Interpolatory biorthogonal multiwavelet transforms based on Hermite splines

A. Averbuch
V. Zheludev

Abstract. We present new multiwavelet transforms for discrete signals. The transforms are implemented in two phases: 1. Pre (post)processing which transform the scalar signal into the vector one (and back). 2.Wavelet transforms of the vector signal. Both phases are performed in a lifting manner. We use the cubic interpolatory Hermite splines as a predicting aggregate in the vector wavelet transform. We present new pre(post)processing algorithms which do not degrade the approximation accuracy of the vector wavelet transforms. As a result we get fast biorthogonal algorithms to transform discrete-time signals which are exact on the sampled cubic polynomials. The bases for the transform are symmetric and have short support.

§1. Introduction

Hermite splines were used for the construction of the first examples of multiwavelets [8, 10, 11]. In these works the generating functions of Hermite splines were employed as scaling functions for the multiresolution analysis, and the corresponding wavelets were derived. Our approach is different. This construction produced biorthogonal bases for $L_2(\mathbb{R})$ or $L_2([a,b])$. But this theory can not be immediately applied to processing of discrete-time signals. While the original signal is scalar, the input stream to the multiwavelet transform must be a vector sequence. This sequence is produced by the so called preprocessing of the signal. Reconstruction of the scalar signal from the vector sequence is called postprocessing. So, a multiwavelet transform of a signal actually consists of two phases: 1).Pre (post)processing which transform the scalar signal into the vector one (and back). 2). Wavelet transforms of the vector signal.

For construction of the interpolatory cubic Hermite splines we have to produce two groups of coefficients. The first group contains samples from the signal while the second group contains samples of its derivative. It is clear that

Trends in Approximation Theory
Kirill Kopotun, Tom Lyche, and Mike Neamtu (eds.), pp. 1–3.

typically the values of the derivatives are unavailable and have to be evaluated from the samples $\{f(k)\}$. If the splines are built with the proper coefficients, the cubic polynomials can be restored. Existing preprocessing algorithms which are similar to the Haar wavelet transform, degrade the approximation accuracy of the cubic Hermite splines (see, for example [12]).

Recently, lifting schemes [13] became very popular for construction of biorthogonal wavelet transforms . They have the advantages of being fast, consume relatively small memory and have inherent tools for custom design of filters and waveforms. In its simplest form, the lifting scheme consists of three steps: 1. Splitting the signal into even and odd sub-arrays. 2. Prediction of the odd array through linear combinations of even samples and extraction of the predicted values from the existing ones. 3. Lifting (Updating) the even array using already updated odd array which eliminates aliasing. The interpolatory lifting schemes, where the odd values are predicted by the values in the midpoints of some function which interpolates the even values, are important for signal processing. If this is the case, then the scheme is purely discrete and there is no need to have a scaling function. Such scheme was first devised in [5, 13] with the use of Deslauriers-Dubuc interpolatory splines . In [14] we presented constructions based on polynomial splines and in [1] it was based on discrete interpolatory splines.

In this paper we use an approach which is close to [1, 14] for a multiwavelet construction. We propose to use the lifting scheme for the construction and implementation of the multiwavelet transforms. We use the cubic interpolatory Hermite splines as a predicting aggregate in the vector wavelet transform. Together with the multiwavelet transforms we present the pre(post)processing algorithms which do not degrade the approximation accuracy of the transforms. These algorithms are derived by lifting the Haar pre(post)processing that was mentioned above. A characteristic feature of our approach is that we are staying completely in the discrete-time setting. Therefore, we abandon the notions of scaling functions and multiresolution analysis. Instead, we tightly link the multiwavelet transforms with the pre(post)processing procedures. As a result we get fast biorthogonal algorithms to transform discrete-time signals which are exact on the sampled cubic polynomials. This approach can be extended in a straightforward manner to handle higher order accuracy.

Recently, other papers such as [3, 7] linked between lifting schemes and multiwavelet transforms. But these papers are not using the Hermite splines and any pre (post)processing algorithms.

§2. Preliminaries

Let S be a cubic Hermite spline constructed on the equidistant grid $\{2hk\}$, $k \in \mathbb{Z}$, $h \in \mathbb{R}$, satisfying the boundary conditions $S(2hk) = s_1(k)$, $hS'(2hk) = s_2(k)$. We need to have the values of the spline S and its derivative in the midpoints of the intervals.

$$S(h(2k+1)) = \frac{1}{2}s_1(k) + \frac{1}{4}s_2(k) + \frac{1}{2}s_1(k+1) - \frac{1}{4}s_2(k+1) \qquad (2.1)$$

$$S'(h(2k+1)) = -\frac{3}{4}s_1(k) - \frac{1}{4}s_2(k) + \frac{3}{4}s_1(k+1) - \frac{1}{4}s_2(k+1). \quad (2.2)$$

Equations (2.1) and (2.2) provide an approximation scheme of the fourth order in the following sense:

