Rational Coefficient Dual-Tree Complex Wavelet Transform: Design and Implementation

Adeel Abbas and Trac D. Tran, *Member, IEEE*

***Abstract*—The dual-tree complex wavelet transform (CWT) has recently received significant interest in the wavelet community, owing primarily to its directional selective and near-shift invariant properties. It has been shown that with two separate maximally decimated and dyadic decompositions where filters are offset by a half sample, the resulting CWT wavelet bases form an approximate Hilbert transform pair. In this paper, we present the design, implementation and applications of several families of orthogonal as well as biorthogonal rational-coefficient wavelet filters that satisfy the Hilbert transform pair condition and meet other desirable properties such as high coding gain, good directional sensitivity, and sufficient degree of regularity. The wavelet filters presented here, which confirm to Selesnick's and Kingsbury's design schemes, are designed and implemented directly in the lattice and lifting domain using VLSI-friendly dyadic coefficients. We confirm the fact that rational-coefficient constraint does not impose a significant loss in terms of energy compaction, wavelet smoothness, time-invariance, or directionality. We also propose the time-reversal relationships between the two CWT filter pairs in lattice and lifting domain, thereby facilitating both the design and implementation process. In the end, we present several applications and evaluations to illustrate the performance of the proposed designs.**

***Index Terms*—Complex wavelet transform, dual-tree, Hilbert transform, lattice structure, lifting scheme, rational coefficient.**

I. Introduction

THE discrete wavelet transform (DWT) has been shown to offer solutions to a wide variety of multiresolution image and signal processing applications, including compression, denoising, classification, and many others. During the last couple of years, there has been significant progress in the theory and design of filter banks [1] and especially wavelets [2]. Since wavelets are now part of the JPEG-2000 standard, the popularity for wavelet-based codecs is expected to grow in future. However, there are well known limitations in the conventional wavelet design, for example the lack of directionality, poor shift invariance and lack of phase information. The aim of research in the domain of complex wavelet transform (CWT) is to explore solutions to these limitations, while benefiting from the existing advantages that wavelets have to offer.

Manuscript received October 31, 2006; revised November 15, 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Thierry Blu. This work has been supported in part by the National Science Foundation under Grant CCF-0093262 and CCF-0430869.

A. Abbas is affiliated with Corporate Research, Thomson Multimedia, Princeton, NJ 08540 USA (e-mail: adeel.abbas@thomson.net).

T. D. Tran is with the Department of Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: trac@jhu.edu).

Digital Object Identifier 10.1109/TSP.2008.925898

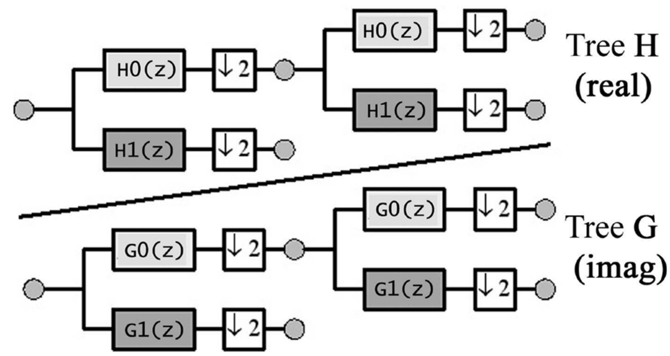

Fig. 1. Kingsbury's dual-tree CWT.

Several authors have proposed that in a formulation where two dyadic wavelet bases form a Hilbert transform pair, CWT can provide answer to some of the aforementioned limitations [3]. Shown in Fig. 1, Kingsbury's pioneering complex dual-tree [4] and [5] has received considerable interest. As shown, two sets of wavelet filters are used. The $\{H_0(z), H_1(z)\}$ pair forms a dyadic tree that implements the real part of the transform. $\{G_0(z), G_1(z)\}$ is another analysis filter pair that generates the imaginary part of the transform. The filter pairs considered here, also referred to as quadrature mirror filter (QMF) pairs, can be orthogonal or biorthogonal, are real-valued and capable of perfect reconstruction (PR). A detailed overview of the dual-tree formulation is available in [6].

We will denote the wavelets associated with real and imaginary filter banks as $\psi_h(t)$ and $\psi_g(t)$ with Fourier transforms $\Psi_h(\omega)$ and $\Psi_g(\omega)$. It has been shown in [3] that if filters in both trees can be made to be offset by half-sample, the wavelets resulting from the filter pair satisfy Hilbert transform condition. In other words, if we have

$$G_0(\omega) \simeq H_0(\omega) \times e^{-j\theta(\omega)} \quad \theta(\omega) = \omega/2$$

then

$$\Psi_g(\omega) \simeq \begin{cases} -j\Psi_h(\omega), & \omega > 0 \\ j\Psi_h(\omega), & \omega < 0. \end{cases}$$

In this paper, we enforce the rational-coefficient constraint on top of the aforementioned desirable properties of the dual-tree CWT. Each of the proposed designs can be implemented efficiently using only binary shift and add operations. All of the popular performance metrics investigated such as coding gain, wavelet smoothness, and Hilbert energy condition show that rational-coefficient filters are a close approximation to previously published irrational CWT filters [4], [5], [7], [8]. Some of these properties, e.g., perfect reconstruction, are not affected if the filter bank is quantized in the lattice/lifting domain; others like wavelet smoothness are affected. The objective is to obtain a tradeoff between these properties and computational complexity. However, since the proposed filters are designed directly in the lattice/lifting domain, their practical implementations have the lowest level of complexity reported in literature. Finally, applications such as image denoising and directional feature extraction confirm that our multiplierless CWT designs retain all practical CWT features such as high energy compaction, good directionality, and improved shift invariance.

This paper is organized as follows. In Section II, we briefly offer preliminary background materials on CWT and review fundamentals of polyphase matrix factorization using lattice structure and lifting scheme. Section III presents the overall design methodology. Section IV presents orthogonal designs in the lattice domain. In Section V, we present biorthogonal designs implemented via the lifting scheme. Section VI presents some of the evaluations and applications that quantify the proposed designs. Finally, Section VII concludes the paper with a summary and several final remarks.

II. Preliminaries

A. Review

Let $\{h_0[n], h_1[n]\}$ represent the filters for the analysis stage of a real coefficient wavelet expansion. Let the corresponding filter pair that implements the synthesis stage be denoted by $\{f_0[n], f_1[n]\}$. The conditions for perfect reconstruction imply that (in the z-transform domain) [2]

$$H_0(-z)F_0(z) + H_1(-z)F_1(z) = 0$$
$$H_0(z)F_0(z) + H_1(z)F_1(z) = 2z^{-d}.$$

For simplicity, we assume that all filters are causal, and therefore, we introduce the term z^{-d} in the above equation.

