A library of biorthogonal wavelet transforms originated from polynomial splines

Amir Z. Averbuch[a] and Valery A. Zheludev [a]

[a]School of Computer Science, Tel Aviv University
Tel Aviv 69978, Israel

ABSTRACT

We present a library of biorthogonal wavelet transforms and the related library of biorthogonal symmetric waveforms. For the construction we use interpolatory, quasiinterpolatory and smoothing splines with finite masks (local splines). On this base we designed a set of perfect reconstruction infinite and finite impulse response filter banks with the linear phase property. The construction is performed in a “lifting” manner. The developed technique allows to construct wavelet transforms with arbitrary prescribed properties such as number of vanishing moments, shape of wavelets, frequency resolution. Moreover, the transforms contain some scalar control parameters which enable their flexible tuning in either time or frequency domains. The transforms are implemented in a fast manner. The transforms based on interpolatory splines are implemented through recursive filtering.We present encouraging results of applicarion of devised wavelet transforms to image compression.

1. BIORTHOGONAL WAVELET TRANSFORM

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

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

Throughout the paper we assume that $z = e^{i\omega}$. We introduce a family of biorthogonal wavelet-type transforms that operate on the signal $\mathbf{x} = \{x(k)\}_{k=-\infty}^{\infty}$, which we construct through lifting steps.

First we recall a basic lifting scheme of wavelet transform. The lifting scheme can be implemented in a primal or dual modes. For brevity we consider only the primal mode.

1.1. Lifting scheme of wavelet transform

1.2. Decomposition

Generally, the primal lifting scheme for decomposition of signals consists of three steps: 1. Split. 2. Predict. 3. Update or lifting.

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

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

Predict - We use the even array $\mathbf{e_1}$ to predict the odd array $\mathbf{d_1}$ and redefine the array $\mathbf{d_1}$ as the difference between the existing array and the predicted one. To be specific, we apply some filter with transfer function $zU(z)$ to the sequence $\mathbf{e_1}$ and predict the function $d_1(z^2)$ which is the z^2-transform of $\mathbf{d_1}$. The $z-$transform of the new $d-$array is defined as follows:

$$d_1^u(z^2) = d_1(z^2) - zU(z)e_1(z^2). \tag{1.1}$$

From now on the superscript u means an *update* operation of the array. Obviously, $zU(z)e_1(z^2)$ should well approximate $d_1(z^2)$.

Further author information: (Send correspondence to Amir Z. Averbuch)
Amir Z. Averbuch: E-mail: amir@math.tau.ac.il
Valery A. Zheludev : E-mail: zhel@math.tau.ac.il

Lifting - We update the even array using the new odd array:

$$e_1^u(z^2) = e_1(z^2) + \beta(z)z^{-1}\, d_1^u(z^2). \tag{1.2}$$

Generally, the goal of this step is to eliminate aliasing which appears while downsampling the original signal $\mathbf{x}$ into $\mathbf{e_1}$. By doing so we have that $\mathbf{e_1}$ is transformed into a low-pass filtered and downsampled replica of $\mathbf{x}$. Further on we will discuss how to achieve this effect by a proper choice of the control filter β , but for now we only require the function $\beta(z)$ to be real-valued and obey the condition $\beta(-z) = -\beta(z)$, so the product $\beta(z)z^{-1}$ is a function of z^2.

1.2.1. Reconstruction

The reconstruction of the signal $\mathbf{x}$ from the arrays $\mathbf{e_1^u}$ and $\mathbf{d_1^u}$ is implemented in reverse order: 1. Undo Lifting. 2. Undo Predict. 3. Unsplit.

Undo Lifting - We restore the even array: $e_1(z^2) = e_1^u(z^2) - \beta(z)z^{-1}\, d_1^u(z^2)$.

Undo Predict - We restore the odd array: $d_1(z^2) = d_1^u(z^2) + zU(z)e_1(z^2)$.

Unsplit - The last step represents the standard restoration of the signal from its even and odd components. In the $z-$ domain it looks as: $x(z) = e_1(z^2) + z^{-1}d_1(z^2)$.

1.3. Filter banks

Lifting schemes, that were presented above, yield efficient algorithms for the implementation of the forward and backward transform of $\mathbf{x} \longleftrightarrow \mathbf{e_1^u} \cup \mathbf{d_1^u}$. But these operations can be interpreted as transformations of the signals by a filter bank that possesses the perfect reconstruction properties.

First we define two filter transfer functions $\Phi^l(z) \triangleq (1 + U(z))/2, \quad \Phi^h(z) \triangleq (1 - U(z))/2$.

Define the filter functions

$$\begin{aligned}
&\tilde{g}(z) \triangleq 2z^{-1}\Phi^h(z), \quad \tilde{h}_\beta(z) \triangleq 1 + 2\beta(z)\Phi^h(z) = 1 + \beta(z)\, z\tilde{g}(z),\\
&h(z) \triangleq 2\Phi^l(z) \quad g_\beta(z) \triangleq z^{-1}\left(1 - 2\beta(z)\Phi^l(z)\right) = z^{-1}\left(1 - \beta(z)\, h(z)\right).
\end{aligned}$$

We call $\tilde{h}_\beta(z)$ and $\tilde{g}(z)$ the transfer functions of the low-pass and high-pass primal decomposition filters, respectively. We call $h(z)$ and $g_\beta(j)$ the transfer functions of the low-pass and high-pass primal reconstruction filters, respectively. These four filters form a perfect reconstruction filter bank (Ref. 5).

Theorem 1.1.

