *and the subdivision scheme $S_q$, whose symbol is $q(z)$, is contractive then this filter generates a continuous scaling function $\varphi_a(t)$, which decays exponentially as $|t| \to \infty$. Namely, if (7.4) holds then for any $\varepsilon > 0$ there exists a constant $\Phi_\varepsilon > 0$ such that the following inequality is true:*

$$|\varphi_a(t)| \le \Phi_\varepsilon \, e^{-(g-\varepsilon)t}. \tag{7.25}$$

**Corollary 7.2** *Let the filter $a(z)$ satisfy the conditions of Corollary 7.1 and in addition be factorized as follows:*

$$a(z) = \frac{(1+z)^{m+1}}{2^m} q^{[m]}(z), \ \text{where } q^{[m]}(1) = 1. \tag{7.26}$$

*If the subdivision scheme $S_q^{[m]}$, whose symbol is $q^{[m]}(z)$, is contractive then the scaling function $\varphi_a(t)$ has $m$ continuous derivatives.*

The analysis and synthesis scaling functions $\widetilde{\varphi}$ and $\varphi$, which satisfy the refinement equations (7.1) generate the analysis and synthesis wavelets $\widetilde{\psi}$ and $\psi$, respectively, via the two-scale equations

$$\widetilde{\psi}(t) = \sum_{k\in\mathbb{Z}} \widetilde{g}_k \widetilde{\varphi}(2t-k), \quad \psi(t) = \sum_{k\in\mathbb{Z}} g_k \varphi(2t-k), \tag{7.27}$$

where $\{\widetilde{g}_k\}$ and $\{g_k\}$ are the impulse responses of the high-pass filters $\widetilde{G}$ and $G$, respectively. Equations (2.11) and (2.12) imply that the denominator of the rational function $\widetilde{G}(z)$ is the same as the denominator of $H(z)$. Therefore, if $H(z) = Q(z)/P(z)$ belongs to Class **P** and Eq. (7.4) holds then the impulse response $\{\widetilde{g}_k\}$ satisfies the inequality $|\widetilde{g}_k| \le e^{-gk}$. If $\widetilde{H}(z) = \widetilde{Q}(z)/\widetilde{P}(z)$ belongs to Class **P** then a similar inequality is true for the impulse response $\{g_k\}$. These inequalities together with equations (7.27) imply that if the scaling functions $\widetilde{\varphi}$ and $\varphi$ decay exponentially then the wavelets $\widetilde{\psi}$ and $\psi$ do the same. Moreover, the smoothness of the wavelets $\widetilde{\psi}$ and $\psi$ is the same as the smoothness of the scaling functions $\widetilde{\varphi}$ and $\varphi$, respectively.

**Definition 7.5** *A wavelet $\psi$ has $n$ vanishing moments if the following relations hold*

$$\int_{-\infty}^{\infty} t^k \psi(t)\, dt = 0, \quad k = 0, \dots n-1. \tag{7.28}$$

**Proposition 7.6 ([38])** *If the low-pass synthesis filter $H(z)$ has $n$ zeros at $z = 1$ then the analysis wavelet $\widetilde{\psi}$ has $n$ vanishing moments. If the low-pass analysis filter $\widetilde{H}(z)$ has $\tilde{n}$ zeros at $z = 1$ then the synthesis wavelet $\psi$ has $\tilde{n}$ vanishing moments.*

In our scheme this fact can be reformulated in a form similar to Proposition 2.3

**Proposition 7.7** *Let the transfer functions $U(z)$ and $V(z)$, which are used for the* predict *and* update *steps, respectively, be rational and have no poles on the unit circle $|z| = 1$. Let $1 + z^{-1}U(z^2)$ comprise the factor $(z + 2 + z^{-1})^r$ and $1 + zV(z^2)$ comprise the factor $(z + 2 + z^{-1})^p$. If the analysis and synthesis scaling functions exist then the analysis wavelet $\tilde{\psi}$ has $2r$ vanishing moments and the synthesis wavelet $\psi$ has $2s$ vanishing moments, where $s = \min(p, r)$.*

## 7.5 Scaling functions generated by polynomial splines of even order

In the case when the prediction filter is derived from a polynomial interpolatory spline of even order, as was described in Section 3.2, the synthesis scaling function and wavelet are splines of the same order.

**Theorem 7.2 1.** *If the prediction filter in the lifting scheme $U_i^{2r}(z)$ is derived from a polynomial interpolatory spline of order $2r$ then the cascade algorithm for the construction of the synthesis scaling function $\varphi(t)$ converges for any $r$ and $\varphi(t) = L^{2r}(t)$, where $L^{2r}(t)$ is the fundamental spline of order $2r$ with nodes on the grid $\{k\}_{k\in\mathbb{Z}}$. This spline was defined in (3.20).*

**2.** *If, in addition, the update filter $V(z)$ is chosen as suggested in Section 5 then the synthesis wavelet $\psi(t)$ is the spline $\sigma^{2r}(t)$, which interpolates the impulse response $\{g_k\}$ of the synthesis high-pass filter*

$$G(z) \triangleq \sqrt{2}z^{-1}\left(1 - \frac{zV(z^2)}{2}(1 + z^{-1}U_i^{2r}(z^2)\right) \tag{7.29}$$

*on the grid $\{k/2\}$, $k \in \mathbb{Z}$.*

**Proof:**

**1.** We consider a subdivision scheme $S_a$ with symbol $a(z) = 1 + z^{-1}U_i^{2r}(z^2)$ derived from an interpolatory spline of degree $2r$. Let the initial data be $\mathbf{f^0} = \{\delta_k\}$ and $\{f^j(t)\},\ j = 0, 1, \ldots$ be a sequence of splines of order $2r$ constructed on the grid $\{2^{-j}k\}$ that interpolate the subsequently refined data at grid points, that is, $f^j(2^{-j}k) = f_k^j,\ k \in \mathbb{Z}$. We show by induction that for all non-negative integers $j$ the splines $f^j(t) = L^{2r}(t)$. Obviously, it is true for $f^0(t)$.

Suppose that for some $j \in \mathbb{N}$ we have $f^{j-1}(t) = L^{2r}(t)$. Due to the well known property of minimal norm [2], the integral

$$\mu \triangleq \int_{-\infty}^{\infty} |(L^{2r})^{(r)}(t)|^2\, dt \le \int_{-\infty}^{\infty} |q^{(r)}(t)|^2\, dt,$$

where $q(t)$ is any function such that $q^{(r)}(t)$ is square summable and $q(k) = \delta_k$. The refined data is $f_k^j = f^{j-1}(2^{-j}k) = L^{2r}(2^{-j}k)$. For the spline $f^j(t)$, which interpolates this data, we have the inequality

$$\nu \triangleq \int_{-\infty}^{\infty} |(f^j)^{(r)}(t)|^2\, dt \le \int_{-\infty}^{\infty} |Q^{(r)}(t)|^2\, dt,$$

where $Q(t)$ is any function such that $Q^{(r)}(t)$ is square summable and $Q(2^{-j}k) = L^{2r}(2^{-j}k)$. Hence, $\nu \le \mu$. On the other hand, $f_k^j = \delta_k$ and, therefore, $\mu \le \nu$. Thus,

$$\int_{-\infty}^{\infty} |(L^{2r})^{(r)}(t)|^2\, dt = \int_{-\infty}^{\infty} |(f^j)^{(r)}(t)|^2\, dt.$$

Hence, it follows that $f^j(t) \equiv L^{2r}(t)$. Thus the first claim of the theorem is proved.

**2.** Under conditions of the theorem the synthesis wavelet $\psi(t)$ is derived from the scaling function $\varphi(t) = L^{2r}(t)$ via the two-scale equation

$$\psi(t) = \sum_{k \in \mathbb{Z}} g_k L^{2r}(2t - k), \tag{7.30}$$

where $\{g_k\}$ is the impulse response of the high-pass filter $G$. But the series in (7.30) represents a spline of order $2r$, which interpolates $\{g_k\}$ on the grid $\{k/2\},\ k \in \mathbb{Z}$.

■

**Remark.** Note that the analysis scaling function and wavelet are not splines. Also for the schemes that use splines of odd order, the synthesis as well as the analysis scaling functions and wavelets are not splines. However, due to the super-convergence property, in this case the synthesis scaling function and wavelet are smoother then the generating splines.

## 7.6 Evaluation of the subdivision mask coefficients via the discrete Fourier transform

The above propositions yield a practical algorithm that establishes the convergence of a subdivision scheme and analyzes its regularity. The key operation is the evaluation of sums of type (7.8) of the mask coefficients. For subdivision schemes with finite masks these sums can be calculated directly. But for infinite masks different methods of evaluation of the coefficients are required.

Again we consider the case when the number $L$ in Eq. (7.8) is 1. The cases $L > 1$ are similarly treated. We assume that $N = 2^p$, $p \in \mathbb{N}$ and $\sum_k^p$ stands for $\sum_{k=-N/2}^{N/2-1}$. The discrete Fourier transform (DFT) of an array $\mathbf{x}^p = \{x_k^p\}_{k=-N/2}^{N/2-1}$ and its inverse (IDFT) are

$$\widehat{x}_n^p = \sum_k^p e^{-2\pi ikn/N}\, x_k^p \quad \text{and } x_k^p = \frac{1}{N}\sum_n^p e^{2\pi ikn/N}\, \widehat{x}_n^p.$$

As before, $y(z)$ denotes the $z$-transform of a sequence $\{y_k\} \in l^1$. We assume that $z = e^{-i\omega}$.

The coefficients of the masks that we deal with are evaluated as follows:

$$|a_k| \le a\gamma^k \Rightarrow \sum_{k=N}^{\infty} |a_k| \le B\gamma^N,\ B = \frac{a}{1-\gamma}, \tag{7.31}$$

where $0 < \gamma < 1$ and $a$ is some positive constant.

We need to evaluate the sums $\sigma_e(a) = \sum_{k=-\infty}^{\infty} |a_{2k}|$, $\quad \sigma_o(a) = \sum_{k=-\infty}^{\infty} |a_{2k+1}|$. We denote

$$A(\omega) = a(e^{-i\omega}) = Q(e^{-i\omega})/P(e^{-i\omega}) = \sum_{k=-\infty}^{\infty} e^{-i\omega k}\, a_k.$$

Let us calculate the function $A$ in the discrete set of points:

$$\widehat{a}_n = A\left(\frac{2\pi n}{N}\right) = \sum_{k=-\infty}^{\infty} e^{\frac{-2\pi ikn}{N}}\, a_k = \sum_{r=-N/2}^{N/2-1} e^{\frac{-2\pi irn}{N}}\, \theta_r,$$

$$\theta_r = \sum_{l=-\infty}^{\infty} a_{r+lN} = a_r + \kappa_r, \quad \kappa_r = \sum_{l\in\mathbb{Z}/0} a_{r+lN}.$$

It follows from (7.31) that

$$|\kappa_r| \le 2B\alpha^N \Rightarrow |a_r| = |\theta_r| + \alpha_r^N, \quad |\alpha_r^N| \le 2B\gamma^N. \tag{7.32}$$

The samples $\theta_k$ are available via the IDFT: $\theta_k = \frac{1}{N}\sum_n^p e^{2\pi ikn/N}\,\widehat{a}_n$. Using (7.32) we can evaluate the sums we are interested in as follows:

$$\sigma_e(a) = \sum_{k=-N/4}^{N/4-1} |a_{2k}| + 2\sum_{k=N/4}^{\infty} |a_{2k}| = \sum_{r=-N/4}^{N/4-1} |\theta_{2k}| + \rho_N,$$

$$\rho_N = \sum_{r=-N/4}^{N/4-1} |\alpha_N(2k)| + 2\sum_{k=N/4}^{\infty} |a_{2k}|, \quad |\rho_N| \le B(N+2)\gamma^N.$$

Hence, it follows that by doubling $N$, we can approximate the infinite series $\sigma_e(a)$ by the finite sum $\sigma_e^N(a) = \sum_{r=-N/4}^{N/4-1} |\theta_{2k}|$, whose terms are available via the DFT. An appropriate value of $N$ can be found theoretically using estimations of the roots of the denominator $P(z)$. But practically, we can iterate calculations by gradually doubling $N$ until the result of calculating $\sigma_e^{2N}(a)$ becomes identical to $\sigma_e^N(a)$ (up to machine precision). The same approach is valid for evaluating the sum $\sigma_o(a)$ and the sums $\sum_k |q_{i-2^L k}^{[L]}|$ for any $L$.

# 8 Examples

In this section we outline the properties and present graphical illustrations of several particular wavelet transforms based on the prediction and update filters, which were designed in Sections 3 and 4. For this illustration we selected transforms that proved to be efficient in our image compression experiments. The results of these experiments are presented in Section 9.

