

Lifting Scheme for Biorthogonal Multiwavelets Originated from Hermite Splines

Amir Z. Averbuch and Valery A. Zheludev

Abstract—We present new multiwavelet transforms of multiplicity 2 for manipulation of discrete-time signals. The transforms are implemented in two phases: 1) Pre (post)-processing, which transforms the scalar signal into a vector signal (and back) and 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 that do not degrade the approximation accuracy of the vector wavelet transforms. We describe two types of vector wavelet transforms that are dual to each other but have similar properties and three pre(post)-processing algorithms. As a result, we get fast biorthogonal algorithms to transform discrete-time signals that are exact on sampled cubic polynomials. The bases for the transform are symmetric and have short support.

Index Terms—Hermite spline, lifting scheme, multifilter, multiwavelet transform.

I. INTRODUCTION

HERMITE splines are piecewise polynomials of degree $2n - 1$ that have n continuous derivatives. These splines were used for the construction of the first examples of multiwavelets [11], [13], [14]. In these works, the generating functions of Hermite splines were employed as scaling functions for the multiresolution analysis, and the corresponding wavelets were derived. Unlike the popular Geronimo, Hardin, Massopust (GHM) multiwavelets [9], the bases generated by these multiwavelets are not orthogonal. In [4], [12], [16], and [17], the dual scaling functions and multiwavelets were constructed. This construction produced biorthogonal bases for $L_2(\mathbb{R})$ or $L_2([a, b])$, but this theory cannot be directly applied to processing of discrete-time signals. While the original signal is scalar, the input stream to the multiwavelet transform must be a vector array. This array is produced by the so-called preprocessing of the signal. Reconstruction of the scalar signal from the vector array is called postprocessing. Therefore, a multiwavelet transform of a signal actually consists of two phases: 1) Pre(post)-processing, which transforms the scalar signal into a vector one (and back), and 2) wavelet transforms of the vector signal.

For the so-called balanced multiwavelets [10], [15], the pre(post)-processing phase is trivial. Relevant pre(post)-processing algorithms were devised for the GHM multiwavelets

and for some other classes of nonbalanced multiwavelets [21], [23], [24]. However, this is not the case for transforms that are based on Hermite multiwavelets. In order to implement a vector wavelet transform, which is based on cubic Hermite splines, we need to have coefficients that are samples from the signal and its derivatives. Usually, the values of the derivatives are unavailable and have to be evaluated from the signal samples. A pre(post)-processing algorithm for multiwavelet transform, which used cubic Hermite spline as scaling functions, was presented in [17]. This algorithm is similar to the forward and backward Haar wavelet transform. However, this preprocessing algorithm degrades the approximation accuracy of the cubic Hermite splines. If the splines are constructed with the proper coefficients, then the cubic polynomials can be restored. The above preprocessing scheme allows only restoration of linear polynomials. We are unaware in the literature of the existence of any pre(post)-processing algorithms for Hermite multiwavelet transforms of degrees higher than three.

Recently, lifting schemes [6], [19], [20] became very popular for construction of biorthogonal wavelet transforms. They are fast, consume relatively little memory, and have inherent tools for custom design of filter banks. 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 an already updated odd array which smooths this array.

It is possible to predict an odd sample through the value of a polynomial that interpolates adjacent even samples [8], [19]. In [1] and [2], we used for prediction polynomial and discrete interpolatory splines, respectively.

In this paper, we utilize an approach that is close to [1] and [2] for the lifting construction of interpolatory multiwavelet transforms. We use the interpolatory cubic Hermite splines as a predicting aggregate in the vector wavelet transform phase rather than for the construction of scaling functions. Recently, other papers such as [7], [10], and [22] linked between lifting schemes and multiwavelet transforms, but these papers were using neither Hermite splines nor any pre(post)-processing procedures.

Characteristic features of our construction of the biorthogonal multiwavelets are as follows.

Union: We treat a multiwavelet transform as a union of the pre(post)-processing phase and the wavelet transforms of vector signals. In this framework, the pre(post)-processing phase is considered to be the multiwavelet transform of the zero level of a discrete-time scalar signal.

Manuscript received August 3, 2000; revised November 26, 2001. This work was supported in part by the Israel Science Foundation under Grant 1999–2003, 258/99-1. The associate editor coordinating the review of this paper and approving it for publication was Dr. Paulo J. S. G. Ferreira.

The authors are with the School of Computer Sciences, Tel Aviv University, Tel Aviv, Israel (e-mail: amir@math.tau.ac.il; zhel@math.tau.ac.il).

Publisher Item Identifier S 1053-587X(02)01338-7.

Interpolation: We construct an interpolatory lifting scheme for the vector wavelet transform using cubic Hermite splines. Therefore, we stay completely in the discrete-time setting.

Pre(post)-Processing: We present 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 previously.

Diversity: We present two types of vector wavelet transforms, which are dual to each other but have similar properties, and three pre(post)-processing algorithms. Each of these vector wavelet transforms can be combined with each of the pre(post)-processing algorithms.

As a result, we get fast biorthogonal algorithms to transform discrete-time signals. The bases for the transform are symmetric and have short support. The analysis wavelets have four vanishing moments. This approach to both the vector wavelet and the pre(post)-processing transforms is generic and allows an extension in a straightforward manner to handle higher order accuracy. This approach can be applied to the construction of multiwavelet transforms using non-Hermite interpolants.

The paper is organized as follows. In Section II, we recall some facts about cubic Hermite splines and establish their approximation properties that are of use in the sequel. In Section III, we introduce the primal and dual modes of the lifting wavelet transform of discrete-time vector signals using the Hermite spline as a predictor. In Section IV, we analyze the structure of multifilter banks that are generated by the above lifting wavelet transform. We determine conditions for the multifilter banks to have the perfect reconstruction property, establish the approximation accuracy of the multifilters, and use the control tools inherent to the lifting scheme in order to bring the properties of the dual filters close to the properties of the primal ones. Section V is devoted to the construction of lifting algorithms for pre/postprocessing. These algorithms are exact on the polynomials up to fourth order. In Section VI, we study the biorthogonal analysis and synthesis bases for the space of discrete-time scalar signals that are generated by the pre/post-processing procedures and by one step of the multiwavelet transforms. In Section VII, we introduce bases of the multiwavelet transforms to all levels. In Section VIII, we present some discussion on the possible application of the developed multiwavelet transforms.

II. PRELIMINARIES

A cubic Hermite spline, which is defined on the grid $\{t_k\}$, $k \in \mathbb{Z}$, is a piecewise polynomial function S belonging to C^1 . Inside the intervals (t_k, t_{k+1}) , it coincides with cubic polynomials that are determined by the boundary conditions

$$\begin{aligned} S(t_k) &= s_1(k), & S'(t_k) &= s_2(k), & S(t_{k+1}) &= s_1(k+1) \\ S'(t_{k+1}) &= s_2(k+1) \end{aligned}$$

where $s_1(k)$ and $s_2(k)$ are predetermined values. If P is a cubic polynomial, $P(t_k) = s_1(k)$ and $hP'(t_k) = s_2(k)$, $k \in \mathbb{Z}$, then $S(t)$ and $P(t)$ are identical.

Throughout the paper, S will denote a Hermite spline constructed on the equidistant grid $\{2hk\}$, $k \in \mathbb{Z}$, $h \in \mathfrak{R}$, satisfying the conditions $S(2hk) = s_1(k)$, $hS'(2hk) = s_2(k)$.

In the sequel, we will need to have the values of the spline S and its derivative in the midpoints of the intervals. They are given by the following formulas:

$$\begin{aligned} S(h(2k+1)) &= \frac{1}{2}s_1(k) + \frac{1}{4}s_2(k) \\ &\quad + \frac{1}{2}s_1(k+1) - \frac{1}{4}s_2(k+1) \\ hS'(h(2k+1)) &= -\frac{3}{4}s_1(k) - \frac{1}{4}s_2(k) \\ &\quad + \frac{3}{4}s_1(k+1) - \frac{1}{4}s_2(k+1). \end{aligned} \quad (2.1)$$

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

Proposition 2.1: Assuming $F \in C^4$. If $s_1(k) = F(2hk)$ and $s_2(k) = hF'(2hk)$, then

$$\begin{aligned} S(h(2k+1)) &= F(h(2k+1)) - \frac{h^4}{24}F^{(4)}(h(2k+1)) \\ &\quad + o(h^4F^{(4)}) \end{aligned} \quad (2.2)$$

$$hS'(h(2k+1)) = hF'(h(2k+1)) + o(h^4F^{(4)}). \quad (2.3)$$

In particular, if F is a cubic polynomial, then $S(h(2k+1)) = F(h(2k+1))$. If F is a polynomial of fourth degree, then $hS'(h(2k+1)) = hF'(h(2k+1))$.

The proof is straightforward.

Notation: Let $\mathbf{M} = \{M(k)\}_{k \in \mathbb{Z}}$ be a matrix sequence

$$M(k) \triangleq \begin{pmatrix} m_{11}(k) & m_{12}(k) \\ m_{21}(k) & m_{22}(k) \end{pmatrix}$$

and $\vec{g} \triangleq \{\vec{g}(k)\} = (g_1(k), g_2(k))^T$ be a vector signal. The following operation

$$\vec{\gamma} = \mathbf{M} * \vec{g} \iff \vec{\gamma}(r) = \sum_k M(r-k)\vec{g}(k) \quad (2.4)$$

is called the matrix-vector convolution or multifiltering the signal \vec{g} . The set $\mathbf{M} = \{M(k)\}_{k \in \mathbb{Z}}$ is called the multifilter.

In the sequel, the z -transform will be frequently used. We introduce the z -transforms of the vector signal $\vec{g} = \{\vec{g}(k)\}$ and of the multifilter $\mathbf{M} = \{M(k)\}_{k \in \mathbb{Z}}$ as follows:

$$\begin{aligned} \vec{g}(z) &\triangleq \sum_{k \in \mathbb{Z}} z^{-k} \vec{g}(k), & |z| &= 1 \\ \mathbf{M}(z) &\triangleq \sum_{k \in \mathbb{Z}} z^{-k} M(k) = \begin{pmatrix} m_{11}(z) & m_{12}(z) \\ m_{21}(z) & m_{22}(z) \end{pmatrix}. \end{aligned}$$

It is easily verified that

$$\vec{\gamma} = \mathbf{M} * \vec{g} \iff \vec{\gamma}(z) = \mathbf{M}(z)\vec{g}(z). \quad (2.5)$$

III. LIFTING SCHEME FOR THE APPLICATION OF WAVELET TRANSFORM ON VECTOR SIGNALS

Assume that a signal $F \in C^1$ is sampled on the grid $\{kh/2\}$ and $\mathbf{F} = \{F(kh/2)\}$, $k \in \mathbb{Z}$. In Section V, we will present

preprocessing algorithms that transform this scalar array into the sampled vector signal

$$\begin{aligned} \vec{f} &\triangleq (f_1(k), f_2(k))^T \quad \text{where } f_1(k) = \Phi(kh) \\ f_2(k) &\approx h\Phi'(kh), \quad k \in \mathbb{Z} \end{aligned} \quad (3.6)$$

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 [19] and [20]. In its simplest form, it consists of three steps:

- 1) *Split*;
- 2) *Predict*;
- 3) *Lifting or Update*.

We apply the wavelet transform on the vector signal \vec{f} , using these three steps, and we add an additional step, which we call 4) *Rescaling*. We introduce two mutually dual versions of the transform.