The decomposition and reconstruction formulas can be represented as follows:

$$\begin{aligned}
e_1^u(z^2) &= \frac{1}{2}\left(\overline{\tilde{h}_\beta(z)}\, x(z) + \overline{\tilde{h}_\beta(-z)}\, x(-z)\right) \quad d_1^u(z^2) = \frac{1}{2}\left(\overline{\tilde{g}(z)}\, x(z) + \overline{\tilde{g}^r(-z)}\, x(-z)\right) &(1.3)\\
x(z) &= h(z)e_1^u(z^2) + g_\beta(z)\, d_1^u(z^2). &(1.4)
\end{aligned}$$

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

$$h(z)\, \overline{\tilde{h}_\beta(z)} + g_\beta(z)\, \overline{\tilde{g}(z)} = 2 \quad h(z)\, \overline{\tilde{h}_\beta(-z)} + g_\beta(z)\, \overline{\tilde{g}(-z)} = 0.$$

The filter functions are linked to each other in the following way:

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

1.4. Bases for the signal space

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

In section 1.3 we introduced a family of filters by their transfer functions $h(z)$, $g_\beta(z)$, $\tilde{h}_\beta(z)$, $\tilde{g}(z)$. We denote by $\varphi^1(k)$, $\psi^1_\beta(k)$, $\tilde{\varphi}^1_\beta(k)$, $\tilde{\psi}^1(k)$ the impulse response functions of the corresponding filters, respectively. It means that, for example $h^r(z) = \sum_{k\in\mathbb{Z}} z^{-k}\varphi^{r,1}(k)$ and similarly for the other functions.

Proposition 1.1. *The shifts of the functions $\varphi^1(k)$, $\psi^1_\beta(k)$, $\tilde{\varphi}^1_\beta(k)$ and $\tilde{\psi}^1(k)$ form a biorthogonal pair of bases for the signal space $\mathcal{S}$. This means that any signal $\mathbf{x} \in \mathcal{S}$ can be represented as:*

$$x(l) = \sum_{k\in\mathbb{Z}} e^u_1(k)\,\varphi^1(l-2k) + \sum_{k\in\mathbb{Z}} d^u_1\,\psi^1_\beta(l-2k).$$

The coordinates $e^u_1(k)$ and $d^u_1(k)$ can be represented as inner products:

$$e^u_1(k) = \langle \mathbf{x}, \tilde{\varphi}^1_{\beta,k}\rangle, \ \textit{where}\ \tilde{\varphi}^1_{\beta,k}(l) = \tilde{\varphi}^1_\beta(l-2k) \quad d^u_1(k) = \langle \mathbf{x}, \tilde{\psi}^1_k\rangle, \ \textit{where}\ \tilde{\psi}^1_k(l) = \tilde{\psi}^1(l-2k).$$

The following biorthogonal relations hold:

$$\langle \tilde{\varphi}^1_{\beta,k}, \varphi^1_l\rangle = \langle \psi^1_{\beta,k}, \tilde{\psi}^1_l\rangle = \delta^l_k, \quad \langle \tilde{\varphi}^1_{\beta,k}, \psi^1_{\beta,l}\rangle = \langle \tilde{\psi}^1_l, \varphi^1_k\rangle = 0, \quad \forall l,\, k.$$

The formulated proposition justifies the following definition.

Definition 1.2. *The functions φ^1 and ψ^1_β, which belong to the space $\mathcal{S}$, are called the low-frequency and high-frequency synthesis wavelets of the first scale, respectively. The functions $\tilde{\varphi}^1_\beta$ and $\tilde{\psi}^1$ are called the primal low-frequency and high-frequency analysis wavelets of the first scale, respectively.*

Repeated applications of the transform can be achieved in an iterative way. In this transform we store the array $\mathbf{d}^u_1$ and decompose the array $\mathbf{e}^u_1$. The transformed arrays $\mathbf{e}^u_2$ and $\mathbf{d}^u_2$ of the second decomposition scale are derived from the even and odd sub-arrays of the array $\mathbf{e}^u_1$ by the same lifting steps as those described above.i As a result we get that the signal $\mathbf{x}$ is transformed into three subarrays: $\mathbf{x} \leftrightarrow \mathbf{d}^u_1 \bigcup \mathbf{d}^u_2 \bigcup \mathbf{e}^u_2$. The reconstruction is performed in the reverse order.

2. POLYNOMIAL SPLINES

We will construct polynomial splines of various kinds using the even subarray of a signal, calculate their values in the midpoints between nodes and use these values for prediction of the odd array. In this section we discuss some properties of such splines and derive the corresponding filters U.

2.1. B-splines

The central $B-$spline of the first order on the grid $\{kh\}$ is defined as follows:

$$M^h_1(x) = \begin{cases} 1/h & \text{if } x \in [-h/2, h/2] \\ 0 & \text{elsewhere.} \end{cases}$$

Its Fourier transform is:

$$\widehat{M^1_h}(\omega) = \int_{-\infty}^{\infty} e^{i\omega x} M^h_1(x)dx = V_h(\omega), \quad V_h(\omega) \triangleq \frac{\sin(\omega h/2)}{\omega h/2} \tag{2.5}$$

The central $B-$spline of order p is the convolution

$$M^p_h(x) = M^{p-1}_h(x) * M^1_h(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\omega x}\,(V_h(\omega)^p\,d\omega \quad p \geq 2. \tag{2.6}$$

Note that the B-spline of order p is supported at the interval $(-ph/2, ph/2)$. It is positive within its support and symmetric around zero. The nodes of B-splines of even orders are located at points $\{kh\}$ and of the odd orders at points $\{h(k+1/2)\}$, $k \in \mathbb{Z}$. It follows from (2.6) that $hM_h^p(hx) = M^p(x), \quad$ where $M^p(x) \triangleq M_1^p(x)$. B-splines can be presented explicitly. Suppose $x_+ \triangleq (x+|x|)/2$. Then

$$M^p(x) = \frac{1}{(m-1)!}\sum_{k=0}^{p}(-1)^k\binom{p}{k}\left(x-k+\frac{p}{2}\right)_+^{p-1}. \tag{2.7}$$