## 8.1 List of filters

We describe transforms, which use the following six filters:

$F_1(z) \triangleq U_i^3(z) = 4\dfrac{1+z}{z+6+z^{-1}}$,

(IIR filter derived from a quadratic interpolatory spline, see (3.29))

$F_2(z) \triangleq U_m^3(z) = \dfrac{-z^{-1}+9+9z-z^2}{16}$

(FIR filter derived from a quadratic minimal quasi-interpolatory spline, see (3.39))

$$F_3(z) \triangleq U_d^6(z) = \frac{(z + 14 + z^{-1})(1 + z)}{6z^{-1} + 20 + 6z)}$$

(IIR filter derived from an interpolatory discrete spline of sixth degree, see (4.18))

$$F_4(z) \triangleq U_e^3(z) = \frac{3z^{-2} - 25z^{-1} + 150 + 150z - 25z^2 + 3z^3}{256}$$

(FIR filter derived from a quadratic extended quasi-interpolatory spline, see (3.40)),

$$F_5(z) \triangleq U_d^8(z) = \frac{8(1 + z)(z^{-1} + 6 + z)}{z^{-2} + 28z^{-1} + 70 + 28z + z^2}$$

(IIR filter derived from an interpolatory discrete spline of eighth degree, see (4.19)),

$$F_6(z) \triangleq U_i^5(z) = \frac{16(z + 10 + z^{-1})(1 + z)}{z^2 + 76z + 230 + 76z^{-1} + z^{-2}},$$

(IIR filter derived from an interpolatory spline of fourth degree, see (3.30)).

**Remark:** It follows from Remark 3.4.1 that the synthesis scaling function generated by the FIR filters $F^2(z)$ and $F_4(z)$ are the autocorrelations of the scaling functions generated by the 4-tap and 6-tap Daubechies filters, respectively (see [37]).

By combining the above six filters we construct a number of transforms, which we label as $\mathbf{P}_p\mathbf{U}_u$. Here $p$ means the index of the filter $F_k$, which is used for the *predict* step in the lifting scheme and $u$ is the index of the filter, which is used for the *update* step. For example, $\mathbf{P}_1\mathbf{U}_3$ designates the transform, which performs the *predict* step using the filter $F^1(z)$ and the *update* step using the filter $F_3(z)/z$.

**Proposition 8.1** *The synthesis $\varphi(x)$ and the analysis $\widetilde{\varphi}(x)$ scaling functions associated with the transforms $\boldsymbol{P}_p\boldsymbol{U}_u$ are continuous and*

**if the filters $F_u$ and $F_p$ are FIR** *then $\varphi(x)$ and $\widetilde{\varphi}(x)$ have compact support;*

**if $F_u$ is IIR and $F_p$ is FIR** *then $\widetilde{\varphi}(x)$ decays exponentially as $|x| \to \infty$ and $\varphi(x)$ has compact support;*

**if $F_p$ is IIR and $F_u$ is FIR** *then $\varphi(x)$ and $\widetilde{\varphi}(x)$ decay exponentially as $|x| \to \infty$;*

**if $F_u$ and $F_p$ are IIR** *then $\varphi(x)$ and $\widetilde{\varphi}(x)$ decay exponentially as $|x| \to \infty$.*

## 8.2 Analysis of regularity of the scaling function: Example

We illustrate the scheme of analysis of the convergence and regularity of the scaling function, that was described in Sections 7.2 and 7.4. As an example, we use the scaling function $\varphi(t)$ generated by the filter $F_1(z) = U_i^3(z)$. We recall that this filter originates from the quadratic interpolatory spline, which belongs to $C^1$. The symbol of the corresponding subdivision scheme is

$$a(z) = 1 + z^{-1}U_i^3(z^2) = \frac{(1+z)^4}{z^4+6z^2+1} = (1+z)\,q(z),\ q(z) = \frac{(1+z)^3}{z^4+6z^2+1}.$$

Obviously, $a(z)$ is a function of class $\mathbf{P}$. To establish the convergence, we have to prove that the scheme $S_q$ with the rational symbol $q(z)$ and the infinite mask $\{q_k\}$ is contractive. For this purpose, we evaluate the norms $\|S_q^L\| = \max\left\{\sum_k |q^{[L]}_{i-2^L k}|\right\}$ using the DFT as it was explained in Section 7.6. We begin with $L = 1$. In this case

$$\widehat{q}_n^{[1]} = q(e^{-2\pi i n/N}) = \frac{2e^{\frac{i\pi n}{N}}\cos^3\frac{i\pi n}{N}}{1+\cos^2\frac{2\pi n}{N}}.$$

The sums $\sum_{k=-\infty}^{\infty}|q^{[1]}_{i-2k}| \simeq \sum_{r=-N/4}^{N/4-1}|\theta_{2k+i}|$, $i = 0, 1$, provided $N$ is sufficiently large. The values $\theta_k$ are calculated via the IDFT: $\theta_k = N^{-1}\sum_n^p e^{2\pi i k n/N}\,\widehat{q}_n^{[1]}$. Direct calculation yields the estimate: $\|S_q^1\| \le 0.7071$. Thus, the scheme converges and there exists a continuous scaling function $\varphi$.

To establish the differentiability of the limit function $f^\infty$ of the scheme $S_a$ we have to prove that the scheme $S_q^{[1]}$ with the symbol $q^{[1]}(z) = 2(1+z)^{-2}\,a(z)$ is contractive. The norm of the operator $S_q^{[1]}$ does not meet the requirement $\|S_q^{[1]}\| < 1$. But, using two iterations, $L = 2$, we found that $\|(S_q^{[1]})^2\| \le 0.6667$. Hence, the scaling function $\varphi \in C^1$.

But an even stronger claim is true: the scaling function $\varphi \in C^2$. To establish this, we prove that the scheme $S_q^{[2]}$ with the symbol $q^{[2]}(z) = 4(1+z)^{-3}\,a(z)$ is contractive. As in the previous case, our calculations with $L = 2$ lead to $\|(S_q^{[2]})^2\| \le 0.6667$, which proves the statement.

## 8.3 Summary of the properties of the wavelet transforms

We summarize in Table 2 the properties of the transforms, which we employed in our image compression experiments. In those experiments we compare the results produced by our

transforms with the results achieved by application of the popular 9/7 biorthogonal transform ([4]), which we denote by **B**9/7. Therefore, we add to the table the properties of **B**9/7.

The following abbreviations are used in the table:

**VmA:** Number of vanishing moments of the analysis wavelet $\widetilde{\psi}$.

**VmS:** Number of vanishing moments of the synthesis wavelet $\psi$.

**RegA:** Regularity of the analysis scaling function $\widetilde{\varphi}$. $C^k$ means that $\widetilde{\varphi}$ is continuous together with its $k$ derivatives.

**RegS:** Regularity of the synthesis scaling function $\varphi$.

**Add:** Number of additions per pixel in the implementation of one step of the two-dimensional transforms.

**Mult:** The same for multiplications.

**LFA:** Length of the analysis filter $\widetilde{H}$.

**LFS:** Length of the synthesis filter $H$.

| Transform | VmA | VmS | RegA | RegS | Add | Mult | LFA | LFS |
|---|---|---|---|---|---|---|---|---|
| $\mathbf{P}_1\mathbf{U}_1$ | 4 | 4 | $C^1$ | $C^2$ | 8 | 6 | $\infty$ | $\infty$ |
| $\mathbf{P}_2\mathbf{U}_2$ | 4 | 4 | $C^0$ | $C^1$ | 8 | 4 | 11 | 5 |
| $\mathbf{P}_3\mathbf{U}_3$ | 6 | 6 | $C^2$ | $C^4$ | 12 | 6 | $\infty$ | $\infty$ |
| $\mathbf{P}_4\mathbf{U}_4$ | 6 | 6 | $C^1$ | $C^2$ | 12 | 6 | 15 | 7 |
| $\mathbf{P}_1\mathbf{U}_3$ | 4 | 6 | $C^1$ | $C^4$ | 10 | 7 | $\infty$ | $\infty$ |
| $\mathbf{P}_2\mathbf{U}_4$ | 4 | 6 | $C^0$ | $C^2$ | 10 | 5 | 13 | 5 |
| $\mathbf{P}_5\mathbf{U}_5$ | 8 | 8 | $C^4$ | $C^5$ | 16 | 12 | $\infty$ | $\infty$ |
| $\mathbf{P}_6\mathbf{U}_6$ | 6 | 6 | $C^2$ | $C^4$ | 16 | 12 | $\infty$ | $\infty$ |
| **B**7/9 | 4 | 4 | $C^0$ | $C^1$ | 8 | 4 | 9 | 7 |

Table 2: Properties of the transforms employed in the image compression experiments.

**Comments on Table 2.**

- The computational complexity is calculated under the assumption that all the presented transforms are carried out through lifting steps and filtering with IIR filters is implemented in cascade mode with special treatment of the boundaries of the images (Section 6).

- Factorization of the $\mathbf{B}9/7$ transform as suggested in [5, 18], speeds up the computation. The number of operations in Table 2 is computed with respect to this factorization algorithm. The two-dimensional wavelet transforms are applied separately to the rows and columns of the 2D array. However, for the lifting implementation of 2D $\mathbf{P}_2\mathbf{U}_2$, $\mathbf{P}_4\mathbf{U}_4$ and $\mathbf{B}9/7$ transforms, which use FIR filters, it is possible to merge row and column operations. This merge reduces the computational cost of the implementation ([5], Appendix I - Section 10).

- It is clear from the table that the cost of the implementation of the transforms $\mathbf{P}_1\mathbf{U}_1$, $\mathbf{P}_3\mathbf{U}_3$ and $\mathbf{P}_1\mathbf{U}_3$ is close to the cost of the implementation of the transform $\mathbf{B}9/7$.

## 8.4 Graphical illustrations

In the next seven figures we display several filters, scaling functions and wavelets associated with the presented transforms. All the figures are identically organized. Each of them contains four pictures. If these pictures are counted from left to right then the first column displays the frequency responses of the low-pass $H$ and the high-pass $G$ synthesis filters, the second column displays the frequency responses of the low-pass $\widetilde{H}$ and the high-pass $\widetilde{G}$ analysis filters, the third column displays the synthesis scaling function $\varphi$ and the wavelet $\psi$, and the fourth column displays the analysis scaling function $\widetilde{\varphi}$ and the wavelet $\widetilde{\psi}$. Figure 1 illustrates the transform $\mathbf{P}_1\mathbf{U}_1$, Figure 2 illustrates the transform $\mathbf{P}_2\mathbf{U}_2$, Figure 3 illustrates the transform $\mathbf{P}_3\mathbf{U}_3$, Figure 4 illustrates the transform $\mathbf{P}_4\mathbf{U}_4$, Figure 5 illustrates the transform $\mathbf{P}_5\mathbf{U}_5$ , Figure 6 illustrates the transform $\mathbf{P}_6\mathbf{U}_6$ and Figure 7 illustrates the transform $\mathbf{B}9/7$.

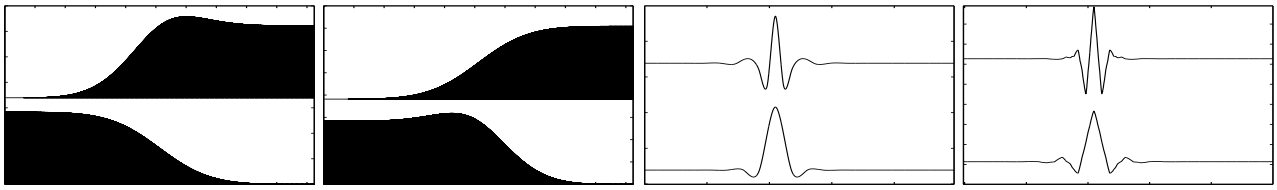

Figure 1: Filters and wavelets associated with the $\mathbf{P}_1\mathbf{U}_1$ transform.

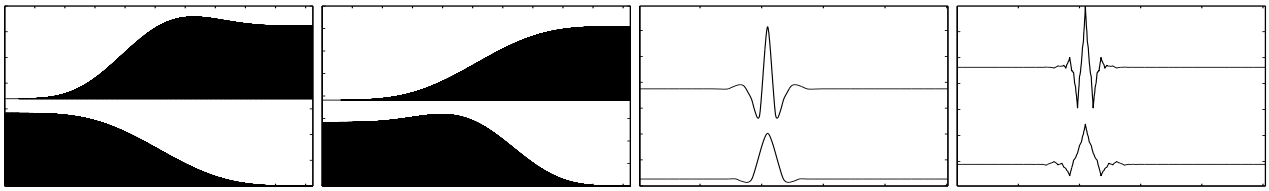

Figure 2: Filters and wavelets associated with the $\mathbf{P}_2\mathbf{U}_2$ transform.

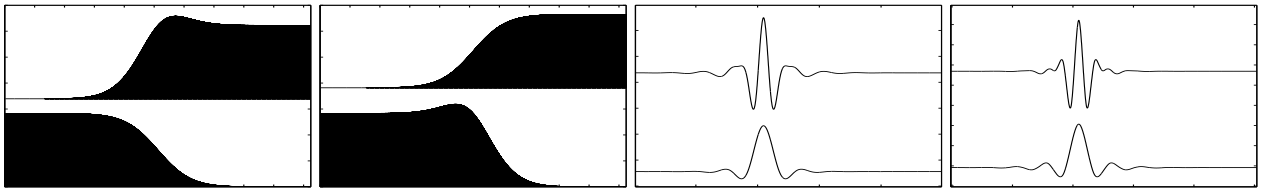

Figure 3: Filters and wavelets associated with the $\mathbf{P}_3\mathbf{U}_3$ transform.