Proposition 2.1. *Assume $F \in C^4$. If $s_1(k) = F(2hk)$ and $s_2(k) = hF'(2hk)$ then $S(h(2k+1))=F(h(2k+1)) + O(h^4F^{(4)})$, $hS'(h(2k+1))=hF'(h(2k+1)) + o(h^4F^{(4)})$.*

§3. Lifting scheme for the vector wavelet transform

Assume that a signal $F \in C^1$ is sampled on the grid $\{kh/2\}$ and $\boldsymbol{F} = \{F(kh/2\},\ k \in \mathbb{Z}$. Later on we will present preprocessing algorithms which transform this scalar array into the vector signal $\vec{f} \triangleq (f_1(k), f_2(k))^T$ where $f_1(k) = \Phi(kh)$, $f_2(k) \approx h\Phi'(kh)$, $k \in \mathbb{Z}$ and Φ is a function derived from F.

For now, assume that we are already given this vector signal $\vec{f}$. The lifting scheme for the wavelet transform was presented originally in [13]. In its simplest form, it consists of three steps: 1.Split, 2.Predict and 3.Lifting or Update. We apply the wavelet transform on the vector signal $\vec{f}$ using these steps. We introduce two versions of the transform which are dual to each other.

Primal mode

Decomposition: 1). We split the vector signal into even and odd sub-arrays: $\vec{f} = \vec{s} + \vec{d}$ where $\vec{s}(k) \triangleq \vec{f}(2k)$, $\vec{d}(k) \triangleq \vec{f}(2k+1)$, $k \in \mathbb{Z}$. The z-transforms of $\vec{s}$ and $\vec{d}$ are: $\vec{s}(z^2) = (\vec{f}(z) + \vec{f}(-z))/2$, $\vec{d}(z^2) = z(\vec{f}(z) - \vec{f}(-z))/2$.

2). We predict the values of the signal $\vec{d}$ by the values of the Hermite spline, which interpolates the signal $\vec{s}$ on the grid $\{2kh\}$, and update the $\vec{d}-$array by extracting the predicted values from the existing ones. From (2.1) and (2.2) we have in the $z-$domain:

$$\vec{d}^u(z) = \vec{d}(z) - \boldsymbol{A}(z)\,\vec{s}(z) \quad \boldsymbol{A}(z) \triangleq \begin{pmatrix} \frac{1}{2}(1+z) & \frac{1}{4}(1-z) \\ -\frac{3}{4}(1-z) & -\frac{1}{4}(1+z) \end{pmatrix}.$$

3). Last step updates the even array $\vec{s}$ using the array $\vec{d}^u$ and some multifilter $\boldsymbol{B}$: $\vec{s}^u = \vec{s} + \boldsymbol{B} * \vec{d}^u \iff \vec{s}^u(z) = \vec{s}(z) + \boldsymbol{B}(z)\vec{d}^u(z)$. Later on we provide suggestions to choosing $\boldsymbol{B}$.

The reconstruction is performed in reverse order. 1). Given two updated vector arrays $\vec{s}^u$ and $\vec{d}^u$, we restore the $\vec{s}-$array: $\vec{s}(z) = \vec{s}^u(z) - \boldsymbol{B}(z)\vec{d}^u(z)$.

2). Restore the values of the signal $\vec{d}$: $\vec{d}(z) = \vec{d}^u(z) + \boldsymbol{A}(z)\,\vec{s}(z)$.

3). Restore the signal $\vec{f}$ from its even and odd sub-arrays. In the $z-$domain we have $\vec{f}(z) = \vec{s}(z^2) + z^{-1}\vec{d}(z^2)$.

Dual mode

Decomposition. 1). As before, we split the vector signal into even and odd sub-arrays. 2). We predict the values of the signal $\vec{s}$ by the values of the

Hermite spline, which interpolates the signal $\vec{d}$ on the grid $\{(2k+1)h\}$. Unlike the primal mode, we update $\vec{s}$ by adding the predicted values to the existing ones. We have in the $z-$domain: $\vec{s}^u(z) = \vec{s}(z) + z^{-1}\boldsymbol{A}(z)\,\vec{d}(z)$. 3). We update d-array using already updated $s-$ array: $\vec{d}^u(z) = \vec{d}(z) - z\boldsymbol{B}(z)\vec{s}^u(z)$.

Reconstruction is implemented in reverse order:

1). $\vec{d}(z) = \vec{d}^u(z) + z\boldsymbol{B}(z)\vec{s}^u(z)$.

2.) $\vec{s}(z) = \vec{s}^u(z) - z^{-1}\boldsymbol{A}(z)\,\vec{d}(z)$.

3.) $\vec{f}(z) = \vec{s}(z^2) + z^{-1}\vec{d}(z^2)$.

§4. Multifilter banks

4.1 Structure of multifilter banks

The operations on vector signals, which we implemented in a lifting manner, can be viewed as processing the signal by biorthogonal perfect reconstruction multifilter banks.

