A New Family of Spline-Based Biorthogonal Wavelet Transforms and Their Application to Image Compression

Amir Z. Averbuch and Valery A. Zheludev

***Abstract*—In this paper. we design a new family of biorthogonal wavelet transforms and describe their applications to still image compression. The wavelet transforms are constructed from various types of interpolatory and quasiinterpolatory splines. The transforms use finite impulse response and infinite impulse response filters that are implemented in a fast lifting mode.**

***Index Terms*—Image compression, lifting scheme, spline, wavelet transform.**

I. Introduction

WAVELET transforms have become one of the basic tools of signal processing, image processing, and numerical computation. In particular, biorthogonal symmetric wavelets manifested remarkable abilities in still image compression. So-called 9/7 wavelets [1] were adopted by the JPEG 2000 image compression standard [26]. In this paper, we present a new family of biorthogonal wavelet transforms, which are proven to be efficient for image compression. The performance of still image compression of the transforms that belong to this family is similar to the performance of the transform with 9/7 wavelets on most of the benchmark images. Moreover, the technique, which we developed for the design of wavelet transforms, is generic. It enables us to construct transforms with predetermined properties and provides a means for flexible adaptation of the transforms to problems under consideration.

Our approach combines versatile custom-design capabilities which are inherent in the lifting schemes [31] with the usage of the well-developed theory of interpolatory, quasiinterpolatory, continuous, and discrete splines [25], [30], [36], [38].

Polynomial splines are a common source for wavelet constructions. Until recently, two approaches governed the construction of wavelet schemes that use splines. One is based on orthogonal (Battle–Lemarié, [7], [19]) and semi-orthogonal wavelets in spline spaces [9], [34], [39]. This approach produces compactly supported spline wavelets. However, their dual wavelets have infinite support. The other approach, which employs splines in wavelet analysis, was introduced by Cohen, Daubechies, and Feauveau [11], who constructed symmetric compactly supported spline wavelets whose dual wavelets remain compactly supported and symmetric but do not belong to a spline space.

However, since the introduction of the lifting scheme for the design of wavelet transforms [31], a new way has opened for the use of splines as a tool for devising wavelet transforms.

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

1) *Split:* The signal is split into even and odd subarrays.
2) *Predict:* The filtered even array is used to predict the odd array. Then, the odd array is redefined as the difference between the existing array and the predicted one. If the predictor is correctly chosen then this step decorrelates the signal and reveals its high-frequency component.
3) *Update:* To eliminate aliasing, which appears while downsampling the original signal, and to obtain the low-frequency component of the signal, the even array is updated using the filtered new odd array.

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

The transform generates biorthogonal wavelet bases for the signal space. The specifics of the transform and its generated wavelets are determined by the choice of the predicting and updating filters. In the construction by Donoho [14], an odd sample is predicted from a polynomial interpolation of neighboring even samples. We propose using various kinds of splines as predicting and updating aggregates in the lifting scheme. In the following, we outline how we use splines for this purpose.

- *Continuous interpolatory splines:* The interpolatory spline of odd order (even degree) with equidistant nodes possesses the super-convergence property at the midpoints of the intervals between grid points [36]. This property increases the number of vanishing moments and the regularity of the corresponding wavelets. Thus, we construct a spline of odd order, which interpolates even samples of the signal, and use the values of the spline at the midpoints between grid points as prediction of odd samples of the signal [5].
- *Discrete interpolatory splines:* Another option is to use the discrete interpolatory splines [3], [4]. In this case, explicit formulas for the transforms that have any number of vanishing moments are established. Moreover, our investigation reveals an interesting relation between discrete splines and Butterworth filters, which are commonly used in signal processing. The filter banks used in our scheme comprise filters which act as bidirectional half-band Butterworth filters.

Manuscript received January 20, 2002; revised November 18, 2002. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ivan W. Selesnick.

The authors are with The School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel (e-mail: amir@math.tau.ac.il).

Digital Object Identifier 10.1109/TIP.2004.827229

The frequency response of a Butterworth filter is maximally flat and we succeed in the construction of dual filters with a similar property. Unlike the construction in [14], the above transforms used causal and anticausal filters with infinite impulse response (IIR). Fortunately, the transfer functions of the employed filters are rational. Therefore, filtering can be performed recursively. One-directional Butterworth filters were used for devising orthogonal nonsymmetric wavelets [18]. The computations there were conducted in time domain using recursive filtering. A scheme that used recursive filters for the construction of biorthogonal symmetric wavelets and their application to image processing was presented in [21] and [24].

- *f) Quasiinterpolatory splines:* There is a way to devise wavelet transforms that employ finite impulse response (FIR) filters whose properties are similar to the properties of interpolatory transforms. It can be done using the so-called quasiinterpolatory splines [36]. These splines also possess the super-convergence property.

The rest of the paper is organized as follows. In Section II, we outline the lifting scheme of wavelet transforms and discuss its relation to the conventional setting of wavelet transforms. In Section III, we describe the derivation of the prediction and update filters from interpolatory and quasiinterpolatory splines. The filters, which stem from interpolatory splines, have rational transfer functions. Note that IIR filters with rational transfer functions, which allow recursive implementation, appear in signal processing algorithms using spline functions. Construction and implementation of these filters was studied in [32], [33]. Our scheme that implements these filters is close to that of [33]. More details are provided in Appendix I (Section I). By combining the devised filters, we construct in Section IV a number of biorthogonal wavelet transforms. Then, in Section V, we present compression results for four benchmark images after applying these transforms. These results are also compared with the results from the 9/7 transform. In Appendix II (Section II), we describe a direct two-dimensional (2-D) implementation of a transform, which uses FIR filters.

II. Biorthogonal Wavelet Transforms: Background

We call the sequences $\{a(k)\}_{k=-\infty}^{\infty}$, which belong to the space $l_1 \cup l_2$, the discrete-time signals. The z transform of a signal $\{a(k)\}$ is defined as $A(z) = \sum_{k=-\infty}^{\infty} z^{-k} a(k)$. Throughout the paper, we assume that $z = e^{i\omega}$. We introduce a family of biorthogonal wavelet-type transforms that operate on the signal $\mathbf{x} = \{x(k)\}_{k=-\infty}^{\infty}$, which we construct through lifting steps.

We recall the basic lifting scheme formulation of a wavelet transform. The lifting scheme [31] can be implemented in either primal or dual mode.

A. Lifting Scheme of the Wavelet Transform

1) Decomposition: Generally, the primal lifting scheme for decomposition of signals consists of four steps: a) split; b) predict; c) update or lifting; and d) normalization. Let $\mathbf{x}$ be a signal and $X(z)$ its z transform.

a) Split: The signal $\mathbf{x}$ is split into even and odd subarrays: $\mathbf{e_1} = \{e_1(k) = x(2k)\}, \mathbf{o_1} = \{o_1(k) = x(2k+1)\}, k \in \mathbb{Z}$. In the z-transform domain, this operation corresponds to the following relation: $E_1(z^2) = (X(z) + X(-z))/2, O_1(z^2) = z(X(z) - X(-z))/2$, where $E_1(z)$ and $O_1(z)$ denote the z transforms of $\mathbf{e_1}$ and $\mathbf{o_1}$, respectively.

b) Predict: The even array $\mathbf{e_1}$ is used to predict the odd array $\mathbf{o_1}$. Then, the new odd array $\mathbf{o_1^{\nu}}$ is defined as the difference between the existing array $\mathbf{o_1}$ and the predicted one. To be specific, we apply some *prediction* filter U to the array $\mathbf{e_1}$, in order for the result to approximate the array $\mathbf{o_1}$. Then, we subtract this result from the array $\mathbf{o_1}$: $O_1^{\nu}(z) = O_1(z) - U(z)E_1(z)$. From now on the superscript ν denotes the new array. Provided that the filter U is properly chosen, this step results in decorrelation of the signal.

c) Update (lifting): The even array is updated using the new odd array that is being convolved with the *update* filter whose transfer function we denote by $V(z)/2$: $E_1^{\nu}(z) = E_1(z) + (1/2)V(z)O_1^{\nu}(z)$. Generally, the goal of this step is to eliminate aliasing which appears when the original signal $\mathbf{x}$ is downsampled into $\mathbf{e_1}$. By doing so $\mathbf{e_1}$ is transformed into a downsampled and smoothed (low-pass filtering) replica $\mathbf{e_1^{\nu}}$ of $\mathbf{x}$.

d) Normalization: Finally, the smoothed array $\mathbf{s_1}$ and the array of details $\mathbf{d_1}$ are obtained from the following operation: $\mathbf{s_1} = \sqrt{2}\mathbf{e_1^{\nu}}, \mathbf{d_1} = \mathbf{o_1^{\nu}}/\sqrt{2}$.

The key issue in this lifting scheme is how to properly choose the filters U and V. We address this issue in subsequent sections.

2) Reconstruction: The reconstruction of the signal $\mathbf{x}$ from the arrays $\mathbf{s_1}$ and $\mathbf{d_1}$ is implemented in a reverse order.

a) Undo Normalization: $\mathbf{e_1^{\nu}} = \mathbf{s_1}/\sqrt{2}\ \mathbf{o_1^{\nu}} = \sqrt{2}\mathbf{d_1}$.

b) Undo lifting: The even array $E_1(z) = E_1^{\nu}(z) - V(z)Q_1^{\nu}(z)/2$ is restored.

c) Undo Predict: The odd array $O_1(z) = O_1^{\nu}(z) + U(z)E_1(z)$ is restored.

d) Undo Split: It is the standard restoration of the signal from its even and odd components. In the z domain, it appears as $X(z) = E_1(z^2) + z^{-1}O_1(z^2)$.

3) Dual Mode:

a) Update: The even array is averaged with the filtered odd array $E_1^{\nu}(z) = (E_1(z) + V(z)O_1(z))/2$.

b) Predict: The odd array is predicted by the filtered new even array $O_1^{\nu}(z) = O_1(z) - U(z)E_1^{\nu}(z)$.. The reconstruction is achieved in a reverse order.

In the primal construction, the update step follows 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, dual mode allows us to use an adaptive nonlinear wavelet transform [10] by choosing different predictors for different fragments of a signal or an image.

B. Filter Banks and Bases for Signal Space

The lifting schemes presented above yield efficient algorithms for the implementation of the forward and backward transforms of the signal $\mathbf{x} \leftrightarrow \mathbf{s_1} \cup \mathbf{d_1}$. For completeness, we briefly outline the links between these schemes and the transforms of the signal by two-channel multirate perfect reconstruction filter banks. We also outline the relations between

the lifting scheme and the expansion of signals with respect to the corresponding biorthogonal bases [23], [35].

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

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

Here, $\tilde{H}(z)$ and $\tilde{G}(z)$ are the low- and high-pass primal analysis filters, respectively, and $H(z)$ and $G(z)$ are the low- and high-pass primal synthesis filters, respectively.

The decomposition and reconstruction equations can be represented as follows:

$$S_1(z^2) = \frac{1}{2}(\tilde{H}(z)X(z) + \tilde{H}(-z)X(-z))$$
$$D_1(z^2) = \frac{1}{2}(\tilde{G}(z)X(z) + \tilde{G}(-z)X(-z)\Big)$$
$$X(z) = H(z)S_1(z^2) + G(z)D_1(z^2)$$

where $S_1(z)$ and $D_1(z)$ are the z transforms of the arrays $\mathbf{s_1}$ (smooth) and $\mathbf{d_1}$ (details), respectively.