The filter pair $\{g_0[n], g_1[n]\}$ represents dyadic expansion in the analysis stage of the imaginary-coefficient tree. Their corresponding synthesis filters are $\{p_0[n], p_1[n]\}$. These filters are defined in a similar way.

In polyphase notation, these filters can be written in terms of their even and odd phases according to the following relations:

$$H_0(z) = H_{00}(z^2) + z^{-1}H_{01}(z^2) \tag{1}$$
$$H_1(z) = H_{10}(z^2) + z^{-1}H_{11}(z^2) \tag{2}$$
$$G_0(z) = G_{00}(z^2) + z^{-1}G_{01}(z^2) \tag{3}$$
$$G_1(z) = G_{10}(z^2) + z^{-1}G_{11}(z^2). \tag{4}$$

Let $\mathbf{H}_p(z)$ and $\mathbf{G}_p(z)$ be the polyphase matrices of $\{H_0(z), H_1(z)\}$ and $\{G_0(z), G_1(z)\}$ pairs, respectively. $\mathbf{H}_p(z)$ and $\mathbf{G}_p(z)$ are usually written in terms of even and odd phases of these filters. For instance, the polyphase matrix $\mathbf{G}_p(z)$ is given by

$$\mathbf{G}_p(z) = \begin{bmatrix} G_{00}(z) & G_{01}(z) \\ G_{10}(z) & G_{11}(z) \end{bmatrix}. \tag{5}$$

The 2×2 identity matrix is indicated with the symbol $\mathbf{I}$. The symbol $\mathbf{J}$ is reserved for the 2×2 antidiagonal (reversal) matrix, i.e., $\mathbf{J} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. Finally, we define the matrix $\mathbf{\Lambda}(z)$ as the delay matrix of high-pass subband such that $\mathbf{\Lambda}(z) = \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix}$ and $\mathbf{\Lambda}^{-1}(z) = \mathbf{\Lambda}(z^{-1})$.

Assume that length of filters $\{h_0[n], f_1[n]\}$ is L_0, while the length of $\{h_1[n], f_0[n]\}$ is L_1. In case when the filters are orthogonal and power-complimentary, all filters have same length L (i.e., $L_0 = L_1 = L$). Orthogonality implies that the resulting polyphase matrices are paraunitary, thus

$$\sum_n h_i[n]h_j[n+2k] = \delta[i-j]\delta[k]$$
$$\sum_n g_i[n]g_j[n+2k] = \delta[i-j]\delta[k]$$

and the high-pass filters are alternate time reversals of the low-pass filters

$$h_1[n] = (-1)^n h_0[L-n-1]$$
$$g_1[n] = (-1)^n g_0[L-n-1].$$

For the case when filters are biorthogonal, the length of filters may be different. One interesting class of biorthogonal solutions is one in which the low-pass filters $\{g_0[n], p_0[n]\}$ are related to $\{h_0[n], f_0[n]\}$ by a time reversal, thus

$$g_0[n] = h_0(L_0 - n - 1)$$
$$p_0[n] = f_0(L_1 - n - 1).$$

The biorthogonal filters that will be presented in this paper confirm to the above constraint. Further insight into the design of such filters is available in [7].

B. Lattice Structure

The lattice structure has been widely studied as a tool for efficient implementation of two-channel PR filter banks [9]. It is well known that every orthogonal QMF filter pair can be represented as a cascade of plane rotations. This implementation can further be optimized using a two-multiplier approach, as shown in Fig. 2 (top). Thus, in matrix notation, the lattice transformation can be written as

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \cos\theta_j \begin{bmatrix} 1 & K_j \\ -K_j & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \cos\theta_j \mathbf{R}_\theta^j \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

where $K_j = \tan\theta_j$ is called the lattice coefficient, and if implemented by only shifts and additions, it can provide a very fast approximation of the original transform. By convention, j will denote the number of the stage (zero being the number of the stage that is applied immediately on the input). The inverse of this lattice is rotation by angle $-\theta$ and that will be denoted by $\mathbf{R}_{-\theta}^j$.

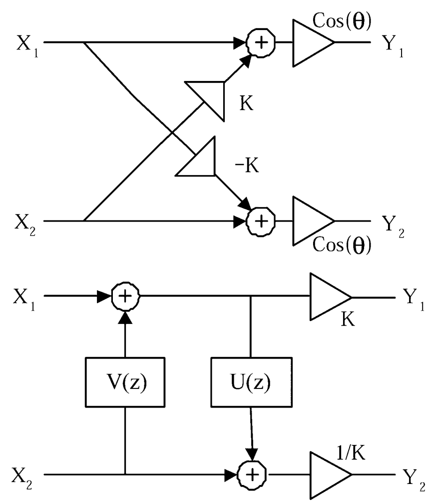

Fig. 2. Lattice structure (top) and lifting scheme (bottom).

To realize higher-order filters, multiple lattice stages need to be cascaded together. In that case, all the $\cos\theta_i$ scaling factors can be combined together and absorbed in the quantization stage. Let C_H and C_G be the scaling parameters for $\mathbf{H}_p(z)$ and $\mathbf{G}_p(z)$, respectively, and they can be expressed in terms of their respective rotation angles according to the following relation:

$$\{C_H, C_G\} \triangleq \prod_{i=0}^{N-1} \cos\theta_i$$

where N is the total number of lattice stages.

C. Lifting Scheme

Shown in Fig. 2 (bottom), the lifting scheme [10] is another popular architecture for building fast and efficient signal decompositions. In matrix notation, this implementation can be written as

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} \begin{bmatrix} 1 & 0 \\ U(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & V(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}.$$

The inverse of lifting step is simple: We simply subtract out what was added in at the forward transform. Like lattice, lifting structure is very robust to quantization and structurally enforces prefect reconstruction. Another advantage of lifting is that it can be used to build wavelet transforms that map integers to integers, a very desirable feature for lossless image compression [11]. It has been shown in [12] that any finite-impulse-response (FIR) wavelet filter pair can be realized using the lifting scheme. It is particularly attractive when used in implementing symmetric/antisymmetric filters, because in that case, the lifting polynomials may be simplified to be of the form $\alpha(1+z^{-1})$. Lifting scheme can be used in several other ways. For example, it can realize any lattice structure [12] and even M-channel filter banks, e.g., in [13], a family of multiplierless approximations of the DCT, called the binDCT, was implemented entirely in the lifting domain.