We define a family of matrix filters through their $z-$transforms:

$$\tilde{\boldsymbol{G}}(\boldsymbol{z}) \triangleq z\left(\boldsymbol{I} - \frac{\boldsymbol{A}(z^2)}{z}\right), \quad \tilde{\boldsymbol{H}}_{\boldsymbol{b}}(\boldsymbol{z}) \triangleq [\boldsymbol{I} + \boldsymbol{B}(z^2)\tilde{\boldsymbol{G}}(z)].$$

$$\boldsymbol{H}(z) \triangleq \left[\boldsymbol{I} + \frac{\boldsymbol{A}(z^2)}{z}\right], \quad \boldsymbol{G}_{\boldsymbol{b}}(z) \triangleq \frac{1}{z}\left(\boldsymbol{I} - \boldsymbol{H}(z)z\boldsymbol{B}(z^2)\right).$$

Theorem 4.1. *The primal decomposition transforms of the vector signal $\vec{f}$ into $\vec{d}^u \bigcup \vec{s}^u$ can be represented as filtering the $\vec{f}$ with the multifilters $\tilde{\boldsymbol{G}}$ and $\tilde{\boldsymbol{H}}_{\boldsymbol{b}}$ followed by downsampling:*

$$\vec{d}^u(z^2) = \frac{1}{2}\left[\tilde{\boldsymbol{G}}(z)\vec{f}(z) + \tilde{\boldsymbol{G}}(-z)\vec{f}(-z))\right],$$
$$\vec{s}^u(z^2) = \frac{1}{2}\left[\tilde{\boldsymbol{H}}_{\boldsymbol{b}}(\boldsymbol{z})\vec{f}(z) + \tilde{\boldsymbol{H}}_{\boldsymbol{b}}(-\boldsymbol{z})\vec{f}(-z))\right].$$

Reconstruction of the signal $\vec{f}$ from $\vec{s}^u$ and $\vec{d}^u$ can be implemented as filtering the arrays with the multifilters $\boldsymbol{H}$ and $\boldsymbol{G}_{\boldsymbol{b}}$, respectively, preceded by upsampling: $\vec{f}(z) = \boldsymbol{H}(z)\vec{s}^u(z^2) + \boldsymbol{G}_{\boldsymbol{b}}(z)\vec{d}^u(z^2)$. The reconstruction multifilters $\boldsymbol{H}$ and $\boldsymbol{G}_{\boldsymbol{b}}$ and the decomposition multifilters $\tilde{\boldsymbol{G}}$ and $\tilde{\boldsymbol{H}}_{\boldsymbol{b}}$ form a perfect reconstruction filter bank. This means that the following relations are true for any $\boldsymbol{B}$:

$$\boldsymbol{H}(z)\tilde{\boldsymbol{H}}_{\boldsymbol{b}}(\mathbf{z}) + \boldsymbol{G}_{\boldsymbol{b}}(z)\tilde{\boldsymbol{G}}(\mathbf{z}) = 2\boldsymbol{I},$$
$$\boldsymbol{H}(z)\tilde{\boldsymbol{H}}_{\boldsymbol{b}}(-\boldsymbol{z}) + \boldsymbol{G}_{\boldsymbol{b}}(z)\tilde{\boldsymbol{G}}(-\boldsymbol{z}) = 0.$$

Proposition 4.1. *Suppose that entries of the matrix $\boldsymbol{B}(z)$ are Laurent polynomials and $\boldsymbol{B}(z)\boldsymbol{A}(z) = \boldsymbol{A}(z)\boldsymbol{B}(z)$. Then, the multifilters $\boldsymbol{H}(z)$, $\tilde{\boldsymbol{H}}_{\boldsymbol{b}}(\boldsymbol{z})$,*

$\boldsymbol{G_b}(z)$, $\tilde{\boldsymbol{G}}(\boldsymbol{z})$ constructed above are finite and the following relations between them hold: $\boldsymbol{H}(z) = (-z)^{-1}\tilde{\boldsymbol{G}}(-z)$, $\quad \tilde{\boldsymbol{H}}_{\boldsymbol{b}}(z) = -z\boldsymbol{G_b}(-z)$.

When the transform is implemented in the dual mode, similar relations hold. The reconstruction multifilters for the dual mode $\boldsymbol{G^d}$, $\boldsymbol{H_b^d}$ coincide (up to a constant factor) with the primal decomposition multifilters for the primal mode and the dual decomposition multifilters $\tilde{\boldsymbol{H}}^{\boldsymbol{d}}$, $\tilde{\boldsymbol{G}}_{\boldsymbol{b}}^{\boldsymbol{d}}$ coincide with the primal reconstruction multifilters.