The filter bank $\tilde{H}(z), \tilde{G}(z), H(z)$, and $G(z)$ possesses the perfect reconstruction property [23], [35]

$$H(z)\tilde{H}(z) + G(z)\tilde{G}(z) = 2$$
$$H(z)\tilde{H}(-z) + G(z)\tilde{G}(-z) = 0.$$

The filters are linked to each other as follows:

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

The shifts of impulse response of the filters $H(z), G(z), \tilde{H}(z)$, and $\tilde{G}(z)$ form biorthogonal pairs of bases in the space of discrete-time signals. We denote these impulse responses by $\varphi^1(k), \psi^1(k), \tilde{\varphi}^1(k)$, and $\tilde{\psi}^1(k)$, respectively. It means, for example, that $H(z) = \sum_{k\in\mathbb{Z}} z^{-k}\varphi^1(k)$ and similar relations hold for the other functions. Any signal $\mathbf{x}$ can be represented as follows:

$$x(l) = \sum_{k\in\mathbb{Z}} s_1(k)\varphi^1(l-2k) + \sum_{k\in\mathbb{Z}} d_1\psi^1(l-2k).$$

The coordinates $s_1(k)$ and $d_1(k)$ are the inner products

$$s_1(k) = \left\langle \mathbf{x}, \tilde{\varphi}^1_k \right\rangle, \quad \text{where} \quad \tilde{\varphi}^1_k(l) = \tilde{\varphi}^1(l-2k)$$
$$d_1(k) = \left\langle \mathbf{x}, \tilde{\psi}^1_k \right\rangle, \quad \text{where } \tilde{\psi}^1_k(l) = \tilde{\psi}^1(l-2k) \quad (2.1)$$

and the following biorthogonal relations hold: $\langle\tilde{\varphi}^1_k, \varphi^1_l\rangle = \langle\psi^1_k, \tilde{\psi}^1_l\rangle = \delta^l_k, \langle\tilde{\varphi}^1_k, \psi^1_l\rangle = \langle\tilde{\psi}^1_l, \varphi^1_k\rangle = 0, \forall l, k$.

The functions φ^1 and ψ^1 are called the low- and high-frequency synthesis wavelets of the first scale, respectively, and the functions $\tilde{\varphi}^1$ and $\tilde{\psi}^1$, the low- and high-frequency analysis wavelets of the first scale, respectively.

We say that a wavelet ψ has m vanishing moments if the following relations hold: $\sum_{k\in\mathbb{Z}} k^s\psi(k) = 0, s = 0, 1, \ldots, m-1$. The following proposition is related to the well-known Fix–Strang condition [17].

Proposition 2.1: Let the filters $U(z)$ and $V(z)$, which are used for the *predict* and *update* steps, respectively, be rational and have no poles on the unit circle $|z| = 1$. If $1 - z^{-1}U(z^2)$ comprises the factor $(z - 2 + z^{-1})^r$, then the high-frequency analysis wavelet $\tilde{\psi}^1$ has $2r$ vanishing moments. If, in addition, $1-zV(z^2)$ comprises the factor $(z-2+z^{-1})^p$ then the high-frequency synthesis wavelet ψ^1 has $2s$ vanishing moments, where $s = \min(p, r)$.

Proof: Let us ignore for a moment the assumption $|z| = 1$ and examine the function

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

of the complex variable z. We have

$$\tilde{q}(z) = z^{-1}(1 - zU(z^{-2}))/\sqrt{2} = (1-z)^{2r}\tilde{Q}(z)$$

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

$$q(z) = z\sqrt{2}(1 - (2z)^{-1}V(z^{-2})(1 + zU(z^{-2})))$$
$$= \frac{z}{\sqrt{2}}((1 - z^{-1}V(z^{-2})) + (1 - zU(z^{-2}))$$
$$+ zU(z^{-2})(1 - z^{-1}V(z^{-2}))) = (1-z)^{2s}Q(z)$$

where $s = \min(p, r)$ and $\tilde{Q}(z)$ is a regular function in some vicinity of $z = 1$. Thus, the wavelet ψ^1 has $2s$ vanishing moments. ∎

Expansion of the transform to coarser scales is implemented recursively by storing the detail arrays $\mathbf{d}$ and decomposing the smooth arrays $\mathbf{s}$ with the same lifting steps as above.

The wavelets of subsequent scales are linked via the two-scale relations

$$\varphi^{m+1}(l) = \sum_{k\in\mathbb{Z}} \varphi^1(k)\varphi^m(l-2k)$$
$$\psi^{m+1}(l) = \sum_{k\in\mathbb{Z}} \psi^1(k)\varphi^m(l-2k)$$
$$\tilde{\varphi}^{m+1}(l) = \sum_{k\in\mathbb{Z}} \tilde{\varphi}^1(k)\tilde{\varphi}^m(l-2k)$$
$$\tilde{\psi}^{m+1}(l) = \sum_{k\in\mathbb{Z}} \tilde{\psi}^1(k)\tilde{\varphi}^m(l-2k).$$

The filters $H(z)$ and $\tilde{H}(z)$ generate two versions of the multiresolution analysis of the space L^2 with the synthesis $\varphi(x)$ and the analysis $\tilde{\varphi}(x)$ scaling functions, respectively [12].

III. Design of Predict and Update Filters

In this section, we first design the filters U for the predict operation. Our goal is to devise filters that when applied to an even sub-array of a signal, will produce a signal, which approximates the odd subarray. We will use polynomial and discrete splines as a tool for the construction of the appropriate filters.

In general, we construct a spline on the grid $\{2k\}$, which interpolates or quasiinterpolates the even samples $\{e_1(k)\} =$

$x(2k)$. The values of this spline at grid points $\{2k+1\}$ are used for the prediction of the odd samples $\{o_1(k) = x(2k+1)\}$.

A. Discrete Splines

In [3] and [4], we give a detailed presentation of the construction of prediction filters using discrete splines. Therefore, here we only recall the basic idea and present three filters which we will use for still image compression.

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

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

Higher order B-splines are defined via iterated discrete convolutions $\beta_{p,n} = \beta_{1,n} * \beta_{p-1,n}, p = 2, \ldots$. Obviously, the z-transform of the B-spline of order p is $\beta_{p,n}(z) = (1 + z^{-1} + z^{-2} + \cdots + z^{-2n+1})^p$.

In this paper, we are interested only in the case when $p = 2r, r \in \mathbb{N}$, and $n = 1$. In this case, we have $\beta_{2r,1}(z) = (1+z^{-1})^{2r}$. The B-spline $\beta_{2r,1}(j)$ is symmetric about the point $j = r$ where it reaches its maximal value. The central B-spline $Q^{2r}(j)$ of order $2r$ is a shift of the B-spline $Q^{2r}(j) \triangleq \beta_{2r,1}(j+r), Q^{2r}(z) = z^r(1+z^{-1})^{2r}$. The discrete spline of order $2r$ is defined as a linear combination with real-valued coefficients of shifts of the central B-spline of order $2r$, as follows:

$$S^{2r}(k) \triangleq \sum_{l=-\infty}^{\infty} c(l) Q^{2r}(k-2l).$$

Definition 3.1: Let $\{e(k)\}$ be a given sequence. The discrete spline S^{2r} is called the interpolatory spline if the following relations hold: $S^{2r}(2k) = e(k), k \in \mathbb{Z}$.

The following proposition shows how to construct an interpolatory splines of any order and to calculate its values at the midpoints between nodes, which we denote by $\sigma(k) = S^{2r}(2k+1), k \in \mathbb{Z}$.

Proposition 3.1 [3]: The interpolatory spline is represented as

$$S^{2r}(k) = \sum_{l=-\infty}^{\infty} c(l) Q^{2r}(k-2l),$$
$$C(z^2) = \frac{2E(z^2)}{z^r(1+z^{-1})^{2r} + (-z)^r(1-z^{-1})^{2r}}.$$

The z transform of the array of the interpolatory spline values at midpoints is

$$\sigma(z) = U_d^{2r}(z)E(z), \quad \text{where}$$
$$U_d^{2r}(z^2) = z\frac{(1+z^{-1})^{2r} - (-1)^r(1-z^{-1})^{2r}}{(1+z^{-1})^{2r} + (-1)^r(1-z^{-1})^{2r}}. \quad (3.2)$$

This proposition provides us with a family of prediction filters $U_d^{2r}(z), r = 1, 2, \ldots$. The transfer functions of these filters are rational functions of its argument (except the case $r = 1$), which do not have poles on the unit circle $|z| = 1$. Consequently, the filters have IIR $\{u_d^{2r}(k)\}$, which decay exponentially as $|k| \to \infty$. In [3], we derived explicit formulas, which enable fast implementation of these filters of any order via a cascade of elementary causal and anti-causal recursive filters. A parallel implementation is also possible.

It is interesting to note that these filters are closely related to the discrete-time Butterworth filters [3], [22], which are commonly used in signal processing. These Butterworth filters have maximally flat frequency responses. The magnitude-squared transfer functions of the low-pass B_l^r and high-pass B_h^r half-band Butterworth filters of order r are

$$|B_l^r(z)|^2 = \frac{(1+z^{-1})^{2r}}{(1+z^{-1})^{2r} + (-1)^r(1-z^{-1})^{2r}}$$
$$|B_h^r(z)|^2 = \frac{(-1)^r(1-z^{-1})^{2r}}{(1+z^{-1})^{2r} + (-1)^r(1-z^{-1})^{2r}}.$$

It is readily seen that the function U, defined in (3.2), is related to these transfer functions

$$1 + z^{-1}U_d^{2r}(z^2) = 2|B_l^r(z)|^2$$
$$1 - z^{-1}U_d^{2r}(z^2) = 2|B_h^r(z)|^2.$$

Examples of Filters:

a) Example 1, $r = 1$: This is the simplest case. We have

$$U_d^2(z) = \frac{1+z}{2}, \quad 1 - z^{-1}U_d^2(z^2) = -\frac{z^{-1} - 2 + z}{2}. \quad (3.3)$$

The filter U_d^2 is FIR. In time domain, this filter operates as follows:

$$y(k) = \frac{1}{2}(x(k) + x(k+1)). \quad (3.4)$$

Due to Proposition 2.1, the high-frequency analysis wavelet $\tilde{\psi}^1$ has two vanishing moments.

b) Example 2, $r = 3$

$$U_d^6(z) = \frac{1}{18}\frac{(z + 14 + z^{-1})(1+z)}{(1+z^{-1}/3)(1+z/3)}$$
$$1 - z^{-1}U_d^6(z^2) = -\frac{(z^{-1} - 2 + z)^3}{6z^2 + 20 + 6z^{-2}}. \quad (3.5)$$

The high-frequency analysis wavelet $\tilde{\psi}^1$ has six vanishing moments. Filtering with U_d^6 can be implemented via the following cascade of operations:

$$y_1(k) = \frac{1}{18}(x(k) + x(k+1)) \quad (3.6)$$
$$y_2(k) = y_1(k-1) + 14y_1(k) + y_1(k+1)$$
$$y_3(k) = y_2(k) - \frac{1}{3}y_3(k-1)) \quad (3.7)$$
$$y(k) = y_3(k) - \frac{1}{3}y(k+1). \quad (3.8)$$

From (3.7) and (3.8), we can see that the cascade comprises the causal and anti-causal elementary recursive blocks with rational coefficients. Another option is a parallel implementation. This option stems from the following decomposition of the function $U_d^6(z)$:

$$U_d^6(z) = \frac{1}{54}\left[-\frac{72}{1+z/3} - \frac{8z^{-1}}{1+z^{-1}/3} + 9(z+105)\right].$$

c) Example 3, $r = 4$:

$$U_d^8(z) = \frac{8(1+z)(z^{-1}+6+z)}{z^{-2}+28z^{-1}+70+28z+z^2}$$
$$1 - z^{-1}U_d^8(z^2) = \frac{(z^{-1}-2+z)^4}{z^{-4}+28z^{-2}+70+28z^2+z^4}. \tag{3.9}$$

d) Case $r = 2$: The filter U_d^4, which stems from the cubic discrete spline, coincides with a filter derived from the quadratic polynomial spline and is presented in Section III-B.