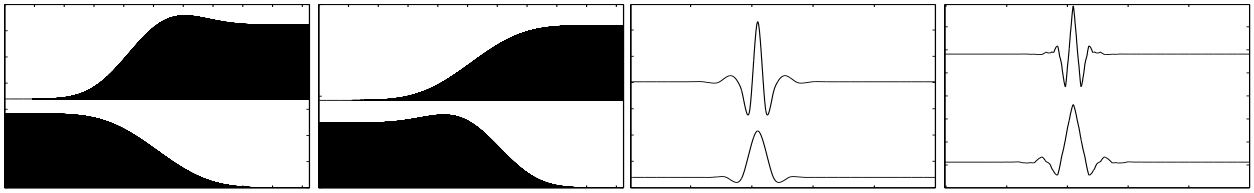

Figure 4: Filters and wavelets associated with the $\mathbf{P}_4\mathbf{U}_4$ transform.

**Comments on Figures 1–7:** We observe that, unlike **B**9/7, all the devised low-pass synthesis filters and high-pass analysis filters have flat frequency responses. It is especially apparent for the IIR filters. The rest of the filters are flat up to a small bump near the cut-off. The filters of the transform $\mathbf{P}_5\mathbf{U}_5$ have the flattest frequency response and the steepest cut-off. Its scaling functions are most regular among the presented transforms.

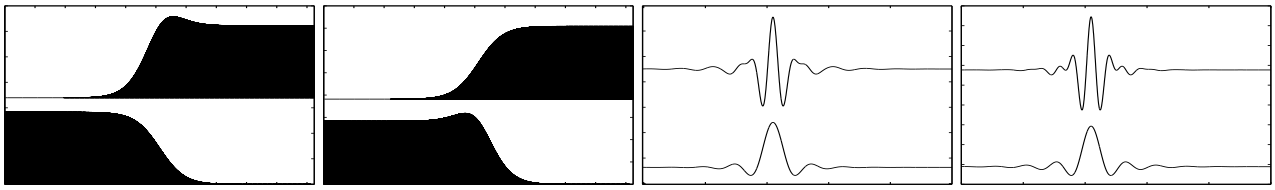

Figure 5: Filters and wavelets associated with the $\mathbf{P}_5\mathbf{U}_5$ transform.

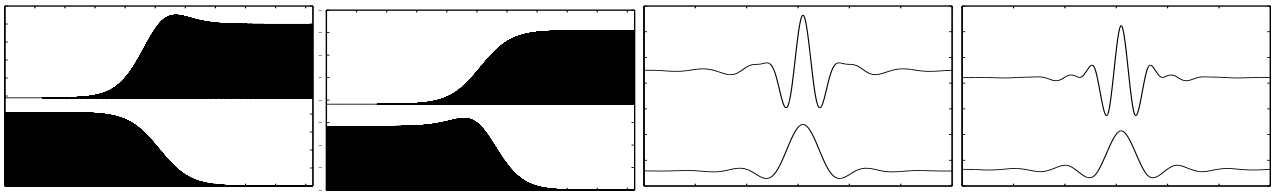

Figure 6: Filters and wavelets associated with the $\mathbf{P}_6\mathbf{U}_6$ transform.

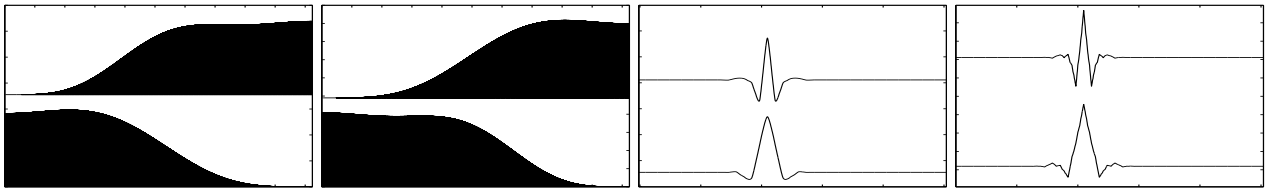

Figure 7: Filters and wavelets associated with the $\mathbf{B}9/7$ transform.

In Figures 8 and 9 we present the derivatives of several scaling functions. Figure 8 displays second derivatives of the synthesis scaling functions of two transforms $\mathbf{P}_2\mathbf{U}_2$ and $\mathbf{P}_4\mathbf{U}_4$, which use FIR filters, and the transform $\mathbf{P}_1\mathbf{U}_1$, which uses IIR filters. Like $\mathbf{P}_2\mathbf{U}_2$, the transform $\mathbf{P}_1\mathbf{U}_1$ operates with wavelets, which have four vanishing moments. The wavelets of the $\mathbf{P}_4\mathbf{U}_4$ transform have four vanishing moments. Note that the synthesis scaling functions of $\mathbf{P}_2\mathbf{U}_2$ and $\mathbf{P}_4\mathbf{U}_4$ are the basic limit functions of the well-known 4-point and 6-point Dubuc and Deslauriers interpolatory subdivision schemes [19, 20].

The second derivative of the synthesis scaling function of the $\mathbf{P}_2\mathbf{U}_2$ transform does not exist. The synthesis scaling function of the $\mathbf{P}_4\mathbf{U}_4$ transform belongs to $C^\alpha$, $\alpha < 2.830$ [16, 19, 20]. In Figure 8 the second derivative of the synthesis scaling function of the $\mathbf{P}_1\mathbf{U}_1$ transform looks smoother than the scaling function of $\mathbf{P}_4\mathbf{U}_4$. Thus, we conjecture that the

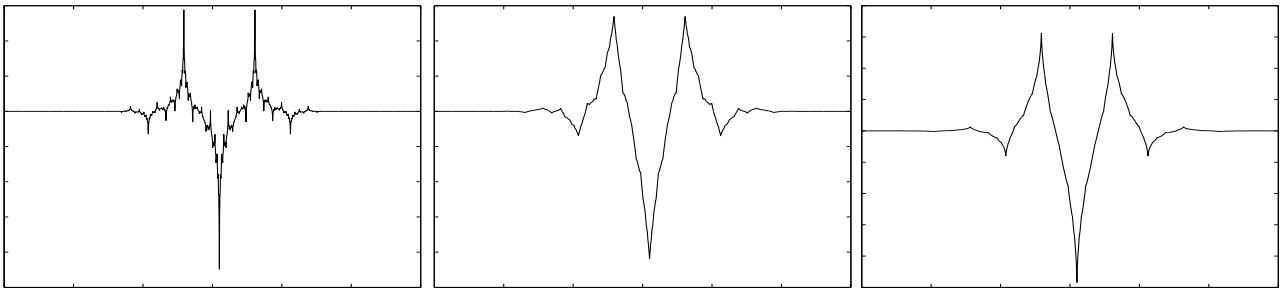

Figure 8: Second derivatives of the synthesis scaling functions: left: $\mathbf{P}_2\mathbf{U}_2$, center: $\mathbf{P}_4\mathbf{U}_4$, right: $\mathbf{P}_1\mathbf{U}_1$.

scaling function of $\mathbf{P}_1\mathbf{U}_1$ belongs to $C^\beta,\ \beta > \alpha$.

In Figure 9 we display the fourth derivatives of the synthesis scaling functions of two transforms $\mathbf{P}_3\mathbf{U}_3$ and $\mathbf{P}_6\mathbf{U}_6$, which use IIR filters. We recall that $\mathbf{P}_3\mathbf{U}_3$ is based on discrete splines of sixth order and $\mathbf{P}_6\mathbf{U}_6$ is based on polynomial splines of fifth order. Corresponding wavelets have six vanishing moments.

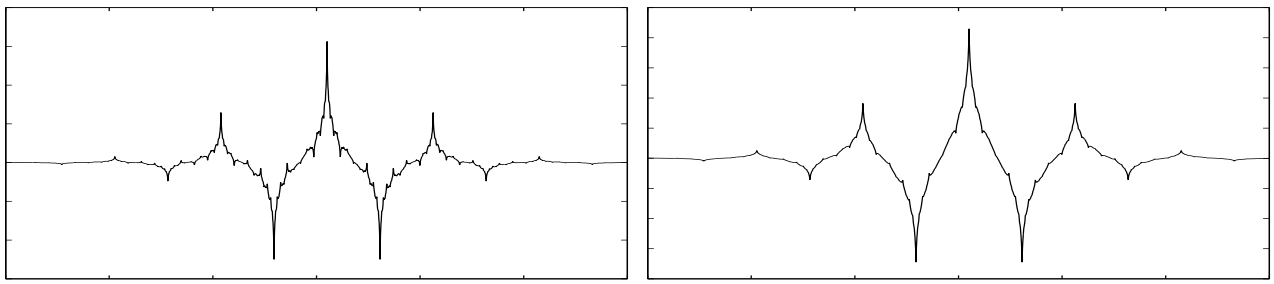

Figure 9: Fourth derivatives of the synthesis scaling functions: left: $\mathbf{P}_3\mathbf{U}_3$, right: $\mathbf{P}_6\mathbf{U}_6$.

We observe that the fourth derivative of the scaling function of $\mathbf{P}_3\mathbf{U}_3$ has a near-fractal appearance. Nevertheless, we proved that it is continuous. Both transforms $\mathbf{P}_3\mathbf{U}_3$ and $\mathbf{P}_6\mathbf{U}_6$ have very similar properties. Also, their performance is very similar. But the computational cost to implement the $\mathbf{P}_6\mathbf{U}_6$ transform is much higher than to implement $\mathbf{P}_3\mathbf{U}_3$, and is equal to that of implementing the $\mathbf{P}_5\mathbf{U}_5$ transform, which has smoother scaling functions and wavelets with more vanishing moments.

# 9 Image compression results

We tested the applicability of the devised transforms to image compression. We report the results of a number of experiments on four still benchmark images. It turns out that the performance of our transforms is comparable with the performance of the $\mathbf{B}9/7$ transform, which is in the JPEG 2000 compression standard [35]. In this section we apply the transforms to the images presented in Figures 10 and 11. These are 512×512 8 bit per pixel (8bpp) images. The following experiments were conducted:


Figure 10: Left: "Lena", right: "Barbara".

1. Each image was decomposed up to six scales with the wavelet transform using the $\mathbf{B}9/7$ transform and the transforms $\mathbf{P}_p\mathbf{U}_u$, $p, u = 1, \ldots, 6$, that are listed in Table 2.

2. The transforms' coefficients were coded using the SPIHT algorithm ([36]). This algorithm enables us to achieve an exact predetermined compression ratio. We coded the coefficients with different compression ratios (CR): 1:10 (0.8 bpp), 1:20 (0.4 bpp), 1:30 (4/15 bpp), 1:40 (0.2 bpp) and 1:50 (4/25 bpp).

3. The reconstructed image was compared with the original image and the peak signal-

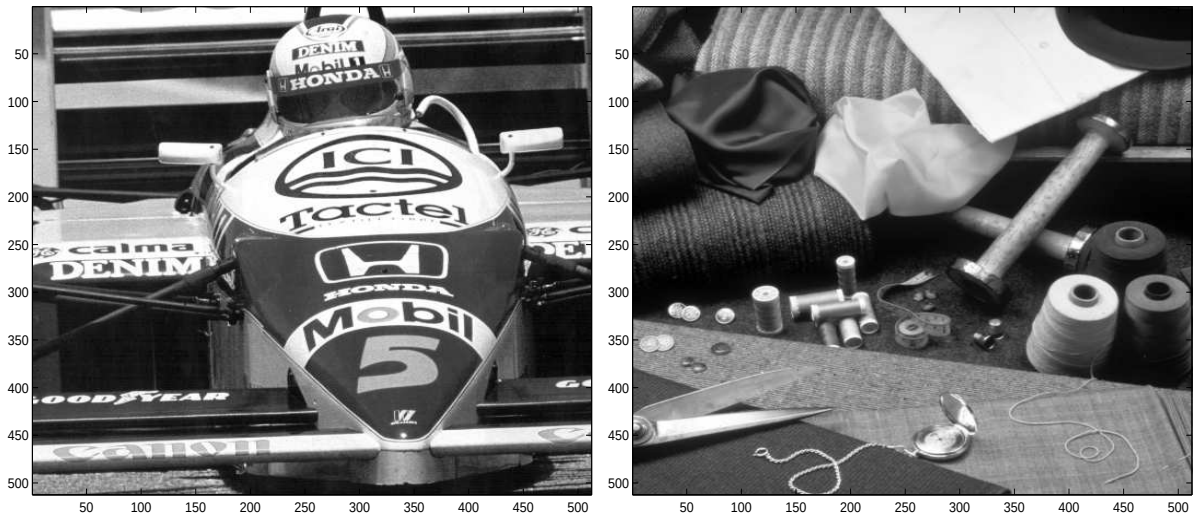


Figure 11: Left: "Car", right: "Fabrics".

to-noise ratio (PSNR) in decibels was computed:

$$PSNR = 10 \log_{10} \left( \frac{N\, 255^2}{\sum_{k=1}^{N} (x_k - \tilde{x}_k)^2} \right) \; dB. \tag{9.1}$$