4.2 Approximation properties of the multifilters

Proposition 4.2. *The dual decomposition multifilter $\tilde{\boldsymbol{H}}^{\boldsymbol{d}}$ (primal reconstruction multifilter $\boldsymbol{H}$) reproduces cubic polynomials. The primal decomposition multifilter $\tilde{\boldsymbol{G}}$ (dual reconstruction multifilter $\boldsymbol{G^d}$) eliminates cubic polynomials. Namely, if $\vec{f} = (P(kh),\, hP'(kh))^T$ then*

$$\tilde{\boldsymbol{H}}^{\boldsymbol{d}} * \vec{f} = 2\vec{f}, \quad \tilde{\boldsymbol{G}} * \vec{f} = \vec{0}. \tag{4.3}$$

Let us turn now to the dual decomposition multifilter $\tilde{\boldsymbol{G}}_{\boldsymbol{b}}^{\boldsymbol{d}}$ (primal reconstruction multifilter $\boldsymbol{G_b}$) and the primal decomposition multifilter $\tilde{\boldsymbol{H}}_{\boldsymbol{b}}$ (dual reconstruction multifilter $\boldsymbol{H_b^d}$) which depend on yet undefined multifilter $\boldsymbol{B}$. So far we required the entries of $\boldsymbol{B}(z)$ to be Laurent polynomials and $\boldsymbol{B}(z)$ to commute with $\boldsymbol{A}(z)$. So, we have a remarkable freedom in choosing $\boldsymbol{B}$. We use this opportunity to supply these $\boldsymbol{B}$-dependent multifilters with properties similar to (4.3). Therefore, the properties of the reconstruction multifilters are getting close to properties of the decomposition multifilters.

Proposition 4.3. *Once the multifilter $\boldsymbol{B}$ is chosen such that $z\boldsymbol{B}(z) = \boldsymbol{A}(z)/2$, the primal decomposition multifilter $\tilde{\boldsymbol{H}}_{\boldsymbol{b}}$ (dual reconstruction multifilter $\boldsymbol{H_b^d}$) reproduces cubic polynomials. The dual decomposition multifilter $\tilde{\boldsymbol{G}}_{\boldsymbol{b}}^{\boldsymbol{d}}$ (primal reconstruction multifilter $\boldsymbol{G_b}$) eliminates cubic polynomials.*

We choose $\boldsymbol{B}$ as the shortest multifilter which supplies dual transform with properties similar to the primal one. However, many more various options for choosing $\boldsymbol{B}$ are possible.

§5. Lifting algorithms for pre/postprocessing phases

We constructed two schemes of the vector wavelet transform, which are dual to each other and are similar in their properties. The multifilters, which are used in these transforms, are finite, symmetric and possess the fourth order approximation property. The input to the transforms must be a sampled vector signal $\vec{f} = (f_1(k), f_2(k))^T$ such that $f_1(k) \stackrel{\Delta}{=} \Phi(kh)$ were samples of a function Φ on the grid $\{kh\}$ and $f_2(k) \approx h\Phi'(kh)$. To achieve that order of approximation, the difference

$$f_2(k) - h\Phi'(kh) = O(h^4\Phi^{(4)}) \ \ \forall\Phi \in C^4. \tag{5.4}$$

The goal of preprocessing in our scheme is to create from the scalar sampled signal $\boldsymbol{F} = \{F(kh/2\}$ the vector signal $\vec{f} = (f_1(k), f_2(k))^T$ with such a property. The postprocessing procedure must restore the scalar signal from the vector which is an output of the inverse wavelet transform.

When we have pre/postprocessing schemes which are finite, symmetric and possess the approximation property (5.4) they fit the vector wavelet transforms constructed in the previous sections. In this section we present such schemes.

An obvious way to produce such a vector $\vec{f}$ is to take $f_1(k) = F(kh)$ and to find $f_2(k)$ as an approximation to $F'(kh)$ from the samples $\{F(lh/2\}$. However, it can be proved that, if we require the postprocessing filters to be finite and symmetric, we can not achieve in this way an approximation order higher than two. Therefore, we chose not to leave $f_1(k) = F(kh)$ unchanged but to update them using neighboring samples $\{F(lh/2\}$. In other words, we will convert the input signal F into a function Φ such that $f_1(k) = \Phi(kh)$, $f_2(k) \approx h\Phi'(kh)$ and (5.4) holds. If $\{F(lh/2\}$ are samples of a cubic polynomial, we will require that $f_1(k) = \Phi(kh)$ and $f_2(k) = h\Phi'(kh)$.

Similarly to the vector wavelet transforms, we construct and implement the preprocessing by lifting steps. It is common in lifting algorithms that the calculations are performed "in place", and postprocessing is performed in a reverse order.

Denote $s(k) = F(kh)$ and $d(k) = F((k+1/2)h)$. First we introduce a simple preparatory scheme which illustrates our approach and serves as a basis for higher order algorithms. Similar algorithm is presented in [12].

An orthogonal scheme of third approximation order. Let us update the $s-$ and $d-$arrays as follows:

$$\begin{aligned} d^u(k) &= d(k) - s(k) = F((k+1/2)h) - F(kh) \\ s^u(k) &= s(k) + d^u(k)/2 = (F((k+1/2)h) + F(kh))/2 \end{aligned} \tag{5.5}$$

and put $f_1(k) = s^u(k),\ f_2(k) = 2d^u(k)$. The postprocessing is implemented in a reverse order: Denote: $\Phi(x) \stackrel{\Delta}{=} (F(x+h/2) + F(x))/2$.