B. Polynomial Splines

In this section, we deal with polynomial splines. We discuss some of their properties and derive the corresponding filters U.

1) Interpolatory Splines: The central B-spline of first order on the grid $\{k\}$ is defined to be the characteristic function of the interval $[-1/2, 1/2]$. The central B-spline of order p is the convolution $M^p(x) = M^{p-1}(x) * M^1(x)$ $p \geq 2$. Note that the B-spline of order p is supported on the interval $(-p/2, p/2)$. It is positive within its support and symmetric around zero. The nodes of even order B-splines are located at points $\{k\}$ and the odd order nodes at points $\{k + 1/2\}, k \in \mathbb{Z}$. Denote

$$\mathbf{t}^p \triangleq \{M^p(k)\}, \quad \mathbf{w}^p \triangleq \left\{M^p\left(k+\frac{1}{2}\right)\right\}, \quad k \in \mathbb{Z}. \tag{3.10}$$

Due to the compact support of B-splines, these sequences are finite. We will use for our constructions only splines of odd order $p = 2r - 1$. In Table I, we present the sequences $\mathbf{t}^p$ and $\mathbf{w}^p$ for some values of p.

The z transforms $T^p(z)$ and $W^p(z)$ of the sequences $\mathbf{t}^p$ and $\mathbf{w}^p$ are Laurent polynomials. They are called the Euler–Frobenius polynomials [28]. As in Section II-A, to predict the odd samples $\{x(2k+1)\}$ of the signal $\mathbf{x}$, we use the values at the midpoints of the splines, which interpolate the even samples $\{x(2k)\}$. Such a construction is described in [5], [6]. In this case, we get for the spline of order p the following prediction filter:

$$U_i^p(z) \triangleq \frac{W^p(z)}{T^p(z)}. \tag{3.11}$$

It follows from (2.1) that if the analysis wavelet $\tilde{\psi}^1$ has p vanishing moments, then applying the wavelet transform to a polynomial of degree $p-1$ produces the zero detail array $\mathbf{d_1}$. It happens if the spline, which interpolates samples of a polynomial of degree $p-1$ on the grid $\{2k\}$, coincides with this polynomial at the points $\{2k+1\}$. Since polynomial splines of order p are piecewise polynomials of degree $p-1$, their maximal order of approximation is p. It means that the spline, which interpolates a polynomial of a degree not exceeding $p-1$, identically coincides with this polynomial. Thus one can expect that the analysis wavelets, that were constructed using interpolatory splines of order p, should have p vanishing moments. This is indeed the case when splines of even order $p = 2r$ are employed. Therefore, the following statement is surprising.

TABLE I
VALUES OF THE SEQUENCES $\mathbf{t}^p$ AND $\mathbf{w}^p$

k	-4	-3	-2	-1	0	1	2	3
$\mathbf{t}^3 \times 8$	0	0	0	1	6	1	0	0
$\mathbf{t}^5 \times 384$	0	0	1	76	230	76	1	0
$\mathbf{t}^7 \times 46080$	0	1	722	10543	23548	10543	722	1
$\mathbf{w}^3 \times 2$	0	0	0	1	1	0	0	0
$\mathbf{w}^5 \times 24$	0	0	1	11	11	1	0	0
$\mathbf{w}^7 \times 720$	0	1	57	302	302	57	1	0

Proposition 3.2 [5]: The rational functions $U_i^p(z)$ are real-valued. If $p = 2r - 1$, then

$$1 - z^{-1}U_i^p(z^2) = \frac{(z - 2 + z^{-1})^r \xi_r(\alpha)}{t^p(z^2)} \tag{3.12}$$

where $\alpha \triangleq z + z^{-1}$ and $\xi_r(\alpha)$ is a polynomial of degree $r-2$. Hence, the high-frequency analysis wavelet $\tilde{\psi}^1$, that was derived from the interpolatory spline of order $p = 2r - 1$, has $2r$ vanishing moments.

The rationale of Proposition 3.2 lies in the property of superconvergence at the midpoints between the nodes, which is inherent to interpolatory splines of odd order $2r-1$ [36]. It means that the spline of order $2r - 1$ (degree $2r - 2$), which interpolates samples of a polynomial of degree $2r - 1$ on the grid $\{2k\}$, coincides with this polynomial at the points $\{2k+1\}$. In other words, the spline of order $2r - 1$ produces wavelets with the same number of vanishing moments as the spline of order $2r$. But the computational complexity of the transform with the spline of order $2r-1$ is lower than the complexity of the transform using the spline of order $2r$. Moreover, wavelets that are based on the spline of order $2r-1$ are better localized in time domain than their counterparts that are based on the spline of order $2r$. Therefore, we prefer to use splines of odd order $p = 2r+1$.

2) Examples:

a) Example 1: $p = 3$ *(quadratic interpolatory spline)*

$$U_i^3(z) = 4\alpha\frac{1+z}{(1+\alpha z)(1+\alpha z^{-1})}$$
$$1 - z^{-1}U_i^3(z^2) = \frac{(z-2+z^{-1})^2}{z^{-2}+6+z^2} \tag{3.13}$$

where $\alpha = 3 - 2\sqrt{2} \approx 0.172$. Filtering can be implemented via the following cascade of operations:

$$y_0(k) = 4\alpha(x(k) + x(k+1))$$
$$y_1(k) = y_0(k) - \alpha y_1(k-1)$$
$$y(k) = y_1(k) - \alpha y(k+1). \tag{3.14}$$

The representation

$$U_i^3(z) = \frac{4\alpha}{1+\alpha}\left(\frac{1}{1+\alpha z^{-1}} + \frac{z}{1+\alpha z}\right)$$

enables a parallel filtering implementation.

Remark: The filter U_i^3 coincides with the filter U_d^4, which was derived from a cubic *discrete* spline. The analysis wavelet

that was generated by the quadratic spline, has four vanishing moments.

b) Example 2: $p = 4$ (cubic polynomial spline)

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

where $\alpha = 2 - \sqrt{3} \approx 0.2679$.

Remark: We observe that the analysis wavelet that was generated from the cubic spline has four vanishing moments like the wavelet that was generated from the quadratic spline. But the structure of the filter U_i^4, which determines the computational cost of the implementation, is more complex than the structure of U_i^3 and is similar to the structure of the filter U_d^6, which is based on the discrete spline of sixth order. However, U_d^6 produces a wavelet with six vanishing moments.

c) Example 3: $p = 5$ (spline of fourth degree)

$$U_i^5(z) = \frac{16(z+10+z^{-1})(1+z)}{z^2+76z+230+76z^{-1}+z^{-2}}$$
$$1 - z^{-1}U_i^5(z^2) = \frac{(z-2+z^{-1})^3(z+z^{-1}-10)}{z^4+76z^2+230+76z^{-2}+z^{-4}}. \tag{3.15}$$

The corresponding analysis wavelet has six vanishing moments.

3) Local Quasi-Interpolatory Splines: The useful property of super-convergence at midpoints is inherent not only to interpolatory splines but also to so-called *local quasiinterpolatory splines of odd order*, which can be constructed using FIR filtering.

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

$$S^p(x) = \sum_l q(l) M^p(x-l). \tag{3.16}$$

Denote $\mathbf{q} \triangleq \{q(l)\}$ and let $Q(z)$ be the z-transform of $\mathbf{q}$.

Definition 3.2: Let a function f be sampled on the grid $\{k\}, \mathbf{f} \triangleq \{f(k)\}$, and $F(z)$ be the z transform of $\mathbf{f}$. A spline S^p of order p given by (3.16) is called local if the array $\mathbf{q}$ of its coefficients is derived by FIR filtering the array $\mathbf{f}$

$$Q(z) = \Gamma(z)F(z) \tag{3.17}$$

where $\Gamma(z)$ is a Laurent polynomial. The local spline of order p is called quasiinterpolatory if it restores polynomials of degree $p-1$.

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

To predict the odd samples we use the values at the midpoints of the splines, which quasiinterpolate the even samples. If $\mathbf{w}^p$ is the sequence defined in (3.10), then the FIR prediction filter, which stems from a local quasiinterpolatory spline of order p, is

$$U_q^p(z) \triangleq \Gamma(z)W^p(z). \tag{3.18}$$

Explicit formulas for the construction of quasiinterpolatory splines were given in [29]. Properties of quasiinterpolatory splines and, in particular, their approximation accuracy were studied in [36]. In this paper, we are interested in splines of odd order $p = 2r-1$. There are many FIR filters, which generate quasiinterpolatory splines, but there is only one filter of minimal length $2r-1$ for each order $p = 2r-1$.

Proposition 3.3 [29]: A quasiinterpolatory spline of order $p = 2r-1$ can be constructed by filtering (3.17) with filters Γ of length no less than $2r-1$. There exists a unique filter Γ_m of length $2r-1$, which produces the minimal quasiinterpolatory spline $S_m^{2r-1}(x)$. Its transfer function is

$$\Gamma_m(z) = 1 + \sum_{k=1}^{r-1} \gamma_k (z^{-1}-2+z)^k.$$

The coefficients γ_k can be derived from the generating function

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

Proposition 3.4 [36]: If $f(x)$ is a polynomial of degree $2r$ and $\mathbf{f} = \{f(2k)\}$, then the following relation holds for the minimal quasiinterpolatory spline S_m^{2r-1} of order $2r-1$ (degree $2r-2$) for $x = (2k+1+\tau), \tau \in [0,1]$:

$$S_m^{2r-1}(x) = f(x) - 2^{2r-1} f^{(2r-1)}(x) \frac{b_{2r-1}(\tau)}{(2r-1)!} + 2^{2r} f^{(2r)} \left(\frac{(2r-1)b_{2r}(\tau)}{(2r)!} - \gamma_r \right)$$

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

We recall that the values $b_s \triangleq b_s(0)$ are called Bernoulli numbers. If $s > 1$ is odd, then $b_s = 0$. Hence, we have

Corollary 3.1: If $f(x)$ is a polynomial of degree $2r-1$ and S_m^{2r-1} is a minimal quasiinterpolatory spline of order $2r-1$ (degree $2r-2$) then $S_m^{2r-1}(2k+1) \equiv f(2k+1), k \in \mathbb{Z}$. If $f(x)$ is a polynomial of degree $2r$ and $\mathbf{f} = \{f(2k)\}$ then

$$S_m^{2r-1}(2k+1) = f(2k+1) + f^{(2r)} A^r$$
$$A^r \triangleq \frac{(2r-1)b_{2r}}{(2r)!} - \gamma_r. \tag{3.19}$$

Equation (3.19) implies the super-convergence property. This property is similar to the property of interpolatory splines. The minimal quasiinterpolatory spline of order $2r-1$ (degree $2r-2$) restores polynomials of degree $2r-1$ at the midpoints between nodes. The asymptotic representation (3.19) allows one to increase the approximation accuracy of the minimal spline at the midpoints. This is achieved via an upgrade of the filter Γ_m. We call the corresponding quasiinterpolatory spline *extended* and denote it S_e^{2r+1}.