A. 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

$$\vec{s}(z^2) = \frac{1}{2} (\vec{f}(z) + \vec{f}(-z)), \quad \vec{d}(z^2) = \frac{z}{2} (\vec{f}(z) + \vec{f}(-z)). \quad (3.7)$$

- 2) We calculate the values in the points $\{(2k+1)h\}$ of the Hermite spline that interpolates \vec{s} on the grid $\{2kh\}$. These values are used for prediction of the signal \vec{d} . Then, we update the \vec{d} array by extracting the predicted values from the existing ones. From (2.1), we have

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

Note that $\mathbf{A}(z)$ is the z -transform of the two-tap multfilter

$$\begin{aligned} \mathbf{A} &= \{A(-1), A(0)\}, \quad A(-1) = \begin{pmatrix} \frac{1}{2} & -\frac{1}{4} \\ \frac{3}{4} & -\frac{1}{4} \end{pmatrix} \\ A(0) &= \begin{pmatrix} \frac{1}{2} & \frac{1}{4} \\ -\frac{3}{4} & -\frac{1}{4} \end{pmatrix}. \end{aligned} \quad (3.8)$$

- 3) The last step updates the even array \vec{s} using the new array \vec{d}^u and some multfilter \mathbf{B}

$$\vec{s}^u = \vec{s} + \mathbf{B} * \vec{d}^u \iff \vec{s}^u(z) = \vec{s}(z) + \mathbf{B}(z) \vec{d}^u(z).$$

In Section IV-B, we explain how to choose \mathbf{B} .

- 4) While the input vector signal $\vec{f} = (f_1(k), f_2(k))^T$ is related [in the spirit of (3.6)] to the grid kh and $f_2(k) \approx h\Phi'(kh)$, the s array is related to the coarse grid $2kh$. Therefore, if we intend to go into coarser scales using the updated s array instead of \vec{f} , we have to double the second

component of the vector $\vec{s}^u = (s_1^u(k), s_2^u(k))^T$. Namely, the rescaled s array and its z -representation are

$$\begin{aligned} \vec{s}^{ur}(z) &\triangleq \begin{pmatrix} s_1^u(k) \\ 2s_2^u(k) \end{pmatrix}, \quad \vec{s}^{ur}(z) = \mathbf{R} \cdot \vec{s}^u(z) \\ \mathbf{R} &\triangleq \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}. \end{aligned} \quad (3.9)$$

Reconstruction: Reconstruction is performed in reverse order. Given two updated vector arrays \vec{s}^{ur} and \vec{d}^u , we restore the \vec{s} array and $\vec{f}(z)$ in the following way.

- 1) Undo rescaling: $\vec{s}^u(z) = \mathbf{R}^{-1} \cdot \vec{s}^{ur}(z)$.
- 2) Undo lifting:

$$\vec{s} = \vec{s}^u - \mathbf{B} * \vec{d}^u \iff \vec{s}(z) = \vec{s}^u(z) - \mathbf{B}(z) \vec{d}^u(z). \quad (3.10)$$

- 3) Undo predict: Restore the values of the signal \vec{d}

$$\vec{d}(z) = \vec{d}^u(z) + \mathbf{A}(z) \vec{s}(z). \quad (3.11)$$

- 4) Unsplit: Restore the signal \vec{f} from its even and odd sub-arrays. In the z -domain, we have

$$\vec{f}(z) = \vec{s}(z^2) + z^{-1} \vec{d}(z^2). \quad (3.12)$$

B. Dual Mode

As before, we split the vector signal into even and odd sub-arrays. For the primal construction, which was described above, the update step followed the prediction. In some applications, it is preferable to have the update step before the prediction step and to control the prediction step. In particular, such a dual scheme is used in adaptive nonlinear wavelet transform [3] by choosing different predictors for different fragments of the signal.

We sum up the even array with its prediction that was derived from the odd array

$$\vec{s}^u(z) = \vec{s}(z) + \frac{\mathbf{A}(z)}{z} \vec{d}(z). \quad (3.13)$$

This update smooths the even array.

We update the odd array using the already updated even array

$$\vec{d}^u(z) = \vec{d}(z) - z\mathbf{B}(z) \vec{s}^u(z). \quad (3.14)$$

Rescaling

$$\vec{s}^{ur}(z) = \mathbf{R} \cdot \vec{s}^u(z). \quad (3.15)$$

Reconstruction is implemented in a reverse order.

$$\begin{aligned} \vec{s}^u(z) &= \mathbf{R}^{-1} \cdot \vec{s}^{ur}(z), \quad \vec{d}(z) = \vec{d}^u(z) + z\mathbf{B}(z) \vec{s}^u(z), \\ \vec{s}(z) &= \vec{s}^u(z) - z^{-1} \mathbf{A}(z) \vec{d}(z), \quad \vec{f}(z) = \vec{s}(z^2) + z^{-1} \vec{d}(z^2). \end{aligned}$$

IV. MULTIFILTER BANKS

A. 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 multfilter banks.

1) *Primal Mode*: We define a family of matrix filters through their z -transforms

$$\begin{aligned}\tilde{\mathbf{G}}(z) &\triangleq z \left(\mathbf{I} - \frac{\mathbf{A}(z^2)}{z} \right) \\ &= z \begin{pmatrix} \frac{1}{2} \left(-\frac{1}{z} + 2 - z \right) & -\frac{1}{4} \left(\frac{1}{z} - z \right) \\ \frac{3}{4} \left(\frac{1}{z} - z \right) & \frac{1}{4} \left(\frac{1}{z} + 4 + z \right) \end{pmatrix} \\ \tilde{\mathbf{H}}_{\mathbf{b}}(z) &\triangleq \mathbf{R}[\mathbf{I} + \mathbf{B}(z^2)\tilde{\mathbf{G}}(z)]\end{aligned}\quad (4.16)$$

$$\begin{aligned}\mathbf{H}(z) &\triangleq \left[\mathbf{I} + \frac{\mathbf{A}(z^2)}{z} \right] \mathbf{R}^{-1} \\ &= \begin{pmatrix} \frac{1}{2} \left(\frac{1}{z} + 2 + z \right) & \frac{1}{8} \left(\frac{1}{z} - z \right) \\ -\frac{3}{4} \left(\frac{1}{z} - z \right) & \frac{1}{8} \left(-\frac{1}{z} + 4 - z \right) \end{pmatrix} \\ \mathbf{G}_{\mathbf{b}}(z) &\triangleq \frac{1}{z} (\mathbf{I} - \mathbf{H}(z)z\mathbf{B}(z^2)).\end{aligned}\quad (4.17)$$

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

$$\begin{aligned}\vec{d}^u(z^2) &= \frac{1}{2} \left[\tilde{\mathbf{G}}(z)\vec{f}(z) + \tilde{\mathbf{G}}(-z)\vec{f}(-z) \right] \iff \vec{d}^u(k) \\ &= \sum_r \tilde{G}(2k-r) \cdot \vec{f}(r)\end{aligned}\quad (4.18)$$

$$\begin{aligned}\vec{s}^{ur}(z^2) &= \frac{1}{2} \left[\tilde{\mathbf{H}}_{\mathbf{b}}(z)\vec{f}(z) + \tilde{\mathbf{H}}_{\mathbf{b}}(-z)\vec{f}(-z) \right] \iff \vec{s}^{u,r}(k) \\ &= \sum_r \tilde{H}_{\mathbf{b}}(2k-r) \cdot \vec{f}(r).\end{aligned}\quad (4.19)$$

Reconstruction of the signal \vec{f} from \vec{s}^{ur} and \vec{d}^u can be implemented as filtering the arrays with the multifilters \mathbf{H} and $\mathbf{G}_{\mathbf{b}}$, respectively, preceded by upsampling

$$\vec{f}(z) = \mathbf{H}(z)\vec{s}^{ur}(z^2) + \mathbf{G}_{\mathbf{b}}(z)\vec{d}^u(z^2).\quad (4.20)$$

The reconstruction multifilters \mathbf{H} and $\mathbf{G}_{\mathbf{b}}$ and the decomposition multifilters $\tilde{\mathbf{G}}$ and $\tilde{\mathbf{H}}_{\mathbf{b}}$ form a perfect reconstruction filter bank. This means that the following relations are true for any \mathbf{B} :

$$\begin{aligned}\mathbf{H}(z)\tilde{\mathbf{H}}_{\mathbf{b}}(z) + \mathbf{G}_{\mathbf{b}}(z)\tilde{\mathbf{G}}(z) &= 2\mathbf{I} \\ \mathbf{H}(z)\tilde{\mathbf{H}}_{\mathbf{b}}(-z) + \mathbf{G}_{\mathbf{b}}(z)\tilde{\mathbf{G}}(-z) &= 0.\end{aligned}\quad (4.21)$$

Proof: Using (3.7), we write

$$\begin{aligned}\vec{d}^u(z^2) &= \frac{1}{2} \left[z \left(\vec{f}(z) - \vec{f}(-z) \right) - \mathbf{A}(z^2) \left(\vec{f}(z) + \vec{f}(-z) \right) \right] \\ &= \frac{1}{2} \left[(z\mathbf{I} - \mathbf{A}(z^2)) \vec{f}(z) + (-z\mathbf{I} - \mathbf{A}(z^2)) \vec{f}(-z) \right] \\ &= \frac{1}{2} \left[\tilde{\mathbf{G}}(z)\vec{f}(z) + \tilde{\mathbf{G}}(-z)\vec{f}(-z) \right].\end{aligned}$$

The even array is treated similarly.

The reconstruction formula (4.20) is derived by substituting (3.10) and (3.11) into (3.12).

The perfect reconstruction relations (4.21) follow immediately from the previous definitions of (4.16) and (4.17). ■

The following assertion is readily verified.

2) *Dual Mode*: Similar to the primal mode, we obtain

$$\begin{aligned}\vec{s}^u(z^2) &= \frac{1}{2} \left[\tilde{\mathbf{H}}^{\mathbf{d}}(z)\vec{f}(z) + \tilde{\mathbf{H}}^{\mathbf{d}}(-z)\vec{f}(-z) \right] \\ \vec{d}^u(z^2) &= \frac{1}{2} \left[\tilde{\mathbf{G}}_{\mathbf{b}}^{\mathbf{d}}(z)\vec{f}(z) + \tilde{\mathbf{G}}_{\mathbf{b}}^{\mathbf{d}}(-z)\vec{f}(-z) \right]\end{aligned}$$

where

$$\begin{aligned}\tilde{\mathbf{H}}^{\mathbf{d}}(z) &\triangleq \mathbf{R} \left[\mathbf{I} + \frac{\mathbf{A}(z^2)}{z} \right] \\ \tilde{\mathbf{G}}_{\mathbf{b}}^{\mathbf{d}}(z) &\triangleq z \left(\mathbf{I} - z\mathbf{B}(z^2)\mathbf{R}^{-1}\tilde{\mathbf{H}}^{\mathbf{d}}(z) \right).\end{aligned}\quad (4.22)$$

The reconstruction formula is

$$\vec{f}(z) = \mathbf{H}_{\mathbf{b}}^{\mathbf{d}}(z)\vec{s}^u(z^2) + \mathbf{G}^{\mathbf{d}}(z)\vec{d}^u(z^2)\quad (4.23)$$

where

$$\begin{aligned}\mathbf{G}^{\mathbf{d}}(z) &\triangleq \frac{1}{z} \left(\mathbf{I} - \frac{\mathbf{A}(z^2)}{z} \right) \\ \mathbf{H}_{\mathbf{b}}^{\mathbf{d}}(z) &\triangleq \left[\mathbf{I} + z^2\mathbf{G}^{\mathbf{d}}(z)\mathbf{B}(z^2) \right] \mathbf{R}^{-1}.\end{aligned}\quad (4.24)$$