**Lena:** The PSNR values of "Lena" are presented in Table 3. All the $\mathbf{P}_p\mathbf{U}_u$ transforms

| CR | $\mathbf{B}9/7$ | $\mathbf{P}_1\mathbf{U}_1$ | $\mathbf{P}_2\mathbf{U}_2$ | $\mathbf{P}_3\mathbf{U}_3$ | $\mathbf{P}_4\mathbf{U}_4$ | $\mathbf{P}_1\mathbf{U}_3$ | $\mathbf{P}_4\mathbf{U}_2$ | $\mathbf{P}_4\mathbf{U}_3$ | $\mathbf{P}_5\mathbf{U}_5$ | $\mathbf{P}_6\mathbf{U}_6$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 37.70 | 37.82 | 37.67 | 37.85 | 37.79 | 37.85 | 37.71 | 37.84 | 37.80 | 37.83 |
| 20 | 34.53 | 34.71 | 34.70 | 34.72 | 34.68 | 34.75 | 34.63 | 34.73 | 34.67 | 34.71 |
| 30 | 32.33 | 32.62 | 32.56 | 32.69 | 32.59 | 32.65 | 32.57 | 32.67 | 32.75 | 32.73 |
| 40 | 31.42 | 31.63 | 31.52 | 31.67 | 31.61 | 31.67 | 31.54 | 31.68 | 31.60 | 31.66 |
| 50 | 30.70 | 30.91 | 30.82 | 30.93 | 30.90 | 30.95 | 30.85 | 30.96 | 30.87 | 30.93 |

Table 3: PSNR of the "Lena" image.

have a slightly better PSNR than the $\mathbf{B}9/7$ transform for most of the compression ratios except when $\mathbf{P}_2\mathbf{U}_2$ is used at CR=10. The complexity of the $\mathbf{P}_2\mathbf{U}_2$ and $\mathbf{B}9/7$ transforms is the same. The transforms $\mathbf{P}_1\mathbf{U}_3$, $\mathbf{P}_4\mathbf{U}_3$ and $\mathbf{P}_3\mathbf{U}_3$ perform equally and outperform other transforms. All these transforms use $F_3$ as the update filter. In

Figure 12 we display a zoom from the reconstructed "Lena" (compressed to 1:40). In the compression we used the $\mathbf{B}9/7$ and $\mathbf{P}_4\mathbf{U}_3$ transforms. Note that, although both portions are almost indistinguishable, there are details which are better restored by the $\mathbf{P}_4\mathbf{U}_3$ transform.

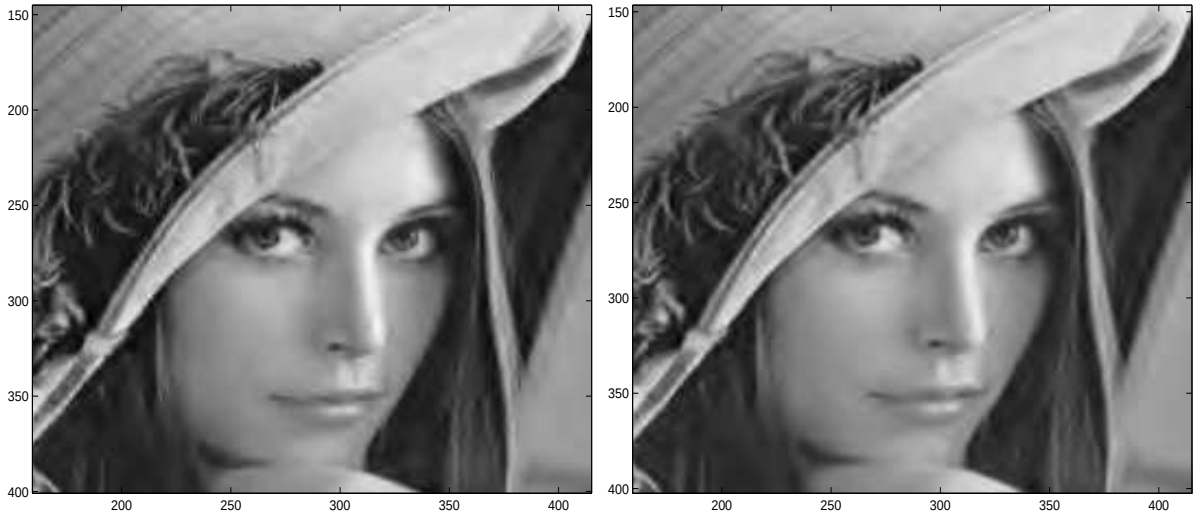


Figure 12: Fragment of "Lena" reconstructed from 40:1 compression. Left: the $\mathbf{B}7/9$ transform was used: PSNR=31.42. Right: $\mathbf{P}_4\mathbf{U}_3$ transform was used: PSNR=31.68.

**Barbara:** The PSNR values of "Barbara" are presented in Table 4. We observe from Table

| CR | $\mathbf{B}9/7$ | $\mathbf{P}_1\mathbf{U}_1$ | $\mathbf{P}_2\mathbf{U}_2$ | $\mathbf{P}_3\mathbf{U}_3$ | $\mathbf{P}_4\mathbf{U}_4$ | $\mathbf{P}_1\mathbf{U}_3$ | $\mathbf{P}_4\mathbf{U}_2$ | $\mathbf{P}_4\mathbf{U}_3$ | $\mathbf{P}_5\mathbf{U}_5$ | $\mathbf{P}_6\mathbf{U}_6$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 33.01 | 33.39 | 32.91 | 33.72 | 33.32 | 33.51 | 33.06 | 33.46 | 33.81 | 33.75 |
| 20 | 28.93 | 29.13 | 28.90 | 29.32 | 29.10 | 29.24 | 28.99 | 29.22 | 29.39 | 29.34 |
| 30 | 26.99 | 27.03 | 26.91 | 27.33 | 27.17 | 27.15 | 27.04 | 27.15 | 27.45 | 27.40 |
| 40 | 25.78 | 25.74 | 25.65 | 25.95 | 25.74 | 25.83 | 25.69 | 25.78 | 26.06 | 25.99 |
| 50 | 25.10 | 25.00 | 24.86 | 25.14 | 24.95 | 25.05 | 24.90 | 25.02 | 25.23 | 25.18 |

Table 4: PSNR of the "Barbara" image.

4 that the transforms $\mathbf{P}_3\mathbf{U}_3$ and $\mathbf{P}_6\mathbf{U}_6$, where the wavelets have six vanishing moments and the scaling functions belong to $C^4$, perform better than the transforms with four

vanishing moments and less regular scaling functions. The $\mathbf{P}_5\mathbf{U}_5$ transform, where both analysis and synthesis wavelets have eight vanishing moments and the scaling functions belong to $C^4$ and $C^5$, respectively, produces the best results. In Figure 13 we display a zoom from the reconstructed "Barbara" (compressed to 1:40). In the compression we used the $\mathbf{B}9/7$ and $\mathbf{P}_5\mathbf{U}_5$ transforms. Note that the texture of the tablecloth and the scarf is better revealed with the $\mathbf{P}_5\mathbf{U}_5$ transform while the leg of the table is displayed more accurately with the $\mathbf{B}9/7$ transform.

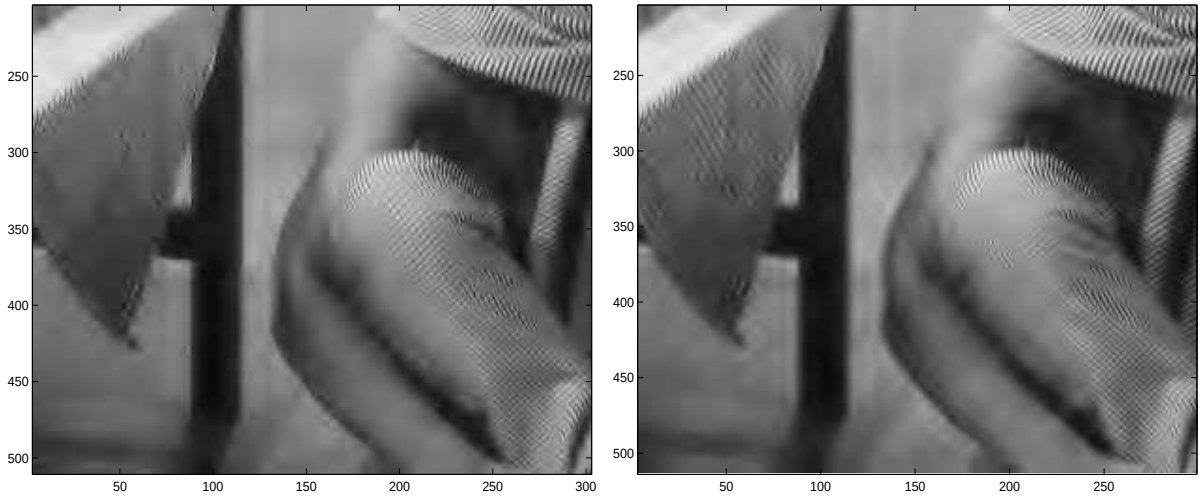


Figure 13: Fragment from "Barbara" reconstructed from 40:1 compression. Left: $\mathbf{B}9/7$ filter was used: PSNR=25.78. Right: $\mathbf{P}_5\mathbf{U}_5$ transform was used: PSNR=26.06.

**Car:** The PSNR values of the "Car" image are presented in Table 5. Here most of the $\mathbf{P}_p\mathbf{U}_u$ transforms produce a slightly better PSNR than the $\mathbf{B}9/7$ transform for any compression ratio. More regular transforms $\mathbf{P}_5\mathbf{U}_5$ and $\mathbf{P}_6\mathbf{U}_6$ perform worse than the transforms $\mathbf{P}_1\mathbf{U}_1$ and $\mathbf{P}_1\mathbf{U}_3$ where the wavelets have less vanishing moments but better spatial localization. The reason for that may lie in the presence of small details in the image, which are better captured by the wavelets with a shorter effective support. We display a zoom from the reconstructed "Car" (compressed to 1:40) in Figure 14. The $\mathbf{B}9/7$ and $\mathbf{P}_1\mathbf{U}_3$ transforms were used. While the horizontal lines on the left-hand side of the fragment are better displayed in the $\mathbf{B}9/7$ picture, the $\mathbf{P}_1\mathbf{U}_3$ transform is

| CR | $\mathbf{B}9/7$ | $\mathbf{P}_1\mathbf{U}_1$ | $\mathbf{P}_2\mathbf{U}_2$ | $\mathbf{P}_3\mathbf{U}_3$ | $\mathbf{P}_4\mathbf{U}_4$ | $\mathbf{P}_1\mathbf{U}_3$ | $\mathbf{P}_4\mathbf{U}_2$ | $\mathbf{P}_4\mathbf{U}_3$ | $\mathbf{P}_5\mathbf{U}_5$ | $\mathbf{P}_6\mathbf{U}_6$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 32.57 | 32.63 | 32.72 | 32.52 | 32.59 | 32.66 | 32.69 | 32.64 | 32.35 | 32.49 |
| 20 | 28.38 | 28.53 | 28.46 | 28.46 | 28.47 | 28.53 | 28.49 | 28.51 | 28.32 | 28.42 |
| 30 | 26.78 | 26.99 | 26.89 | 26.92 | 26.94 | 27.02 | 26.95 | 27.0 | 26.83 | 26.90 |
| 40 | 25.05 | 25.19 | 25.11 | 25.15 | 25.17 | 25.22 | 25.16 | 25.20 | 25.12 | 25.14 |
| 50 | 24.40 | 24.52 | 24.43 | 24.46 | 24.49 | 24.56 | 24.48 | 24.55 | 24.37 | 24.44 |

Table 5: PSNR of the "Car" image.

slightly advantageous in restoring the arcs and the letters on the right-hand side.

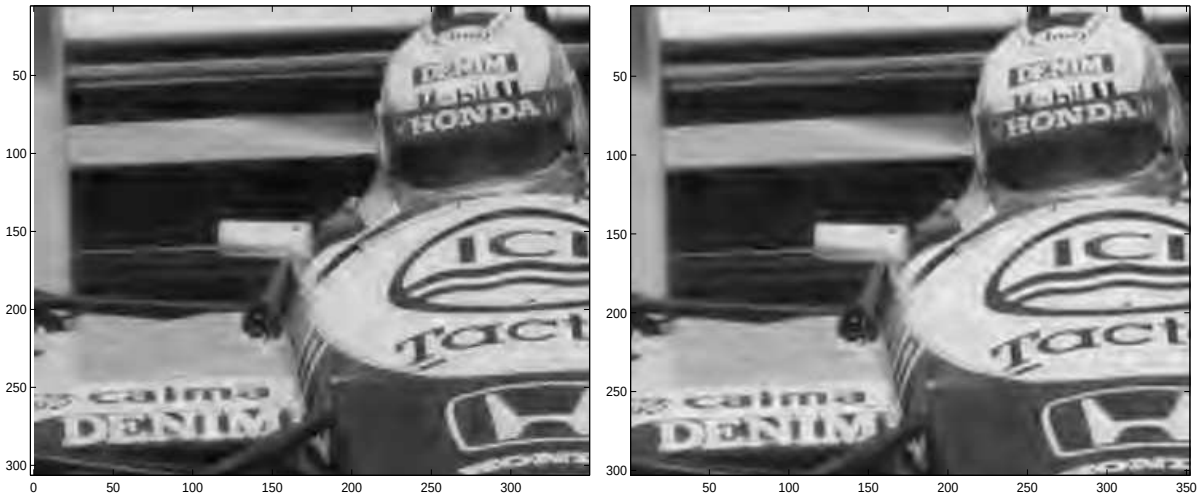


Figure 14: Fragment of "Car" reconstructed from 40:1 compression. Left: B9/7 filter was applied: PSNR=25.05. Right: $\mathbf{P}_1\mathbf{U}_3$ transform: PSNR=25.22.