D. Rational-Coefficient Design Considerations

It is well known that quantizing filter coefficients directly in FIR form satisfies aliasing cancellation, because filter coefficients get quantized by the same amount. However distortion elimination fails because filters do not remain power complimentary any more [9]. As a result, PR cannot be achieved. However, when quantization is performed in lattice domain, the power-complementary property is preserved and PR is possible. The price paid is mostly in the degree of regularity, because zeros for vanishing moments get perturbed from their actual locations. Therefore, the design goal is to keep these zeros as close as possible to $z=-1$, while maintaining minimum hardware cost. Another important consideration is that the round-off noise between filter coefficients and the quantized coefficients is more sensitive to the quantization of lifting/lattice structures that are located closer to the input. Thus, more quantization accuracy is needed for structures that are closer to the input, while the later stages in the cascade structure can be aggressively quantized. There has also been interest in designing filters with reduced implementation complexity. Approaches in which coefficients are based on sum of power of two coefficients are particularly interesting because floating-point multiplications can be transformed into simple VLSI-friendly shifts and additions. In [14], an effective design approach is discussed, according to which a multiplierless approximation can be found by using coefficients of the form $k/2^n$, where n indicates the resolution of the approximation.

III. Design Procedure

Our design procedure is governed by global optimization of four parameters: i) coding gain; ii) Hilbert transform properties; iii) dc leakage; and iv) hardware complexity. These quantities will be introduced shortly.

We used the software Singular [15] to develop a framework to solve the nonlinear equations that would enable us to switch from filter domain to lattice/lifting domain, and vice versa. Two filter sets presented here are based upon Selesnick's and Kingsbury's schemes, quantized and exhaustively searched to optimize the design parameters, while another set is obtained by searching directly in lattice or lifting domain. Furthermore, we have also presented designs that are based upon the time-reversal theorems. The time-reversal theorem in the lattice domain establishes that for orthogonal designs, the computational complexity of filters $\{h_0[n], h_1[n]\}$ and $\{f_0[n], f_1[n]\}$ is similar. Similarly, the time-reversal theorem in lifting domain states that for biorthogonal designs, the computational complexity of filters $\{h_0[n], h_1[n]\}$ and $\{g_0[n], g_1[n]\}$ is similar. Finally the designs are applied in several image processing applications.

A. Coding Gain

Transform coding gain is a very desirable property in building transforms, especially the ones that are intended for compression applications. Coding gain relates to the ability of a subband coder to compress most of the signal energy in the least number of bands, also referred to as energy compaction property. As a result, the quantization and entropy coding can be tailored appropriately to obtain the highest rate-distortion performance.

The biorthogonal coding gain $\mathbf{C}_g$, is defined as [16]

$$\mathbf{C}_g = 10 \times \log_{10} \frac{\sigma_x^2}{\left(\prod_{i=0}^{M-1} \sigma_{xi}^2 \times \Gamma f_i \Gamma^2\right)^{1/M}}$$

where

$M \triangleq$ number of subbands;
$\sigma_x^2 \triangleq$ variance of input;
$\sigma_{xi}^2 \triangleq$ variance of ith subband;
$\Gamma f_i \Gamma^2 \triangleq$ L-2 norm of ith synthesis basis function.

In above calculation, the input is assumed to be a first-order Gaussian–Markov process with unit variance and correlation coefficient of 0.95 (usually a good approximation for natural images). The input variance does not affect the optimization results; however, unit input variance simplifies calculation.

B. Hilbert Transform Properties

In signal processing theory, for the CWT wavelet bases to form an ideal Hilbert transform pair, the frequency response of the function $\psi_h(t)+j\psi_g(t)$ should have high attenuation for all frequencies in the region $-\infty < \omega < 0$. Due to the fact that it is not possible to design an FIR filter $g_0[n]$ that is an exact half-sample delay of $h_0[n]$, a perfect Hilbert transform pair is not possible [7]. Thus, one has to compromise for a Hilbert transform approximation. There are several ways to measure this approximation. For instance, for the function $\Psi_h(\omega) + j\Psi_g(\omega)$, we can measure i) the stop-band attenuation for negative frequencies; ii) the peak ripple in the region $-\infty < \omega < 0$; or iii) the total energy present in the negative frequency range of $-\infty < \omega < 0$.

We propose two measures that are related to the energy and maxima of the function $\Psi_h(\omega)+j\Psi_g(\omega)$ in the passband. These quantities are termed as Hilbert Energy (HE) and Hilbert PSNR (HPSNR). They are defined as

$$\mathrm{HE} = \sqrt{\int_{-\infty}^{0} |\Psi_h(\omega) + j\Psi_g(\omega)|^2 d\omega}$$

$$\mathrm{HPSNR} = 10 \times \log_{10} \left(\frac{\max |\Psi_h(\omega) + j\Psi_g(\omega)|^2}{\int_{-\infty}^{0} |\Psi_h(\omega) + j\Psi_g(\omega)|^2 d\omega} \right).$$

Hilbert transform properties directly correspond to better directional and denoising performances, as will be explained towards later part of this letter.

C. DC Leakage or Vanishing Moments

A key property that distinguishes wavelets from classical filter banks is the number of zeroes at $z = -1$ [2], [17]. It is also referred to as regularity or the vanishing moments. In particular, for a K-regular filter bank, the high-pass filter has K vanishing moments, i.e., $\sum_n n^l h_1[n] = 0$ for $l = 0, 1, \ldots K-1$. In other words, the high-pass filter attenuates all the polynomials of degree less than K. As perfect reconstruction has to be ensured, all of these polynomials have to be captured by the low-pass filter.

By enforcing certain constraints on the filters, we can ensure that at least one zero is always located at $z = -1$. This would imply that no dc component would be picked up in the high-pass subband. If $h_1[n]$ is the high-pass filter, then this also implies that $\sum_n h_1[n] = 0$. In the lattice domain, the same can be achieved by utilizing the fact that sum of all the lattice angles, θ_i must confirm to the following:

$$\sum_i \theta_i = \pi/4 \pm k\pi.$$

To measure the energy picked up in the high pass sub-band when a constant is presented at the input, we define dc leakage (DCPSNR). It is mathematically defined as

$$\mathrm{DCPSNR} = 10 \times \log_{10} \left(\frac{255^2}{|\sum_n h_1[n]|^2} \right).$$

D. Hardware Complexity

When a number is multiplied by an integer, this multiplication can be transformed into additions of its bit-shifted versions. The number of additions can then be minimized by using the classical Booth's algorithm for signed multiplication [18]. For example, a multiplication by $-7/8$ can be implemented by only one shift and one addition, as it can be written as $-1 + 1/8$.

The hardware complexity of an algorithm is generally determined by the number of additions and shifts required in its implementation. From a hardware perspective, adders are considered to be the most expensive modules in a CPU's arithmetic logic unit. Likewise, smaller amount of shifts allow the algorithm to be mapped on a CPU with narrower bus width, a very attractive feature for portable devices where bus width and battery power are limited. This topic is further explored in [14], where the authors present an algorithm to generate multiplierless approximation of transforms.