Proposition 3.5: If the coefficients of a spline S_e^{2r+1} of order $2r-1$ are derived according to (3.17) using the filter $\Gamma_e(z) = \Gamma_m(z) - A^r(z^{-1}-2+z)^r$ of length $2r+1$, then the spline restores polynomials of degree $2r+1$ at the midpoints between nodes.

4) Examples:

a) Example 1: Minimal quadratic spline: The filters are

$$\begin{aligned}\Gamma_m(z) &= 1 - \frac{z^{-1} - 2 + z}{8}\\ U_m^3(z) &= \frac{-z^{-1} + 9 + 9z - z^2}{16}\\ 1 - z^{-1}U_m^3(z^2) &= \frac{(z^{-1} - 2 + z)^2(z^{-1} + 4 + z)}{16}.\end{aligned} \tag{3.20}$$

The corresponding analysis wavelet has four vanishing moments.

b) Example 2: Extended quadratic spline

$$\begin{aligned}\Gamma_e(z) &= \Gamma_m(z) + \frac{3}{128}(z^{-1} - 2 + z)^2\\ U_e^3(z) &= \frac{3(z^{-2} + z^3) - 25(z^{-1} + z^2) + 150(1 + z)}{256}\\ 1 - z^{-1}U_e^3(z^2) &= \frac{(z^{-1} - 2 + z)^3 P_e^3(z)}{256},\\ P_e^3(z) &\triangleq 3z^{-2} + 18z^{-1} + 38 + 18z + 3z^2.\end{aligned} \tag{3.21}$$

The corresponding analysis wavelet has six vanishing moments.

Remark: Donoho [14] presented a scheme where an odd sample is predicted by the value at the central point of a polynomial of odd degree which interpolates adjacent even samples. One can observe that our filter U_m^3 (3.20) coincides with the filter derived by Donoho's scheme using a cubic interpolatory polynomial. The filter U_e^3 (3.21) coincides with the filter derived using an interpolatory polynomial of fifth degree. Note that in Donoho's construction, the *update* step does not exist.

c) Example 3: Minimal spline of fifth order (fourth degree)

$$\begin{aligned}\Gamma_m(z) &= 1 - \frac{5}{24}(z^{-1} - 2 + z) + \frac{47}{1152}(z^{-1} - 2 + z)^2\\ U_m^5(z) &= \frac{1}{27\,648}(47(z^{-3} + z^4) + 89(z^{-2} + z^3)\\ &\quad - 2277(z^{-1} + z^2) + 15\,965(1 + z))\\ 1 - z^{-1}U_m^5(z^2) &= \frac{-(z^{-1} - 2 + z)^3 P_m^5(z)}{27\,648}\\ P_m^5(z) &\triangleq 47(z^{-4} + z^4) + 282(z^{-3} + z^3)\\ &\quad + 1076(z^{-2} + z^2) + 3166\alpha + 5414.\end{aligned} \tag{3.22}$$

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

C. Filters for the Update Step

In Sections II-A and III-B, we presented a family of filters U for the *predict* step, which was derived from splines of various types. To complete the construction of the transform, we need to define the filter V for the *update* part. The fact that any choice of these filters retains the perfect reconstruction property of the transform is a great advantage of the lifting scheme. Proposition 2.1 indicates that, in order to produce synthesis and analysis filters with similar properties, it is advisable to choose $V(z) = \breve{U}(z)/z$, where $\breve{U}$ is one of the filters U presented above. In particular, the filter $\breve{U}$ may coincide with the filter U, which is used for the prediction.

IV. Application of the Transforms to Image Compression

We carried out a series of experiments to evaluate the suitability of the wavelet transforms that are based on the constructed filters for the compression of still images. The experiments reveal that the performance of these transforms is comparable to the performance of the 9/7 Daubechies transform in the quality of the reconstructed image and computational complexity. Moreover, due to the diversity of the devised transforms, one can choose a transform, which is most adaptable to a given class of images or to a specific image.

A. Employed Wavelet Transforms

We explore various combinations of filters for *prediction* and *update* steps in the lifting methodology to obtain different transforms. We describe here several transforms, which use the following five filters:

$$\begin{aligned}F_1(z) &\triangleq U_i^3(z) = 4\alpha\frac{1 + z}{(1 + \alpha z)(1 + \alpha z^{-1})}\\ F_2(z) &\triangleq U_m^3(z) = \frac{-z^{-1} + 9 + 9z - z^2}{16}\\ F_3(z) &\triangleq U_d^6(z) = \frac{1}{18}\frac{(z + 14 + z^{-1})(1 + z)}{(1 + z^{-1}/3)(1 + z/3)}\\ F_4(z) &\triangleq U_e^3(z)\\ &= \frac{1}{256}(3z^{-2} - 25z^{-1} + 150 + 150z - 25z^2 + 3z^3)\\ F_5(z) &\triangleq U_d^8(z) = \frac{8(1 + z)(z^{-1} + 6 + z)}{z^{-2} + 28z^{-1} + 70 + 28z + z^2}.\end{aligned}$$

We recall that the IIR filter F_1 is derived from a quadratic interpolatory spline [see (3.13)], the FIR filter F_2 is derived from a quadratic minimal quasiinterpolatory spline [see (3.20)], the IIR filter F_3 is derived from interpolatory discrete spline of sixth degree [see (3.5)], the FIR filter F_4 is derived from a quadratic extended quasiinterpolatory spline [see (3.21)], and the IIR filter F_5 is derived from an interpolatory discrete spline of eighth degree [see (3.9)].

d) Remark: The performance of the transforms that use the filter U_i^5 is almost identical to the performance of the transforms that use the filter U_d^6, but the computational complexity of the former is higher. Therefore, we do not present here the results of the application of the filter U_i^5.

By combining the above filters, we construct a number of transforms, which we label as $\mathbf{P}p\mathbf{U}u$. Here, p means the index of the filter F_k, which is used for the *update* step in the lifting scheme and u is the index of the filter, which is used for the *update* step. For example, **P**1**U**3 designates the transform, which

performs the *predict* step using the filter $F_1(z)$ and the *update* step using the filter $F_3(z)/z$. We denote by **B**9/7 the 9/7 Daubechies's transform.

Proposition 4.1: The synthesis $\varphi(x)$ and the analysis $\tilde{\varphi}(x)$ scaling functions associated with the transforms $\mathbf{P}p\mathbf{U}u$ are continuous and:

- *if the filters F_u and F_p are FIR*, then $\varphi(x)$ and $\tilde{\varphi}(x)$ have compact support;
- *if the filter F_u is IIR and F_p is FIR*, then $\tilde{\varphi}(x)$ decays exponentially as $|x| \to \infty$ and $\varphi(x)$ has compact support;
- *if the filter F_p is IIR and F_u is FIR*, then $\tilde{\varphi}(x)$ and $\varphi(x)$ decay exponentially as $|x| \to \infty$;
- *if the filters F_u and F_p are IIR* then $\varphi(x)$ and $\tilde{\varphi}(x)$ decay exponentially as $|x| \to \infty$.

We summarize in Table II the properties of the transforms, which we employed in our image compression experiments. The following abbreviations are used in the table.

VmA Number of vanishing moments of the analysis high-frequency wavelet.
VmS Number of vanishing moments of the synthesis high-frequency wavelet.
RA Regularity of the analysis scaling function $\tilde{\varphi}$. C^k means that $\tilde{\varphi}$ is continuous together with its k derivatives.
RS Regularity of the synthesis scaling function φ.
Add Number of additions per pixel in the implementation of one of the transforms (cascade mode for IIR filters)
Mult The same for multiplications.
LFA Length of the analysis filter $\tilde{H}$.
LFS Length of the synthesis filter H.

Comments on Table II

- All the presented transforms are carried out through lifting steps. Filtering with IIR filters is implemented in either cascade or parallel mode with special treatment of the boundaries of the images (Appendix I—Section I). Parallel filtering on several processors can reduce the CPU time of the implementation.
- Factorization of the **B**9/7 transform, as suggested in [2] and [13], speeds up the computation. The number of operations in Table II is computed with respect to this factorization algorithm. The 2-D wavelet transforms are applied separately to the rows and columns of the 2-D array. However, for lifting implementation of 2-D **P**2**U**2, **P**4**U**4, and **B**9/7 transforms, which use FIR filters, it is possible to merge row and column operations. This merge reduces the computational cost of the implementation ([2, App. II, Sec. II]).
- It is clear from the table that even by using a cascade computation, the cost of the implementation of the transforms **P**1**U**1, **P**3**U**3, and **P**1**U**3 is close to the cost of the application of the transform **B**9/7.
- It is well known that the scaling function of a wavelet transform is the limit function of a corresponding subdivision scheme [12], provided that the Kroneker delta is the initial data. Therefore, we base our investigation of the regularity of the scaling functions of the devised transforms on the technique that was developed by Dyn *et al.* [15], [16] for the analysis of the regularity of the limit function of subdivision schemes. Unlike [15] and [16], we deal with the IIR filters. Therefore, a proper modification of the technique is required. This investigation will be reported in a different paper.

TABLE II
PROPERTIES OF THE TRANSFORMS EMPLOYED IN THE IMAGE COMPRESSION EXPERIMENTS

Trans.	VmA	VmS	RA	RS	Add	Mult	LFA	LFS
P1**U**1	4	4	C^1	C^2	8	6	∞	∞
P2**U**2	4	4	C^0	C^1	8	4	11	5
P3**U**3	6	6	C^2	C^4	12	6	∞	∞
P4**U**4	6	6	C^1	C^2	12	6	15	7
P1**U**3	4	6	C^1	C^4	10	7	∞	∞
P2**U**4	4	6	C^0	C^2	10	5	13	5
P5**U**5	8	8	C^4	C^5	16	12	∞	∞
B7/9	4	4	C^0	C^1	8	4	9	7

In the next six figures, we display a few filters and wavelets associated with the presented transforms. All the figures are identically organized. Each of them contains four pictures. If these pictures are counted from left to right, then the first column displays the frequency responses of low-pass H and high-pass G synthesis filters, the second column displays the frequency responses of low-pass $\tilde{H}$ and high-pass $\tilde{G}$ analysis filters, the third column displays the synthesis scaling function φ and wavelet ψ, the fourth column displays the analysis scaling function $\tilde{\varphi}$ and wavelet $\tilde{\psi}$. Fig. 1 illustrates the transform **P**1**U**1, Fig. 2 illustrates the transform **P**2**U**2, Fig. 3 illustrates the transform **P**3**U**3, Fig. 4 illustrates the transform **P**4**U**4, Fig. 5 illustrates the transform **P**5**U**5, and Fig. 6 illustrates the transform **B**9/7.