**Fabric:** The PSNR values of "Fabric"are presented in Table 6. As in the above examples, the transform $\mathbf{P}_1\mathbf{U}_1$ produces a better PSNR than the $\mathbf{B}9/7$ transform for any compression ratio. The same is true for more complicated transforms such as $\mathbf{P}_1\mathbf{U}_3$ and $\mathbf{P}_4\mathbf{U}_3$. The latter transform produces the best PSNR among all the listed transforms. In Figure 15 we display a zoom from the reconstructed "Fabric" (compressed to 1:40). In the compression we used the transforms $\mathbf{B}9/7$ and $\mathbf{P}_4\mathbf{U}_3$. Although it is difficult to

| CR | $\mathbf{B}9/7$ | $\mathbf{P}_1\mathbf{U}_1$ | $\mathbf{P}_2\mathbf{U}_2$ | $\mathbf{P}_3\mathbf{U}_3$ | $\mathbf{P}_4\mathbf{U}_4$ | $\mathbf{P}_1\mathbf{U}_3$ | $\mathbf{P}_4\mathbf{U}_2$ | $\mathbf{P}_4\mathbf{U}_3$ | $\mathbf{P}_5\mathbf{U}_5$ | $\mathbf{P}_6\mathbf{U}_6$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 34.95 | 34.96 | 34.86 | 34.97 | 34.96 | 34.99 | 34.89 | 34.99 | 34.92 | 34.96 |
| 20 | 31.53 | 31.53 | 31.44 | 31.53 | 31.51 | 31.55 | 31.47 | 31.57 | 31.47 | 31.53 |
| 30 | 29.62 | 29.79 | 29.73 | 29.75 | 29.77 | 29.81 | 29.77 | 29.82 | 29.83 | 29.74 |
| 40 | 28.90 | 29.00 | 28.93 | 29.02 | 28.98 | 29.02 | 28.97 | 29.04 | 28.99 | 29.02 |
| 50 | 28.41 | 28.52 | 28.44 | 28.52 | 28.50 | 28.54 | 28.46 | 28.55 | 28.47 | 28.51 |

Table 6: PSNR of the "Fabric" image.

distinguish between them, it seems that $\mathbf{P}_4\mathbf{U}_3$ produces a superior visual display such as the scissors and the texture of the tablecloth.

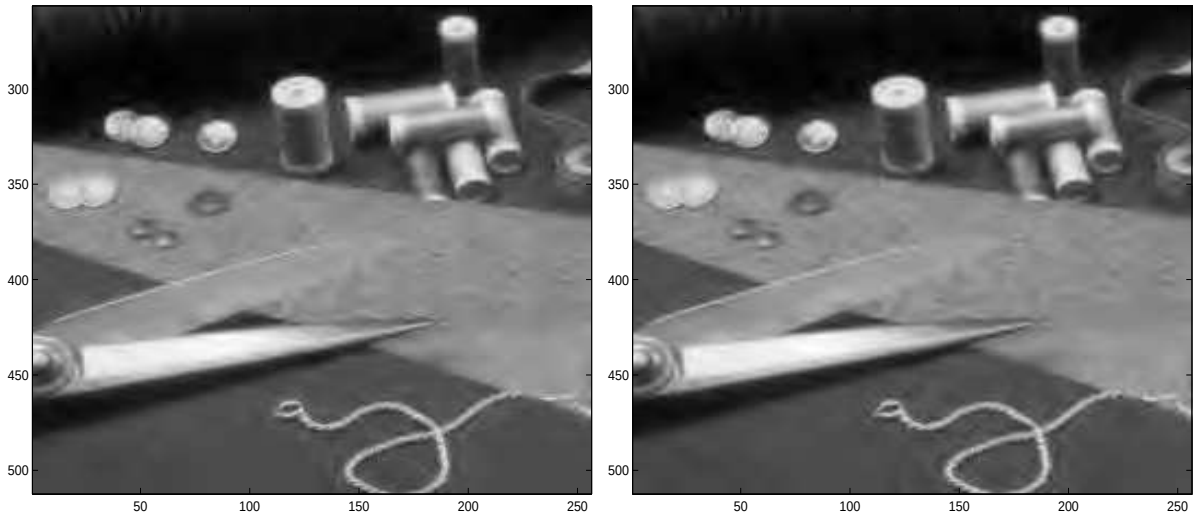


Figure 15: Fragment of "Fabric" reconstructed from 40:1 compression. Left: $\mathbf{B}9/7$ transform was applied: PSNR=28.90. Right: $\mathbf{P}_4\mathbf{U}_3$ transform: PSNR=29.04.

The $\mathbf{P}_3\mathbf{U}_3$ transform, which uses filters that are derived from the discrete spline of sixth order, produces a high PSNR on all test images. Almost identical results are displayed by the $\mathbf{P}_6\mathbf{U}_6$ transform, which stems from polynomial splines of fourth degree, but the computational cost of the $\mathbf{P}_6\mathbf{U}_6$ implementation is significantly higher than the cost of the implementation of $\mathbf{P}_3\mathbf{U}_3$. But there are singular cases when it is outperformed. The $\mathbf{P}_1\mathbf{U}_1$

transform combines low computational cost with a good performance. It uses filters that are derived from the quadratic interpolatory spline. The performance of the transforms $\mathbf{P}_1\mathbf{U}_3$ and $\mathbf{P}_4\mathbf{U}_3$ suggests that in some cases it is better to use different filters for the *predict* and *update* steps. The transform $\mathbf{P}_5\mathbf{U}_5$, which uses filters that are derived from the discrete spline of eighth order produces good results for the compression of "Barbara". Therefore, it justifies the use of highly regular wavelets with a high number of vanishing moments.

# Conclusion

In this paper we proposed an efficient technique that generates a wide range of new biorthogonal symmetric wavelet transforms. This technique is based on using discrete and polynomial interpolatory and quasi-interpolatory splines for the design of filters for the *predict* and *update* operations in lifting schemes of the wavelet transform. These are the linear phase filters which have flat frequency responses. By combining different designed filters for the *predict* and *update* steps, we can devise practically unlimited types of wavelets which have a predetermined number of vanishing moments that are as smooth as required. When transforms, which are based on splines of higher orders are implemented, it is advisable to switch from time-domain to frequency-domain implementation. Then, an increase in the number of vanishing moments and the regularity of wavelets does not affect the computational cost of the implementation.

We analyzed scaling functions and wavelets associated with the devised wavelet transforms. In particular, we established conditions for exponential decay of the scaling functions and determined the rate of decay. We found that the synthesis scaling functions derived from interpolatory polynomial splines of even degrees coincide with fundamental splines.

We explored the applicability of the newly designed transforms to still image compression. The new transforms and the biorthogonal 9/7 transform were incorporated into SPIHT in order to compare their performances. The performance (quality and computational cost) of the presented transforms proved to be comparable with the 9/7 transform. Most of the filters that are employed in the transforms are IIR.

A number of authors [4, 51] studied the performance of families of wavelet transforms that use FIR filters for image compression. The most comprehensive investigation on this topic

was reported in [27]. The results of these studies as well as the results of our experiments suggest that it is difficult to single out properties of the transforms that are most valuable for image compression. We believe that a trade-off between properties such as the number of vanishing moments of analysis wavelets, the regularity of synthesis wavelets, the flatness of the frequency response and the steepness of its transition band, and the spatial localization of the wavelets, is the key to achieve better compression. This is exemplified by the $\mathbf{B}9/7$ transform as well as by the transforms $\mathbf{P}_3\mathbf{U}_3$ and $\mathbf{P}_1\mathbf{U}_1$.

In the future we intend to find accurate estimates of smoothness (Hölder exponents) of the designed scaling functions and wavelets. We plan to extend our technique to the irregularly sampled signals. The designed transforms will be tested on processing video sequences and seismic data.

**Acknowledgement:** This research was partially supported by the grant, *Construction of spline based filter banks for the wavelet analysis and image processing application,* of Tel Aviv University.

# 10 Appendix I: 2D implementation of transforms with FIR filters

The 2D array $\mathbf{x} \stackrel{\Delta}{=} \{x(n,m)\}$, $n,m = 0 \dots N-1$, is to be transformed into four subarrays: $\mathbf{x} \rightarrow \mathbf{ss} \bigcup \mathbf{sd} \bigcup \mathbf{ds} \bigcup \mathbf{dd}$, where **ss** is the smoothed low-frequency section in horizontal and vertical directions, **sd** – low-frequency in vertical direction, high-frequency in horizontal direction, **ds** – low-frequency in horizontal direction, high-frequency in vertical direction, **dd** – high-frequency section of details in both horizontal and vertical directions. Conventionally, the 2D transform of **x** is implemented in a separable way (tensor product). In [5] a direct 2D lifting implementation of the transform $\mathbf{B}9/7$ is presented. We describe the 2D lifting implementation of the transform $\mathbf{P}_2\mathbf{U}_2$.

NOTATION: Let $\mathbf{a} = \{a(n,m)\}$, $n,m = 0 \dots M-1$, be a 2D array and $\tilde{\mathbf{a}}$ be a 2M-periodic **HH**- extension of the array in both vertical and horizontal directions. Then $\mathbf{L_r} \cdot \mathbf{a}$ means the left shift of the extended array. Namely, $L_r \cdot a(n,m) = \tilde{a}(n, m+r)$, $m = 0 \dots M-1$.

Similarly, the right shift is $\mathbf{R_r} \cdot \mathbf{a} = \{R_r \cdot a(n,m)\} = \{\tilde{a}(n, m-r)\},\ m = 0 \dots M-1$.

The upper shift is $\mathbf{U_r} \cdot \mathbf{a} = \{U_r \cdot a(n,m)\} = \{\tilde{a}(n+r, m)\},\ n = 0 \dots M-1$. The lower shift is $\mathbf{D_r} \cdot \mathbf{a} = \{D_r \cdot a(n,m)\} = \{\tilde{a}(n-r, m)\},\ n = 0 \dots M-1$.

## 10.1 $\mathbf{P_2U_2}$ transform

The parameters are: $\alpha = -9/16,\ \beta = 1/16,\ \gamma = -\alpha/2,\ \delta = -\beta/2$.

**Decomposition**

**Split:** $\mathbf{x} = \mathbf{ee} \bigcup \mathbf{eo} \bigcup \mathbf{oe} \bigcup \mathbf{oo}$.

**Predict:**
1. $\mathbf{de} = \mathbf{oe} + \alpha(\mathbf{ee} + \mathbf{U_1} \cdot \mathbf{ee}) + \beta(\mathbf{D_1} \cdot \mathbf{ee} + \mathbf{U_2} \cdot \mathbf{ee})$.
2. $\mathbf{dd} = \mathbf{oo} + \alpha(\mathbf{de} + \mathbf{L_1} \cdot \mathbf{de} + \mathbf{eo} + \mathbf{U_1} \cdot \mathbf{eo}) + \beta(\mathbf{R_1} \cdot \mathbf{de} + \mathbf{L_2} \cdot \mathbf{de} + \mathbf{D_1} \cdot \mathbf{eo} + \mathbf{U_2} \cdot \mathbf{eo})$.

**Update:**
1. $\mathbf{sd} = \mathbf{eo} + \alpha(\mathbf{ee} + \mathbf{L_1} \cdot \mathbf{ee}) + \beta(\mathbf{R_1} \cdot \mathbf{ee} + \mathbf{L_2} \cdot \mathbf{ee}) + \gamma(\mathbf{dd} + \mathbf{D_1} \cdot \mathbf{dd}) + \delta(\mathbf{U_1} \cdot \mathbf{dd} + \mathbf{D_2} \cdot \mathbf{dd})$
2. $\mathbf{ss} = \mathbf{ee} + \gamma(\mathbf{de} + \mathbf{D_1} \cdot \mathbf{de} + \mathbf{sd} + \mathbf{R_1} \cdot \mathbf{sd}) + \delta(\mathbf{U_1} \cdot \mathbf{de} + \mathbf{D_2} \cdot \mathbf{de} + \mathbf{L_1} \cdot \mathbf{sd} + \mathbf{R_2} \cdot \mathbf{sd})$.
3. $\mathbf{ds} = \mathbf{de} + \gamma(\mathbf{dd} + \mathbf{R_1} \cdot \mathbf{dd}) + \delta(\mathbf{L_1} \cdot \mathbf{dd} + \mathbf{R_2} \cdot \mathbf{dd})$.