IV. Orthogonal Designs via Lattice Structure

A. Design via Exhaustive Search

Our first design is termed Lattice-6 and it is obtained by an exhaustive search involving three lattice stages. This design conforms to the Kingsbury's Q-shift scheme, so the filter set $\{G_0(z), G_1(z)\}$ is taken to be a time reversal of $\{H_0(z), H_1(z)\}$. We will soon show that under this constraint, the lattice coefficients of $\{G_0(z), G_1(z)\}$ can be directly derived from those of $\{H_0(z), H_1(z)\}$. The dc leakage condition introduced above allows us to express one of the angles in terms of the other two. Thus, the optimization problem reduces to a search in 2-D space (shown in Fig. 3). In this search, the coding gain and Hilbert PSNR are plotted in 3-D, against first two lattice coefficients (K_0 and K_1). The point that corresponds to the best overall compromise is marked with a black cylinder (lattice values corresponding to that point are $K_0 = 3/16, K_1 = -37/8$ and $K_2 = -5/2$). The polyphase factorization of this implementation is as follows:

$$\mathbf{H}_p(z) = \begin{bmatrix} -5/64 & 0 \\ 0 & -5/64 \end{bmatrix} \begin{bmatrix} 1 & 3/16 \\ -3/16z^{-1} & z^{-1} \end{bmatrix} \cdots \times \begin{bmatrix} 1 & -37/8 \\ 37/8z^{-1} & z^{-1} \end{bmatrix} \begin{bmatrix} 1 & -5/2 \\ 5/2z^{-1} & z^{-1} \end{bmatrix}$$

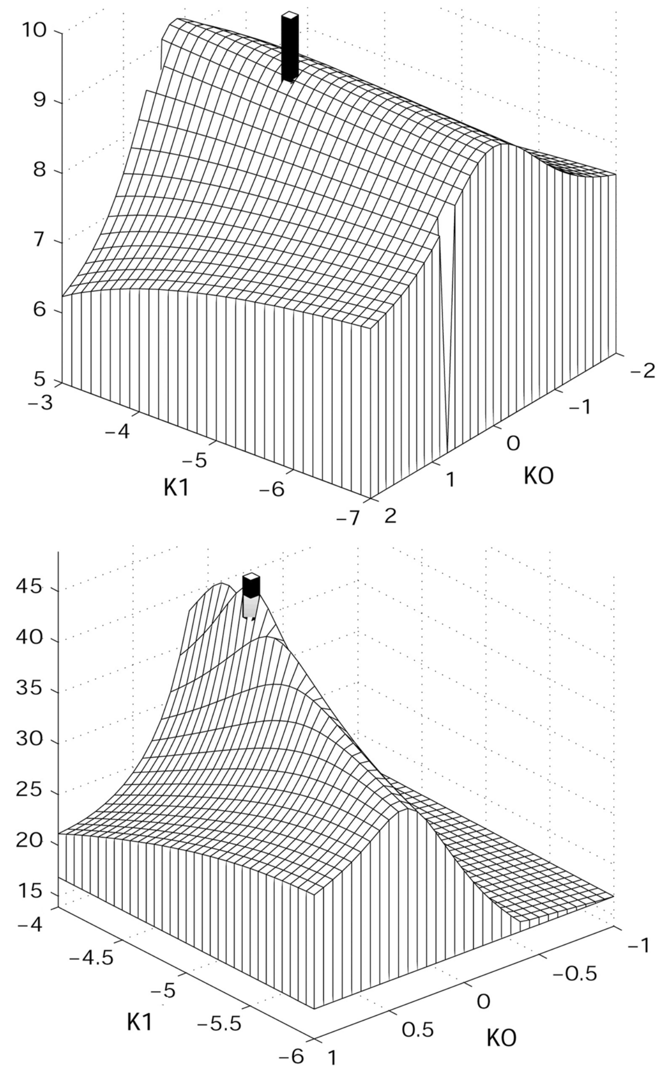

Fig. 3. Exhaustive search results for coding gain (top) and Hilbert PSNR (bottom). The black cylinders mark the point that corresponds to Lattice-6 implementation.

$$\mathbf{G}_p(z) = \begin{bmatrix} -1/64 & 0 \\ 0 & -1/64 \end{bmatrix} \begin{bmatrix} 1 & 85/16 \\ -85/16z^{-1} & z^{-1} \end{bmatrix} \cdots \times \begin{bmatrix} 1 & 37/8 \\ -37/8z^{-1} & z^{-1} \end{bmatrix} \begin{bmatrix} 1 & 5/2 \\ -5/2z^{-1} & z^{-1} \end{bmatrix}.$$

The filter coefficients are displayed in Table IV (column 3). In Table IV, we also list N_q and N_u, which are quantized and unquantized values of normalization constants. These have to be multiplied to filter coefficients to enforce normalization, so in this case, $N_u \sum_n h_0[n] = \sqrt{2}$ and $N_q \sum_n h_0[n] \simeq \sqrt{2}$.

B. Design via Approximation

The next design, called Lattice-12, is an approximation of one of the filters designed using Selesnick's approach in [7] where the author described a systematic design procedure based on spectral factorization. In this procedure, a flat-delay all-pass filter is used to approximate half-sample delay between $H_0(z)$ and $G_0(z)$. The problem reduces to the design of only two filters: $H_0(z)$ and $G_0(z)$. Presented as Example 1A in [7], the two filters have 12-taps with four vanishing moments and second-order all-pass characteristics. We first map the original filter coefficients to the lattice-coefficient domain, then subsequently approximate each resulting irrational lattice coefficient by a dyadic rational, making sure that filters obtained are the best compromise of the four design parameters. As shown in Table I, the dyadic rationals of Lattice-12 are very good approximations of the irrationals originally proposed by Selesnick. The filter coefficients are displayed in Table IV (column 1). As expected, for the original filters, the zeros at $Z = -1$ move from their original positions, however the DC leakage condition enforces one zero to be in the vicinity of $Z = -1$.