Proposition 5.1. *Assume $F \in C^3$. Then, $f_1(k) = \Phi(kh)$ and $f_2(k) = h\Phi'(kh) + O(h^3 F^{(3)})$. In particular, if $F(x)$ is a quadratic polynomial then $\Phi(x)$ is a quadratic polynomial as well and $f_2(k) = h\Phi'(kh)$.*

Equations (5.6) define a finite-length vector signal $\vec{\varphi}^0 = (\varphi_1^0, \varphi_2^0)^T$ and $\tilde{\vec{\varphi}}^0 = (\tilde{\varphi}_1^0, \tilde{\varphi}_2^0)^T$ which we call the postprocessing and the preprocessing multiwavelets of zero level, respectively.

$$\vec{\varphi}^0(0) = \left(1, -\frac{1}{4}\right)^T,\ \vec{\varphi}^0(1) = \left(1, \frac{1}{4}\right)^T,\quad \tilde{\vec{\varphi}}^0(0) = \left(\frac{1}{2}, -2\right)^T,\ \vec{\varphi}^0(1) = \left(\frac{1}{2}, 2\right)^T. \tag{5.6}$$

We can observe that $\tilde{\varphi}_1^0(k) = \varphi_1^0(k)/2$ and $\tilde{\varphi}_2^0(k) = 2\varphi_2^0(k)$. This pre/postprocessing scheme is, actually, the orthogonal Haar wavelet transform of a signal.

Now we derive two different schemes of fifth order. To get it, we update the arrays s^u and d^u produced in Eq. (5.5) in order to increase their approximation order while retaining the symmetry and finite support of the corresponding multiwavelets.

Scheme I of fifth order. Let us update the s^u-array as follows: $s^{uu}(k) = s^u(k) - (d^u(k+1) - d^u(k-1))/48$ while $f_1(k) \stackrel{\Delta}{=} s^{uu}(k)/2$ and $f_2(k) \stackrel{\Delta}{=} d^u(k)$

Postprocessing is implemented in a reverse order:

$$s^{uu}(k)=2f_1(k),\ d^u(k)=f_2(k),\ s^u(k)=s^{uu}(k)+\frac{1}{48}\left(d^u(k+1)-d^u(k-1)\right),$$
$$s(k)=s^u(k)-\frac{1}{2}d^u(k),\ d(k)=d^u(k)+s(k),\ F((k+\frac{1}{2})h)=d(k),\ F(kh)=s(k).$$

Denote: $\Phi(x) \stackrel{\Delta}{=} \frac{1}{96}\Big(- F(x+\frac{3}{2}h) + F(x+h) + 24F(x+\frac{1}{2}h) + 24F(x) - F(x-h) + F(x-\frac{1}{2}h)\Big)$.

Proposition 5.2. *Assume $F \in C^5$. Then, $f_1(k) = \Phi(kh)$ and $f_2(k) = h\Phi'(kh) + O(h^5 F^{(5)})$. In particular, if $F(x)$ is a polynomial of fourth degree then $\Phi(x)$ is a polynomial of fourth degree as well and $f_2(k) = h\Phi'(kh)$.*

We define the postprocessing $\vec{\varphi}^0 = (\varphi_1^0, \varphi_2^0)^T$ and preprocessing multiwavelets $= (\tilde{\varphi}_1^0, \tilde{\varphi}_2^0)^T$ as follows:

$$\vec{\varphi}^0(-2) = \begin{pmatrix} 0 \\ \frac{1}{48} \end{pmatrix} = \vec{\varphi}^0(-1) = -\vec{\varphi}^0(2) = -\vec{\varphi}^0(3),\ \vec{\varphi}^0(1) = \begin{pmatrix} 2 \\ \frac{1}{2} \end{pmatrix}\ \vec{\varphi}^0(0) = \begin{pmatrix} 2 \\ -\frac{1}{2} \end{pmatrix}.$$
$$\tilde{\vec{\varphi}}^0(2) = \begin{pmatrix} \frac{1}{96} \\ 0 \end{pmatrix} = \tilde{\vec{\varphi}}^0(-1) = -\tilde{\vec{\varphi}}^0(-2) = -\tilde{\vec{\varphi}}^0(3),\ \tilde{\vec{\varphi}}^0(1) = \begin{pmatrix} \frac{1}{4} \\ 1 \end{pmatrix}\ \tilde{\vec{\varphi}}^0(0) = \begin{pmatrix} \frac{1}{4} \\ -1 \end{pmatrix}.$$

Scheme II of fifth order. Another way to increase the approximation order of the Haar preprocessing scheme is to update the d^u- rather than the s^u- array. We do it in the following way: $d^{uu}(k) = d^u(k)+\frac{1}{32}\left(s^u(k+1) - s^u(k-1)\right)$ while taking $f_1(k) = \frac{9}{6}s^u(k)$ and $f_2(k) = d^{uu}(k)$.