Comments on Figs. 1–6: We observe that, unlike **B**9/7, all the devised low-pass synthesis filters and high-pass analysis filters have flat frequency responses. In particular, it is true for the IIR filters. The rest of the filters are flat up to a small bump near the cutoff. The filters of the transform **P**5**U**5 have the flattest frequency response and the steepest cut-off. Its scaling functions are most regular among the presented transforms.

V. IMAGE COMPRESSION RESULTS

The devised transforms were applied to achieve compression for a number of still benchmark images. Their performance is comparable with the popular **B**9/7. In this section we apply the transforms to four images presented in Figs. 7 and 8. These are 512×512 8 bit per pixel (8 bpp) images. The following experiments were conducted.

1) Each image was decomposed up to six scales with the wavelet transform using the **B**9/7 transforms and the transforms $\mathbf{P}p\mathbf{U}u, p, u = 1, \ldots, 5$, that are listed in Table II.
2) The transforms' coefficients were coded using the SPIHT algorithm [27]. This algorithm enables us to achieve an exact predetermined compression ratio. We coded the coefficients with different compression ratios (CR): 1:10 (0.8 bpp), 1:20 (0.4 bpp), 1:30 (4/15 bpp), 1:40 (0.2 bpp), and 1:50 (4/25 bpp).

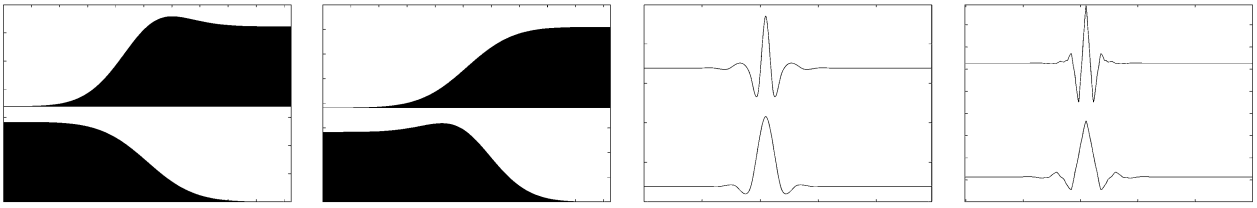


Fig. 1. Filters and wavelets associated with the **P**1**U**1 transform.

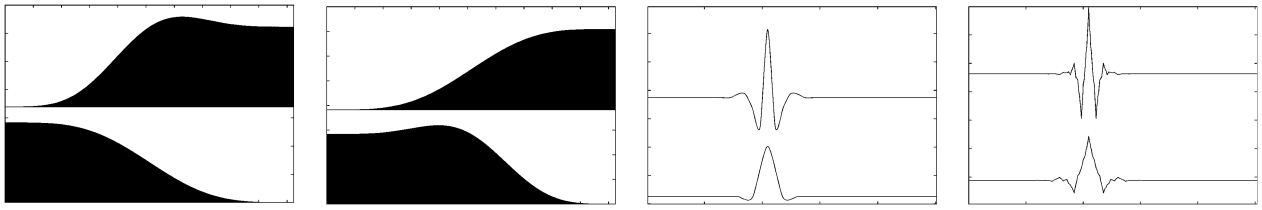


Fig. 2. Filters and wavelets associated with the **P**2**U**2 transform.

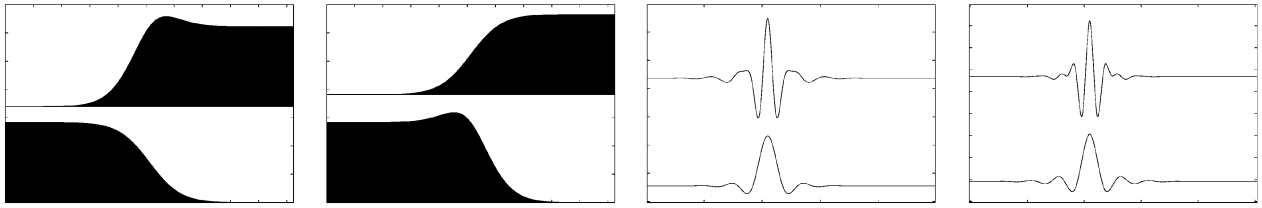


Fig. 3. Filters and wavelets associated with the **P**3**U**3 transform.

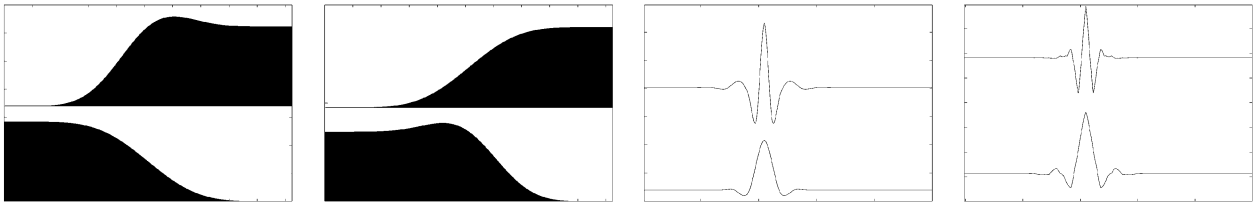


Fig. 4. Filters and wavelets associated with the **P**4**U**4 transform.

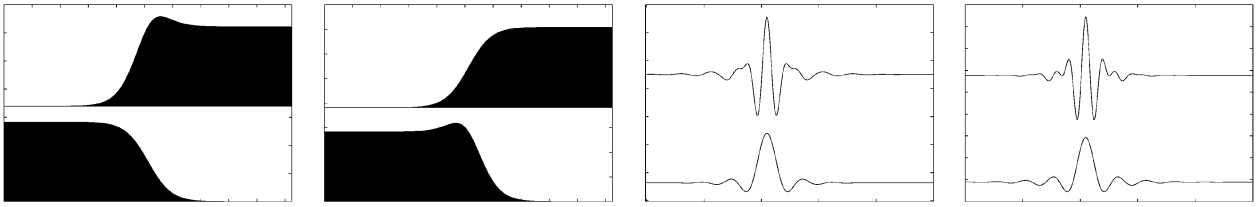


Fig. 5. Filters and wavelets associated with the **P**5**U**5 transform.

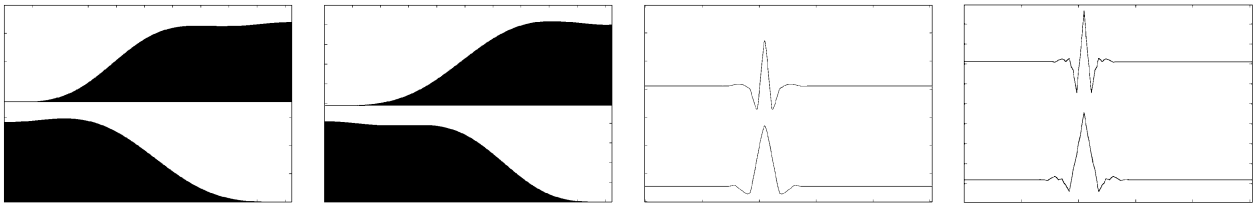


Fig. 6. Filters and wavelets associated with the **B**9/7 transform.

Fig. 7. Left: "Lena." Right: "Barbara."

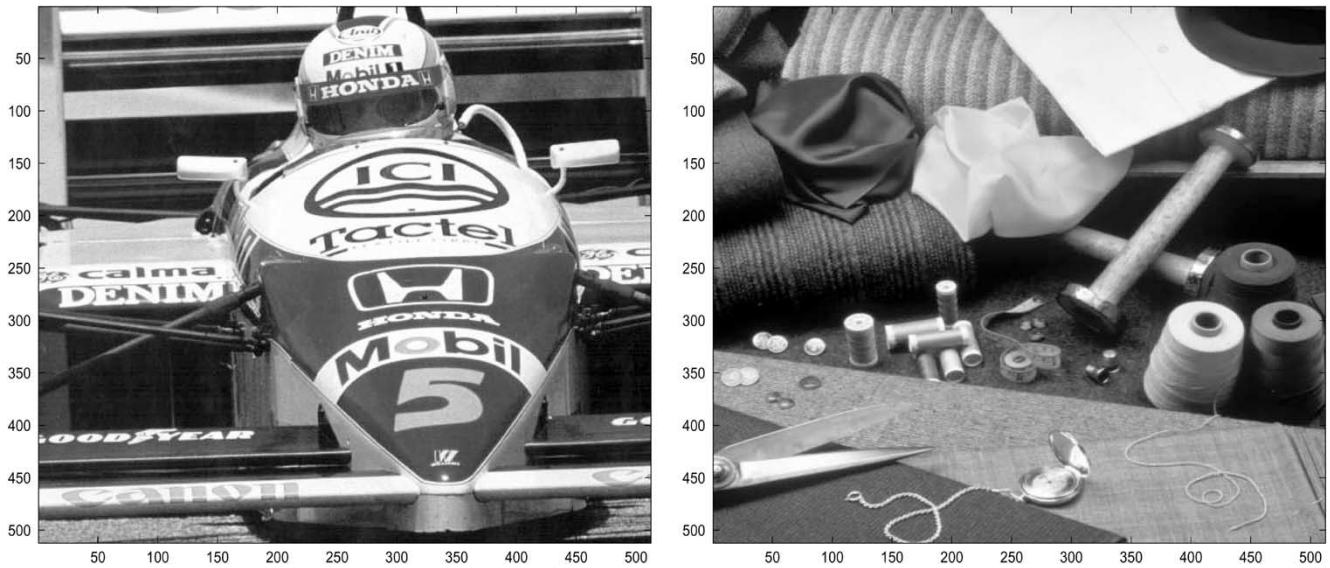


Fig. 8. Left: "Car." Right: "Fabrics."

3) The reconstructed image was compared with the original image and the peak signal-to-noise ratio (PSNR) in decibels was computed

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{N255^2}{\sum_{k=1}^{N}(x(k)-\tilde{x}(k))^2}\right)\ \mathrm{dB}. \quad (5.23)$$

Lena: The PSNR values of "Lena" are presented in Table III. All the $\mathbf{P}p\mathbf{U}u$ transforms have a slightly better PSNR than the **B**9/7 transform for most of the compression ratios except when **P**2**U**2 is used at $\mathrm{CR} = 10$. The complexity for both transforms is the same. The transforms **P**1**U**3, **P**4**U**3, and **P**3**U**3 perform equally. Their synthesis scaling functions have four continuous derivatives. We can see that if we go beyond or below the smoothness of these transforms (**P**1**U**1 and **P**5**U**5) then the performance is degraded. In Fig. 9 we display a zoom from the reconstructed "Lena" (compressed to 1:40). In the compression we used the **B**9/7 and **P**4**U**3 transforms. Note that, although both portions are almost indistinguishable, there are details which are better restored by the **P**4**U**3 transform.

Barbara: The PSNR values of "Barbara" are presented in Table IV. We observe from Table IV that the transforms **P**3**U**3 and **P**1**U**3, where the synthesis wavelets have six vanishing moments and the scaling functions belong to C^4, perform better than the transforms with four vanishing moments and less regular scaling functions. The **P**5**U**5 transform, where both analysis and synthesis wavelets have eight vanishing moments and the scaling functions belong to C^4 and C^5, respectively, produces the best results. In Fig. 10, we display a zoom from the reconstructed "Barbara" (compressed to 1:40). In the compression, we used the **B**9/7 and **P**5**U**5 transforms. Note that the texture of the tablecloth and the scarf is better revealed with the **P**5**U**5 while the leg of the table is displayed more accurately with the **B**9/7 transform.