TABLE I
SELESNICK'S LATTICE AND LATTICE-12 APPROXIMATIONS

	$\{H_0(z), H_1(z)\}$ Lattice Coefficients		$\{G_0(z), G_1(z)\}$ Lattice Coefficients	
	Selesnick-12	Lattice-12	Selesnick-12	Lattice-12
κ_0	-7.48299980	$-31/4$	0.51739997	$2/4$
κ_1	-2.31345916	$-9/4$	7.54474640	$30/4$
κ_2	-2.40704036	$-10/4$	-0.25932312	$-1/4$
κ_3	1.75770926	$7/4$	-3.55918646	$-14/4$
κ_4	3.28926611	$13/4$	-2.48548698	$-10/4$
κ_5	-1.27999997	$-5/4$	-32.0000000	$-100/4$

TABLE II
KINGSBURY'S LATTICE AND LATTICE-14 APPROXIMATIONS

	$\{H_0(z), H_1(z)\}$ Lattice Coefficients		$\{G_0(z), G_1(z)\}$ Lattice Coefficients	
	Kingsbury-14	Lattice-14	Kingsbury-14	Lattice-14
κ_0	-1.19400000	$-19/16$	1.19400000	$20/16$
κ_1	0.31583467	$5/16$	-0.31583467	$-4/16$
κ_2	12.6443805	$200/16$	-12.6443805	$-144/16$
κ_3	0.03376979	$1/16$	-0.03376979	$-2/16$
κ_4	-0.75500833	$-12/16$	0.75500833	$12/16$
κ_5	6.80293321	$84/16$	-6.80293321	$-108/16$
κ_6	-1.40100002	$-22/16$	-0.71377586	$-12/16$

The second approximation, called Lattice-14, is based on a 14-tap Q-shift filter proposed by Kingsbury [8].[1] Q-shift filters are surprisingly neat filters, in a sense that they are orthogonal, the imaginary part of the complex wavelet is the time-reverse of the real part, (i.e., $\psi_g(t) = \psi_h(N-1-t)$) and the filter $G_0(z)$ is the time-reverse of $H_0(z)$. The Q-shift filters have a group delay of 1/4 or 3/4. This condition ensures that the difference in delay between filter and its time reverse is always 1/2. The original and quantized coefficients of Lattice-14 are listed in Table II.

The next approximation is Lattice-14 T. It utilizes the fact that filters in imaginary CWT tree are the time-flips of the filters in real CWT tree. This approximation uses the time reversal theorem in the lattice domain, according to which, the lattice coefficients of filters that are time reversals of each other are related. The theorem and its proof are presented in the next subsection. For Lattice-14T, the filter pair $\{H_0(z), H_1(z)\}$ is the same as Lattice-14; however, lattice coefficients of $\{G_0(z), G_1(z)\}$ are directly derived from those of $\{H_0(z), H_1(z)\}$. Table III displays the lattice coefficients for this design while in Fig. 4, the exact implementation is shown. It can be seen that one of the lattice coefficients is $-8/11$, which is not a dyadic rational number. The most common method to deal with the rational

[1]Matlab files for generating these filters are available at http://www-sigproc.eng.cam.ac.uk/~ngk/

TABLE III
KINGSBURY'S LATTICE AND LATTICE-14T APPROXIMATIONS

	$\{H_0(z), H_1(z)\}$ Lattice Coefficients		$\{G_0(z), G_1(z)\}$ Lattice Coefficients	
	Kingsbury-14	Lattice-14	Kingsbury-14	Lattice-14
κ_0	-1.19400000	$-19/16$	1.19400000	$19/16$
κ_1	0.31583467	$5/16$	-0.31583467	$-5/16$
κ_2	12.6443805	$200/16$	-12.6443805	$-200/16$
κ_3	0.03376979	$1/16$	-0.03376979	$-1/16$
κ_4	-0.75500833	$-12/16$	0.75500833	$12/16$
κ_5	6.80293321	$84/16$	-6.80293321	$-84/16$
κ_6	-1.40100002	$-22/16$	-0.71377586	$-8/11$

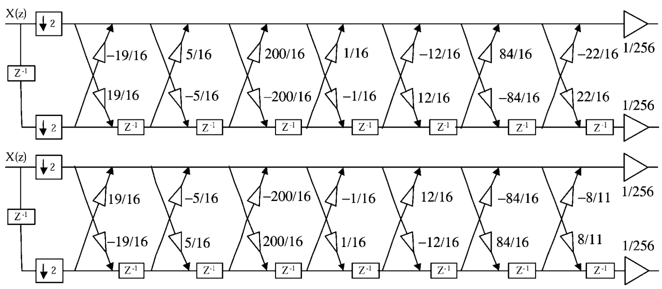

Fig. 4. Implementation of the Lattice-14T. Top structure implements $\{H_0(z), H_1(z)\}$ while bottom one implements $\{G_0(z), G_1(z)\}$.

8/11 in a practical multiplierless implementation is to approximate it by a close dyadic-rational such as 93/128. Since this is simply a scaling factor, the only sacrifice we have to make is that the $\{G_i(z), P_i(z), i = 0, 1\}$ filters will not be the exact time-reversed versions of the $\{H_i(z), F_i(z), i = 0, 1\}$ filters: They will be off by a tiny margin in a scaling sense. The rounding results in movement of zero at $Z = -1$ by less than 1%, and it does not affect the magnitude of filter coefficients, directionality, coding gain, or HPSNR.

C. Time-Reversal Relationship in the Lattice Domain

Theorem 1: For Kingsbury's Q-shift designs [19], the filter pair $\{G_0(z), G_1(z)\}$ is related to $\{H_0(z), H_1(z)\}$ by a time reversal. Let $K_0, K_1, \ldots K_{N-2}, K_{N-1}$ be a set of lattice coefficients that realizes the filter pair $\{H_0(z), H_1(z)\}$, where K_{N-1} is associated with the last rotation (closest to the filters' outputs). Then, the time-reversed filter pair $\{G_0(z), G_1(z)\}$ can be realized with the following set of lattice coefficients: $-K_0, -K_1, \ldots -K_{N-2}, 1/K_{N-1}$. ■

In other words, inverting the last lattice coefficient and reversing the polarity of the rest yield the time-reversed filters. The lattice-domain time-reversal theorem is also evident from observing unquantized coefficients in Table III (and also from Fig. 4), where the product of the K_6 coefficients for $H_0(z)$ and $G_0(z)$ is one. For $K_0 \ldots K_5$, the rotations have opposite sign but same magnitude.

Two important implications result from this elegant relationship. First, $\{G_0(z), G_1(z)\}$ pair has nearly the same computational complexity as the $\{H_0(z), H_1(z)\}$ pair. Second, if there exists a wavelet system where all coefficients are sums of power of two (e.g., the Daubechies-5/3), the $\{G_0(z), G_1(z)\}$ pair will be equally efficient in complexity.