Postprocessing is implemented in reverse order:

$$d^{uu}(k)=f_2(k),\ s^u(k)=\frac{16}{9}f_1(k),\ d^u(k)=d^{uu}(k) - \frac{1}{32}\left(s^u(k+1)-s^u(k-1)\right)$$
$$s(k)=s^u(k) - \frac{1}{2}d^u(k),\ d(k)=d^u(k)+s(k),\ F((k+\frac{1}{2})h)=d(k),\ F(kh)=s(k).$$

Denote: $\Phi(x) \stackrel{\Delta}{=} 9/32\left(F(x+1/2h) + F(x)\right)$.

Proposition 5.3. *Assume $F \in C^5$. Then $f_1(k) = \Phi(kh)$ and $f_2(k) = h\Phi'(kh) + O(h^5 F^{(5)})$. In particular, if $F(x)$ is a polynomial of fourth degree then $\Phi(x)$ is a polynomial of fourth degree as well and $f_2(k) = h\Phi'(kh)$.*

We define the postprocessing $\vec{\varphi}^0 = (\varphi_1^0, \varphi_2^0)^T$ and preprocessing multiwavelets $= (\tilde{\varphi}_1^0, \tilde{\varphi}_2^0)^T$ as follows:

$$\vec{\varphi}^0(-2) = \begin{pmatrix} \frac{1}{36} \\ 0 \end{pmatrix} = \vec{\varphi}^0(-1) = -\vec{\varphi}^0(2) = \vec{\varphi}^0(3),\ \vec{\varphi}^0(1) = \begin{pmatrix} \frac{16}{9} \\ \frac{1}{2} \end{pmatrix}, \vec{\varphi}^0(0) = \begin{pmatrix} \frac{16}{9} \\ -\frac{1}{2} \end{pmatrix}.$$

$$\tilde{\vec{\varphi}}^0(2)=\begin{pmatrix}0\\ \frac{1}{64}\end{pmatrix}=-\tilde{\vec{\varphi}}^0(-1)=-\tilde{\vec{\varphi}}^0(-2)=\tilde{\vec{\varphi}}^0(3),\ \tilde{\vec{\varphi}}^0(1)=\begin{pmatrix}\frac{9}{32}\\ 1\end{pmatrix},\ \tilde{\vec{\varphi}}^0(0)=\begin{pmatrix}\frac{9}{32}\\ -1\end{pmatrix}.$$

§6. Bases for the space of discrete-time signals

6.1 Bases of zero level

In the previous section we introduced three postprocessing and three preprocessing multiwavelets of zero level $\vec{\varphi}^0=(\varphi_1^0,\varphi_2^0)^T$ and $\tilde{\vec{\varphi}}^0=(\tilde{\varphi}_1^0,\tilde{\varphi}_2^0)^T$ respectively. These multiwavelets correspond to the preprocessing algorithm of third approximation order and to two algorithms of fifth order. Their translations form three biorthogonal bases for the space of scalar discrete signals. Namely, the following assertion holds.

Proposition 6.1. *The signal* $\boldsymbol{F}=\{F(kh/2\}_{k\in\mathbb{Z}}$ *can be represented as follows:*

$$\boldsymbol{F}(l)=F(hl/2)=\sum_k f_1(k)\varphi_1^0(l-2k)+\sum_k f_2(k)\varphi_2^0(l-2k). \tag{6.7}$$

The coefficients are: $f_1(k)=\langle\boldsymbol{F},\tilde{\varphi}_1^0(\cdot-2k)\rangle,\quad f_2(k)=\langle\boldsymbol{F},\tilde{\varphi}_2^0(\cdot-2k)\rangle$. *The following biorthogonal relations hold:* $\langle\varphi_1^0(\cdot-2l),\tilde{\varphi}_1^0(\cdot-2k)\rangle=\delta_{k,l}$, $\langle\varphi_2^0(\cdot-2l),\tilde{\varphi}_2^0(\cdot-2k)\rangle=\delta_{k,l}$, $\langle\varphi_1^0(\cdot-2l),\tilde{\varphi}_2^0(\cdot-2k)\rangle=0$, $\langle\varphi_2^0(\cdot-2l),\tilde{\varphi}_1^0(\cdot-2k)\rangle=0$.