We introduce the following sequences

$$\mathbf{u}^p \triangleq \{hM_h^p(hk) = M^p(k)\}, \quad \mathbf{w}^p \triangleq \left\{hM_h^p\left(h(k+\frac{1}{2})\right) = M^p\left(k+\frac{1}{2}\right)\right\},\ k \in \mathbb{Z} \tag{2.8}$$

comprised of values of B-splines in the grid points and in the midpoints. Due to the compact support of B-splines, these sequences are finite. By the reasons to be explained further on, we will use for our constructions only splines of odd orders $p = 2r-1$. In Tables 1, 2 we present the sequences for initial values r which are of practical importance. We need the z^2-transforms of the sequences $\mathbf{u}^p$ and $\mathbf{w}^p$:

k	-4	-3	-2	-1	0	1	2	3	4
$\mathbf{u}^3 \times 8$	0	0	0	1	6	1	0	0	0
$\mathbf{u}^5 \times 384$	0	0	1	76	230	76	1	0	0
$\mathbf{u}^7 \times 46080$	0	1	722	10543	23548	10543	722	1	0

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

k	-4	-3	-2	-1	0	1	2	3
$\mathbf{w}^3 \times 2$	0	0	0	1	1	0	0	0
$\mathbf{w}^5 \times 24$	0	0	1	11	11	1	0	0
$\mathbf{w}^7 \times 720$	0	1	57	302	302	57	1	0

Table 2. Values of the sequences $\mathbf{w}^p$.

$$u^p(z) \triangleq \sum_{k=-\infty}^{\infty} z^{-2k}u^p(k), \quad w^p(z) \triangleq \sum_{k=-\infty}^{\infty} z^{-2k}w^p(k).$$

These functions are Laurent polynomials. They are called the Euler-Frobenius polynomials Ref. 2 and were extensively studied in Ref. 3. In particular, the following important facts Ref. 3 were discovered:

Proposition 2.1. *On the circle $z = e^{i\omega}$ the Laurent polynomials $u^p(z)$ are strictly positive. Their roots are all simple and negative. Each root ζ can be paired with a dual root θ such that $\zeta\,\theta = 1$. Thus if $p = 2r+1$ is odd then $u^p(z)$ can be represented as follows:*

$$u^p(z) = \prod_{n=1}^{r}\frac{1}{\gamma_n}(1+\gamma_n z^2)(1+\gamma_n z^{-2}), \tag{2.9}$$

where $0 < \gamma_n < 1$.

We denote

$$U_i^p(z) \triangleq z^{-1}\frac{w^p(z)}{u^p(z)}. \tag{2.10}$$

Proposition 2.2. *The rational functions $U_i^p(z)$ are real-valued and $U_i^p(-z) = -U_i^p(z)$. If $p = 2r+1$ is odd then*

$$1 - U_i^p(z) = \frac{(\alpha-2)^{r+1}\xi_r(\alpha)}{u^p(z)}, \quad 1 + U_i^p(z) = \frac{(-\alpha-2)^{r+1}\xi_r(-\alpha)}{u^p(z)}, \tag{2.11}$$

where $\alpha \triangleq z + z^{-1}$ and $\xi_r(\alpha)$ is a polynomial of degree $r-1$.

2.2. Interpolatory splines

The shifts of B-splines form a basis in the space $\mathbf{S}_h^p$ of splines of order p on the grid kh. Namely, any spline $S_h^p \in \mathbf{S}_h^p$ has the following representation:

$$S_h^p(x) = h \sum_l q(l)\, M_h^p(x - lh). \tag{2.12}$$

Denote $\mathbf{q} \stackrel{\Delta}{=} \{q(l)\}$ and let $q(z^2)$ be the z^2-transform of $\mathbf{q}$. We introduce also the sequences $\mathbf{s}^p \stackrel{\Delta}{=} h\{S_h^p(hk) = S_1^p(k)\}$ and $\mathbf{m}^p \stackrel{\Delta}{=} \{S_h^p(h(k+1/2)) = S_1^p(k+1/2)\}$ of values of the spline in the grid points and in the midpoints. Let $s^p(z^2)$ and $m^p(z^2)$ be corresponding z^2-transforms. We have

$$S_1^p(k) = \sum_l q(l)\, M_h^p(k-l), \quad S_1^p\left(k + \frac{1}{2}\right) = \sum_l q(l)\, M_h^p\left(k - l + \frac{1}{2}\right). \tag{2.13}$$

Respectively, $s^p(z^2) = q(z^2)u(z), \quad m^p(z^2) = q(z^2)w(z)$.

From these formulas we can derive expression for the coefficients of a spline which interpolates a given sequence $\mathbf{e} \stackrel{\Delta}{=} \{e(k)\}$ at grid points:

$$hS_h^p(hk) = e(k),\ k \in \mathbb{Z}, \Longleftrightarrow q(z^2)u^p(z) = e(z^2) \Longleftrightarrow q(z^2) = \frac{e(z^2)}{u^p(z)}. \tag{2.14}$$

The z^2-transform of the sequence $\mathbf{m}^p$ is:

$$m^p(z^2) = q(z^2)w^p(z) = zU_i^p(z)e(z^2). \tag{2.15}$$

Our further construction exploits the super-convergence property of the interpolatory splines of odd orders (even degrees).

Theorem 2.1 (Ref. 7). *Let a function $f \in L^1(-\infty, \infty)$ has $p+1$ continuous derivatives and $S_h^p \in \mathbf{S}_h^p$ interpolates f on the grid $\{kh\}$. Denote $\tilde{f}_k = f(k+1/2)h)$. Then in the case of odd $p = 2r+1$, the following asymptotic relation holds*

$$S_h^p(h(k+1/2) = \tilde{f}_k - h^{2r+2} f^{(2r+2)}(h(k+1/2)(2r+1)\frac{b_{2r+2}(0) - b_{2r+2}(1/2)}{(2r+2)!} + o(h^{2r+2}f^{(2r+2)}(x)), \tag{2.16}$$

where $b_s(x)$ is the Bernoulli polynomial of degree s.