Proof: Since the length of all filters is L (where L is even), the number of lattice stages required to realize the filters is $N = L/2$. Since the filter $g_0[n]$ is a time-reverse of $h_0[n]$, we have

$$g_0[n] = h_0(2N - n - 1).$$

In the z-transform domain, we have equivalently

$$G_0(z) = z^{-(2N-1)} H_0(z^{-1}).$$

Since the filters are of even length and time-flip of each other, the polyphase components are also related. That is

$$G_{00}(z) = z^{-(N-1)} H_{01}(z^{-1}) \tag{6}$$

$$G_{01}(z) = z^{-(N-1)} H_{00}(z^{-1}) \tag{7}$$

$$G_{10}(z) = z^{-(N-1)} H_{11}(z^{-1}) \tag{8}$$

$$G_{11}(z) = z^{-(N-1)} H_{10}(z^{-1}). \tag{9}$$

Using (5) and from (6)–(9), even/odd phases of $\{G_0(z), G_1(z)\}$ can be expressed in terms of even/odd phases of $\{H_0(z), H_1(z)\}$, according to the following:

$$\mathbf{G}_p(z) = z^{-(N-1)} \begin{bmatrix} H_{01}(z^{-1}) & H_{00}(z^{-1}) \\ H_{11}(z^{-1}) & H_{10}(z^{-1}) \end{bmatrix}.$$

Therefore $\mathbf{G}_p(z)$ can be expressed in terms of $\mathbf{H}_p(z)$ as

$$\mathbf{G}_p(z) = z^{-(N-1)} \mathbf{H}_p(z^{-1}) \mathbf{J}. \tag{10}$$

In terms of lattice factorization, the polyphase matrix $\mathbf{H}_p(z)$ can be written as

$$\begin{aligned} \mathbf{H}_p(z) &= C_H \begin{bmatrix} 1 & K_{N-1} \\ -K_{N-1} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \cdots \\ &\quad \times \begin{bmatrix} 1 & K_1 \\ -K_1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \begin{bmatrix} 1 & K_0 \\ -K_0 & 1 \end{bmatrix} \\ &= C_H \mathbf{P}_H \mathbf{\Gamma}_H(z) \end{aligned} \tag{11}$$

where

$$\begin{aligned} \mathbf{P}_H &\triangleq \begin{bmatrix} 1 & K_{N-1} \\ -K_{N-1} & 1 \end{bmatrix} \\ \mathbf{\Gamma}_H(z) &\triangleq \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \begin{bmatrix} 1 & K_{N-2} \\ -K_{N-2} & 1 \end{bmatrix} \cdots \\ &\quad \begin{bmatrix} 1 & K_0 \\ -K_0 & 1 \end{bmatrix} \\ &= \mathbf{\Lambda}(z) \mathbf{R}_\theta^{N-2} \mathbf{\Lambda}(z) \mathbf{R}_\theta^{N-3} \ldots \mathbf{\Lambda}(z) \mathbf{R}_\theta^{0}. \end{aligned}$$

Using a very similar argument, $\mathbf{G}_p(z)$ can be written in terms of lattice coefficients $\bar{K}_0, \bar{K}_1, \ldots \bar{K}_{N-1}$ and scaling factor C_G as follows:

$$\begin{aligned} \mathbf{G}_p(z) &= C_G \begin{bmatrix} 1 & \bar{K}_{N-1} \\ -\bar{K}_{N-1} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \cdots \\ &\quad \times \begin{bmatrix} 1 & \bar{K}_1 \\ -\bar{K}_1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \begin{bmatrix} 1 & \bar{K}_0 \\ -\bar{K}_0 & 1 \end{bmatrix} \\ &= C_G \mathbf{P}_G \mathbf{\Gamma}_G(z) \end{aligned} \tag{12}$$

where

$$\mathbf{P}_G \triangleq \begin{bmatrix} 1 & \bar{K}_{N-1} \\ -\bar{K}_{N-1} & 1 \end{bmatrix} \tag{13}$$

$$\begin{aligned} \mathbf{\Gamma}_G(z) &\triangleq \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \begin{bmatrix} 1 & \bar{K}_{L-2} \\ -\bar{K}_{L-2} & 1 \end{bmatrix} \cdots \\ &\quad \begin{bmatrix} 1 & \bar{K}_0 \\ -\bar{K}_0 & 1 \end{bmatrix} \\ &= \mathbf{\Lambda}(z) \bar{\mathbf{R}}_\theta^{N-2} \mathbf{\Lambda}(z) \bar{\mathbf{R}}_\theta^{-3} \ldots \mathbf{\Lambda}(z) \bar{\mathbf{R}}_\theta^{0}. \end{aligned} \tag{14}$$

Substituting $\mathbf{H}_p(z)$ and $\mathbf{G}_p(z)$ from (11) and (12) in (10) yields

$$C_G\mathbf{P}_G\mathbf{\Gamma}_G(z) = z^{-(N-1)}C_H\mathbf{P}_H\mathbf{\Gamma}_H(z^{-1})\mathbf{J}. \quad (15)$$

From (15), we can simplify $z^{-(N-1)}\mathbf{\Gamma}_H(z^{-1})$ to be

$$\begin{aligned} z^{-(N-1)}&\mathbf{\Gamma}_H(z^{-1}) \\ &= \begin{bmatrix} z^{-1} & z^{-1}K_{N-2} \\ -K_{N-2} & 1 \end{bmatrix} \cdots \\ &\quad\times \begin{bmatrix} z^{-1} & z^{-1}K_1 \\ -K_1 & 1 \end{bmatrix} \begin{bmatrix} z^{-1} & z^{-1}K_0 \\ -K_0 & 1 \end{bmatrix}. \end{aligned}$$

Since $\begin{bmatrix} z^{-1} & z^{-1}K_j \\ -K_j & 1 \end{bmatrix} = \mathbf{J\Lambda}(z)\mathbf{R}^j_{-\theta}\mathbf{J}$, and by using the fact that $\mathbf{JJ} = \mathbf{I}$, the above equation can be simplified as

$$z^{-(N-1)}\mathbf{\Gamma}_H(z^{-1}) = \mathbf{J\Lambda}(z)\mathbf{R}^{N-2}_{-\theta} \dots \mathbf{\Lambda}(z)\mathbf{R}^0_{-\theta}\mathbf{J}. \quad (16)$$

From the value of $z^{-(N-1)}\mathbf{\Gamma}_H(z^{-1})$ above, (15) now can be written as

$$C_G\mathbf{P}_G\mathbf{\Gamma}_G(z) = C_H\mathbf{P}_H\mathbf{J\Lambda}(z)\mathbf{R}^{N-2}_{-\theta} \dots \mathbf{\Lambda}(z)\mathbf{R}^0_{-\theta}. \quad (17)$$

Equating the scaling terms separately from (17), we get

$$\mathbf{P}_G = \frac{C_H}{C_G}\mathbf{P}_H\mathbf{J} = \frac{C_H}{C_G}\begin{bmatrix} K_{N-1} & 1 \\ 1 & -K_{N-1} \end{bmatrix}. \quad (18)$$