TABLE III
PSNR OF THE "LENA" IMAGE

CR	**B**9/7	**P**1**U**1	**P**2**U**2	**P**3**U**3	**P**4**U**4	**P**1**U**3	**P**4**U**2	**P**4**U**3	**P**5**U**5
10	37.70	37.82	37.67	37.85	37.79	37.85	37.71	37.84	37.80
20	34.53	34.71	34.70	34.72	34.68	34.75	34.63	34.73	34.67
30	32.33	32.62	32.56	32.69	32.59	32.65	32.57	32.67	32.75
40	31.42	31.63	31.52	31.67	31.61	31.67	31.54	31.68	31.60
50	30.70	30.91	30.82	30.93	30.90	30.95	30.85	30.96	30.87

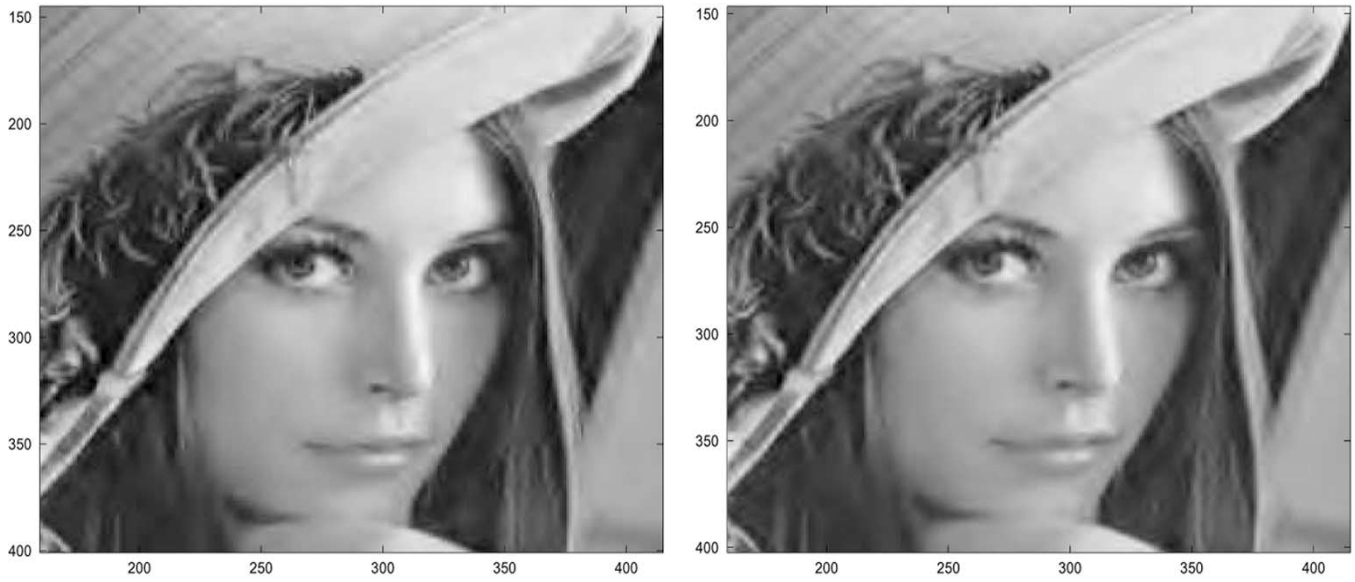


Fig. 9. Fragment of "Lena" reconstructed from 40:1 compression. Left: the **B**7/9 transform was used: $\mathrm{PSNR} = 31.42$. Right: **P**4**U**3 transform was used: $\mathrm{PSNR} = 31.68$.

TABLE IV
PSNR OF THE "BARBARA" IMAGE

CR	**B**9/7	**P**1**U**1	**P**2**U**2	**P**3**U**3	**P**4**U**4	**P**1**U**3	**P**4**U**2	**P**4**U**3	**P**5**U**5
10	33.01	33.39	32.91	33.72	33.32	33.51	33.06	33.46	33.81
20	28.93	29.13	28.90	29.32	29.10	29.24	28.99	29.22	29.39
30	26.99	27.03	26.91	27.33	27.17	27.15	27.04	27.15	27.45
40	25.78	25.74	25.65	25.95	25.74	25.83	25.69	25.78	26.06
50	25.10	25.00	24.86	25.14	24.95	25.05	24.90	25.02	25.23

Car: The PSNR values of "Car" are presented in Table V. As in "Lena," here the $\mathbf{P}p\mathbf{U}u$ transforms, except for the highly regular transform **P**5**U**5, produce a slightly better PSNR than the **B**9/7 transform in any compression ratio. The reason for that may lie in the presence of small details in the image, which are better captured by the wavelets with shorter effective support. We display a zoom from the reconstructed "Car" (compressed to 1:40) in Fig. 11. The **B**9/7 and **P**1**U**3 transforms were used. While the horizontal lines on the left hand side of the fragment are better displayed in the **B**9/7 picture, the **P**1**U**3 transform is slightly advantageous in restoring arcs and letters on the right-hand side.

Fabric: The PSNR values of "Fabric"are presented in Table VI. As in the above examples, the transform **P**1**U**1 produces a better PSNR than the **B**9/7 transform for any compression ratio. The same is true for more complicated transforms such as **P**1**U**3 and **P**4**U**3. The latter transform produces the best PSNR among all the listed transforms. In Fig. 12, we display a zoom from the reconstructed "Fabric" (compressed to 1:40). In the compression we used the transforms **B**9/7 and **P**4**U**3. Although it is difficult to destinguish between them, it seems that **P**4**U**3 produces a superior visual display such as the scissors and the texture of the tablecloth.

The **P**3**U**3 transform, which uses filters that are derived from discrete spline of sixth order, produces a high PSNR on all test images. But there are singular cases when it was outperformed. The **P**1**U**1 transform combines low computational costs with a good performance. **P**1**U**1 transform uses filters that were derived from the quadratic interpolatory spline. The performance of the transforms **P**1**U**3 and **P**4**U**3 suggests that it in some cases it is better to use different filters for the *predict* and *update* steps. The transform **P**5**U**5, which uses filters derived from the discrete spline of eighth order produces good results for the compression of "Barbara". Therefore, it justifies the usage of highly regular wavelets with a high number of vanishing moments.

VI. CONCLUSION

In this paper, we proposed an efficient technique that generates a wide range of new biorthogonal symmetric wavelet transforms.

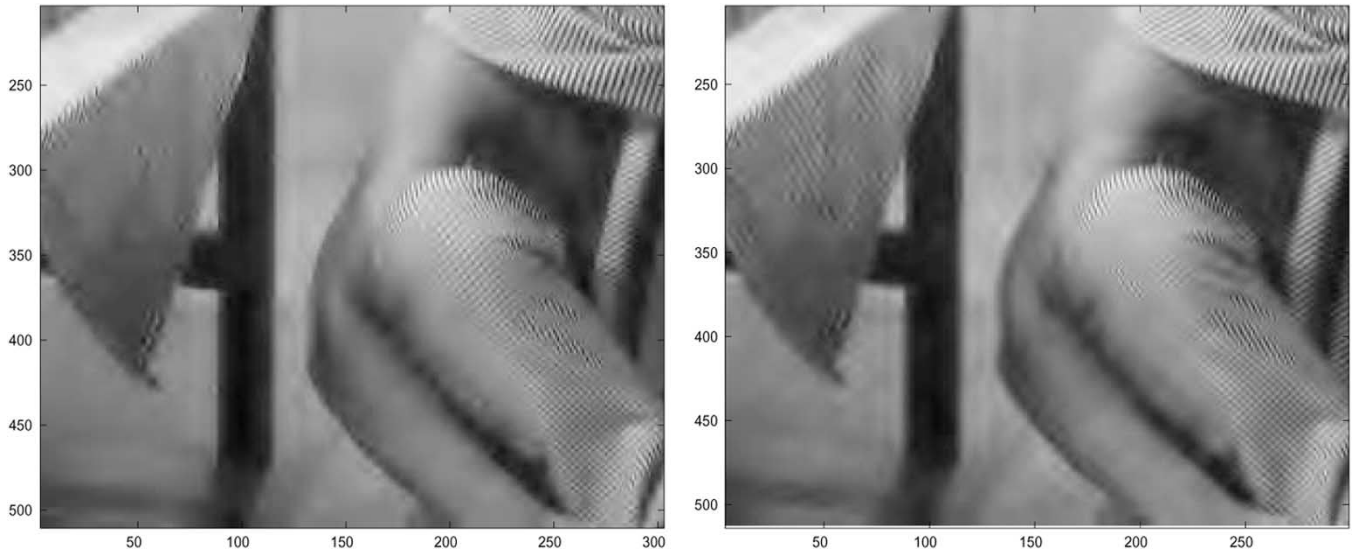


Fig. 10. Fragment from "Barbara" reconstructed from 40:1 compression. Left: **B**9/7 filter was used: $\mathrm{PSNR} = 25.78$. Right: **P**5**U**5 transform was used: $\mathrm{PSNR} = 26.06$.

TABLE V
PSNR OF THE "CAR" IMAGE

CR	**B**9/7	**P**1**U**1	**P**2**U**2	**P**3**U**3	**P**4**U**4	**P**1**U**3	**P**4**U**2	**P**4**U**3	**P**5**U**5
10	32.57	32.63	32.72	32.52	32.59	32.66	32.69	32.64	32.35
20	28.38	28.53	28.46	28.46	28.47	28.53	28.49	28.51	28.32
30	26.78	26.99	26.89	26.92	26.94	27.02	26.95	27.0	26.83
40	25.05	25.19	25.11	25.15	25.17	25.22	25.16	25.20	25.12
50	24.40	24.52	24.43	24.46	24.49	24.56	24.48	24.55	24.37

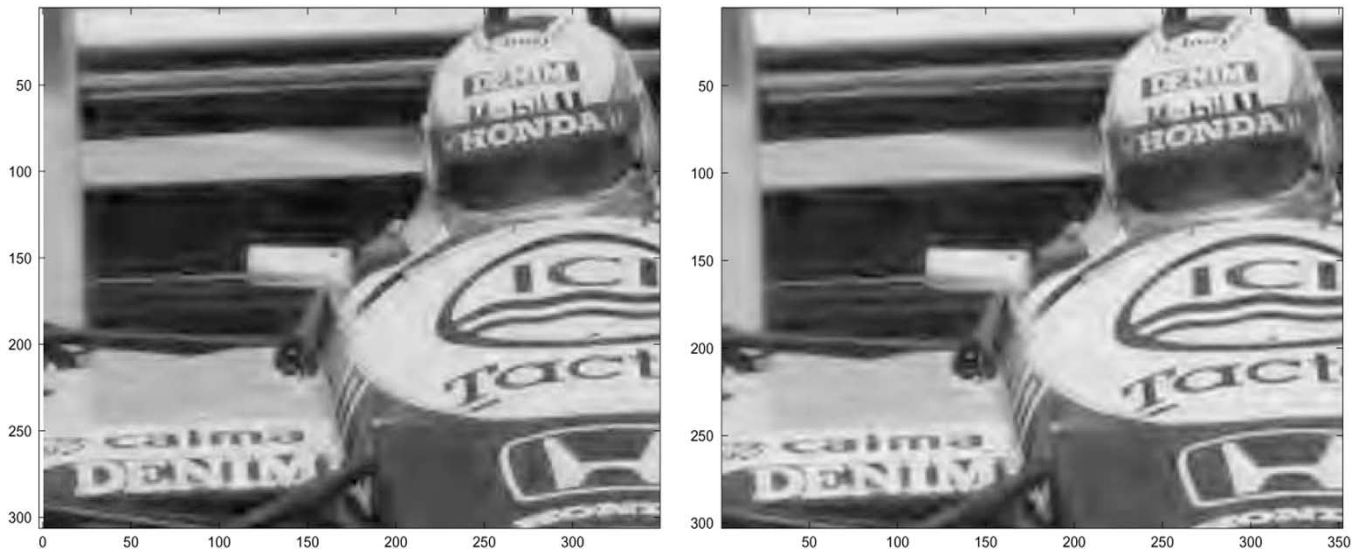


Fig. 11. Fragment of "Car" reconstructed from 40:1 compression. Left: B9/7 filter was applied: $\mathrm{PSNR} = 25.05$. Right: **P**1**U**3 transform: $\mathrm{PSNR} = 25.22$.