Recall, that in general the interpolatory spline of order $2r+1$ approximates the function f with accuracy of h^{2r+1}. Therefore, we may claim that $\{(k+1/2)h\}$ are points of super-convergence of the spline S_h^p. Note, that the spline of order $2r+1$ which interpolates the values of a polynomial of degree $2r$ coincides with this polynomial. However, the spline of order $2r+1$ which interpolates the values of a polynomial of degree $2r+1$ on the grid $\{kh\}$ restores the values of this polynomial in the mid-points $\{(k+1/2)h\}$. This property will result in the vanishing moments property of the wavelets to be constructed further on.

2.3. Quasi-interpolatory splines

It is seen from Eqs. (2.14), (2.15) that, in order to find coefficients of the spline interpolating a signal $\mathbf{e}$ and the values of the spline in the midpoints, the signal has to be filtered with the filters, whose transfer functions are $1/u^p(z)$ and $zU_i^p(z)$, respectively. These filters have infinite impulse responses (IIR). However, the property of super-convergence in the midpoints is not exclusive attribute of the interpolatory splines. It is also inherent to the so called local quasi-interpolatory splines of odd orders, which can be constructed using finite impulse response (FIR) filtering.

Definition 2.2. *Let a function f has p continuous derivatives and $\mathbf{f} \stackrel{\Delta}{=} \{f_k = f(hk)\},\ k \in \mathbb{Z}$. A spline $S_h^p \in \mathbf{S}_h^p$ of order p given by (2.12) is said to be the local quasi-interpolatory spline if the array $\mathbf{q}$ of its coefficients is derived by FIR filtering the array of samples $\mathbf{f}$*

$$\mathbf{q}(z^2) = \Gamma(z^2)\mathbf{f}(z^2), \tag{2.17}$$

where $\Gamma(z^2)$ is a Laurent polynomial, and the difference $|f(x) - S_h^p(x)| = O(f^{(p)}h^p)$.Provided f is a polynomial of degree $p-1$, the spline $S_h^p(x) \equiv f(x)$.

If $\mathbf{w}^p$ is the sequence defined in (2.8) then the values of the spline S_h^p in the midpoints between grid points $\mathbf{m}^p$ are produced by the following FIR filtering the array of samples $\mathbf{f}$

$$\mathbf{m^p}(z^2) = zU^p(z)\mathbf{f}(z^2), \quad U^p(z) \stackrel{\Delta}{=} z^{-1}\Gamma(z^2)w^p(z^2). \tag{2.18}$$

In Ref. 7 explicit formulas for construction of quasi-interpolatory splines are established as well as the estimations of the differences. In the present work we are interested in the splines of odd orders $p = 2r+1$. There are many FIR filters which generate quasi-interpolatory splines but only one filter of minimal length $2r+1$ for each order $p = 2r+1$. Denote $\lambda(z) \stackrel{\Delta}{=} z^{-2} - 2 + z^2$..

Theorem 2.3. *A quasi-interpolatory spline of order $p = 2r+1$ can be produced by filtering (2.17) with filters Γ of length no less than $2r+1$. There exists a unique filter Γ_m^r of length $2r+1$, which produces the minimal quasi-interpolatory spline $\tilde{S}_h^{2r+1}(x)$. Its transfer function is:*

$$\Gamma_m^r(z^2) = 1 + \sum_{k=1}^{r} \beta_k^r \lambda^k(z). \tag{2.19}$$

The coefficients β_k^r can be derived from the generating function

$$\left(\frac{2\arcsin t/2}{t}\right)^{2r+1} = \sum_{k=0}^{\infty} (-1)^k \beta_k^r t^{2k}. \tag{2.20}$$

If the function f has $2r+3$ derivatives then the following asymptotic relations hold for the minimal quasi-interpolatory spline $\tilde{S}_h^{2r+1}(x)$ for $x = h(k+1/2+\tau)$, $\tau \in [0,1]$:

$$\tilde{S}_h^{2r+1}(x) = f(x) - h^{2r+1}f^{(2r+1)}(x)\frac{b_{2r+1}(\tau)}{(2r+1)!} + h^{2r+2}f^{(2r+2)}(x)\left(\frac{(2r+1)b_{2r+2}(\tau)}{(2r+2)!} - \beta_{r+1}^r\right) + O(f^{(2r+3)}h^{2r+3}), \tag{2.21}$$

where $b_s(x)$ is the Bernoulli polynomial of degree s.

We recall that the values $b_s \stackrel{\Delta}{=} b_s(0)$ are called the Bernoulli numbers. If s is odd and $s > 1$ then $b_s = 0$. Hence, values of the minimal quasi-interpolatory spline in the midpoints between nodes are the following :

$$\tilde{S}_h^{2r+1}(h(k+\frac{1}{2})) = f(h(k+\frac{1}{2})) + h^{2r+2}f^{(2r+2)}(h(k+\frac{1}{2}))A^r + O(f^{(2r+3)}h^{2r+3}), \; A^r \stackrel{\Delta}{=} \frac{(2r+1)b_{2r+2}}{(2r+2)!} - \beta_{r+1}^r (2.22)$$

This implies the super-convergence property similar to that of the interpolatory splines. The minimal quasi-interpolatory spline of order $2r+1$ (of degree $2r+1$) restores in the midpoints between nodes polynomials of degree $2r+1$. The asymptotic representation (2.22) allows to enhance the approximation accuracy of the minimal spline in the midpoints upgrading the filter Γ_m^r.

Proposition 2.3. *If the coefficients of a spline $\check{S}_h^{2r+1} \in \mathbf{S}_h^{2r+1}$ of order $2r+1$ are derived in line with (2.17) using the filter Γ_e^r of length $2r+3$ with the transfer function $\Gamma_e^r(z^2) = \Gamma_m^r(z^2) - A^r\lambda^{r+1}(z)$ then the spline restores polynomials of degree $2r+3$ in the midpoints between nodes.*

There exists one more way to upgrade the filter Γ_m^r, which retains the approximation accuracy of the minimal spline in the midpoints but allows custom design of splines.