Comparing (18) with (13) yields

$$\bar{K}_{N-1} = \frac{C_H}{C_G} = \frac{1}{K_{N-1}}. \quad (19)$$

Similarly, by equating z-terms in (17), we get

$$\mathbf{\Gamma}_G(z) = \mathbf{\Lambda}(z)\mathbf{R}^{N-2}_{-\theta}\mathbf{\Lambda}(z)\mathbf{R}^{N-3}_{-\theta} \dots \mathbf{\Lambda}(z)\mathbf{R}^0_{-\theta}. \quad (20)$$

So, from (14) and (20), it is clear that $\bar{\mathbf{R}}^j_\theta = \mathbf{R}^j_{-\theta}$, and therefore for $j = 0, 1, \dots, N-2$,

$$\bar{K}_j = -K_j.$$

Thus the last lattice coefficient of the imaginary tree can be simply inverted while the polarity of the remaining can be simply reversed. ∎

V. Biorthogonal Designs via Lifting Scheme

The biorthogonal families of dual-tree CWT that are studied here are based on the spectral factorization approach proposed by Selesnick in [7]. Using this algorithm, biorthogonal filters with certain regularity and approximation to the flat all pass delay filter can be constructed. One important constraint that this design imposes is that $G_0(z)$ is always a time reversal of $H_0(z)$. Because of constraints discussed in the cited paper, Selesnick's design procedure cannot produce linear phase filters. However, for the interested reader, biorthogonal linear phase filters for CWT are discussed in [20] and [4].

A. Design Via Exhaustive Search

The first biorthogonal approximation is called Lifting-5/3. Its analysis low-pass and high-pass filters are of order four and two, respectively. Similar to the Daubechies-5/3 filters, Lifting-5/3 is very efficient. The polyphase matrices for Lifting-5/3 have the following form:

$$\begin{aligned} \mathbf{H}_p(z) &= \begin{bmatrix} -\frac{44}{32} & 0 \\ 0 & -\frac{23}{32} \end{bmatrix} \begin{bmatrix} z^{-1} & c+dz^{-1} \\ 0 & z^{-1} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ a+bz^{-1} & 1 \end{bmatrix} \\ \mathbf{G}_p(z) &= \begin{bmatrix} -\frac{44}{32} & 0 \\ 0 & -\frac{23}{32} \end{bmatrix} \begin{bmatrix} z^{-1} & d+cz^{-1} \\ 0 & z^{-1} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ b+az^{-1} & 1 \end{bmatrix}. \end{aligned}$$

By utilizing that $\sum_n h_1[n] = 0$ and $\sum_n f_1[n] = 0$, we can represent b and d in terms of a and c, respectively,

$$a + b = -1 \quad (21)$$

$$c + d = 1/2. \quad (22)$$

The above conditions also enforce one zero at exactly $z = -1$ (i.e., no dc component leaks into the high pass subband). Therefore, the filters can be designed using an exhaustive search that maximizes coding gain and Hilbert performance from two free parameters. Fig. 5 depicts the results from this search. Coding gain and Hilbert PSNR are plotted in 3-D against variables a and c. The optimal area is marked with black cylinder and this corresponds to values $a = -3/4, b = -1/4, c = 1/8$ and $d = 3/8$. It turns out that this is also a very close solution to the 5/3 filter pair that can be designed using Selesnick's construction (called Selesnick-5/3). With one vanishing moment and all pass filter of order one, Selesnick-5/3 is one of the simplest designs that can be achieved using dual-tree CWT construction. Thus, if we start with Selesnick's 5/3 filters and perform Euclidean factorization algorithm [12], the result is a solution that is close to the our 5/3 solution.

Shown in Fig. 6, the implementation has one update and one prediction stage, followed by a scaling factor. As expected, the lifting polynomials for time-reversed filters are also related. This relationship will be described and proved in the next section.

B. Design via Approximation

Our final design via approximation is called Lifting-9/3. Lifting-9/3 uses the same high-pass filter as the 5/3, but improves on the number of vanishing moments of the low-pass filter. Table V shows filter coefficients for this approximation. The polyphase factorization is given by

$$\begin{aligned} \mathbf{H}_p(z) &= \begin{bmatrix} -\frac{44}{32} & 0 \\ 0 & -\frac{23}{32} \end{bmatrix} \begin{bmatrix} -z^{-1} & \frac{1}{64}(1+4z^{-3}) \\ 0 & z^{-1} \end{bmatrix} \cdots \\ &\quad\times \begin{bmatrix} -z^{-1} & \frac{1}{16}(2+7z^{-2}) \\ 0 & z^{-1} \end{bmatrix} \begin{bmatrix} -z^{-1} & 0 \\ -\frac{1}{4}(3+1z^{-1}) & 1 \end{bmatrix} \\ \mathbf{G}_p(z) &= \begin{bmatrix} -\frac{44}{32} & 0 \\ 0 & -\frac{23}{32} \end{bmatrix} \begin{bmatrix} -z^{-1} & \frac{1}{64}(4+z^{-3}) \\ 0 & z^{-1} \end{bmatrix} \cdots \\ &\quad\times \begin{bmatrix} -z^{-1} & \frac{1}{16}(7+2z^{-2}) \\ 0 & z^{-1} \end{bmatrix} \begin{bmatrix} -z^{-1} & 0 \\ -\frac{1}{4}(1+3z^{-1}) & 1 \end{bmatrix}. \end{aligned}$$

C. Time-Reversal Relationship in the Lifting Domain

Theorem 2: Consider a biorthogonal dual-tree CWT design where the low-pass filter pair $\{G_0(z), P_0(z)\}$ is related to the low-pass filter pair $\{H_0(z), F_0(z)\}$ by a time reversal. If