This technique is based on the usage of discrete and polynomial interpolatory and quasiinterpolatory splines for the design of filters for the *predict* and *update* operations in lifting schemes of the wavelet transform. These are the linear phase filters which have flat frequency responses. By combining different designed filters for the *predict* and *update* steps, we can devise practically unlimited forms of wavelets which have a predetermined number of vanishing moments that are as smooth as required. When transforms that are based on splines of higher orders are implemented, it is advisable to switch from time-domain to frequency-domain implementation as described in [4] and [5]. Then, an increase in the number of vanishing moments and the regularity of wavelets does not affect the computational cost of the implementation.

We explored the applicability of the newly designed transforms to still image compression. The new transforms and the biorthogonal 9/7 transform were incorporated into SPIHT in order to compare their performances. The performance (quality

TABLE VI
PSNR OF THE "FABRIC" IMAGE

CR	**B**9/7	**P**1**U**1	**P**2**U**2	**P**3**U**3	**P**4**U**4	**P**1**U**3	**P**4**U**2	**P**4**U**3	**P**5**U**5
10	34.95	34.96	34.86	34.97	34.96	34.99	34.89	34.99	34.92
20	31.53	31.53	31.44	31.53	31.51	31.55	31.47	31.57	31.47
30	29.62	29.79	29.73	29.75	29.77	29.81	29.77	29.82	29.83
40	28.90	29.00	28.93	29.02	28.98	29.02	28.97	29.04	28.99
50	28.41	28.52	28.44	28.52	28.50	28.54	28.46	28.55	28.47

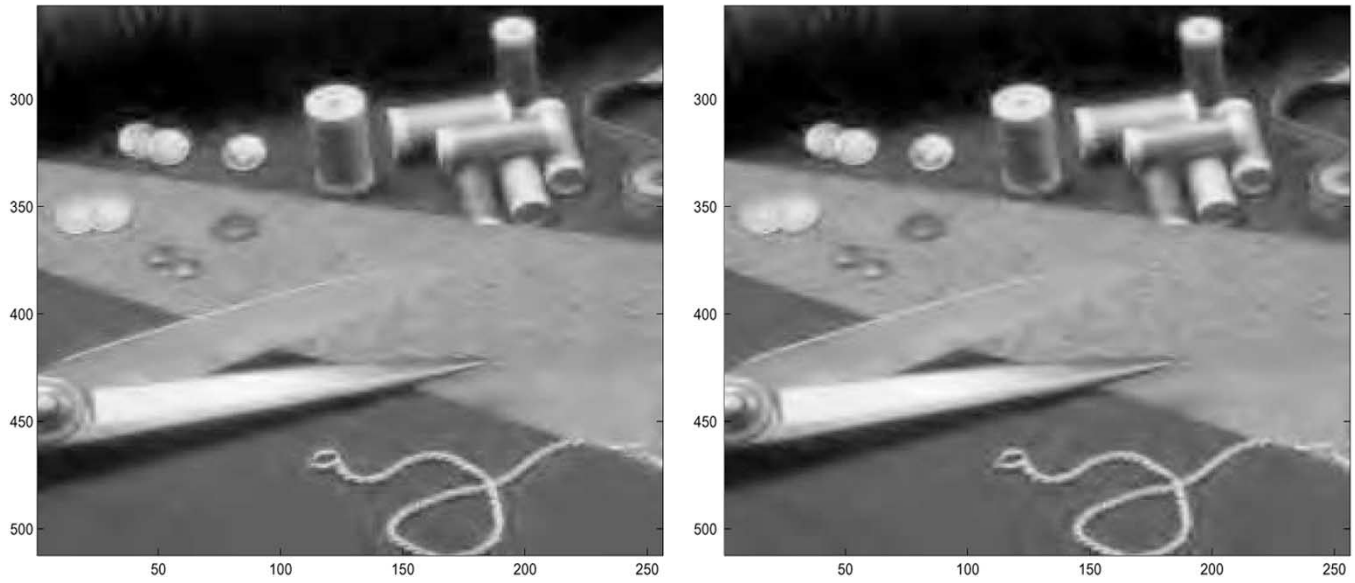


Fig. 12. Fragment of "Fabric" reconstructed from 40:1 compression. Left: **B**9/7 transform was applied: $\mathrm{PSNR} = 28.90$. Right: **P**4**U**3 transform: $\mathrm{PSNR} = 29.04$.

and computational cost) of the presented transforms proved to be comparable with the 9/7 transform. Most of the filters that are employed in the transforms are IIR.

A number of authors [1], [37] studied the performance of families of wavelet transforms that use FIR filters for image compression. The most comprehensive investigation on this topic was reported in [20]. The results of these studies as well as the results of our experiments suggest that it is difficult to single out properties of the transforms that are most valuable for image compression. We believe that a trade-off between properties such as the number of vanishing moments of analysis wavelets, the regularity of synthesis wavelets, the flatness of the frequency response and the steepness of its transition band, and the spatial localization of the wavelets, is the key to achieve better compression. This is exemplified by the popular **B**9/7 transform as well as the transforms **P**3**U**3 and **P**1**U**1.

Analysis of the regularity of the devised wavelets will be reported in an another paper. In addition, we plan to design a coding scheme which can utilize specific properties of the newly constructed transforms. These transforms will be tested on video sequences and seismic data.

APPENDIX I
IMPLEMENTATION OF RECURSIVE FILTERS

Application of wavelet transforms to finite-length signals and, in particular, to images, requires an extension of signals beyond their boundaries [8]. The extension is even more important in our scheme since we implicitly assumed in our construction that the signals are defined on infinite intervals. When the filter banks are symmetric, the **HH** extension in the terminology of [8] is most efficient. It means that the signal $\mathbf{x} = \{x(k)\}, k = 1, \dots N$, is symmetrically extended with the repetition of boundary samples through both ends of the interval. Namely, $x(0) \triangleq x(1), x(-1) \triangleq x(2), \dots x(-k) \triangleq x(k+1)$ and $x(N+1) \triangleq x(N)$, $x(N+2) \triangleq x(N-1), \dots x(N+k) \triangleq x(N-k+1)$. This results in periodization of the signal with period $2N$. This extended signal is denoted by $\tilde{\mathbf{x}}$ and its z transform is $\tilde{X}(z)$.

Recursive filtering of finite-length signals requires additional treatment of the boundaries. Each of the recursive filters $F_1(z) \triangleq U_i^3(z)$ (3.13), $F_3(z) \triangleq U_d^6(z$, and $F_5(z) \triangleq U_d^8(z)$ (3.9) comprises a block of type

$$F(z) = \frac{1+z}{(1+\alpha z)(1+\alpha z^{-1})}. \tag{1.24}$$

We describe the application of the filter F to a finite-length signal $\mathbf{x}$. We begin with the parallel mode. Equation (1.24) is equivalent to

$$F(z) = \frac{1}{1+\alpha}\left(\frac{1}{1+\alpha z^{-1}} + \frac{z}{1+\alpha z}\right).$$

Denote

$$Y_1(z) = \frac{\tilde{X}(z)}{1+\alpha z^{-1}} = \sum_{n=0}^{\infty}(-\alpha)^n z^{-n}\tilde{X}(z)$$

$$Y_2(z) = \frac{z\tilde{X}(z)}{1+\alpha z} = \sum_{n=0}^{\infty}(-\alpha)^n z^{n+1}\tilde{X}(z). \tag{1.25}$$

Then, the signals $y_1(k)$ and $y_2(k)$ can be computed in time domain in either recursive or nonrecursive modes

$$\begin{aligned} y_1(k) &= \tilde{x}(k) - \alpha y_1(k-1) \\ y_2(k) &= \tilde{x}(k+1) - \alpha y_2(k+1) \end{aligned} \tag{1.26}$$

$$\begin{aligned} \Longleftrightarrow y_1(k) &= \tilde{x}(k) + \sum_{n=1}^{\infty} (-\alpha)^n \tilde{x}(k-n) \\ y_2(k) &= \tilde{x}(k+1) + \sum_{n=1}^{\infty} (-\alpha)^n \tilde{x}(k+n+1). \end{aligned} \tag{1.27}$$

We can use (1.26) for the computation of $y_1(k)$ and $y_2(k)$ on the interval $k = 1, 2, \ldots, N$ provided we know $y_1(0)$ and $y_2(N+1)$, respectively. To evaluate these samples, we employ the nonrecursive (1.27). We have

$$\begin{aligned} y_1(0) &= \tilde{x}(0) + \sum_{n=1}^{\infty} (-\alpha)^n \tilde{x}(-n) \\ &\approx x(1) + \sum_{n=2}^{d} (-\alpha)^n x(n) \end{aligned} \tag{1.28}$$

$$\begin{aligned} y_2(N+1) &= \tilde{x}(N+2) + \sum_{n=1}^{\infty} (-\alpha)^n \tilde{x}(N+n+2) \\ &\approx x(N-1) + \sum_{n=2}^{d} (-\alpha)^n x(N-n) \end{aligned} \tag{1.29}$$

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

Recursive parallel filtering of the finite-length signal $\mathbf{x}$ by the filter F is implemented as follows.

1) Evaluate $y_1(0)$ from (1.28) and $y_2(N+1)$ from (1.29).
2) Calculate $y_1(k) = x(k) - \alpha y_1(k-1), k = 1, \ldots N$ and $y_2(k) = x(k+1) - \alpha y_2(k+1), k = N, \ldots 1$.
3) The result of filtering is $y(k) = (y_1(k) + y_2(k))/(1+\alpha), k = 1 \ldots N$.

Equations (1.27) and (1.29) imply that $y_2(N) = y_1(N)$. Hence, it follows that

$$y(N) = \frac{y_1(N) + y_2(N)}{1+\alpha} = \frac{2y_1(N)}{1+\alpha}. \tag{1.30}$$

The cascade algorithm has the following form.

1) Evaluate $y_1(0)$ from (1.28).
2) Calculate $y_1(k) = x(k) - \alpha y_1(k-1), k = 2, \ldots N$.
3) Evaluate $y(N)$ from (1.30).
4) Calculate $y(k) = y_1(k) + y_1(k+1) - \alpha y(k+1), k = N-1, \ldots 1$. Note that the depth of the initialization does not affect the perfect reconstruction property of the transforms since in lifting schemes the reconstruction steps are the reversed decomposition steps. But the results of lossy compression deteriorate if this depth is insufficient.