Proposition 4.1: If the commutation property $\mathbf{B}(z)\mathbf{A}(z) = \mathbf{A}(z)\mathbf{B}(z)$ holds, then the multifilters of the dual mode are related to the reconstruction multifilters of the primal mode in the following way:

$$\begin{aligned}\tilde{\mathbf{H}}_{\mathbf{b}}(z) &= \mathbf{R}\mathbf{H}_{\mathbf{b}}^{\mathbf{d}}(z)\mathbf{R}, \quad \mathbf{H}(z) = \mathbf{R}^{-1}\tilde{\mathbf{H}}^{\mathbf{d}}(z)\mathbf{R}^{-1} \\ \mathbf{G}_{\mathbf{b}}(z) &= \frac{1}{z^2} \tilde{\mathbf{G}}_{\mathbf{b}}^{\mathbf{d}}(z), \quad \tilde{\mathbf{G}}(z) = z^2 \mathbf{G}^{\mathbf{d}}(z).\end{aligned}\quad (4.25)$$

B. Approximation Properties of the Multifilters

Definition 4.1: Let Φ be a differentiable function and

$$\begin{aligned}\vec{f}(k) &= \begin{pmatrix} \Phi(hk) \\ h\Phi'(hk) \end{pmatrix}, \quad \vec{f}^r(k) = \begin{pmatrix} f_1(k) \\ 2f_2(k) \end{pmatrix} \\ \vec{f} &= \{ \vec{f}(k) \}, \quad \vec{f}^r = \{ \vec{f}^r(k) \}\end{aligned}\quad (4.26)$$

$$\begin{aligned}\vec{s}(k) &= \vec{f}(2k), \quad \vec{d}(k) = \vec{f}(2k+1), \quad \vec{s} = \{ \vec{s}(k) \} \\ \vec{d} &= \{ \vec{d}(k) \}, \quad \vec{d}_- = \{ \vec{d}(k-1) \}.\end{aligned}\quad (4.27)$$

We say that a multifilter \mathbf{C} possesses the approximation property of order $r+1$ if $\mathbf{C} * \vec{s} = \vec{d}$, $\mathbf{C} * \vec{d}_- = \vec{s}$ for any $\Phi = P_r$, where P_r is a polynomial of degree r . We say that a multifilter \mathbf{D} reproduces polynomials of degree r if there exists a number α such that for any $\Phi = P_r$, $\mathbf{D} * \vec{f} = \alpha \vec{f}^r$. A multifilter \mathbf{D}_0 eliminates such polynomials if for any $\Phi = P_r$, $\mathbf{D}_0 * \vec{f} = \vec{0}$. We say that a two-tap multifilter $\mathbf{Q} = \{Q(-1), Q(0)\}$ is symmetric if $|Q(-1)_{ij}| = |Q(0)_{ij}|$, $i, j = 1, 2$.

Proposition 4.2:

- 1) The two-tap multifilter $\mathbf{A} = \{A(-1), A(0)\}$ defined in (3.8) is symmetric and possesses the fourth-order approximation property.
- 2) There is no symmetric two-tap multifilter other than \mathbf{A} that possesses the fourth-order approximation property.
- 3) There is no symmetric two-tap multifilter that possesses the fifth-order approximation property.

Proof:

- 1) The approximation property of the symmetric two-tap multifilter \mathbf{A} follows from Proposition 2.1. In the z -domain, it is equivalent to the relation

$$\frac{\mathbf{A}(z^2)}{z} \vec{f}(z) = \mathbf{I} \vec{f}(z) \quad (4.28)$$

which must hold when Φ is a cubic polynomial and $\vec{f}(z)$ is constructed as in (4.26).

- 2) Let $\mathbf{C} = \{C(-1), C(0)\}$, where

$$C(0) = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad C(-1) = \begin{pmatrix} \check{a} & \check{b} \\ \check{c} & \check{d} \end{pmatrix} \quad (4.29)$$

be a symmetric two-tap multifilter with the fourth-order approximation property. Here, \check{q} denotes a number whose absolute value is equal to the absolute value of the number q . Assume that $\Phi(x) = x^3$, $h = 1$ and that the vectors \vec{s} , \vec{d} are defined as in (4.27). Then, when $t = 2k$, we have

$$\begin{aligned} C(0) \vec{d}(k) + C(-1) \vec{d}(k+1) &= \vec{s}(k) \iff \\ &\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} (t-1)^3 \\ 3(t-1)^2 \end{pmatrix} \\ &+ \begin{pmatrix} \check{a} & \check{b} \\ \check{c} & \check{d} \end{pmatrix} \begin{pmatrix} (t+1)^3 \\ 3(t+1)^2 \end{pmatrix} = \begin{pmatrix} t^3 \\ 3t^2 \end{pmatrix} \iff \\ &(a + \check{a})t^3 + 3(-a + \check{a} + b + \check{b})t^2 \\ &+ 3(a + \check{a} - 2b + 2\check{b})t - a + \check{a} + 3b + 3\check{b} = t^3 \\ &(c + \check{c})t^3 + 3(-c + \check{c} + d + \check{d})t^2 \\ &+ 3(c + \check{c} - 2d + 2\check{d})t - c + \check{c} + 3d + 3\check{d} = 3t^2. \end{aligned}$$

To satisfy these identities, we have to choose $a = 1/2 = \check{a}$, $b = 1/4 = -\check{b}$, $c = -3/4 = -\check{c}$, $d = -1/4 = \check{d}$. Hence, it follows that $\mathbf{C} = \mathbf{A}$.

- 3) Assume $\Phi(x) = x^4$. Then, we have $A(0) \vec{s}(0) + A(-1) \vec{s}(1) - \vec{d} = (h^4/2 \mathbf{0})^T$. ■

Corollary 4.1: The dual decomposition multifilter $\tilde{\mathbf{H}}^d$ (primal reconstruction multifilter \mathbf{H}) reproduces cubic polynomials. The primal decomposition multifilter $\tilde{\mathbf{G}}$ (dual reconstruction multifilter \mathbf{G}^d) eliminates cubic polynomials. Namely, if $\vec{f} = (P(kh), hP'(kh))^T$, then

$$\tilde{\mathbf{H}}^d * \vec{f} = 2\vec{f}, \quad \tilde{\mathbf{G}} * \vec{f} = \vec{0}. \quad (4.30)$$

Proof: By using definitions (4.22) and (4.16), relations (4.30) follow immediately from (4.28). ■

Let us turn now to the dual decomposition multifilter $\tilde{\mathbf{G}}^d$ (primal reconstruction multifilter \mathbf{G}_b) and the primal decomposition multifilter $\tilde{\mathbf{H}}_b$ (dual reconstruction multifilter \mathbf{H}_b^d) that depend on yet undefined multifilter \mathbf{B} . Thus far, we required the entries of $\mathbf{B}(z)$ to be Laurent polynomials and $\mathbf{B}(z)$ to commute with $\mathbf{A}(z)$. We have a remarkable freedom in choosing \mathbf{B} . We use this opportunity to supply these \mathbf{B} -dependent multifilters with properties similar to (4.30). Therefore, the properties of the reconstruction multifilters are getting close to properties of the decomposition multifilters.

To get insight on how to choose \mathbf{B} , we observe (3.13) and (3.14). From (3.13), we see that for any cubic polynomial, $\vec{s}^u =$

$2\vec{s}$. Therefore, as it is clear from (3.14), to supply the dual decomposition multifilter $\tilde{\mathbf{G}}_b^d$ with the property of elimination of cubic polynomials and to stay with two-tap symmetric multifilters, we have to choose \mathbf{B} such that $z\mathbf{B}(z) = \mathbf{A}(z)/2$. This means that

$$\mathbf{B} = \{B(0), B(1)\}, \quad B(0) = A(-1)/2 = B(1) = A(0)/2. \quad (4.31)$$

Proposition 4.3: Once the multifilter \mathbf{B} is chosen as in (4.31), the primal decomposition multifilter $\tilde{\mathbf{H}}_b$ (dual reconstruction multifilter \mathbf{H}_b^d) reproduces cubic polynomials. The dual decomposition multifilter $\tilde{\mathbf{G}}_b^d$ (primal reconstruction multifilter \mathbf{G}_b) eliminates cubic polynomials. Namely, if $\vec{f} = (P(kh), hP'(kh))^T$, then $\tilde{\mathbf{H}}_b * \vec{f} = \vec{f}$, $\tilde{\mathbf{G}}_b^d * \vec{f} = \vec{0}$.

Proof: If $\vec{f}(z)$ is constructed as in (4.27), we have

$$\begin{aligned} \tilde{\mathbf{G}}_b^d(z) \vec{f}(z) &= z \left(\vec{f}(z) - \frac{\mathbf{A}(z^2)}{2z} \mathbf{R}^{-1} \mathbf{H}^d(z) \vec{f}(z) \right) \\ &= z \left(\vec{f}(z) - \frac{\mathbf{A}(z^2)}{z} \vec{f}(z) \right) = \vec{0} \\ \tilde{\mathbf{H}}_b(z) \vec{f}(z) &= \mathbf{R} \left(\vec{f}(z) + \mathbf{B}(z^2) \tilde{\mathbf{G}}(z) \vec{f}(z) \right) = \vec{f}(z). \end{aligned}$$

Remark: We choose \mathbf{B} as the shortest multifilter that supplies dual transform with properties similar to the primal one. However, many more various options for choosing \mathbf{B} are possible. A detailed discussion of these options will be described in our next paper.

Thus, we constructed two schemes of the wavelet transform of vector-signals, 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 in the sense of Definition 4.1. The input to the transforms must be a sampled vector $\vec{f} = (f_1(k), f_2(k))^T$ such that $f_1(k) = \Phi(kh)$ are samples of a function Φ on the grid $\{kh\}$ and $f_2(k) \approx h\Phi'(kh)$. To retain the fourth-order approximation, the difference

$$f_2(k) - h\Phi'(kh) = O(h^4\Phi^{(4)}) \quad \forall \Phi \in C^4. \quad (4.32)$$

To perform the multiwavelet transforms of scalar signals, we have to design pre- and post-processing algorithms. The goal of the preprocessing is to create from the scalar sampled signal $\mathbf{F} = \{F(kh/2)\}$ the vector $\vec{f} = (f_1(k), f_2(k))^T$ that satisfies the property (4.32). The post-processing procedure must restore the scalar signal from the vector that is an output of the inverse wavelet transform.

V. LIFTING ALGORITHMS FOR PRE/POST-PROCESSING PHASES

When we have pre/post-processing schemes that are finite, symmetric, and possess the approximation property (4.32), they fit the vector wavelet transforms constructed in the previous sections. In this section, we will derive these 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 post-processing filters to be finite and symmetric, we cannot 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 (4.32) holds. If $\{F(lh/2)\}$ are samples of a cubic polynomial, we will require that $f_1(k) = \Phi(kh)$ and $f_2(k) = h\Phi'(kh)$.

Similar to the multiwavelet transforms, we construct and implement the preprocessing by lifting steps. It is common in lifting algorithms that the calculations are performed “in place,” and post-processing is performed in a reverse order.

Denote $s(k) = F(kh)$ and $d(k) = F((k+1/2)h)$. We start the construction of the preprocessing algorithms with a simple scheme that illustrates our approach and serves as a basis for higher order algorithms. Similar algorithm is presented in [17].

A. Orthogonal Scheme of Third Approximation Order (Haar Algorithm)

Let us update the s and d arrays as follows:

$$\begin{aligned} d^u(k) &= d(k) - s(k) = F((k+1/2)h) - F(kh) \\ s^u(k) &= s(k) + d^u(k)/2 \\ &= (F((k+1/2)h) + F(kh))/2 \end{aligned} \quad (5.33)$$

and put

$$f_1(k) = s^u(k), \quad f_2(k) = 2d^u(k). \quad (5.34)$$

The post-processing is implemented in reverse order

$$\begin{aligned} d^u(k) &= f_2(k)/2, \quad s^u(k) = f_1(k) \\ s(k) &= s^u(k) - d^u(k)/2 \\ d(k) &= d^u(k) + s(k), \quad F((k+1/2)h) = d(k) \\ F(kh) &= s(k). \end{aligned} \quad (5.35)$$

Denote that $\Phi(x) \triangleq (1/2)(F(x+h/2) + F(x))$. The following assertion is readily verified by using Taylor expansion.