**Normalization:** $\mathbf{ss} = 2\mathbf{ss},\ \mathbf{dd} = \mathbf{dd}/2$.

**Reconstruction** of the 2D array $\mathbf{x}$ from four subarrays of coefficients:
$\mathbf{ss} \bigcup \mathbf{sd} \bigcup \mathbf{ds} \bigcup \mathbf{dd}$, $\mathbf{ss} \to \mathbf{x}$ is conducted in a reverse order.

# 11 Appendix II: Periodic setting

A parallel version of the above theory can be developed for periodic signals using the discrete Fourier transform (DFT). The construction is carried out in the Fourier domain and calculations are performed via the fast Fourier transform (FFT). We outline briefly the periodic scheme. A detailed presentation of the scheme is given in [7, 8].

In the sequel we assume that $N = 2^j$, $j \in N$ and $\sum_k^j$ stands for $\sum_{k=0}^{2^j-1}$. Denote $\omega_j = e^{2\pi i 2^{-j}}$. The discrete Fourier transform (DFT) of an array $\mathbf{x^j} = \{x_k\}_{k=0}^{2^j-1}$ and its inverse (IDFT) are

$$\widehat{x}^j(n) = \sum_k^j \omega_j^{-nk}\, x_k, \qquad x_k = 2^{-j} \sum_n^j \omega_j^{nk}\, \widehat{x}^j(n).$$

Denote $e_k \stackrel{\Delta}{=} x_{2k},\ o_k \stackrel{\Delta}{=} x_{2k+1},\ k = 0 \ldots N/2-1$. We recall the following properties of DFT:

$$\begin{aligned}
&\text{If } x_k = \sum_l^j b_{k-l} c_l \text{ then } \widehat{x}^j(n) = \widehat{b}^j(n)\widehat{c}^j(n)\\
&\widehat{e}^{j-1}(n) = \frac{1}{2}\left(\widehat{x}^j(n) + \widehat{x}^j(n+N/2)\right)\\
&\widehat{o}^{j-1}(n) = \frac{\omega_j^n}{2}\left(\widehat{x}^j(n) - \widehat{x}^j(n+N/2)\right)\\
&\widehat{x}^j(n) = \widehat{e}^{j-1}(n) + \omega_j^{-n}\widehat{o}^{j-1}(n).
\end{aligned} \tag{11.1}$$

## 11.1 Biorthogonal periodic transforms

### 11.1.1 Primal lifting scheme

**One step of decomposition**

**Split -** The array $\mathbf{x^j}$ is split into even and odd sub-arrays: $\mathbf{x^j} = \mathbf{e^{j-1}} \bigcup \mathbf{o^{j-1}}$.

**Predict -** The even array $\mathbf{e^{j-1}}$ is filtered by some *prediction* filter $\mathbf{U^{j-1}}$ and subtracted from $\mathbf{o^{j-1}}$ producing the new odd array $\mathbf{o_\nu^{j-1}}$. In the Fourier domain the operations are described as follows:

$$\widehat{o_\nu}^{j-1}(n) = \widehat{o}^{j-1}(n) - \widehat{U}^{j-1}(n)\widehat{e}^{j-1}(n),$$

where the subscript $\nu$ designates the new array.

**Update (lifting) -**

$$\widehat{e_\nu}^{j-1}(n) = \widehat{e}^{j-1}(n) + \frac{1}{2}\widehat{V}^{j-1}(n)\widehat{o_\nu}^{j-1}(n),$$

where $\mathbf{V^{j-1}}$ is some *update* filter.

**Normalization -** Finally, the smoothed array $\mathbf{s^{j-1}}$ and the array of details $\mathbf{d^{j-1}}$ are obtained by the following operation: $\mathbf{s^{j-1}} = \sqrt{2}\,\mathbf{e_\nu^{j-1}}, \quad \mathbf{d^{j-1}} = \mathbf{o_\nu^{j-1}}/\sqrt{2}$.

**One step of reconstruction** The reconstruction of the signal $\mathbf{x^j}$ from the arrays $\mathbf{s^{j-1}}$ and $\mathbf{d^{j-1}}$ is implemented by the reverse of the decomposition:

**Undo Normalization -** $\mathbf{e_\nu^{j-1}} = \mathbf{s^{j-1}}/\sqrt{2}$ $\mathbf{o_\nu^{j-1}} = \sqrt{2}\,\mathbf{d^{j-1}}$.

**Undo Lifting -** The even array

$$\widehat{e}^{j-1}(n) = \widehat{e}_\nu^{j-1}(n) - \frac{1}{2}\widehat{V}^{j-1}(n)\widehat{o}_\nu^{j-1}(n)$$

is restored.

**Undo Predict -** The odd array

$$\widehat{o}^{j-1}(n) = \widehat{o}_\nu^{j-1}(n) + \widehat{U}^{j-1}(n)\widehat{e}^{j-1}(n)$$

is restored.

**Undo Split -** The last step is performed using (11.1).

### 11.1.2 Periodic filter banks

Let $\hat{\Phi}^j(n) \triangleq (1+\omega_j^{-n}\widehat{U}^{j-1}(n))/2$ and define the following $2^j$-periodic filters $\mathbf{\tilde{g}^j}$, $\mathbf{\tilde{h}^j}$, $\mathbf{g^j}$ and $\mathbf{h^j}$ via their DFT's:

$$\begin{aligned}
&\widehat{\tilde{g}}^j(n) \triangleq \sqrt{2}\omega_j^{-n}\widehat{\Phi}^j(n+N/2), \quad \widehat{\tilde{h}}^j(n) \triangleq \sqrt{2}(1+\omega_j^n\widehat{V}^{j-1}(n)\widehat{\Phi}^j(n+N/2)),\\
&\widehat{h}^j(n) \triangleq \sqrt{2}\widehat{\Phi}^j(n), \quad \widehat{g}^j(n) \triangleq \sqrt{2}(1-\omega_j^n\widehat{V}^{j-1}(n)\widehat{\Phi}^j(n)).
\end{aligned}$$

Here $\mathbf{\tilde{h}^j}$ and $\mathbf{\tilde{g}^j}$ are the low- and high-pass primal analysis filters, respectively, and $\mathbf{h^j}$ and $\mathbf{g^j}$ are the low- and high-pass primal synthesis filters, respectively. These four filters form a $2^j$-periodic perfect reconstruction filter bank.

**Proposition 11.1** *If $\omega_j^{-n}\widehat{U}^{j-1}(n)$ and $\omega_j^n\widehat{V}^{j-1}(n)$ are real valued as $n \in \mathbb{Z}$ then the decomposition and reconstruction equations can be represented as follows:*

$$\begin{aligned}
\widehat{s}^{j-1}(n) &= \frac{1}{2}\Big(\bar{\widehat{\tilde{h}}}^j(n)\,\widehat{x}^j(n) + \bar{\widehat{\tilde{h}}}^j(n+N/2)\,\widehat{x}^j(n+N/2)\\
\widehat{d}^{j-1}(n) &= \frac{1}{2}\Big(\bar{\widehat{\tilde{g}}}^j(n)\,\widehat{x}^j(n) + \bar{\widehat{\tilde{g}}}^j(n+N/2)\,\widehat{x}^j(n+N/2)\Big)\\
\widehat{x}^j(n) &= \widehat{h}^j(n)\,\widehat{s}^{j-1} + \widehat{g}^j(n)\,\widehat{d}^{j-1}.
\end{aligned}$$

### 11.1.3 Bases

As in the non-periodic case, the filter bank $\tilde{\mathbf{h}}^{\mathbf{j}}$, $\tilde{\mathbf{g}}^{\mathbf{j}}$, $\mathbf{h}^{\mathbf{j}}$ and $\mathbf{g}^{\mathbf{j}}$ generate a biorthogonal pair of bases of the space of $2^j$-periodic signals.

**Proposition 11.2** *Any signal* $\mathbf{x}^{\mathbf{j}}$ *can be represented as follows:*

$$x_l = \sum_k^{j-1} s_k^{j-1}\, \varphi_{l-2k}^{j-1} + \sum_k^{j-1} d_k^{j-1}\, \psi_{l-2k}^{j-1},$$

*where* $\varphi_l^{j-1}$ *and* $\psi_l^{j-1}$ *are IDFT's of the arrays* $\{\widehat{h}^{j-1}(n)\}$ *and* $\{\widehat{g}^{j-1}(n)\}$*, respectively.*

*The coordinates* $s_k^{j-1}$ *and* $d_k^{j-1}$ *are the following inner products:*

$$s_k^{j-1} = \sum_l^j x_l\, \widetilde{\varphi}_{l-2k}^{j-1}, \quad d_k^{j-1} = \sum_l^j x_l\, \widetilde{\psi}_{l-2k}^{j-1}, \tag{11.2}$$

*where* $\widetilde{\varphi}_l^{j-1}$ *and* $\widetilde{\psi}_l^{j-1}$ *are IDFT's of the arrays* $\{\hat{\tilde{h}}^{j-1}(n)\}$ *and* $\{\hat{\tilde{g}}^{j-1}(n)\}$*, respectively.*

*The following biorthogonal relations hold:*

$$\sum_l^j \tilde{\psi}_{l-2k}^{j-1}\, \psi_{l-2m}^{j-1} = \sum_l^j \tilde{\varphi}_{l-2k}^{j-1}\, \varphi_{l-2m}^{j-1} = \delta_k^m, \quad \sum_l^j \tilde{\psi}_{l-2k}^{j-1}\, \varphi_{l-2m}^{j-1} = \sum_l^j \tilde{\varphi}_{l-2k}^{j-1}\, \psi_{l-2m}^{j-1} = 0,\ \forall m,\, k.$$

Although periodic wavelets cannot have vanishing moments in a traditional sense, we can define *quasi-vanishing moments* (QVM). QVM means that the high-frequency filter $\mathbf{g}^{\mathbf{j}}$ (or $\tilde{\mathbf{g}}^{\mathbf{j}}$) in our construction comprises a finite difference block of order $p$. It means that the filter can be represented as the convolution

$$g_k^j = \sum_l^j \Delta_{k-l}^p\, \theta_l^j, \text{ where } \Delta_k^p \triangleq \sum_{m=0}^p \binom{p}{m} (-1)^m \delta_{k-m}$$

and $\theta^j$ is a $2^j$-periodic signal. If some fragments of the signal are close to (or coincide with) a polynomial of a degree not exceeding $p$, then application of the filter $\mathbf{g}^{\mathbf{j}}$ (or $\tilde{\mathbf{g}}^{\mathbf{j}}$) turns these fragments close to zero. Recall that in a non-periodic setting such a property of the filter guarantees that the corresponding wavelet has $p$ vanishing moments.

**Proposition 11.3** *Let the DFT's* $\widehat{U}^{j-1}(n)$ *and* $\widehat{V}^{j-1}(n)$ *of filters, which are used for the* predict *and* update *steps, respectively, be rational functions with respect to* $\cos\frac{2\pi n}{2^j}$*. If the following factorization holds*

$$(1 + \omega_j^{-n} \widehat{U}^{j-1}(n)) = \left(\cos\frac{\pi n}{2^j}\right)^{2r} \vartheta^j(n)$$

*then the high-frequency analysis wavelet* $\tilde{\psi}^{j-1}$ *has* $2r$ *QVM. If, in addition,*

$$1 + \omega_j^n \widehat{V}^{j-1}(n) = \left(\cos\frac{\pi n}{2^j}\right)^{2q} \varrho^j(n)$$

*then the high-frequency synthesis wavelet* $\psi^1$ *has* $2s$ *QVM, where* $s = \min(q, r)$*. Here* $\vartheta^j(n)$ *and* $\varrho^j(n)$ *are rational functions with respect to* $\cos\frac{2\pi n}{2^j}$*.*

### 11.1.4 Multiscale periodic wavelet transforms

Expansion of the transform to the coarser scales is implemented in an iterative way. The scheme of the transform of the original signal $\mathbf{x^j}$ into $L \leq j$ scales is the following:

1. Calculate $\{\widehat{x}^j(n)\}$, which is the DFT of $\mathbf{x^j}$.
2. The array $\{\widehat{x}^j(n)\}$ is transformed via lifting steps into the arrays $\{\widehat{s}^{j-1}(n)\}$ and $\{\widehat{d}^{j-1}(n)\}$, which are the DFT's of the smoothed and detail subarrays $\mathbf{s^{j-1}}$ and $\mathbf{d^{j-1}}$, respectively.
3. Derive the detail subarray $\mathbf{d^{j-1}}$ by calculating the IDFT of the array $\{\widehat{d}^{j-1}(n)\}$. The subarray $\mathbf{d^{j-1}}$ is stored.
4. The array $\{\widehat{s}^{j-1}(n)\}$ is transformed into $\{\widehat{s}^{j-2}(n)\}$ and $\{\widehat{d}^{j-2}(n)\}$. The subarray $\mathbf{d^{j-2}}$ is stored.
5. Steps 3–4 are iterated until the arrays $\{\widehat{s}^{j-L+1}(n)\}$ and $\{\widehat{d}^{j-L+1}(n)\}$ are produced. The array $\mathbf{d^{j-L+1}}$ is stored.
6. The array $\{\widehat{s}^{j-L+1}(n)\}$ is transformed into $\{\widehat{s}^{j-L}(n)\}$ and $\{\widehat{d}^{j-L}(n)\}$.
7. Derive the smoothed and detail subarrays $\mathbf{s^{j-L}}$ and $\mathbf{d^{j-L}}$ calculating the IDFT's of the arrays $\{\widehat{s}^{j-L}(n)\}$ and $\{\widehat{d}^{j-L}(n)\}$. Both the subarrays $\mathbf{s^{j-L}}$ and $\mathbf{d^{j-L}}$ are stored.