Proposition 6.2. *Assume* $\boldsymbol{F}$ *is a quadratic polynomial sampled on the grid* $kh/2$ *and* $\tilde{\varphi}_1^0$ *and* $\tilde{\varphi}_2^0$ *are the preprocessing wavelets for the Haar algorithm. Then the inner products* $f_1(k)=\langle\boldsymbol{F},\tilde{\varphi}_1^0(\cdot-2k)\rangle,\ k\in\mathbb{Z}$ *are the values of a quadratic polynomial sampled on the grid* kh, *and the products* $f_2(k)=\langle\boldsymbol{F},\tilde{\varphi}_2^0(\cdot-2k)\rangle$ *are the values of its derivative times* h. *In the case when the wavelets* $\tilde{\varphi}_1^0$ *and* $\tilde{\varphi}_2^0$ *are derived from the schemes of fifth order and* $\boldsymbol{F}$ *is a sampled polynomial of fourth degree, the inner products are samples of polynomials of fourth degree and of their derivatives times* h, *respectively.*

Summarizing, the sets of signals $\{\varphi_1^0(\cdot-2l),\ \varphi_2^0(\cdot-2l)\}$, $\{\tilde{\varphi}_1^0(\cdot-2k),\ \tilde{\varphi}_2^0(\cdot-2k)\}$ form a biorthogonal pair of bases for the signal space. Once we have the expansion (6.7) we can proceed with the vector wavelet transforms described in Section 3. The basis signals $\vec{\varphi}^0=(\varphi_1^0,\varphi_2^0)^T$ and $\tilde{\vec{\varphi}}^0=(\tilde{\varphi}_1^0,\tilde{\varphi}_2^0)^T$ are, to some extent, discrete counterparts of the synthesis and analysis scaling functions used in the theoretical multiwavelet analysis.

6.2 Bases of the first level

We show that the transformations of the vector signal $\vec{f}$ presented in Section 3 lead to re-expansion of the original signal $\boldsymbol{F}$ with respect to new biorthogonal pairs of bases.

Given one of three sets of the postprocessing and preprocessing multiwavelets of zero level $\vec{\varphi}^0=(\varphi_1^0,\varphi_2^0)^T$ and $\tilde{\vec{\varphi}}^0=(\tilde{\varphi}_1^0,\tilde{\varphi}_2^0)^T$, we define the basis multiwavelets of the first level by the two-scale equations in the $z-$domain. The primal synthesis multiwavelets $\vec{\varphi}^1(z) = (\varphi_1^1(z), \varphi_2^1(z))^T$ are defined as:

$$\vec{\varphi}^1(z) \triangleq \boldsymbol{H}(z^2)^T \cdot \vec{\varphi}^0(z) \Longleftrightarrow \begin{array}{l} \varphi_1^1(z) \triangleq H_{11}(z^2)\varphi_1^0(z) + H_{21}(z^2)\varphi_2^0(z) \\ \varphi_1^2(z) \triangleq H_{12}(z^2)\varphi_1^0(z) + H_{22}(z^2)\varphi_2^0(z). \end{array}$$

$$\vec{\psi}_b^1(z) = \boldsymbol{G_b}(z^2)^T \cdot \vec{\varphi}^0(z) \Longleftrightarrow \begin{array}{l} \psi_{1,b}^1(z) \triangleq G_{11,b}(z^2)\varphi_1^0(z) + G_{21,b}(z^2)\varphi_2^0(z), \\ \psi_{2,b}^1(z) \triangleq G_{12,b}(z^2)\varphi_1^0(z) + G_{22,b}(z^2)\varphi_2^0(z). \end{array}$$

Then the dual synthesis multiwavelets are defined similarly.

The primal analysis multiwavelets are defined as:

$$\tilde{\vec{\varphi}}_b^1(z) = \tilde{\boldsymbol{H}}_{\boldsymbol{b}}(z^{-2})^T \cdot \tilde{\vec{\varphi}}^0(z) \Longleftrightarrow \begin{array}{l} \tilde{\varphi}_{1,b}^1(z) \triangleq \tilde{H}_{11,b}(z^{-2})\tilde{\varphi}_1^0(z) + \tilde{H}_{21,b}(z^{-2})\tilde{\varphi}_2^0(z), \\ \tilde{\varphi}_{2,b}^1(z) \triangleq \tilde{H}_{12,b}(z^{-2})\tilde{\varphi}_1^0(z) + \tilde{H}_{22,b}(z^{-2})\tilde{\varphi}_2^0(z). \end{array}$$

$$\tilde{\vec{\psi}}^1(z) = \tilde{\boldsymbol{G}}(z^{-2})^T \cdot \tilde{\vec{\varphi}}^0(z) \Longleftrightarrow \begin{array}{l} \tilde{\psi}_1^1(z) \triangleq \tilde{G}_{11}(z^{-2})\tilde{\varphi}_1^0(z) + \tilde{G}_{21}(z^{-2})\tilde{\varphi}_2^0(z), \\ \tilde{\psi}_2^1(z) \triangleq \tilde{G}_{12}(z^{-2})\tilde{\varphi}_1^0(z) + \tilde{G}_{22}(z^{-2})\tilde{\varphi}_2^0(z). \end{array}$$

Then the dual multiwavelets are defined similarly. Actually we have six different pairs of bases. They correspond to three pre/postprocessing schemes and to two modes of multiwavelet transform.