Proposition 2.4 (Ref. 8). *If the coefficients of a spline $S_{h,\rho}^{2r+1} \in \mathbf{S}_h^{2r+1}$ of order $2r+1$ are derived in line with (2.17) using the filter Γ_ρ^r of length $2r+3$ with the transfer function $\Gamma_\rho^r(z^2) = \Gamma_m^r(z^2) + \rho\lambda^{r+1}(z)$ then the spline restores polynomials of degree $2r+1$ in the midpoints between nodes for any real value ρ.*

If the parameter ρ is chosen such that $\rho = (-1)^r|\rho|$ than the spline $S_{h,\rho}^{2r+1}$ possesses the smoothing property. It means that it approximates values of the function f in the midpoints between nodes being constructed from the noised samples of the function: $\tilde{\mathbf{f}} \stackrel{\Delta}{=} \{\tilde{f}_k = f(hk) + \varepsilon_k\}, \; k \in \mathbb{Z}$.

2.4. Examples

First we present a few values of the coefficients β_k^r:

$$\beta_1^r = -\frac{2r+1}{24},\ \beta_2^r = \frac{(2r+1)(10r+27)}{5760},\ \ \beta_3^r = -\frac{(2r+1)(140r^2+1064r+2025)}{2903040}.$$

For the construction of the extended splines the values of the Bernoulli numbers b_s are required. The initial values are: $b_2 = 1/6,\ b_4 = -1/30,\ b_6 = 1/42$.

2.4.1. Quadratic splines

Interpolatory spline Denote $\alpha = z^{-1} + z$.

$$U_i^1(z) = \frac{4\alpha}{z^2+6+z^{-2}},\ 1 - U_i^1(z) = \frac{(\alpha-2)^2}{z^{-2}+6+z^2}, \tag{2.23}$$

Minimal spline The filters are

$$\Gamma_m^1(z^2) = 1 - \frac{1}{8}\lambda(z),\ U_m^1(z) = \frac{-z^{-3}+9z^{-1}+9z-z^3}{16},\ 1 - U_m^1(z) = \frac{(\alpha-2)^2(z^{-1}+4+z)}{16}, \tag{2.24}$$

.

Extended spline

$$\Gamma_e^1(z) = \Gamma_m^1(z^2) + \frac{3}{128}\lambda^2(z),\ U_e^1(z) = \frac{3z^{-5}-25z^{-3}+150z^{-1}+150z-25z^3+3z^5}{256}, \tag{2.25}$$
$$1 - U_e^1(z) = \frac{(\alpha-2)^3(3z^{-2}+18z^{-1}+38+18z+3z^2)}{256},$$

Remark Donoho Ref. 1 presented a scheme where an odd sample is predicted by the value in the central point of the polynomial of an odd degree which interpolates adjacent even samples. One can observe that our filter U_m^1 coincides with the filter derived by Donoho's scheme using the cubic interpolatory polynomial. The filter U_e^1 coincides with the filter derived using the interpolatory polynomial of fifth degree. On the other hand the filter U_i^1 is closely related to the commonly used Butterworth filter Ref. 4. Namely, in this case the filter transfer functions $\Phi_i^{1,l}(z) \triangleq (1+U_i^1(z))/2,\quad \Phi_i^{1,h}(z) \triangleq (1-U_i^1(z))/2$. coincide with magnitude squared the transfer functions of the discrete-time low-pass and high-pass half-band Butterworth filters of order 4, respectively.

2.4.2. Splines of fifth order (fourth degree)

Interpolatory spline

$$U_i^2(z) = \frac{16(z^3+11z+11z^{-1}+z^{-3})}{z^4+76z^2+230+76z^{-2}+z^{-4}},\ 1 - U_i^2(z) = \frac{(\alpha-2)^3\,(\alpha-10)}{z^4+76z^2+230+76z^{-2}+z^{-4}}. \tag{2.26}$$

Minimal spline The filters are

$$\Gamma_m^2(z^2) = 1 - \frac{5}{24}\lambda(z) + \frac{47}{1152}\lambda^2(z),\ U_m^2(z) = \frac{47(z^{-7}+z^7)+89(z^{-5}+z^5)-2277(z^{-3}+z^3)+15965\alpha}{27648}$$
$$1 - U_m^2(z) = \frac{-(\alpha-2)^3\,(47(z^{-4}+z^4)+282(z^{-3}+z^3)+1076(z^{-2}+z^2)+3166\alpha+5414}{27648}.$$

3. WAVELET TRANSFORMS USING SPLINE FILTERS

3.1. Choosing the filters for the *lifting* step

In the previous section we presented a family of the filters U for the *predict* step which originate from splines of various types. But, as it is seen from (1.2), to accomplish the transform we have to define the filter β. There is a remarkable freedom of the choice of these filters. The only requirement needed to guarantee the perfect reconstruction property of the transform is $\beta(-z) = \beta(z)$. In order to make synthesis and analysis filters similar in their properties, we choose $\beta(z) = \breve{U}(z)/2$, where $\breve{U}$ means one of filters U presented above. In particular, $\breve{U}$ may coincide with the filters U which was used for the prediction. We say that a wavelet ψ 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 3.1. *Suppose that for the* predict *and* lifting *steps filters $U(z)$ and $\beta(z) = \breve{U}(z)/2$ are used, respectively. If $1 - U(z)$ comprises the factor $(z - 2 + 1/z)^r$ then the high-frequency analysis wavelets $\tilde{\psi}^1$ have $2r$ vanishing moments. If, in addition $1 - \breve{U}(z)$ comprises the factor $(z - 2 + 1/z)^p$ then the synthesis wavelet ψ^1_β has $2q$ vanishing moments , where $q = \min\{p, r\}$.*

3.2. Implementation of transforms

Once we have chosen the filter $\beta = \breve{U}/2$, the decomposition procedure is the following (see (1.1), (1.2)):

$$d_1^u(z^2) = d_1(z^2) - zU(z)e_1(z^2), \quad e_1^u(z^2) = e_1(z^2) + \frac{1}{2}z^{-1}\breve{U}(z)\, d_1^u(z^2). \tag{3.27}$$

The transfer function $z^{-1}\breve{U}(z)/2$ differs from $z\breve{U}(z)/2$ only by the factor z^{-2}, that is the impulse responses of the corresponding filters are identical up to a shift. Thus both operations of the decomposition are, in principle, identical. For the reconstruction the same operation are conducted in a reverse order.