By this means the signal $\mathbf{x^j}$ is transformed into the subarrays of wavelet coefficients : $\mathbf{x^j} \rightarrow \mathbf{d^{j-1}} \bigcup \mathbf{d^{j-2}} \bigcup \ldots \bigcup \mathbf{d^{j-L}} \bigcup \mathbf{s^{j-L}}$. The inverse transform $\mathbf{d^{j-1}} \bigcup \mathbf{d^{j-2}} \bigcup \ldots \bigcup \mathbf{d^{j-L}} \bigcup \mathbf{s^{j-L}} \rightarrow \mathbf{x^j}$ is implemented in a reverse order.

1. Calculate $\{\widehat{s}^{j-L}(n)\}$ and $\{\widehat{d}^{j-L}(n)\}$, which are the DFT's of the arrays $\mathbf{s^{j-L}}$ and $\mathbf{d^{j-L}}$, respectively.

2. Restore the array $\{\widehat{s}^{j-L+1}(n)\}$ by lifting steps and calculate $\{\widehat{d}^{j-L+1}(n)\}$, which is the DFT of the array $\mathbf{d^{j-L+1}}$.

3. Iterate steps 1–2 until the array $\{\widehat{s}^{j-1}(n)\}$ is restored and $\{\widehat{d}^{j-1}(n)\}$ is calculated.

4. Derive the array $\{\widehat{x^j}(n)\}$ from $\{\widehat{s}^{j-1}(n)\}$ and $\{\widehat{d^{j-1}}(n)\}$ by lifting steps and restore $\mathbf{x^j}$ by calculating the IDFT of $\{\widehat{x^j}(n)\}$.

Major computational expenses in this processing are allocated to calculate the direct and inverse DFT's via the FFT. Therefore the computational complexity of implementing periodic transforms hardly grows with the increase in order of splines used for the creation of the prediction and update filters. When splines of low order are employed, a time-domain recursive implementation has a lower computational cost than the FFT algorithm. But, unlike the FFT, this cost grows rapidly as the order of the splines increases. The higher the order of the splines used for the filter design, the more QVM have corresponding wavelets. When splines of higher order are involved, the FFT implementation has a lower computational complexity than the time-domain implementation.

### 11.1.5 Father and mother periodic wavelets

As in the non-periodic case, the periodic wavelet transforms generate a biorthogonal pair of bases $\widetilde{\varphi}_l^{j-m}$, $\widetilde{\psi}_l^{j-m}$ and $\varphi_l^{j-m}$, $\psi_l^{j-m}$ at the $m-$th decomposition scale. These wavelets are derived from wavelets of the previous scale via the discrete two-scale relations

$$\widetilde{\varphi}_l^{j-m} = \sum_k^{j-m+1} \widetilde{h}_k^{j-m+1}\, \widetilde{\varphi}_{l-2k}^{j-m+1}, \quad \widetilde{\psi}_l^{j-m} = \sum_k^{j-m+1} \widetilde{g}_k^{j-m+1}\, \widetilde{\varphi}_{l-2k}^{j-m+1},$$
$$\varphi_l^{j-m} = \sum_k^{j-m+1} h_k^{j-m+1}\, \varphi_{l-2k}^{j-m+1}, \quad \psi_l^{j-m} = \sum_k^{j-m+1} g_k^{j-m+1}\, \varphi_{l-2k}^{j-m+1}.$$

A significant difference from the non-periodic case lies in the fact that the filter bank $\widetilde{\mathbf{h}}^{\mathbf{j}}$, $\widetilde{\mathbf{g}}^{\mathbf{j}}$ and $\mathbf{h^j}$, $\mathbf{g^j}$ depends on the scale. Therefore the multiresolution analysis in spaces of periodic functions has some specifics, that are described in [32]. In particular, there are no scaling functions in such spaces. Given a set of filter banks $\widetilde{\mathbf{h}}^{\mathbf{j}}$, $\widetilde{\mathbf{g}}^{\mathbf{j}}$ and $\mathbf{h^j}$, $\mathbf{g^j}$ associated with the scale $j \in \mathbb{Z}_+$, we can define a set of father $\widetilde{\varphi}^j(t)$, $\varphi^j(t)$ and mother $\widetilde{\psi}^j(t)$, $\psi^j(t)$ wavelets related

to this scale $j$. Provided the 1-periodic father wavelets of the scale $j+1$ are available, the 1-periodic wavelets of the scale $j$ are derived via the discrete two-scale relations

$$\widetilde{\varphi}^j(t) = \sum_k^{j+1} \widetilde{h}_k^{j+1}\, \widetilde{\varphi}^{j+1}(t - k2^{-j}), \quad \widetilde{\psi}^j(t) = \sum_k^{j+1} \widetilde{g}_k^{j+1}\, \widetilde{\varphi}^{j+1}(t - k2^{-j}),$$
$$\varphi^j(t) = \sum_k^{j+1} h_k^{j+1}\, \widetilde{\varphi}^{j+1}(t - k2^{-j}), \quad \psi^j(t) = \sum_k^{j+1} g_k^{j+1}\, \widetilde{\varphi}^{j+1}(t - k2^{-j}).$$

Therefore, if the father wavelet $\widetilde{\varphi}^j(t)$ (or $\widetilde{\varphi}^j(t)$) on an initial scale $J$ is available then all the wavelets on the scales $J-1, J-2, \dots 0$ can be derived. The wavelet $\widetilde{\varphi}^j(t)$ (or $\widetilde{\varphi}^j(t)$) can be constructed using a cascade subdivision algorithm similar to the algorithm described in Section 7. Sometimes the wavelets can be presented explicitly. For example, it can be done for the multiresolution analysis in spaces of periodic polynomial splines. This case was studied in [57].

## 11.2 Filter design using periodic splines

### 11.2.1 Polynomial interpolatory splines

The scheme of the filter design in the periodic case is similar to the non-periodic scheme. Given a signal $\mathbf{x^j}$, we construct a 1-periodic spline of order $p$, which interpolates even samples $\mathbf{e^{j-1}} = \{e_k = x_{2k}\}_{k=0}^{2^{j-1}-1}$ on the grid $\{k2^{-j+1}\}$. Then odd samples $\mathbf{o^{j-1}} = \{o_k = x_{2k+1}\}_{k=0}^{2^{j-1}-1}$ are predicted by values of the spline at the midpoints $\{(2k+1)2^{-j}\}$ between the interpolation points. Such a construction produces the prediction filter $\mathbf{U_{i,p}^{j-1}}$, whose DFT is

$$\widehat{U}_{i,p}^{j-1}(n) = \frac{\widehat{w^p}(2^{2-j}\pi n)}{\widehat{u^p}(2^{2-j}\pi n)}. \tag{11.3}$$

The functions $\widehat{u^p}(\omega)$ and $\widehat{w^p}(\omega)$ are defined in (3.3). Note that periodic interpolatory splines of odd order possess the super-convergence property.

**Examples:** Denote $\nu = \cos 2^{1-j}\pi n$.

**Interpolatory periodic quadratic spline:** $p = 3$

$$U_{i,3}^{j-1}(n) = \frac{2\omega_j^n \nu}{1+\nu^2}, \quad 1 + \omega_j^{-n} U_{i,3}^{j-1}(n) = \frac{(1+\nu)^2}{1+\nu^2} = \frac{4\cos^4(2^{-j}\pi n)}{1+\nu^2}.$$

**Interpolatory periodic cubic spline:** $p = 4$

$$U_{i,4}^{j-1}(n) = \frac{\omega_j^n \nu(5+\nu^2)}{2(1+2\nu^2)}, \quad 1 + \omega_j^{-n} U_{i,4}^{j-1}(n) = \frac{2\cos^4(2^{-j}\pi n)(2+\nu)}{1+2\nu^2}.$$

**Interpolatory periodic spline of fourth degree:** $p = 5$

$$U_{i,5}^{j-1}(n) = \frac{\omega_j^n 8\nu(2+\nu^2)}{5+18\nu^2+\nu^4}, \quad 1 + \omega_j^{-n} U_{i,5}^{j-1}(n) = \frac{8\cos^6(2^{-j}\pi n)(5+\nu)}{5+18\nu^2+\nu^4}.$$

$p = 7$ **:**

$$U_{i,7}^{j-1}(n) = \frac{\omega_j^n 16\nu(17+26\nu^2+2\nu^4)}{61+479\nu^2+179\nu^4+\nu^6}, \quad 1 + \omega_j^{-n} U_{i,7}^{j-1}(n) = \frac{16\cos^8(2^{-j}\pi n)(\nu^2+28\nu+61)}{61+479\nu^2+179\nu^4+\nu^6}.$$

$p = 9$ **:**

$$U_{i,9}^{j-1}(n) = \frac{\omega_j^n 128\nu(192\nu^2+62+60\nu^4+\nu^6)}{1385+19028\nu^2+18270\nu^4+1636\nu^6+\nu^8},$$
$$1 + \omega_j^{-n} U_{i,9}^{j-1}(n) = \frac{32\cos^{10}(2^{-j}\pi n)(\nu^3+123\nu^2+1011\nu+1385)}{1385+19028\nu^2+18270\nu^4+1636\nu^6+\nu^8}.$$

$p = 11$ **:**

$$U_{i,11}^{j-1}(n) = \frac{\omega_j^n 256\nu(1382+7192\nu^2+5097\nu^4+502\nu^6+2\nu^8)}{50521+1073517\nu^2+1949762\nu^4+540242\nu^6+14757\nu^8+\nu^{10}},$$
$$1 + \omega_j^{-n} U_{i,11}^{j-1}(n) = \frac{64\cos^{12}(2^{-j}\pi n)(\nu^4+506\nu^3+11706\nu^2+50666\nu+50521)}{50521+1073517\nu^2+1949762\nu^4+540242\nu^6+14757\nu^8+\nu^{10}}.$$

We display in Figure 16 frequency responses of the synthesis and analysis filters $\mathbf{g^j}$, $\mathbf{h^j}$ $\widetilde{\mathbf{g}}^{\mathbf{j}}$, and $\widetilde{\mathbf{h}}^{\mathbf{j}}$, and the synthesis and analysis father $\widetilde{\varphi}^j(t)$, $\varphi^j(t)$ and mother $\widetilde{\psi}^j(t)$, $\psi^j(t)$ wavelets of the transform, which uses the prediction and update filters derived from the periodic interpolatory spline of eleventh order. The mother wavelets have eleven QVM's. We can observe that time-domain localization of wavelets is worse than the localization of wavelets displayed in Section 8.4 but the frequency responses are closer to a rectangular shape.

### 11.2.2 Discrete periodic splines

A $2^j$-periodic discrete B-spline of first order $b_k^{1,n}$ is a $2^j-$periodization of the B-spline $B_k^{1,n}$ defined in (4.1). B-splines of higher order are defined recursively via the circular discrete

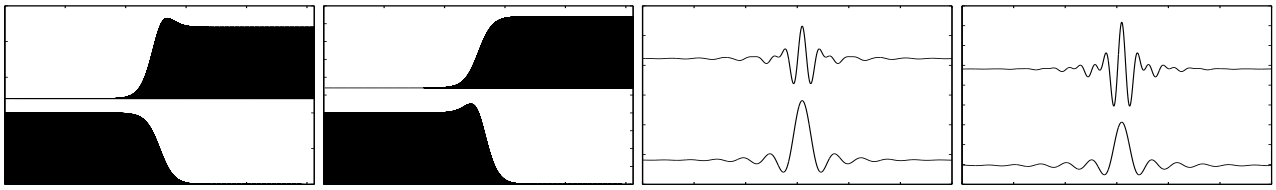

Figure 16: Filters and wavelets originating from the interpolatory periodic spline of eleventh order. First and second columns: frequency responses of synthesis and analysis filters, respectively. Third and fourth columns: synthesis and analysis wavelets, respectively.