Theorem 6.1. *As the result of the first step of the primal or dual vector wavelet transform the signal $\boldsymbol{F} = \{F(lh/2\}_{l\in\mathbb{Z}}$ is expanded as follows:*

$$\begin{aligned} \boldsymbol{F}(l) = F(hl/2) = \sum_k & s_1^u(k)\varphi_1^1(k-4l) + s_2^u(k)\varphi_2^1(k-4l) \\ & + d_1^u(k)\psi_1^1(k-4l) + d_2^u(k)\psi_2^1(k-4l) \end{aligned}$$

and the coefficients are: $s_1^u(k) = \langle \boldsymbol{F}, \tilde{\varphi}_1^1(\cdot - 4k)\rangle$, $s_2^u(k) = \langle \mathbf{F}, \tilde{\varphi}_2^1(\cdot - 4k)\rangle$, $d_1^u(k) = \langle \mathbf{F}, \tilde{\psi}_1^1(\cdot - 4k)\rangle$, $d_2^u(k) = \langle \mathbf{F}, \tilde{\psi}_2^1(\cdot - 4k)\rangle$. The corresponding biorthogonal relations between the synthesis $(\varphi_1^1, \varphi_2^1)^T$, $(\psi_1^1, \psi_2^1)^T$ and the analysis $(\tilde{\psi}_1^1, \tilde{\psi}_2^1)^T$, $(\tilde{\varphi}_1^1, \tilde{\varphi}_2^1)^T$ multiwavelets hold. Once the preprocessing algorithms of fifth order are used, the analysis wavelets $\tilde{\psi}_1^1$ have 4 vanishing moments and the wavelets $\tilde{\psi}_2^1$ have 5 vanishing moments.

Once we applied the multiwavelet transform of the first level on the vector signal $\vec{f}$, we are in a position to decompose the signal into coarser levels. To arrive at the second level we should apply our lifting procedure described above on the vector array $\vec{s}^u$ rather than on $\vec{f}$. Note that at the second step of the decomposition we are free to apply the primal or the dual multiwavelet transform, regardless whether the primal or the dual transform was applied at the first step. So, on this stage we have four different ways to represent the signal $\boldsymbol{F}$. Similarly the basis multiwavelets of the second and subsequent scales are defined. As in the first level, all analysis multiwavelets have vanishing moments property.

Note that our approach allows construction of multiwavelet transforms with higher approximation accuracy using Hermite splines of higher degrees.

More detailed presentation of the stuff and illustrations can be found in the preprint [2].

References

1. A. Averbuch, A. Pevnyi, V. Zheludev, *Butterworth wavelets derived from discrete interpolatory splines*, Preprint,www.math.tau.ac.il/∼amir (∼zhel).
2. A. Averbuch V. Zheludev, *Biorthogonal multiwavelet transforms originated from Hermite splines*, Preprint, www.math.tau.ac.il/∼amir (∼zhel).
3. G. Davis, V. Strela and R. Turcajova, *Multiwavelet construction via the lifting scheme*, in Wavelet Analysis and Multiresolution Methods, T.-X. He (ed.), Lecture Notes in Pure and Applied Mathematics, Marcel Dekker, (1999).
4. G. Deslauriers and S. Dubuc, *Symmetric iterative interpolation processes*, Constructive Approximation, **5**, (1989), 49–68.
5. D. L. Donoho, *Interpolating wavelet transform*, Preprint 408, Department of Statistics, Stanford University, (1992).
6. J. S. Geronimo, D. P. Hardin and P. R. Massopust, *Fractal functions and wavelet expansions based on several scaling functions*, J. Approx. Theory, **78**, (1994), 373–401.
7. S. S. Goh, Q. Jiang, T. Xia, *Construction of biorthogonal multiwavelets using the lifting scheme*, Preprint.
8. T. N. T. Goodman and S. L. Lee, *Wavelets of multiplicity r*, Trans. Amer. Math. Soc., **342**, (1994), 307–324.
9. B. Han, Q. Jiang, *Multiwavelets on the interval*, Preprint, (1999).
10. L. Hervé, *Multi-resolution analysis of multiplicity d: Application to dyadic interpolation*, Appl. Comput. Harm. Anal. **1**, (1994), 299–315.
11. G. Strang and V. Strela, *Finite element multiwavelets*, in Approximation Theory, Wavelets and Applications, S. P. Singh (ed.), Kluwer Academic Publ. (1995) 485–496.
12. V. Strela and A. T. Walden, *Orthogonal and biorthogonal multiwavelets for signal denoising and image compression*, Wavelet Applications V, Proceedings of SPIE, Vol. 3391, (1998).
13. W. Sweldends, *The lifting scheme: A custom design construction of biorthogonal wavelets*, Appl. Comput. Harm. Anal. **3(2)**, (1996), 186–200.
14. V. Zheludev, A. Averbuch, *A biorthogonal wavelet scheme based on interpolatory splines*, Proceedings Second Int. Conf. "Tools for Math. Modeling 99", V. 4, St. Petersburg: SPTU, (1999), 214–231.

Amir Z. Averbuch, Valery A. Zheludev
Department of Computer Science
Tel Aviv University
Tel Aviv 69978, ISRAEL
amir@math.tau.ac.il, zhel@math.tau.ac.il
http://www.math.tau.ac.il/∼amir (∼zhel)