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

Table I defines a finite-length vector signal $\vec{\varphi}^0 = (\varphi_1^0, \varphi_2^0)^T$ and $\tilde{\varphi}^0 = (\tilde{\varphi}_1^0, \tilde{\varphi}_2^0)^T$, which we call the post-processing and the pre-processing multiwavelets of zero level, respectively. 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/post-processing scheme is, actually, the orthogonal Haar wavelet transform of a signal.

B. Schemes of Fifth Order

Now, we update the arrays s^u and d^u produced in (5.33) in order to increase their approximation order while retaining the symmetry and finite support of the corresponding multiwavelets.

Scheme I: Let us update the s^u -array as follows: $s^{uu}(k) = s^u(k) - (d^u(k+1) - d^u(k-1))/48$, whereas $f_1(k) \triangleq s^{uu}(k)/2$ and $f_2(k) \triangleq d^u(k)$.

TABLE I
POST-PROCESSING (LEFT) AND PREPROCESSING (RIGHT) MULTIWAVELETS OF ZERO LEVEL FOR THE HAAR ALGORITHM

k	0	1	k	0	1
$\varphi_1^0(k)$	1	1	$\tilde{\varphi}_1^0(k)$	1/2	1/2
$\varphi_2^0(k)$	-1/4	1/4	$\tilde{\varphi}_2^0(k)$	-2	2

Post-processing is implemented in reverse order

$$\begin{aligned} s^{uu}(k) &= 2f_1(k), \quad d^u(k) = f_2(k) \\ s^u(k) &= s^{uu}(k) + \frac{1}{48}(d^u(k+1) - d^u(k-1)) \\ s(k) &= s^u(k) - \frac{1}{2}d^u(k) \\ d(k) &= d^u(k) + s(k), \quad F((k+\frac{1}{2})h) = d(k) \\ F(kh) &= s(k). \end{aligned} \quad (5.36)$$

It is readily seen that

$$\begin{aligned} f_1(k) &= \frac{1}{96}(-F((k+\frac{3}{2})h) + F((k+1)h) \\ &\quad + 24F((k+\frac{1}{2})h) + 24F(kh) \\ &\quad + F((k-\frac{1}{2})h) - F((k-1)h)) \\ f_2(k) &= F((k+1/2)h) - F(kh). \end{aligned} \quad (5.37)$$

Proposition 5.2: Assume that $F \in C^5$, and

$$\Phi(x) \triangleq \frac{1}{96}(-F(x+\frac{3}{2}h) + F(x+h) + 24F(x+\frac{1}{2}h) + 24F(x) - (F(x-h) + F(x-\frac{1}{2}h))).$$

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

Proof: Let $F(x) = x^4$. Then

$$\Phi(x) = \frac{1}{2}x^4 + 1/2hx^3 + \frac{1}{4}h^2x^2 + \frac{1}{16}h^3x - \frac{7}{192}h^4$$

and

$$\Phi'(x) = 2x^3 + \frac{3}{2}hx^2 + \frac{1}{2}h^2x + \frac{1}{16}h^3.$$

In turn

$$(x+1/2h)^4 - x^4 = 2hx^3 + \frac{3}{2}h^2x^2 + \frac{1}{2}h^3x + \frac{1}{16}h^4 = h\Phi'(x).$$

We have similar equations for $F(x) = x^r$, $r = 0, 1, 2, 3$. Hence, the assertion of the proposition follows. ■

Let us define the post-processing and preprocessing multiwavelets $\vec{\varphi}^0 = (\varphi_1^0, \varphi_2^0)^T$ and $\tilde{\varphi}^0 = (\tilde{\varphi}_1^0, \tilde{\varphi}_2^0)^T$, respectively, in Table II.

Scheme II: Another way to increase the approximation order of the Haar scheme is to update the d^u —rather than the s^u —array. We do it in the following way. $d^{uu}(k) = d^u(k) + (s^u(k+1) - s^u(k-1))/32$ while taking $f_1(k) = 9/16s^u(k)$ and $f_2(k) = d^{uu}(k)$.

Post-processing is implemented in reverse order

$$\begin{aligned} d^{uu}(k) &= f_2(k), \quad s^u(k) = \frac{16}{9}f_1(k) \\ d^u(k) &= d^{uu}(k) - \frac{1}{32}(s^u(k+1) - s^u(k-1)) \\ s(k) &= s^u(k) - \frac{1}{2}d^u(k) \\ d(k) &= d^u(k) + s(k), \quad F((k+\frac{1}{2})h) = d(k) \\ F(kh) &= s(k). \end{aligned} \quad (5.38)$$

TABLE II
POST-PROCESSING (TOP) AND PREPROCESSING (BOTTOM) MULTIWAVELETS
OF ZERO LEVEL FOR SCHEME I ALGORITHM OF THE FIFTH ORDER

k	-2	-1	0	1	2	3
$\varphi_1^0(k)$	0	0	2	2	0	0
$\varphi_2^0(k)$	1/48	1/48	-1/2	1/2	-1/48	-1/48
$\tilde{\varphi}_1^0(k)$	-1/96	1/96	1/4	1/4	1/96	-1/96
$\tilde{\varphi}_2^0(k)$	0	0	-1	1	0	0

It is readily seen that

$$\begin{aligned} f_1(k) &= \frac{9}{32} (F((k + \frac{1}{2})h) + F(kh)) \\ f_2(k) &= -\frac{1}{64} F((k-1)h) - \frac{1}{64} F((k - \frac{1}{2})h) - F(kh) \\ &\quad + F((k + \frac{1}{2})h) + \frac{1}{64} F((k+1)h) \\ &\quad + \frac{1}{64} F((k + \frac{3}{2})h). \end{aligned} \quad (5.39)$$

Denote that $\Phi(x) \triangleq 9/32(F(x+1/2h) + F(x))$.

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

Proof: This proof is similar to the proof of Proposition 5.2. The post-processing and pre-processing multiwavelets $\tilde{\varphi}^0 = (\varphi_1^0, \varphi_2^0)^T$ and $\tilde{\tilde{\varphi}}^0 = (\tilde{\varphi}_1^0, \tilde{\varphi}_2^0)^T$, respectively, are as shown in Table III.

VI. BASES FOR THE SPACE OF DISCRETE-TIME SIGNALS

A. Bases of Zero Level

In the previous section, we introduced three post-(pre)-processing pairs of multiwavelets of zero level $\tilde{\varphi}^0 = (\varphi_1^0, \varphi_2^0)^T$ and $\tilde{\tilde{\varphi}}^0 = (\tilde{\varphi}_1^0, \tilde{\varphi}_2^0)^T$, respectively. Their translations form three biorthogonal bases for the space of scalar discrete-time signals. Namely, the following assertion holds.

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

$$\mathbf{F}(l) = F(hl/2) = \sum_k f_1(k)\varphi_1^0(l-2k) + \sum_k f_2(k)\varphi_2^0(l-2k). \quad (6.40)$$

The coefficients are

$$f_1(k) = \langle \mathbf{F}, \tilde{\varphi}_1^0(\cdot - 2k) \rangle, \quad f_2(k) = \langle \mathbf{F}, \tilde{\varphi}_2^0(\cdot - 2k) \rangle. \quad (6.41)$$

Moreover, the following biorthogonal relations hold.

$$\begin{aligned} \langle \varphi_1^0(\cdot - 2l), \tilde{\varphi}_1^0(\cdot - 2k) \rangle &= \delta_k^l \\ \langle \varphi_2^0(\cdot - 2l), \tilde{\varphi}_2^0(\cdot - 2k) \rangle &= \delta_k^l \end{aligned} \quad (6.42)$$

$$\begin{aligned} \langle \varphi_1^0(\cdot - 2l), \tilde{\varphi}_2^0(\cdot - 2k) \rangle &= 0 \\ \langle \varphi_2^0(\cdot - 2l), \tilde{\varphi}_1^0(\cdot - 2k) \rangle &= 0. \end{aligned} \quad (6.43)$$

Proof: We present the proof only for the algorithm of third order that was introduced in Section V-A. The other two cases can be similarly treated.

Let $\mathbf{F}(z) = \sum_l z^{-l} \mathbf{F}(l)$ be the z -transform of the signal \mathbf{F} , and $s(z)$, $d(z)$, $f_1(z)$, and $f_2(z)$ are the z -transforms of the arrays $s(k)$, $d(k)$, $f_1(k)$, and $f_2(k)$. From (5.35), we derive

TABLE III
POST-PROCESSING (TOP) AND PREPROCESSING (BOTTOM) MULTIWAVELETS
OF ZERO LEVEL FOR SCHEME II ALGORITHM OF FIFTH ORDER

k	-2	-1	0	1	2	3
$\varphi_1^0(k)$	1/36	-1/36	16/9	16/9	-1/36	1/36
$\varphi_2^0(k)$	0	0	-1/2	1/2	0	0
$\tilde{\varphi}_1^0(k)$	0	0	9/32	9/32	0	0
$\tilde{\varphi}_2^0(k)$	-1/64	-1/64	-1	1	1/64	1/64

$s(z) = f_1(z) - f_2(z)/4$, $d(z) = f_1(z) + f_2(z)/4$. Substituting these relations in the identity $\mathbf{F}(z) = s(z^2) + z^{-1}d(z^2)$, we obtain

$$\mathbf{F}(z) = f_1(z^2)\varphi_1^0(z) + f_2(z^2)\varphi_2^0(z) = \vec{f}(z^2)^T \cdot \tilde{\varphi}^0(z). \quad (6.44)$$

Here, $\varphi_1^0(z) = z^{-1} + 1$ and $\varphi_2^0(z) = (z^{-1} - 1)/4$ are the z -transforms of the post-processing signals φ_1^0 and φ_2^0 defined by Table I, and $\tilde{\varphi}^0(z) = (\varphi_1^0(z), \varphi_2^0(z))^T$. Going from the z -domain in (6.44) to the time domain yields (6.40). Equations (6.41) stem immediately from (5.33). The biorthogonal relations (6.42) are readily verified. ■

Corollary 6.1: Assume that \mathbf{F} is a quadratic polynomial sampled on the grid $kh/2$ and $\tilde{\varphi}_1^0$ and that $\tilde{\varphi}_2^0$ are the pre-processing wavelets for the Haar algorithm. Then, the inner products $f_1(k) = \langle \mathbf{F}, \tilde{\varphi}_1^0(\cdot - 2k) \rangle$, $k \in \mathbb{Z}$ are the values of a quadratic polynomial sampled on the grid kh , and the products $f_2(k) = \langle \mathbf{F}, \tilde{\varphi}_2^0(\cdot - 2k) \rangle$ 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 \mathbf{F} is a sampled polynomial of fourth degree, the inner products are samples of a polynomial of fourth degree and of its derivative 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 (6.40), we can proceed with the vector wavelet transforms described in Section III. The basis signals $\tilde{\varphi}^0 = (\varphi_1^0, \varphi_2^0)^T$ and $\tilde{\tilde{\varphi}}^0 = (\tilde{\varphi}_1^0, \tilde{\varphi}_2^0)^T$ are, to some extent, discrete counterparts of the synthesis and analysis scaling functions used in the conventional multiwavelet analysis.