APPENDIX II
2-D IMPLEMENTATION OF TRANSFORMS WITH FIR FILTERS

The 2-D array $\mathbf{x} \triangleq \{x(n,m)\}, n, m = 0 \ldots N-1$ is to be transformed into four subarrays: $\mathbf{x} \rightarrow \mathbf{ss} \cup \mathbf{sd} \cup \mathbf{ds} \cup \mathbf{dd}$, where $\mathbf{ss}$ is the smoothed low-frequency section in horizontal and vertical directions, $\mathbf{sd}$ is low frequency in the vertical direction and high frequency in horizontal direction, $\mathbf{ds}$ is low frequency in the horizontal direction and high frequency in the vertical direction, and $\mathbf{dd}$ is the high-frequency section of details in both the horizontal and vertical directions. Conventionally, the 2-D transform of $\mathbf{x}$ is implemented in a separable way (tensor product). In [2], a direct 2-D lifting implementation of the transformB9/7 is presented. We describe the 2-D lifting implementation of the transform T_2^2.

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

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

T_2^2 Transform.

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

Decomposition:

a) *Split:* $\mathbf{x} = \mathbf{ee} \cup \mathbf{eo} \cup \mathbf{oe} \cup \mathbf{oo}$.
b) *Predict:*
   i) $\mathbf{de} = \mathbf{oe} + \alpha(\mathbf{ee} + \mathbf{U_1} \cdot \mathbf{ee}) + \beta(\mathbf{D_1} \cdot \mathbf{ee} + \mathbf{U_2} \cdot \mathbf{ee})$.
   ii) $\mathbf{dd} = \mathbf{oo} + \alpha(\mathbf{de} + \mathbf{L_1} \cdot \mathbf{de} + \mathbf{eo} + \mathbf{U_1} \cdot \mathbf{eo}) + \beta(\mathbf{R_1} \cdot \mathbf{de} + \mathbf{L_2} \cdot \mathbf{de} + \mathbf{D_1} \cdot \mathbf{eo} + \mathbf{U_2} \cdot \mathbf{eo})$.
c) *Update:*
   i) $\mathbf{sd} = \mathbf{eo} + \alpha(\mathbf{ee} + \mathbf{L_1} \cdot \mathbf{ee}) + \beta(\mathbf{R_1} \cdot \mathbf{ee} + \mathbf{L_2} \cdot \mathbf{ee}) + \gamma(\mathbf{dd} + \mathbf{D_1} \cdot \mathbf{dd}) + \delta(\mathbf{U_1} \cdot \mathbf{dd} + \mathbf{D_2} \cdot \mathbf{dd})$.
   ii) $\mathbf{ss} = \mathbf{ee} + \gamma(\mathbf{de} + \mathbf{D_1} \cdot \mathbf{de} + \mathbf{sd} + \mathbf{R_1} \cdot \mathbf{sd}) + \delta(\mathbf{U_1} \cdot \mathbf{de} + \mathbf{D_2} \cdot \mathbf{de} + \mathbf{L_1} \cdot \mathbf{sd} + \mathbf{R_2} \cdot \mathbf{sd})$.
   iii) $\mathbf{ds} = \mathbf{de} + \gamma(\mathbf{dd} + \mathbf{R_1} \cdot \mathbf{dd}) + \delta(\mathbf{L_1} \cdot \mathbf{dd} + \mathbf{R_2} \cdot \mathbf{dd})$.
d) *Normalization:* $\mathbf{ss} = 2\mathbf{ss}, \mathbf{dd} = \mathbf{dd}/2$. Reconstruction of the 2-D array $\mathbf{x}$ from four subarrays of coefficients: $\mathbf{ss} \cup \mathbf{sd} \cup \mathbf{ds} \cup \mathbf{dd}, \mathbf{ss} \rightarrow \mathbf{x}$ is conducted in reverse order.

REFERENCES

[1] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Processing*, vol. 1, pp. 205–220, Feb. 1992.

[2] A. Z. Averbuch, F. Meyer, and J.-O. Stromberg, "Fast adaptive wavelet packet image compression," *IEEE Trans. Image Processing*, vol. 9, pp. 792–800, 2000.

[3] A. Z. Averbuch, A. B. Pevnyi, and V. A. Zheludev, "Butterworth wavelet transforms derived from discrete interpolatory splines: Recursive implementation," *Signal Processing*, vol. 81, pp. 2363–2382, 2001.

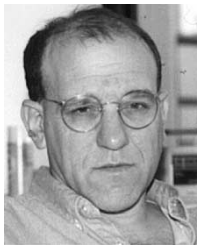
[4] ——, "Biorthogonal Butterworth wavelets derived from discrete interpolatory splines," *IEEE Trans. Signal Processing.*, vol. 49, pp. 2682–2692, Nov. 2001.

[5] A. Averbuch and V. Zheludev, "Construction of biorthogonal discrete wavelet transforms using interpolatory splines," *Appl. Comp. Harmonic Anal.*, vol. 12, pp. 25–56, 2002.

[6] ——, "Splines: A new contribution to wavelet analysis," in *Proc. Conf. Algorithms for Approximation IV*.

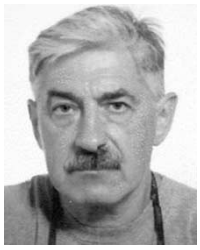
[7] G. Battle, "A block spin construction of ondelettes. Part I. Lemarié functions," *Comm. Math. Phys.*, vol. 110, pp. 601–615, 1987.

[8] C. M. Brislawn, "Classification of nonexpansive symmetric extension transforms for multirate filter banks," *Appl. Comput. Harmonic Anal.*, vol. 3, no. 4, pp. 337–357, 1996. 337–357.
[9] C. K. Chui and J. Z. Wang, "On compactly supported spline wavelets and a duality principle," *Trans. Amer. Math. Soc.*, vol. 330, pp. 903–915, 1992.
[10] R. L. Claypoole Jr., J. M. Davis, W. Sweldens, and R. Baraniuk, "Nonlinear wavelet transforms for image coding via lifting," *IEEE Trans. Signal Processing*.
[11] A. Cohen, I. Daubechies, and J. C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Comm. Pure Appl. Math.*, vol. 45, pp. 485–560, 1992.
[12] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: SIAM, 1992.
[13] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Appl.*, vol. 4, pp. 247–269, 1998.
[14] D. L. Donoho, "Interpolating Wavelet Transform," Dept. Statistics, Stanford Univ., Stanford, CA, 1992. preprint 408.
[15] N. Dyn, J. A. Gregory, and D. Levin, "Analysis of uniform binary subdivision schemes for curve design," *Constr. Approx.*, vol. 7, pp. 127–147, 1991.
[16] N. Dyn, "Analysis of convergence and smoothness by the formalism of Laurent polynomials," in *Tutorials on Multiresolution in Geometric Modeling*, A. Iske, E. Quak, and M. S. Floater, Eds. New York: Springer, 2002, pp. 51–68.
[17] G. Fix and G. Strang, "Fourier analysis of the finite element method in Ritz-Galerkin theory," *Stud. Appl. Math.*, vol. 48, pp. 265–273, 1969.
[18] C. Herley and M. Vetterli, "Wavelets and recursive filter banks," *IEEE Trans. Signal Processing*, vol. 41, pp. 2536–2556, Dec. 1993.
[19] P. G. Lemarié, "Ondelettes à localization exponentielle," *J. de Math. Pure et Appl.*, vol. 67, 1988.
[20] M. Lightstone, E. Majani, and S. K. Mitra, "Low bit-rate design considerations for wavelet-based image coding," *Multidimensional Syst. Signal Processing*, vol. 8, pp. 111–128, 1997.
[21] D. Marpe, G. Heising, A. P. Petukhov, and H. L. Cycon, "Video coding using a bilinear image warping motion model and wavelet-based residual coding," in *Proc. SPIE Conf. Wavelet Applications in Signal and Image Processing VII*, vol. 3813, Denver, CO, July 1999, pp. 401–408.
[22] A. V. Oppenheim and R. W. Shafer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[23] G. Strang and T. Nguen, *Wavelets and Filter Banks*: Wellesley-Cambridge Press, 1996.
[24] A. P. Petukhov, "Biorthogonal wavelet bases with rational masks and their applications," in *Proc. St. Petersburg Mathematical Society*, vol. 7, 1999, pp. 168–193. Russian.
[25] A. B. Pevnyi and V. A. Zheludev, "Construction of wavelet analysis in the space of discrete splines using Zak transform," *J. Fourier Anal. Applic.*, vol. 8, no. 1, pp. 55–77, 2002.
[26] M. Rabbani and R. Joshi, "An overview of the JPEG 2000 still image compression standard," *Signal Processing, Image Commun.*, vol. 17, pp. 3–48, 2002.
[27] A. Said and W. W. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 243–250, June 1996.
[28] I. J. Schoenberg, *Cardinal Spline Interpolation*. Philadelphia, PA: CBMS, SIAM, 1973, vol. 12.
[29] ——, "Contribution to the problem of approximation of equidistant data by analytic functions," *Quart. Appl. Math.*, vol. 4, pp. 45–99, 112–141, 1946.
[30] L. L. Schumaker, *Spline Functions: Basic Theory*. New York: Wiley, 1981.
[31] W. Sweldens, "The lifting scheme: A custom design construction of biorthogonal wavelets," *Appl. Comput. Harm. Anal.*, vol. 3, no. 2, pp. 186–200, 1996.
[32] M. Unser, A. Aldroubi, and M. Eden, "B-spline signal processing: Part I—Theory," *IEEE Trans. Signal Processing*, vol. 41, pp. 821–832, Feb. 1993.
[33] ——, "B-spline signal processing: Part II—Efficient design and applications," *IEEE Trans. Signal Processing*, vol. 41, pp. 834–848, Feb. 1993.
[34] ——, "A family of polynomial spline wavelet transforms," *Signal Processing*, vol. 30, pp. 141–162, 1993.
[35] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[36] V. A. Zheludev, "Local spline approximation on a uniform grid," *U.S.S.R. Comput. Math. & Math. Phys.*, vol. 27, pp. 8–19, 1987.
[37] J. D. Villasenor, B. Belzer, and J. Liao, "Filter evaluation and selection in wavelet compression algorithms," in *Proc. Data Compression Conf.*, Snowbird, UT, 1994, pp. 478–488.
[38] V. A. Zheludev, "Local smoothing splines with a regularizing parameter," *Comput. Math. & Math Phys.*, vol. 31, pp. 193–211, 1991.
[39] ——, "Periodic splines, harmonic analysis, and wavelets," in *Signal and Image Representation in Combined Spaces, Wavelet Anal. Appl.*, Y. Y. Zeevi and R. Coifman, Eds. San Diego, CA: Academic, 1998, vol. 7, pp. 477–509.



Amir 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, in 1983.

In 1967–1971 and 1973–1976, he served in the Israeli Defense Forces. During 1976–1986, 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 (which is now the School of Computer Science), Tel Aviv University, where he is currently a Professor. His research interests include wavelets, signal/image processing, and numerical computation and scientific computing.



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

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

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