convolution: $b^{p,n} = b^{1,n} * b_k^{p-1,n}$. In the case $n = 1,\ p = 2r$, which we are interested in, its DFT is $(\widehat{b^{2r,1}})^j(m) = \left(1 + \omega_j^{-m}\right)^{2r}$. We introduce the central periodic B-spline of order $2r$: $q_k^{2r} \triangleq b_{k+r}^{2r,1}$. Its DFT is $(\widehat{q}^{2r})^j(m) = \left(2\cos \pi m/2^j\right)^{2r}$. A $2^j$-periodic discrete spline with nodes at points $\{2k\}$ is defined as a combination of the central B-splines:

$$s_k^{2r} \triangleq \sum_l^{j-1} c_l q_{k-2l}^{2r}.$$

As in the case of polynomial periodic splines, given a signal $\mathbf{x^j}$, we construct a $2^j$-periodic discrete spline of order $2r$, which interpolates even samples $\mathbf{e^{j-1}} = \{e_k = x_{2k}\}$ on the grid $\{2k\}$: $s^{2r}(2k) = e_k,\ k = 0, \dots 2^{j-1} - 1$. Then odd samples $\mathbf{o^{j-1}} = \{o_k = x_{2k+1}\}_{k=0}^{2^{j-1}-1}$ are predicted by values of the spline at midpoints $\{(2k+1)\}$ between the interpolation points. Such a construction produces the prediction filter $\mathbf{U_{d,2r}^{j-1}}$, whose DFT is

$$\widehat{U}_{d,2r}^{j-1}(m) = \omega_j^m \frac{\left(\cos \pi m/2^j\right)^{2r} - \left(\sin \pi m/2^j\right)^{2r}}{\left(\cos \pi m/2^j\right)^{2r}) + \left(\sin \pi m/2^j\right)^{2r}}. \tag{11.4}$$

Hence we have

$$1 + \omega_j^{-m} \widehat{U}_{d,2r}^{j-1}(m) = \frac{2\left(\cos \pi m/2^j\right)^{2r}}{\left(\cos \pi m/2^j\right)^{2r} + \left(\sin \pi m/2^j\right)^{2r}}. \tag{11.5}$$

The explicit formulas (11.4) and (11.5) enable us to easily construct biorthogonal periodic wavelet transforms with any number of QVM's. The computational complexity of the implementation of these transforms does not grow with the increase in the number of QVM's. We display in Figure 17 frequency responses of the synthesis and analysis filters $\mathbf{g^j}$, $\mathbf{h^j}$ $\tilde{\mathbf{g}}^{\mathbf{j}}$, and $\widetilde{\mathbf{h}}^{\mathbf{j}}$, and the synthesis and analysis father $\widetilde{\varphi}^j(t)$, $\varphi^j(t)$ and mother $\widetilde{\psi}^j(t)$, $\psi^j(t)$ wavelets of the transforms, which use the prediction and update filters derived from the discrete spline

of fiftieth order. The mother wavelets have 50 QVM's. We can observe a poor time-domain localization of wavelets but almost rectangular frequency responses.

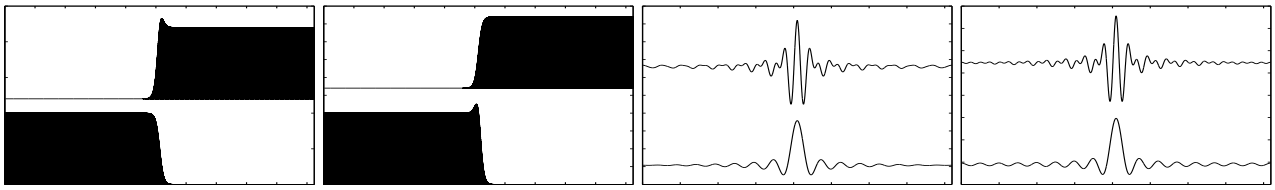

Figure 17: Filters and wavelets originating from the discrete periodic spline of fiftieth order. First and second columns: frequency responses of synthesis and analysis filters, respectively. Third and fourth columns: synthesis and analysis wavelets, respectively.

# References


[1] M. Abramovitz and I. Stegun, *Handbook of mathematical functions*, Dover Publ. Inc., New York, 1972.

[2] J. H. Ahlberg, E. N. Nilson and J. L. Walsh, *The theory of splines and their applications*, Acad. Press, New York, 1967.

[3] A. Aldroubi, M. Eden and M. Unser, *Discrete spline filters for multiresolutions and Wavelets of* $l_2$, SIAM Math. Anal., **25**, (1994), 1412-1432.

[4] M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, *Image coding using wavelet transform*, IEEE Transaction on Image Processing, **1**(2) (1992),205-220.

[5] A. Z. Averbuch, F. G. Meyer, J.-O. Strömberg, *Fast adaptive wavelet packet image compression,* IEEE Trans. on Image Processing, **9**(5), (2000), 792-800.

[6] A. Z. Averbuch, A. B. Pevnyi and V. A. Zheludev *Butterworth wavelet transforms derived from discrete interpolatory splines: Recursive implementation*, Signal Processing, **81**, (2001), 2363-2382.

[7] A. Z. Averbuch, A. B. Pevnyi and V. A. Zheludev, *Biorthogonal Butterworth wavelets derived from discrete interpolatory splines*, IEEE Trans. Sign. Proc., **49**, No.11,(2001), 2682-2692.

[8] A. Averbuch and V. Zheludev, *Construction of biorthogonal discrete wavelet transforms using interpolatory splines*, Applied and Comp. Harmonic Analysis, **12** (2002), 25-56.

[9] A. Averbuch and V. Zheludev, *Splines: a new contribution to wavelet analysis*, to appear in Proc. Conference Algorithms for Approximation IV, Huddersfield, 2002.

[10] G. Battle, *A block spin construction of ondelettes. Part I. Lemarié functions*, Comm. Math. Phys. **110** (1987), 601-615.

[11] C. M. Brislawn, *Classification of nonexpansive symmetric extension transforms for multirate filter banks*, Applied and Computational Harmonic Analysis, **3** (1996), 337-357.

[12] C. K. Chui and J. Z. Wang, *On compactly supported spline wavelets and a duality principle*, Trans. Amer. Math. Soc. **330**(1992), 903-915.

[13] R. L. Claypoole Jr., J. M. Davis, W. Sweldens and R. Baraniuk, *Nonlinear wavelet transforms for image coding via lifting*, submitted to IEEE Trans. Signal Proc.

[14] A. Cohen, I. Daubechies and J.-C. Feauveau, *Biorthogonal bases of compactly supported wavelets*, Commun. on Pure and Appl. Math. **45** (1992), 485-560.

[15] A. Cohen and I. Daubechies, *A new technique to estimate the regularity of refinable functions,* Revista Matematica Iberoamericana, **12** (1996), 527–591.

[16] I. Daubechies, Ten lectures on wavelets, SIAM. Philadelphia, PA, 1992.

[17] I. Daubechies and Y. Huang, *A decay theorem for refinable functions,* Applied Mathematics Letters, **7**, (1994), 1–4.

[18] I. Daubechies, and W. Sweldens, Factoring wavelet transforms into lifting steps, *J. Fourier Anal. Appl.*, **4** (1998) 247–269.

[19] G. Deslauriers and S. Dubuc, *Interpolation dyadique*. In "Fractals, Dimensions nonentières et applications". Paris: Masson, 1987.

[20] G. Deslauriers and S. Dubuc, *Symmetric iterative interpolation processes*, Constructive Approximation, **5** (1989), 49-68.

[21] D. L. Donoho, *Interpolating wavelet transform,* Preprint 408, Department of Statistics, Stanford University, 1992.

[22] N. Dyn, J. A. Gregory and D. Levin, *Analysis of uniform binary subdivision schemes for curve design*, Constr. Approx., **7** (1991), 127–147.

[23] N. Dyn, *Analysis of convergence and smoothness by the formalism of Laurent polynomials*, in Tutorials on Multiresolution in Geometric Modelling, A. Iske, E. Quak, M.S. Floater, eds., Springer 2002, 51-68.

[24] G. Fix and G. Strang, *Fourier analysis of the finite element method in Ritz-Galerkin theory*, Stud. Appl. Math., **48** (1969), 265-273.

[25] C. Herley and M. Vetterli, *Wavelets and recursive filter banks*, IEEE Trans. Signal Proc., **41(12)** (1993), 2536-2556.

[26] P.G. Lemarié, *Ondelettes à localization exponentielle*, J. de Math. Pures et Appl. **67**(1988), 227-236.

[27] M. Lightstone, E. Majani and S. K. Mitra, *Low bit-rate design considerations for wavelet-based image coding*, Multidimensional Systems and Signal Processing, **8** (1997), 111-128.

[28] S. Mallat and Z. Zhang, *Matching pursuit with time-frequency dictionaries,* IEEE Trans. Sign. Proc., **41**(12) (1993), 3397-3415.

[29] D. Marpe, G. Heising, A. P. Petukhov, and H. L. Cycon, Video coding using a bilinear image warping motion model and wavelet-based residual coding, *Proc. SPIE Conf. on Wavelet Applications in Signal and Image Processing VII*, Denver, CO, July 1999, SPIE, Vol. 3813, pp. 401 – 408.

[30] A. V. Oppenheim and R. W. Shafer, Discrete-time signal processing, Englewood Cliffs, New York, Prentice Hall, 1989.

[31] A. B. Pevnyi and V. A. Zheludev, *On the interpolation by discrete splines with equidistant nodes*, J. Appr. Th., **102** (2000), 286-301.

[32] A. P. Petukhov, *Periodic wavelets,* Sb. Math. **188**(10) (1997), 1481–1506.

[33] A. P. Petukhov, *Biorthogonal wavelet bases with rational masks and their applications*, Proc. of St. Petersburg Math. Soc., Vol. **7** (1999), 168 – 193. (Russian)

[34] A. B. Pevnyi and V. A. Zheludev, *Construction of wavelet analysis in the space of discrete splines using Zak transform*, J. Fourier Analysis and Application, **8**(1), (2002) 55-77.

[35] M. Rabbani and R. Joshi, *An overview of the JPEG 2000 still image compression standard*, Signal Processing, Image communication, **17**, (2002), 3-48.

[36] A. Said and W. W. Pearlman, *A new, fast and efficient image codec based on set partitioning in hierarchical trees*, IEEE Trans. on Circ. and Syst. for Video Tech., **6** (1996) 243-250.

[37] N. Saito and G. Beylkin, *Multiresolution representations usung the auto-correlation functions of compactly supported wavelets*, IEEE Trans. Signal Proc., **41**(12), (1993), 3594-3590.

[38] G. Strang, and T. Nguen, Wavelets and filter banks, Wellesley-Cambridge Press, 1996.

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

[40] I. J. Schoenberg, *On spline functions*, In "Inequalities", (O. Shisha ed.) (1967), 255-291.

[41] I. J. Schoenberg , *Cardinal interpolation and spline functions*, J. Approx. Th., **2**, (1969), 167-206.

[42] I. J. Schoenberg, *Cardinal interpolation and spline functions* II, *Interpolation of data of power growth,* J. Approx. Th., **6**, (1972), 404-420.

[43] L. L. Schumaker, Spline functions: Basic theory, Wiley, 1981.

[44] J. O. Strömberg, *A modified Franklin system and higher order spline systems on* $\mathbb{R}^n$ *as unconditional bases for Hardy spaces*. In W. Becker, A. P. Calderón, R. Fefferman and P. W. Jones, editors, Proc. Conf. in honor of Antoni Zygmund, volume II, pp 475-493, New York, 1981, Wadsworth.

[45] Yu. N. Subbotin, *On the relation between finite differences and the corresponding derivatives*, Proc.Steklov Inst. Math. **78** (1965), 24-42.

[46] W. Sweldens *The lifting scheme: A custom design construction of biorthogonal wavelets*, Appl. Comput. Harm. Anal. **3(2)**, (1996), 186-200.

[47] M. Unser, A. Aldroubi and M. Eden, *B-spline signal processing: Part I–theory*, IEEE Trans. Signal Process, **41**(2) (1993), 821-832.

[48] M. Unser, A. Aldroubi and M. Eden, *B-spline signal processing: Part II–Efficient design and applications*, IEEE Trans. Signal Process, **41**(2) (1993), 834-848.

[49] M. Unser, A. Aldroubi and M. Eden, *A family of polynomial spline wavelet transforms*, Signal Processing, **30**, (1993), 141-162.

[50] M. Vetterli and C. Herley, *Wavelets and filter banks: Theory and design,* IEEE Trans. Signal Proc., **40(9)** (1992), 2207-2232.

[51] J. D. Villasenor, B. Belzer and J. Liao, Filter evaluation and selection in wavelet compression algorithms, in Proc. of Data Compression Conf., Snowbird, UT, (1994), 478-488.

[52] W. V. Wickerhauser, Adapted wavelet analysis from theory to software, AK Peters, Wellesley, Massachusetts, 1994.

[53] V. A. Zheludev, *Periodic splines and the fast Fourier transform,* Comput. Math. & Math. Phys., **32**(2) (1992), 149-165.

[54] V. A. Zheludev *Spline harmonic analysis and wavelet bases,* Proc. Symp. Appl. Math., **48** (ed. W. Gautcshi), AMS 1994, 415-419.

[55] V. A. Zheludev, *Local spline approximation on a uniform grid,* U.S.S.R. Comput. Math. & Math. Phys., **27**(5) (1987), 8-19.

[56] V.A. Zheludev, *Local smoothing splines with a regularizing parameter*, Comput. Math. & Math Phys., **31**, (1991), 193-211.

[57] V. A. Zheludev, *Periodic splines, harmonic analysis, and wavelets.* In "Signal and image representation in combined spaces, 477–509, Wavelet Anal. Appl., 7," (eds. Y. Y. Zeevi and R. Coifman), Academic Press, San Diego, CA, 1998.