B. Bases of the First Level

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

1) *Synthesis Bases:* We begin with the primal reconstruction formula (4.20), which we rewrite as

$$\vec{f}(z) = \vec{f}_h(z) + \vec{f}_g(z) \quad \text{where } \vec{f}_h(z) \triangleq \mathbf{H}(z) \cdot \vec{s}^{ur}(z^2)$$

and

$$\vec{f}_g(z) \triangleq \mathbf{G}_b(z) \cdot \vec{d}^u(z^2). \quad (6.45)$$

Correspondingly, the z -transform of the signal \mathbf{F} , which is given by (6.44), is split into the sum

$$\begin{aligned} \mathbf{F}(z) &= \mathbf{F}_h(z) + \mathbf{F}_g(z), \quad \mathbf{F}_h(z) \triangleq \vec{f}_h(z^2)^T \cdot \tilde{\varphi}^0(z) \\ \mathbf{F}_g(z) &\triangleq \vec{f}_g(z^2)^T \cdot \tilde{\varphi}^0(z). \end{aligned} \quad (6.46)$$

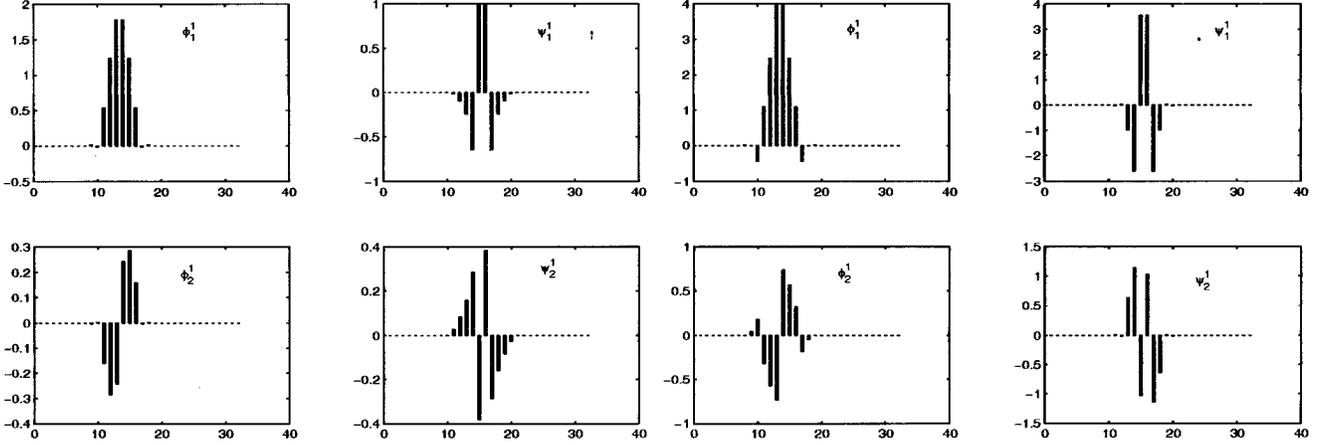


Fig. 1. Primal (left four pictures) and dual (right four pictures) synthesis multiwavelets of first level. The post-processing algorithm of Scheme II was used.

Let us consider $\mathbf{F}_h(z)$. Due to (6.45), we have $\mathbf{F}_h(z) = \vec{s}^{ur}(z^4)^T \cdot \mathbf{H}(z^2)^T \cdot \vec{\varphi}^0(z) = \vec{s}^{ur}(z^4)^T \cdot \vec{\varphi}^1(z)$, where

$$\begin{aligned} \vec{\varphi}^1(z) &\triangleq \mathbf{H}(z^2)^T \cdot \vec{\varphi}^0(z) = (\varphi_1^1(z), \varphi_2^1(z))^T \\ \varphi_1^1(z) &\triangleq H_{11}(z^2)\varphi_1^0(z) + H_{21}(z^2)\varphi_2^0(z) \\ \varphi_2^1(z) &\triangleq H_{12}(z^2)\varphi_1^0(z) + H_{22}(z^2)\varphi_2^0(z). \end{aligned} \quad (6.47)$$

Finally, we have in the z -domain

$$\mathbf{F}_h(z) = s_1^u(z^4)\varphi_1^1(z) + s_2^u(z^4)\varphi_2^1(z). \quad (6.48)$$

Similarly, we can derive the representation in (6.49) and (6.50), shown at the bottom of the page. By switching into the time domain, we observe that the z -vectors $\vec{\varphi}^1(z)$ and $\vec{\psi}_b^1(z)$ are the z -transforms of the vector signals

$$\begin{aligned} \vec{\varphi}^1(l) &= \sum_k H(k)^T \cdot \vec{\varphi}^0(l-2k) \\ \vec{\psi}_b^1(l) &= \sum_k G_b(k)^T \cdot \vec{\varphi}^0(l-2k). \end{aligned} \quad (6.51)$$

We call the vector signal $\vec{\varphi}^1(l) = (\varphi_1^1(l), \varphi_2^1(l))^T$ and $\vec{\psi}_b^1(l) = (\psi_{1,b}^1(l), \psi_{2,b}^1(l))^T$ the primal synthesis multiwavelets of the first level.

Remark: The matrices $H(k)^T$ that participate in the two-scale equation (6.51)

$$\begin{aligned} H(1)^T &= \begin{pmatrix} \frac{1}{2} & -\frac{3}{4} \\ \frac{1}{8} & -\frac{1}{8} \end{pmatrix}, & H(0)^T &= \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \\ H(-1)^T &= \begin{pmatrix} \frac{1}{2} & \frac{3}{4} \\ -\frac{1}{8} & -\frac{1}{8} \end{pmatrix} \end{aligned}$$

coincide with the mask of the refinement equation for the cubic Hermite scaling functions (see, for example, [4]). Then, (6.48) and (6.49) lead us to the following expansion of the signal.

$$\begin{aligned} F(r) &= \sum_l s_1^{ur}(l)\varphi_1^1(l-4r) + s_2^{ur}(l)\varphi_2^1(l-4r) \\ &\quad + d_{1,b}^u(l)\psi_{1,b}^1(l-4r) + d_{2,b}^u(l)\psi_{2,b}^1(l-4r). \end{aligned} \quad (6.52)$$

A similar expansion holds with the usage of the dual synthesis multiwavelets of the first level.

$$\begin{aligned} \vec{\varphi}_b^1(l) &= \sum_k H_b^d(k)^T \cdot \vec{\varphi}^0(l-2k) \\ \vec{\psi}^1(l) &= \sum_k G^d(k)^T \cdot \vec{\varphi}^0(l-2k). \end{aligned} \quad (6.53)$$

Actually, (6.51) and (6.53) define three pairs of primal and three pairs of dual synthesis multiwavelets of the first level according to the one choice from three available post-processing multiwavelets $\vec{\varphi}^0$. We display the synthesis multiwavelets related to the post-processing Scheme II in Fig. 1.

2) *Analysis Bases:* To interpret the coefficients in the expansion formulas (6.52), we turn to the decomposition formula (4.18). Switching into the time domain, we obtain

$$\vec{d}^u(k) = \sum_r \tilde{G}(2k-r) \cdot \vec{f}(r). \quad (6.54)$$

However, (6.41) implies that

$$\vec{f}(r) = \sum_l \mathbf{F}(l)\tilde{\varphi}^0(l-2r). \quad (6.55)$$

Substituting (6.55) into (6.54), we have

$$\begin{aligned} \vec{d}^u(k) &= \sum_r \tilde{G}(2k-r) \cdot \sum_l \mathbf{F}(l)\tilde{\varphi}^0(l-2r) \\ &= \sum_l \mathbf{F}(l) \sum_r \tilde{G}(2k-r) \cdot \tilde{\varphi}^0(l-2r) \\ &= \sum_l \mathbf{F}(l) \sum_s \tilde{G}(s) \cdot \tilde{\varphi}^0(l+2s-4k). \end{aligned} \quad (6.56)$$

Assume that we have the vector signal

$$\vec{\tilde{\varphi}}^1(l) \triangleq \sum_s \tilde{G}(s) \cdot \tilde{\varphi}^0(l+2s) = (\tilde{\psi}_1^1(l), \tilde{\psi}_2^1(l))^T. \quad (6.57)$$

$$\mathbf{F}_g(z) = \vec{d}^u_b(z^4)^T \cdot \vec{\psi}_b^1(z) = d_{1,b}^u(z^4)\psi_{1,b}^1(z) + d_{2,b}^u(z^4)\psi_{2,b}^1(z) \quad (6.49)$$

$$\begin{aligned} \vec{\psi}_b^1(z) = \mathbf{G}_b(z^2)^T \cdot \vec{\varphi}^0(z) &\iff \psi_{1,b}^1(z) \triangleq G_{11,b}(z^2)\varphi_1^0(z) + G_{21,b}(z^2)\varphi_2^0(z) \\ \psi_{2,b}^1(z) &\triangleq G_{12,b}(z^2)\varphi_1^0(z) + G_{22,b}(z^2)\varphi_2^0(z) \end{aligned} \quad (6.50)$$

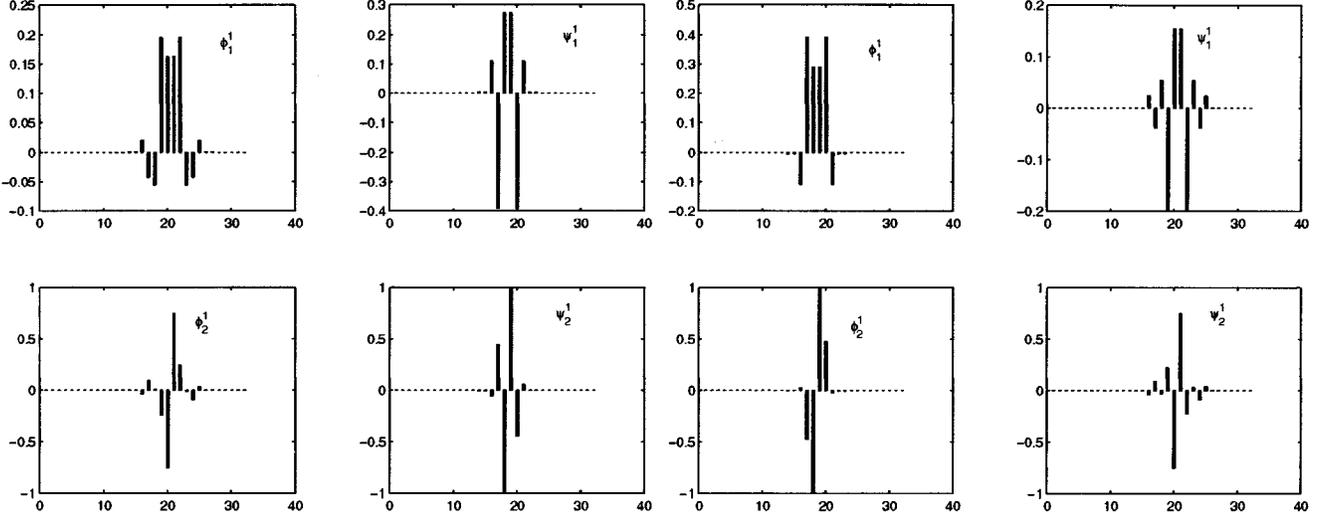


Fig. 2. Primal (left four pictures) and dual (right four pictures) analysis multiwavelets of the first level. The preprocessing algorithm of Scheme II was used.