Therefore, it is sufficient to outline implementation of filtering with the function $zU(z)$. This function depends on z^2 and we denote it as $F(z^2) \triangleq zU(z)$. Then the decomposition formulas (3.27) are equivalent to the following

$$d_1^u(z) = d_1(z) - F(z)e_1(z), \quad e_1^u(z) = e_1(z) + \frac{1}{2z}\breve{F}(z)\, d_1^u(z). \tag{3.28}$$

Equation (3.28) means that in order to obtain the detail array $\mathbf{d}_1^u$, we must process the even array $\mathbf{e}_1$ by the filter F with the transfer function $F(z)$ and extract the filtered array from the odd array $\mathbf{d}_1$. In order to obtain the smoothed array $\mathbf{e}_1^u$, we must process the detail array $\mathbf{d}_1^u$ by the filter Φ that has the transfer function $\Phi(z) = z^{-1}\breve{F}(z)/2$ and add the filtered array to the even array $\mathbf{e}_1$. But the filter Φ differs from $\breve{F}_r/2$ only by one-sample delay and it operates similarly.

Implementation of FIR filters originated from local splines is straightforward and, therefore we only make a few remarks on IIR filters originated from interpolatory splines. Equations (2.9) and (2.10) imply that, while the interpolatory spline of order $2r+1$ is used, the transfer function

$$F(z) = \frac{P(z)}{\prod_{n=1}^{r} \frac{1}{\gamma_n}(1 + \gamma_n z)(1 + \gamma_n z^{-1})}, \tag{3.29}$$

where $P(z)$ is a Laurent polynomial. It means that the IIR filter F can be split into a cascade consisting of a FIR filter with the transfer function $P(z)$, r elementary causal recursive filters, denoted by $\overrightarrow{R(n)}$, and r elementary anti-causal recursive filters, denoted by $\overleftarrow{R(n)}$. On the other hand, the filters can be decomposed into sums of elementary recursive filters. Such a decomposition also allows parallel implementation of the transform.

The causal and anti-causal filters operate as follows:

$$\mathbf{y} = \overrightarrow{R(n)}\mathbf{x} \Longleftrightarrow y(l) = x(l) + \gamma_n y(l-1), \quad \mathbf{y} = \overleftarrow{R(n)}\mathbf{x} \Longleftrightarrow y(l) = x(l) + \gamma_n y(l+1). \tag{3.30}$$

3.3. Examples of recursive filters

Now we present IIR filers derived from the interpolatory splines of third and fifth orders.

$r = 1$. Denote $\gamma_1^1 = 3 - 2\sqrt{2} \approx 0.172$. Then

$$F_i^1(z) = 4\gamma_1^1 \frac{1+z}{(1+\gamma_1^1 z)(1+\gamma_1^1 z^{-1})}.$$

The filter can be implemented with the following cascade:

$$x_0(k) = 4\gamma_1^1(x(k) + x(k+1)), \quad x_1(k) = x_0(k) - \gamma_1^1 x_1(k-1), \quad y(k) = x_1(k) - \gamma_1^1 y(k+1).$$

Another option stems from the following decomposition of the function $F_i^1(z)$:

$$F_i^1(z) = \frac{4\gamma_1^1}{1+\gamma_1^1}\left(\frac{1}{1+\gamma_1^1 z^{-1}} + \frac{z}{1+\gamma_1^1 z}\right).$$

Then the filter is implemented in parallel mode:

$$y_1(k) = x(k) - \gamma_1^1 y_1(k-1), \quad y_2(k) = x(k+1) - \gamma_1^1 y_2(k+1), \quad y = \frac{4\gamma_1^1}{1+\gamma_1^1}\left(y_1 + y_2\right).$$

We note that elementary filters, which produce y_1 and y_2, are operating in opposite directions.

$r = 2$. Denote $\xi = 5-2\sqrt{6}$, $\gamma_1^2 = 19-4\sqrt{19}-2\sqrt{166-38\sqrt{19}} \approx 0.361$, $\gamma_2^2 = 19+4\sqrt{19}-2\sqrt{166+38\sqrt{19}} \approx 0.0137$. So, we have the cascade representation:

$$F_i^2(z) = \frac{16\gamma_1^2\alpha_2^2}{\xi} \cdot \frac{1+\xi z}{1+\gamma_1^2 z} \cdot \frac{1+\xi z^{-1}}{1+\gamma_1^2 z^{-1}} \cdot \frac{1}{1+\gamma_2^2 z} \cdot \frac{1}{1+\gamma_2^2 z^{-1}} \cdot (1+z).$$

4. APPLICATION TO IMAGE COMPRESSION

We carried out a series of experiments on compression of various images using a number of wavelet transforms constructed above.

4.1. Employed wavelet transforms

We explored various combinations of filters for the *prediction* and *lifting* steps obtaining different transforms. Most of them are performing satisfactory on compression of images. We display in the following figures wavelets and their spectra for the three types of transforms:

T_i^1- The transform where for the *prediction* and *lifting* steps the IIR filters U_i^1 are used (see (2.23) We display in Figure 1 the synthesis and analysis wavelets related to the transform T_i^1 and their Fourier spectra. One can observe that the analysis wavelets are regular and, to some extent, are similar to the synthesis ones. Both the analysis and synthesis high-frequency wavelets have 4 vanishing moments.

T_m^1- The transform where for the *prediction* and *lifting* steps the FIR filters U_m^1 are used (see (2.24) We display in Figure 2 the compactly supported synthesis and analysis wavelets related to the transform T_m^1 and their Fourier spectra. One can observe that, unlike the analysis wavelets, the synthesis wavelets are regular and are similar to the synthesis wavelets shown in Figure 1 ones. Both the analysis and synthesis high-frequency wavelets have 4 vanishing moments.