Then, we derive the following representation for the expansion coefficients.

$$\begin{aligned} \vec{d}^u(k) &= \sum_l \mathbf{F}(l) \tilde{\psi}^1(l - 4k) \\ &\iff \begin{aligned} d_1^u(k) &= \langle \mathbf{F}, \tilde{\psi}_1^1(\cdot - 4k) \rangle \\ d_2^u(k) &= \langle \mathbf{F}, \tilde{\psi}_2^1(\cdot - 4k) \rangle. \end{aligned} \end{aligned} \quad (6.58)$$

Using similar reasoning, we obtain the relations for the s coefficients, provided that the vector signal

$$\tilde{\varphi}_b^1(l) \triangleq \sum_s \tilde{H}_b(s) \cdot \tilde{\varphi}^0(l + 2s) = (\tilde{\varphi}_{1,b}^1(l), \tilde{\varphi}_{2,b}^1(l))^T \quad (6.59)$$

was defined. The s coefficients are

$$\vec{s}_b^{ur}(k) = \sum_l \mathbf{F}(l) \tilde{\varphi}_b^1(l - 4k). \quad (6.60)$$

Equations (6.60) and (6.58) imply that the biorthogonal relations between the signals of the first level

$$\begin{aligned} \langle \varphi_1^1(\cdot - 4l), \tilde{\varphi}_{1,b}^1(\cdot - 4k) \rangle &= \delta_{k,l} \\ \langle \varphi_2^1(\cdot - 4l), \tilde{\varphi}_{2,b}^1(\cdot - 4k) \rangle &= \delta_{k,l} \\ \langle \varphi_1^1(\cdot - 4l), \tilde{\varphi}_{2,b}^1(\cdot - 4k) \rangle &= 0 \\ \langle \varphi_2^1(\cdot - 4l), \tilde{\varphi}_{1,b}^1(\cdot - 4k) \rangle &= 0. \end{aligned} \quad (6.61)$$

$$\begin{aligned} \langle \tilde{\psi}_1^1(\cdot - 4l), \psi_{1,b}^1(\cdot - 4k) \rangle &= \delta_{k,l} \\ \langle \tilde{\psi}_2^1(\cdot - 4l), \psi_{2,b}^1(\cdot - 4k) \rangle &= \delta_{k,l} \\ \langle \tilde{\psi}_1^1(\cdot - 4l), \psi_{2,b}^1(\cdot - 4k) \rangle &= 0 \\ \langle \tilde{\psi}_2^1(\cdot - 4l), \psi_{1,b}^1(\cdot - 4k) \rangle &= 0. \end{aligned} \quad (6.62)$$

Moreover, each signal $\tilde{\psi}^1$ is orthogonal to any signal φ^1 , and each $\tilde{\varphi}^1$ is orthogonal to any ψ^1 signal.

Summarizing, we may argue that the sets of signals

$$\begin{aligned} \Gamma_p^1 &\triangleq \{\varphi_1^1(\cdot - 4l), \varphi_2^1(\cdot - 4l), \psi_{1,b}^1(\cdot - 4k), \psi_{2,b}^1(\cdot - 4k)\} \\ \tilde{\Gamma}_p^1 &\triangleq \{\tilde{\varphi}_{1,b}^1(\cdot - 4k), \tilde{\varphi}_{2,b}^1(\cdot - 4k), \tilde{\psi}_1^1(\cdot - 4l), \tilde{\psi}_2^1(\cdot - 4l)\} \end{aligned}$$

form a biorthogonal pair of bases for the signal space. We call the signals belonging to the set Γ_p^1 the primal synthesis wavelets of the first level. The signals from the set $\tilde{\Gamma}_p^1$ are called the primal analysis wavelets of the first level.

Similarly, we can derive the dual pairs of bases

$$\begin{aligned} \Gamma_d^1 &\triangleq \{\varphi_{1,b}^{1,d}(\cdot - 4l), \varphi_{2,b}^{1,d}(\cdot - 4l) \\ &\quad \psi_1^{1,d}(\cdot - 4k), \psi_2^{1,d}(\cdot - 4k)\} \\ \tilde{\Gamma}_d^1 &\triangleq \{\tilde{\varphi}_1^{1,d}(\cdot - 4k), \tilde{\varphi}_2^{1,d}(\cdot - 4k) \\ &\quad \tilde{\psi}_{1,b}^{1,d}(\cdot - 4l), \tilde{\psi}_{2,b}^{1,d}(\cdot - 4l)\} \end{aligned}$$

and the corresponding expansions of the signal F . To get it, we replace the primal multifilters \mathbf{H} , \mathbf{G}_b , $\tilde{\mathbf{H}}_b$, and $\tilde{\mathbf{G}}$ in the two-scale relations (6.51), (6.57), and (6.59) by the dual multifilters \mathbf{H}_b^d , \mathbf{G}^d , $\tilde{\mathbf{H}}^d$, and $\tilde{\mathbf{G}}_b^d$. These multifilters are defined in Section IV-A.

Actually, we have six different pairs of bases. They correspond to three pre/post-processing schemes and to two modes of the multiwavelet transform (see Fig. 2).

Remark: The synthesis multiwavelets, produced with the usage of post-processing Schemes I and II, are very similar in their appearance. The same can be said about the analysis multiwavelets.

Theorem 6.1: In the case when the Haar preprocessing algorithm is used, the primal and dual analysis wavelets of the first level $\tilde{\psi}_1^1$, $\tilde{\psi}_2^1$ and $\tilde{\psi}_{1,b}^{1,d}$, $\tilde{\psi}_{2,b}^{1,d}$ have three vanishing moments. When the fifth-order preprocessing algorithms are used, the analysis wavelets have four vanishing moments.

Proof: First, we consider the primal wavelets $\tilde{\psi}_1^1$ and $\tilde{\psi}_2^1$. Assume that $\mathbf{F}(k) = k^r$, and $k \in \mathbb{Z}$. This array can be viewed as a polynomial of degree r sampled on the grid $kh/2$, $h = 2$. Equation (6.58) implies that

$$\begin{aligned} \sum_k \tilde{\psi}_1^1(k) k^r &= \langle \mathbf{F}, \tilde{\psi}_1^1 \rangle = d_1^u(0) \\ \sum_k \tilde{\psi}_2^1(k) k^r &= \langle \mathbf{F}, \tilde{\psi}_2^1 \rangle = d_2^u(0). \end{aligned} \quad (6.63)$$

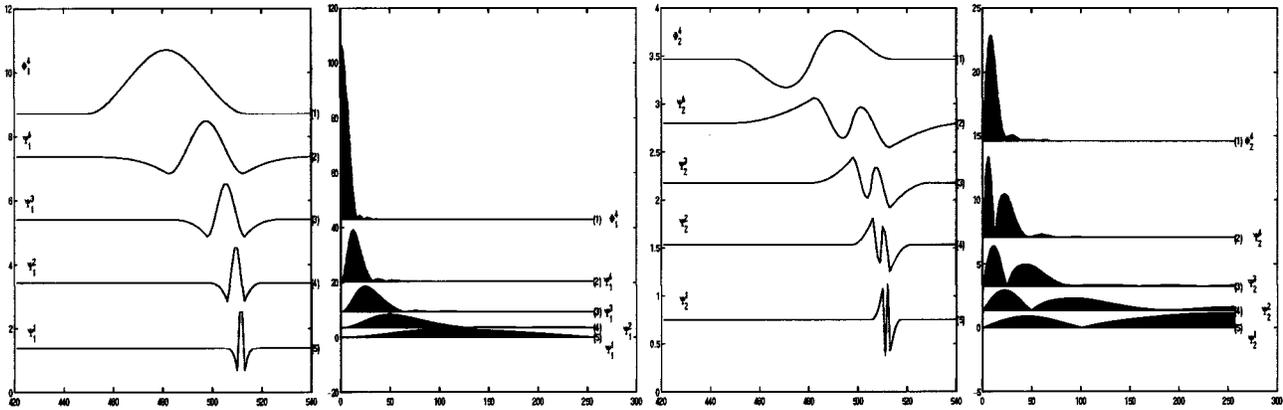


Fig. 3. Left pair of pictures: Primal synthesis wavelets $\varphi_1^i, \psi_{1,b}^i, i = 4, 3, 2, 1$, and their Fourier spectra. Right pair of pictures: Wavelets $\varphi_2^i, \psi_{2,b}^i, i = 4, 3, 2, 1$, and their Fourier spectra.

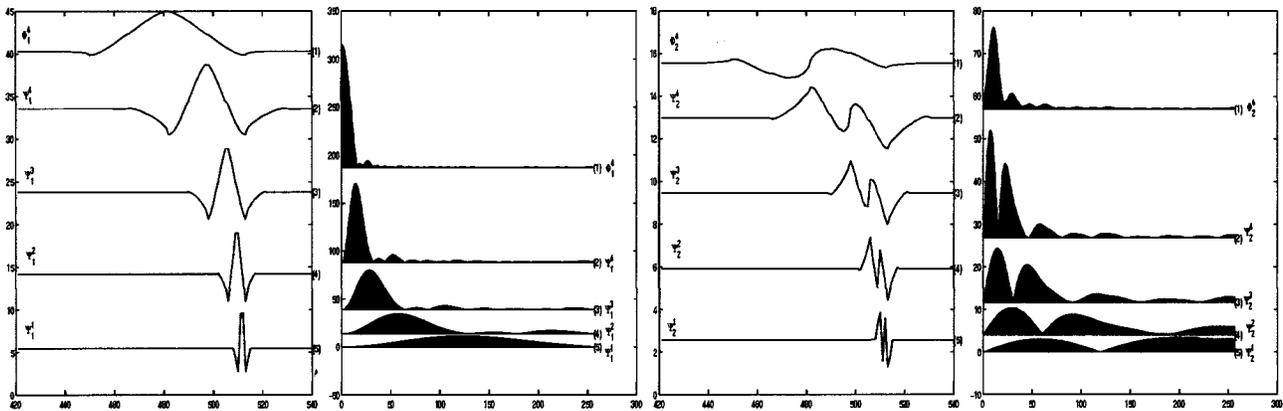


Fig. 4. Left pair of pictures: Dual synthesis wavelets $\varphi_{1,b}^i, \psi_{1,b}^i, i = 4, 3, 2, 1$, and their Fourier spectra. Right pair of pictures: Wavelets $\varphi_{2,b}^i, \psi_{2,b}^i, i = 4, 3, 2, 1$, and their Fourier spectra.

From (6.54), we have

$$\begin{pmatrix} d_1^u(0) \\ d_2^u(0) \end{pmatrix} = \sum_r \tilde{G}(l) \cdot \begin{pmatrix} f_1(-l) \\ f_2(-l) \end{pmatrix}.$$

Here, \vec{f} is a vector signal derived from \mathbf{F} by pre-processing. Due to Corollary 6.1, $f_1(-l)$ are the values of a polynomial of degree r sampled on the grid $-lh$, and $f_2(-l)$ are the values of its derivative multiplied by h . From Corollary 4.1, we have $d_1^u(0) = d_2^u(0) = 0$.

For the dual wavelets, the proof is similar, with the difference that Proposition 4.3 should be employed instead of Corollary 6.1. ■

VII. EXTENSION OF THE MULTIWAVELET TRANSFORMS TO COARSER LEVELS

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 in Section III on the vector array \vec{s}^{ur} given by (3.9) or (3.15), 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 of whether the primal or the dual transform was

applied at the first step. Therefore, in this stage, we have four different ways to represent the signal \mathbf{F} . We outline the situation when both first and second steps of the multiwavelet transform are implemented in the primal mode. Other three cases are similarly treated.

Proposition 7.1: As a result of the three steps of the multiwavelet transform (pre-processing and two steps of vector wavelet transform), the signal $\mathbf{F} = \{F(kh/2)\}$ is expanded as follows.

$$\begin{aligned} F(r) = & \sum_l d_{1,b}^u(l) \psi_{1,b}^1(l-4r) + d_{2,b}^u(l) \psi_{2,b}^1(l-4r) \\ & + \sum_l s_1^{ur,2}(l) \varphi_1^2(l-8r) + s_2^{ur,2}(l) \varphi_2^2(l-8r) \\ & + d_{1,b}^{u,2}(l) \psi_{1,b}^2(l-8r) + d_{2,b}^{u,2}(l) \psi_{2,b}^2(l-8r). \end{aligned} \quad (7.64)$$

The synthesis multiwavelets of the second level $\vec{\varphi}^2$ and $\vec{\psi}_b^2$ are derived from the first level multiwavelets $\vec{\varphi}^1$ through the two-scale equations

$$\begin{aligned} \vec{\varphi}^2(l) &= \sum_k H(k)^T \cdot \vec{\varphi}^1(l-4k) \\ \vec{\psi}_b^2(l) &= \sum_k G_b(k)^T \cdot \vec{\varphi}^1(l-4k). \end{aligned} \quad (7.65)$$

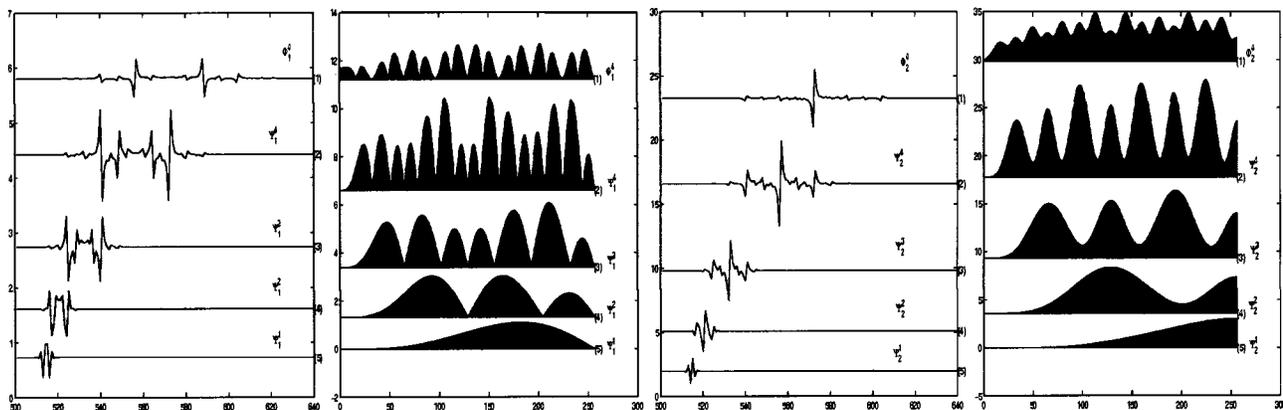


Fig. 5. Left pair of pictures: Primal analysis wavelets $\varphi_{1,b}^i, \psi_{1,b}^i, i = 4, 3, 2, 1$, and their Fourier spectra. Right pair of pictures: Wavelets $\varphi_{2,b}^i, \psi_{2,b}^i, i = 4, 3, 2, 1$, and their Fourier spectra.

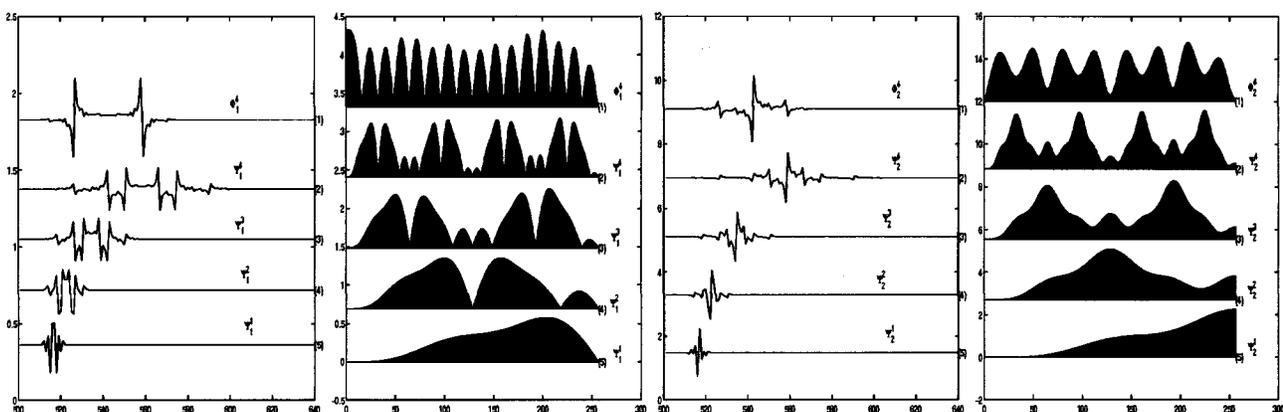


Fig. 6. Left pair of pictures: Dual analysis wavelets $\tilde{\varphi}_{1,b}^i, \tilde{\psi}_{1,b}^i, i = 4, 3, 2, 1$, and their Fourier spectra. Right pair of pictures: Wavelets $\tilde{\varphi}_{2,b}^i, \tilde{\psi}_{2,b}^i, i = 4, 3, 2, 1$, and their Fourier spectra.

The coefficients of the expansion are the inner products of the signal \mathbf{F} with the analysis multiwavelets

$$\begin{aligned} \vec{d}^{u,2}(k) &= \sum_l \mathbf{F}(l) \tilde{\psi}^2(l-8k) \\ \vec{s}_b^{ur,2}(k) &= \sum_l \mathbf{F}(l) \tilde{\varphi}_b^2(l-8k). \end{aligned}$$

The analysis multiwavelets of the second-level $\tilde{\varphi}_b^2$ and $\tilde{\psi}^2$ are derived from the first-level multiwavelets $\tilde{\varphi}_b^1$ through the two-scale equations

$$\begin{aligned} \tilde{\psi}^2(l) &= \sum_s \tilde{G}(s) \cdot \tilde{\varphi}^1(l+4s) \\ \tilde{\varphi}_b^2(l) &= \sum_s \tilde{H}_b(s) \cdot \tilde{\varphi}^1(l+4s). \end{aligned} \quad (7.66)$$

As in the first level, all analysis multiwavelets of the second level have the vanishing moments property. Again, we consider the case when both first and second steps of the multiwavelet transform are implemented in the primal mode.

Theorem 7.1: In the case when the Haar preprocessing algorithm is used, the analysis wavelets of the second level $\tilde{\psi}_1^2$ and $\tilde{\psi}_2^2$ have three vanishing moments. When the fifth-order preprocessing algorithms are used, the analysis wavelets have four vanishing moments.

Proof: Assume that $\mathbf{F}(k) = k^r, k \in \mathbb{Z}$. This is a polynomial of degree r sampled on the grid $kh/2, h = 2$. Equation (6.58) implies that

$$\begin{aligned} \sum_k \tilde{\psi}_1^2(k) k^r &= \langle \mathbf{F}, \tilde{\psi}_1^2 \rangle = d_1^{u,2}(0) \\ \sum_k \tilde{\psi}_2^2(k) k^r &= \langle \mathbf{F}, \tilde{\psi}_2^2 \rangle = d_2^{u,2}(0). \end{aligned}$$

From (6.54) and (4.19), we have

$$\begin{aligned} \begin{pmatrix} d_1^{u,2}(0) \\ d_2^{u,2}(0) \end{pmatrix} &= \sum_l \tilde{G}(l) \cdot \begin{pmatrix} s_{1,b}^{ur}(-l) \\ s_{2,b}^{ur}(-l) \end{pmatrix} \\ \vec{s}^{ur}(-l) &= \sum_k \tilde{H}_b(-2l-k) \cdot \vec{f}(k) \end{aligned}$$

where \vec{f} is a vector signal derived from \mathbf{F} by preprocessing. Due to Corollary 6.1, $f_1(-l)$ are the values of a polynomial of degree r sampled on the grid $-lh$, and $f_2(-l)$ are the values of its derivative multiplied by h . Then, from Proposition 4.3, we conclude that $s_{1,b}^{ur}(-l)$ are the values of a polynomial of degree r sampled on the grid $-2lh$, and $s_{2,b}^{ur}(-l)$ are the values of its derivative multiplied by $2h$. From Corollary 4.1, we have $d_1^{u,2}(0) = d_2^{u,2}(0) = 0$. ■

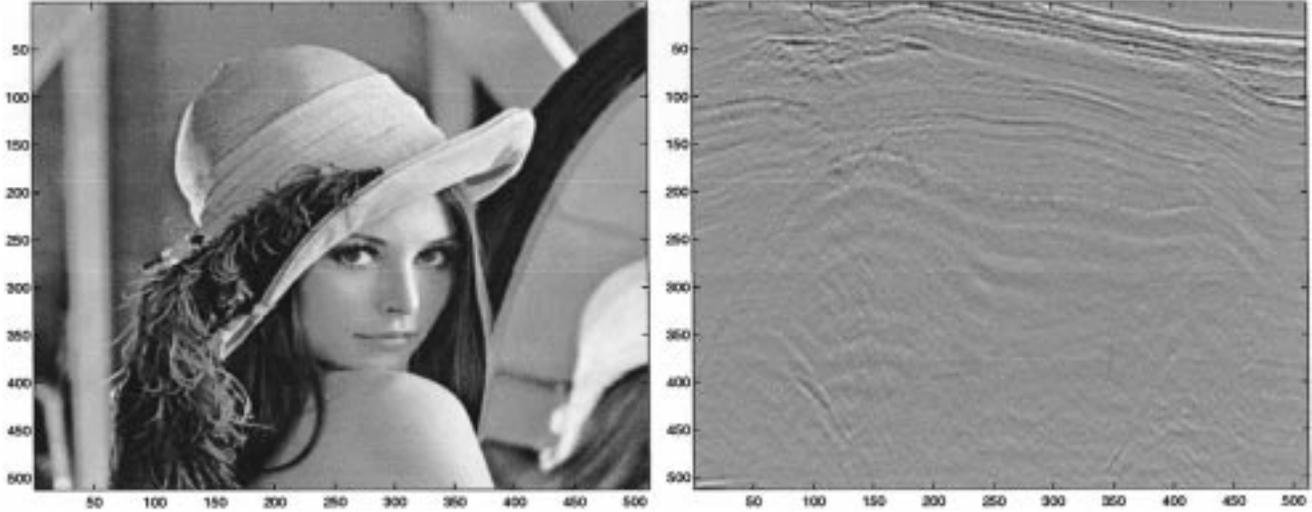


Fig. 7. (Left) *Lena* and (right) a fragment of a stacked seismic section. Their sizes are 512×512 pixels.

TABLE IV
ENTROPY AND ENERGY COMPACTION RATIOS AFTER THE APPLICATION OF WAVELET AND MULTIWAVELET TRANSFORMS INTO FIVE LEVELS

Transform:	original	D8	B7/9	VP/1	VP/2	VP/3	VD/1	VD/2	VD/3
$\chi(\text{Lena})$	7.49	4.51	4.49	4.31	4.32	4.23	3.66	3.67	3.60
$\chi(\text{Seismic})$	6.57	5.75	5.62	5.061	5.06	5.057	5.033	5.029	5.025
$E(\text{Lena})$	0.34	0.9914	0.9917	0.9507	0.9510	0.9557	0.9760	0.9760	0.9789
$E(\text{Seismic})$	0.7	0.8713	0.8729	0.9392	0.9392	0.9396	0.9195	0.9199	0.9207

Decomposition into coarser levels of the multiwavelet transform is similarly performed. The property of vanishing moments is retained at coarser levels.

In Figs. 3 to 6, we display some examples of synthesis and analysis wavelets. Here, the algorithm for post-processing of Scheme I is used. We do not display the corresponding pictures for other schemes of post-processing because they are almost indistinguishable from the displayed ones.

VIII. DISCUSSION

An obvious direction for further research activity is in the application of the devised multiwavelet transforms to signal and image processing. One such application is image compression. To evaluate the potential of the transforms for data compression, we compared the energy compaction [see (8.67)] and the entropies [see (8.68)] of the coefficients of the multiwavelet transforms of *Lena* and a stacked seismic section (see Fig. 7) with those of some classical wavelet transforms.