T_i^2- The transform where for the *prediction* and *lifting* steps the IIR filters U_i^2 are used (see (2.26) We display in Figure 3 the synthesis and analysis wavelets related to the transform T_i^2 and their Fourier spectra. One can observe that the analysis wavelets are regular and are similar to the synthesis ones. Both the analysis and synthesis high-frequency wavelets have 6 vanishing moments.

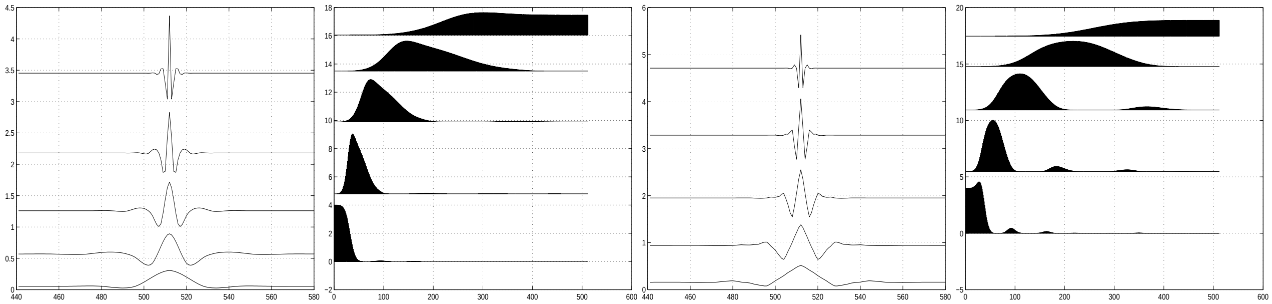

Figure 1. The left pair bottom to top: The synthesis wavelets used by the transform T_i^1: φ^4 and $\psi_\beta^k,\ k = 4, 3, 2, 1$ and their Fourier spectra. The right pair bottom to top: The analysis wavelets used by the transformT_i^1: $\tilde{\varphi}_\beta^4$ and $\psi_\beta^k,\ k = 4, 3, 2, 1$ and their Fourier spectra.

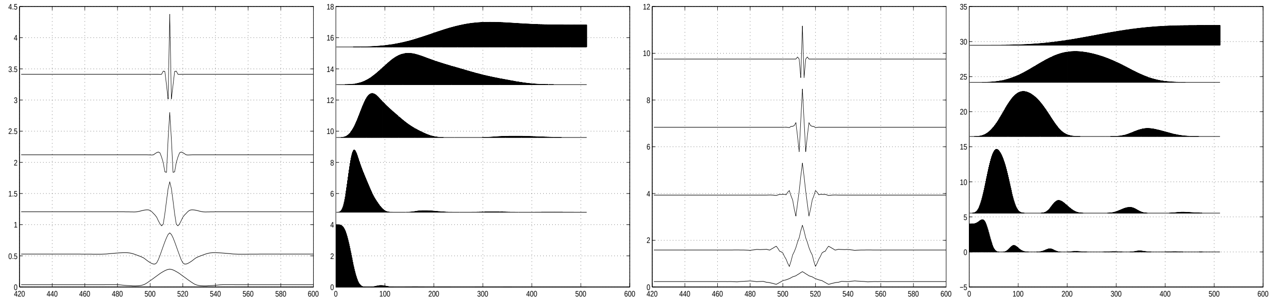

Figure 2. The left pair bottom to top: The synthesis wavelets used by the transform T_m^1: φ^4 and $\psi_\beta^k,\ k = 4, 3, 2, 1$ and their Fourier spectra. The right pair bottom to top: The analysis wavelets used by the transformT_i^1: $\tilde{\varphi}_\beta^4$ and $\psi_\beta^k,\ k = 4, 3, 2, 1$ and their Fourier spectra.

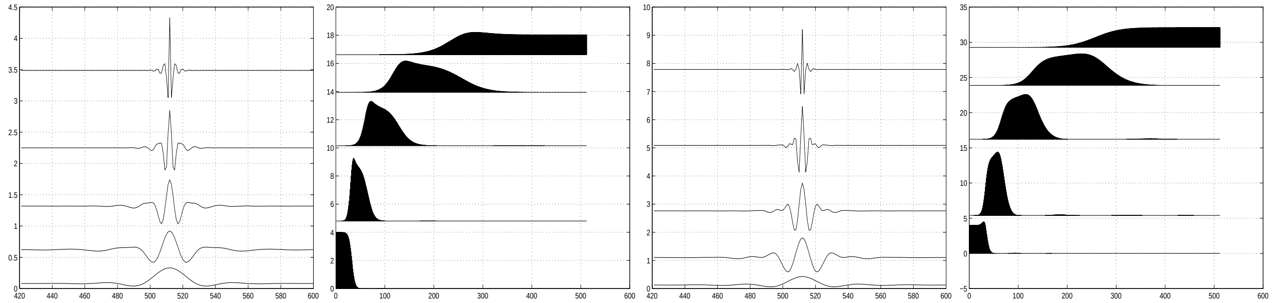

Figure 3. The left pair bottom to top: The synthesis wavelets used by the transform T_i^2: φ^4 and $\psi_\beta^k,\ k = 4, 3, 2, 1$ and their Fourier spectra. The right pair bottom to top: The analysis wavelets used by the transformT_i^1: $\tilde{\varphi}_\beta^4$ and $\psi_\beta^k,\ k = 4, 3, 2, 1$ and their Fourier spectra.

4.2. Experiments with image compression

We conducted a wide series of experiments with compression of multimedia images using the above transforms. In most experiments they outperform the popular B9/7 biorthogonal filter bank. In this section we present the results after compression and decompression of of four images. Two of them are commonly used images "Lena"and "Barbara"and other two "Car" and "Fabrics" are displayed in Figure 4. These are 512×512 8 bit (8bpp) images.