The values of the pixels are centered around the mean values. As a measure for the energy compaction of the data array $\mathbf{a} = \{a(k)\}_{k=1}^M$, we use the ratio

$$E(\mathbf{a}) \triangleq \frac{\sum_{k=1}^{M/10} b^2(k)}{\sum_{k=1}^M a^2(k)} \quad (8.67)$$

where $\{b(k)\}_{k=1}^M$ is the result after sorting the array $\{|a(k)|\}_{k=1}^M$, in descending order. The entropy is calculated as follows.

$$\chi(\mathbf{a}) \triangleq - \sum_{k=1}^M \frac{a^2(k)}{\sum_{l=1}^M a^2(l)} \log_e \frac{a^2(k)}{\sum_{l=1}^M a^2(l)}. \quad (8.68)$$

We denote the primal vector wavelet transforms by VP/ k and the dual ones by VD/ k , where k is the index of the pre-processing algorithm: $k = 1$ stands for the Haar scheme, $k = 2$ for Scheme I of fifth order, and $k = 3$ for Scheme II of fifth order (see Section V-B). By D8, we denote the orthogonal wavelet transform with the eight-tap Daubechies filters and, by B7/9, the biorthogonal wavelet transform with the 7/9 filters [5], which is frequently used in image compression. We applied the 1-D transforms up to the fifth decomposition level to columns of both images. We selected six columns (5 105 205 305 405 505) from each image. Note that in the seismic section, the columns are seismic traces. We present the results of the entropy and the energy compaction ratio of the coefficients of the multiscale expansion in Table IV. For comparison, we give the same information on the original arrays.

Remarks:

- 1) Since the data we are operating on are given on finite intervals, implementation of the above transforms near the boundaries requires a special treatment. We performed symmetric extension on the boundaries of the

scalar signals. Then, preprocessing algorithms produced vector signals whose first component was symmetrically extended. The second component was antisymmetrically extended. We used such an extension in the subsequent steps of the vector wavelet transform.

- 2) Lifting realization of the transforms enables fast implementation. The computational cost of the five-level transform is lower than the cost of implementation of the B7/9 transform, even if the latter is conducted via a fast factorized algorithm [6]. Therefore, the decomposition into five levels by the B7/9 transform requires $31/8$ multiplications and $31/4$ additions per sample. The multiwavelet transform with Haar preprocessing scheme requires $35/8$ multiplications and $39/8$ additions per sample. When a scheme of fifth order is used, these figures grow to $43/8$ and $47/8$, respectively.

We can observe from the table that in both cases, and especially for the *Lena* image, the dual vector wavelet transforms produce smaller entropies than the primal ones. The results depend on the pre-processing algorithm. The pre-processing Scheme II of the fifth order produces the most efficient entropy description of the coefficients. The coefficients of the multiwavelet transforms have smaller entropies than the coefficients of the wavelet transforms. The results of the calculation of the energy compaction are different for different data. For *Lena*, the wavelet transforms result in better energy compaction in comparison to multiwavelet transforms, but for the seismic traces, the multiwavelet transforms outperform the wavelet transforms. We note that the computational cost of the implementation of the presented multiwavelet transforms is lower in comparison with the cost of the lifting implementation of the B7/9 wavelet transform [6] and is significantly lower than the cost of the standard implementation of the B7/9 and D8 wavelet transforms.

IX. CONCLUSIONS

We constructed a new family of biorthogonal multiwavelet transforms for discrete-time signals. A characteristic feature of our approach is that we do not use scaling functions but, instead, treat the multiwavelet transforms of the signals as a tight union of two phases: 1) Pre (post)-processing algorithms that convert the scalar signal into the vector signal and back and 2) the wavelet transform of the vector-valued signal. The construction, as well as the implementation, of both phases of the transforms are conducted in a lifting manner. This allows fast computation and low use of memory and yields tools for a custom design of the transforms. We use cubic Hermite splines as a predicting aggregate in the vector wavelet transform. The scheme is interpolating, and, as such, it yields a relevant tool for processing discrete signals.

We use the ability to customize the vector wavelet transform in order to bring the properties of the dual transform close to the properties of the primal one. By doing so, we produce two vector wavelet transforms with similar basic properties. Each of the two modes of the wavelet transform is accompanied by any one of three pre (post)-processing algorithms, which we present

in the paper, but many more ways for using the customizing abilities are possible.

We devised new pre (post)-processing algorithms that are "custom tailored" to the vector wavelet transforms and allow the latter to perform with the optimal approximation accuracy. Namely, the analysis multiwavelets constructed using our pre-processing algorithms have four vanishing moments. The basis wavelets we constructed are symmetric and have compact and short supports.

The presented transforms have a potential for a successful application in image compression and especially to compression of seismic data. It is worth noting that actually, the transforms are integer-to-integer. This is very useful for fast ASIC implementation of the transforms for digital signal processing. Our approach is generic. It allows construction of multiwavelet transforms with higher approximation accuracy using Hermite splines of higher degrees and appropriate pre (post)-processing algorithms. Moreover, development of completely discrete multiwavelet transforms based on discrete analogs of Hermite splines is promising.

REFERENCES

- [1] A. Z. Averbuch and V. A. Zheludev, "Construction of biorthogonal discrete wavelet transforms using interpolatory splines," *Appl. Comput. Harmon. Anal.*, vol. 12, pp. 25–26, 2002.
- [2] A. Z. Averbuch, A. B. Pevnyi, and V. A. Zheludev, "Biorthogonal Butterworth wavelets derived from discrete interpolatory splines," *IEEE Trans. Signal Processing*, vol. 49, pp. 2682–2692, Nov. 2001.
- [3] R. L. Claypoole, Jr., J. M. Davis, W. Sweldens, and R. Baraniuk, "Non-linear wavelet transforms for image coding via lifting," *IEEE Trans. Signal Processing*, to be published.
- [4] W. Dahmen, B. Han, R.-Q. Jia, and A. Kunoth, "Biorthogonal multiwavelets on the interval: Cubic Hermite splines," *Constr. Approx.*, vol. 16, pp. 221–259, 2000.
- [5] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: SIAM, 1992.
- [6] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Appl.*, vol. 4, no. 3, pp. 247–269, 1998.
- [7] G. Davis, V. Strela, and R. Turcajova, "Multiwavelet construction via the lifting scheme," in *Wavelet Analysis and Multiresolution Methods*, T.-X. He, Ed. New York: Marcel Dekker, 1999, Lecture Notes in Pure and Applied Mathematics.
- [8] D. L. Donoho, "Interpolating wavelet transform," Dept. Statist., Stanford Univ., Stanford, CA, Preprint 408, 1992.
- [9] J. S. Geronimo, D. P. Hardin, and P. R. Massopust, "Fractal functions and wavelet expansions based on several scaling functions," *J. Approx. Theory*, vol. 78, pp. 373–401, 1994.
- [10] S. S. Goh, Q. Jiang, and T. Xia, "Construction of biorthogonal multiwavelets using the lifting scheme," *Appl. Comput. Harmonic Anal.*, vol. 9, pp. 336–352, 2000.
- [11] T. N. T. Goodman and S. L. Lee, "Wavelets of multiplicity r ," *Trans. Amer. Math. Soc.*, vol. 342, pp. 307–324, 1994.
- [12] B. Han and Q. Jiang, "Multiwavelets on the interval," Preprint, 1999.
- [13] L. Hervé, "Multi-resolution analysis of multiplicity d : Application to dyadic interpolation," *Appl. Comput. Harmonic Anal.*, vol. 1, pp. 299–315, 1994.
- [14] G. Strang and V. Strela, "Finite element multiwavelets," in *Approximation Theory, Wavelets and Applications*, S. P. Singh, Ed. Boston, MA: Kluwer, 1995, pp. 485–496.
- [15] J. Lebrun and M. Vetterli, "Balanced multiwavelets, theory and design," *IEEE Trans. Signal Processing*, vol. 46, pp. 1119–1125, June 1998.
- [16] V. Strela, "A note on construction of biorthogonal multi-scaling functions," in *Contemporary Mathematics*, A. Aldroubi and E. B. Lin, Eds. New York: Amer. Math. Soc., 1998, pp. 149–157.
- [17] V. Strela and A. T. Walden, "Orthogonal and biorthogonal multiwavelets for signal denoising and image compression," *Proc. SPIE*, vol. 3391, 1998.
- [18] V. Strela, "Multiwavelets: Theory and applications," Ph.D. dissertation, Mass. Inst. Technol., Cambridge, 1996.

- [19] W. Sweldens, "The lifting scheme: A custom design construction of biorthogonal wavelets," *Appl. Comput. Harmon. Anal.*, vol. 3, no. 2, pp. 186–200, 1996.
- [20] W. Sweldens and P. Schröder, "Building your own wavelets at home," in *Wavelets in Computer Graphics*. New York: ACM, 1996, pp. 15–87.
- [21] J. Y. Tham, L. Shen, S. L. Lee, and H. H. Tan, "A general approach for analysis and application of discrete multiwavelet transforms," *IEEE Trans. Signal Processing*, vol. 48, pp. 457–464, Feb. 2000.
- [22] R. Turcajova, "Construction of symmetric biorthogonal multiwavelets by lifting," *Proc. SPIE*, vol. 813, pp. 443–453, 1999.
- [23] X.-G. Xia, J. S. Geronimo, D. P. Hardin, and B. W. Suter, "Design of prefilters for discrete multiwavelet transforms," *IEEE Trans. Signal Processing*, vol. 44, pp. 25–35, Jan. 1996.
- [24] X.-G. Xia, "A new prefilter design for discrete multiwavelet transforms," *IEEE Trans. Signal Processing*, vol. 46, pp. 1558–1570, June 1998.



Amir Z. Averbuch was born in Tel Aviv, Israel. He received the B.Sc. and M.Sc. degrees in mathematics from the Hebrew University, Jerusalem, Israel, in 1971 and 1975, respectively, and the Ph.D. degree in computer science from Columbia University, New York, NY, in 1983.

From 1966 to 1970 and from 1973 to 1976, he served in the Israeli Defense Forces. From 1976 to 1986, he was a Research Staff Member with the Department of Computer Science, IBM T. J. Watson Research Center, Yorktown Heights, NY. In 1987, he

joined the Department of Computer Science, School of Mathematical Sciences, Tel Aviv University, where he is now Associate Professor of computer science. His research interests include wavelet applications for signal/image processing and numerical computation, multiresolution analysis, and scientific computing (fast algorithms and parallel and supercomputing software and algorithms).



Valery A. Zheludev received the M.S. degree in mathematical physics in 1963 from St. Petersburg University, St. Petersburg, Russia. In 1968, he received the Ph.D. degree in mathematical physics from Steklov Mathematical Institute of the Academy of Sciences of the USSR and, in 1991, received the Dr. Sci. degree in computational mathematics from Siberia Branch of the Academy of Sciences of the USSR.

From 1963 to 1965, he was a Lecturer at Pedagogical University, St. Petersburg. From 1965 to 1968 he was a Ph.D. student at St. Petersburg University. From 1968 to 1970, he was an Assistant Professor with Kaliningrad University. From 1970 to 1975, he was a Senior Researcher with the Research Institute for Electric Measuring Devices, St. Petersburg. From 1975 to 1995, he was an Associate and then Full Professor at the St. Petersburg Military University for Construction Engineering. Since 1995, he has been a Researcher and then a Senior Researcher with the School of Computer Science, Tel Aviv University, Tel Aviv, Israel, and with Paradigm Geophysical, Ltd. His fields of research include wavelet analysis, approximation theory, signal and image processing, geophysics, and pattern recognition.

Dr. Zheludev received the best Ph.D. thesis prize from St. Petersburg University.