The following experiments were done:

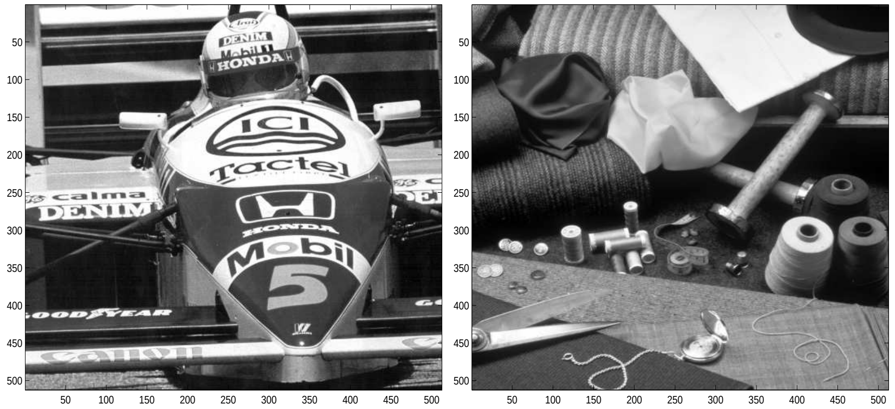

Figure 4. Left: “Car”, right: “Fabrics”.

1. We decomposed the image with the wavelet transform up to 6-th scale using the B9/7 filters and the filters listed in Section 4.1.
2. The transform coefficients were coded using the SPIHT algorithm by Said and Pearlman Ref. 6. This algorithm enables to compress data exactly to a prescribed rate. We coded the coefficients with 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 to noise ratio (PSNR) in decibels was computed.

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

Lena and Barbara: The PSNR values of “Lena”and “Barbara” images are presented in Table 3. One can observe that for “Lena’ the transform T_i^1 outperforms the B9/7 filter in any compression rate. The computational complexity of this transform is slightly higher than that of the B9/7 filter but using parallel mode it can be implemented faster. The FIR transform T_m^1, outperform the B9/7 filters in all CR except 10. The computational complexity of this transform is the same as that of the B9/7 filter but the advantage is that T_m^1 can be implemented in integers. Computationally more expensive transform T_i^2 in the parallel mode can be implemented faster than the transform with B9/7 filter. It is most efficient in the sense of PSNR.

Lena					
CR	B9/7	T_i^1	T_m^1	T_i^2	
10	37.70	37.82	37.67	37.83	
20	34.53	34.71	34.57	34.70	
30	32.33	32.62	32.56	32.72	
40	31.42	31.63	31.52	31.66	
50	30.70	30.91	30.82	30.92	

Barbara					
CR	B9/7	T_i^1	T_m^1	T_i^2	
10	33.01	33.39	32.91	33.74	
20	28.93	29.13	28.90	29.34	
30	26.99	27.03	26.91	27.39	
40	25.78	25.74	25.65	25.98	
50	25.10	25.00	24.86	25.17	

Table 3. PSNR of the “Lena” (left table) and “Barbara” (right table) images after the application of SPIHT where the decomposition of the wavelet transforms was into 6 scales.

Even more it is true for the compression of “Barbara”. Here T_i^2 remarkably outperforms all other transforms. The T_m^1 transform gives way to B9/7.

Car and Fabrics : The PSNR values of “Car” and “Fabrics” are presented in Table 4. On both images T_i^1 transform outperforms the B9/7 filter for any compression ratio. On “Car”, which comprise high-frequency components,the FIR transform T_m^1 proved to be efficient.

Car					
CR	B9/7	T_i^1	T_m^1	T_i^2	
10	32.57	32.63	32.72	32.49	
20	28.38	28.53	28.46	28.42	
30	26.78	26.99	26.89	26.90	
40	25.05	25.19	25.11	25.15	
50	24.40	24.52	24.43	24.44	

Fabrics					
CR	B9/7	T_i^1	T_m^1	T_i^2	
10	34.95	34.96	34.86	34.95	
20	31.53	31.53	31.44	31.52	
30	29.62	29.79	29.73	29.73	
40	28.90	29.00	28.93	29.01	
50	28.41	28.52	28.44	28.50	

Table 4. PSNR of the “Car” (left table) and “Fabrics” (right table) images.

5. CONCLUSIONS

In this paper we proposed an efficient method that produces new filters for wavelet transforms. These wavelet transforms were designed by the usage of polynomial interpolatory and quasi-interpolatory splines. Being applied to the compression of multimedia images, these filters outperform the traditional biorthogonal 9/7 filters which are frequenty used in wavelet based compression. The new filters and the biorthogonal 9/7 are incorporated into SPIHT in order to measure and compare their performance with one well known codec. We presented also wavelet filters with controlling parameter, which can be used, in particular for processing noised signals and images.

In the future we plan to explore this opportunity and to check the performance of devised filters on video and seismic data.

REFERENCES

1. D. L. Donoho, “ Interpolating wavelet transform”, Preprint 408, Department of Statistics, Stanford University, 1992.
2. I. J. Schoenberg, Cardinal spline interpolation, CBMS, **12**, SIAM, Philadelphia, 1973.
3. I. J. Schoenberg, “ Contribution to the problem of approximation of equidistant data by analytic functions” , *Quart. Appl. Math.*, **4** (1946), 45-99, 112-141.
4. A. V. Oppenheim, R. W. Shafer, *Discrete-time signal processing*, Englewood Cliffs, New York, Prentice Hall, 1989.
5. G. Strang, and T. Nguen, *Wavelets and filter banks*, Wellesley-Cambridge Press, 1996.
6. 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: 243-250, June 1996.
7. V. A. Zheludev “ Local spline approximation on a uniform grid”, *U.S.S.R. Comput. Math. & Math. Phys.*, **27** No.5, (1987), 8-19.
8. V.A. Zheludev, “Local smoothing splines with a regularizing parameter”, *Comput. Math. & Math Phys.*, **31**, (1991), 193-211.