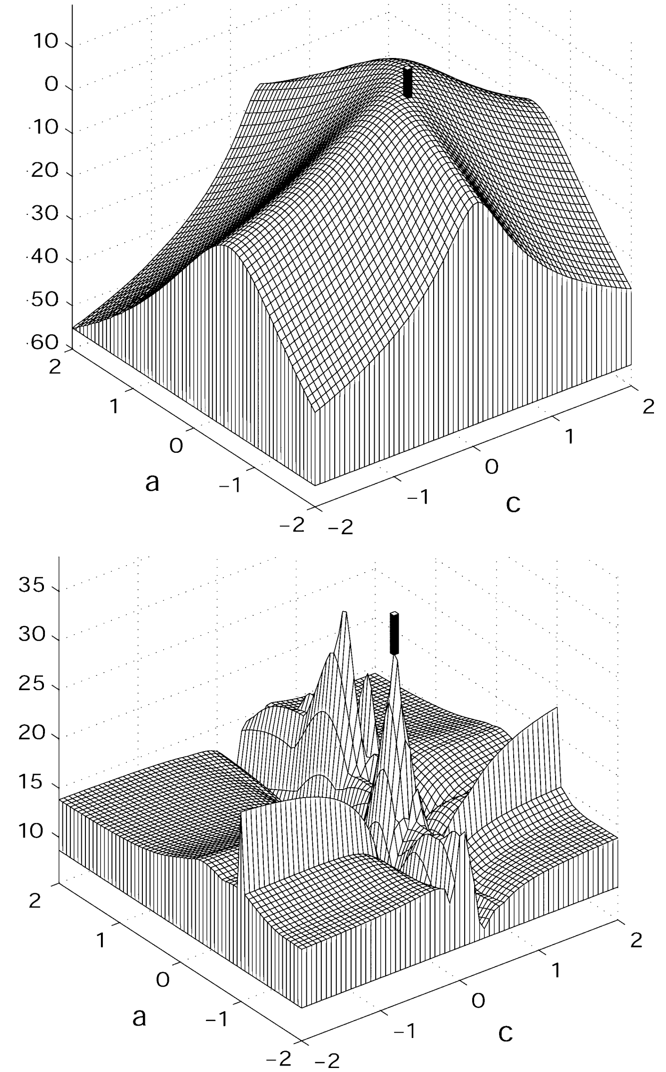

Fig. 5. Exhaustive search results for coding gain (top) and Hilbert PSNR (bottom). The black cylinders mark the point that corresponds to Lifting-5/3 implementation.

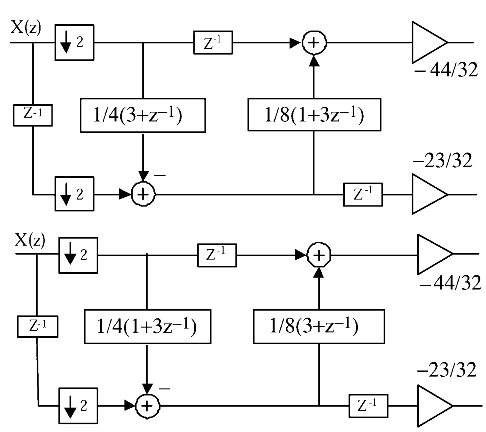

Fig. 6. Implementation of the Lifting-5/3. Top structure implements $\{H_0(z), H_1(z)\}$ while bottom structure implements $\{G_0(z), G_1(z)\}$.

$U_0(z), U_1(z), \ldots, U_{N-1}(z)$ and $V_0(z), V_1(z), \ldots, V_{M-1}(z)$ are respectively update and prediction polynomials that implement $\{H_0(z), H_1(z)\}$, then the set of update/predict polynomials that can realize $\{G_0(z), G_1(z)\}$ are $U_0(z^{-1})$, $U_1(z^{-1}), \ldots, U_{N-1}(z^{-1})$ and $V_0(z^{-1}), V_1(z^{-1}), \ldots, V_{M-1}(z^{-1})$. ■

In other words, by time-reversing all update/predict stages in a lifting scheme, we can implement time-reversed filters. Note that to satisfy causality, some delay has to be introduced into subbands. This holds for odd-length as well as even-length filters, except that in even-length case, the polyphase matrix has to be postmultiplied by antidiagonal (reversal) matrix.

Proof: Since low-pass filter pair $\{G_0(z), P_0(z)\}$ is a time reversal of $\{H_0(z), F_0(z)\}$, we have

$$G_0(z) = z^{-(L_0-1)} H_0(z^{-1})$$
$$P_0(z) = z^{-(L_1-1)} F_0(z^{-1})$$

From simple algebra, it can easily be calculated that

$$G_1(z) = -z^{-(L_1-1)} H_1(z^{-1}).$$

First, consider the case when filters have odd lengths. For this case, since filters are odd-length and time-reversed, their even and odd phases are related by

$$G_{00}(z) = z^{-(L_0-1)/2} H_{00}(z^{-1}) \quad (23)$$
$$G_{01}(z) = z^{-(L_0-3)/2} H_{01}(z^{-1}) \quad (24)$$
$$G_{10}(z) = -z^{-(L_1-1)/2} H_{10}(z^{-1}) \quad (25)$$
$$G_{11}(z) = -z^{-(L_1-3)/2} H_{11}(z^{-1}). \quad (26)$$

The polyphase matrix $\mathbf{G}_p(z)$ can be written in terms of $\mathbf{H}_p(z)$ by

$$\mathbf{G_p(z)} = \boldsymbol{\Delta}(z) \mathbf{H}_p(z^{-1}) \boldsymbol{\Lambda}(z^{-1}) \quad (27)$$

where

$$\boldsymbol{\Delta}(z) \triangleq \begin{bmatrix} z^{-(L_0-1)/2} & 0 \\ 0 & -z^{-(L_1-1)/2} \end{bmatrix}.$$

For the case when filters have even-lengths, even and odd phases are related by

$$G_{00}(z) = z^{-(L_0-2)/2} H_{01}(z^{-1}) \quad (28)$$
$$G_{01}(z) = z^{-(L_0-2)/2} H_{00}(z^{-1}) \quad (29)$$
$$G_{10}(z) = -z^{-(L_1-2)/2} H_{11}(z^{-1}) \quad (30)$$
$$G_{11}(z) = -z^{-(L_1-2)/2} H_{10}(z^{-1}). \quad (31)$$

Again, the polyphase matrix $\mathbf{G}_p(z)$ takes a very similar form

$$\mathbf{G_p(z)} = \boldsymbol{\Delta}(z) \mathbf{H}_p(z^{-1}) \mathbf{J} \quad (32)$$

where $\boldsymbol{\Delta}(z)$ is slightly different

$$\boldsymbol{\Delta}(z) \triangleq \begin{bmatrix} z^{-(L_0-2)/2} & 0 \\ 0 & -z^{-(L_1-2)/2} \end{bmatrix}.$$

Equations (27) and (32) imply that if $\mathbf{H}_p(z)$ has lifting decomposition of the form

$$\mathbf{H}_p(z) = \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} \prod_{i=0}^{N-1} V_i(z) U_i(z)$$

then $\mathbf{G}_p(z)$ can be factored as follows:

$$\mathbf{G}_p(z) = \boldsymbol{\Delta}(z) \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} \prod_{i=0}^{N-1} V_i(z^{-1}) U_i(z^{-1}).$$

$\mathbf{G}_p(z)$ has to be postmultiplied by $\mathbf{J}$ if filters are even-length and by $\boldsymbol{\Lambda}(z^{-1})$ if filters are odd-length. ■

VI. Evaluations and Applications

In this section, we evaluate the performance of proposed designs with the originals and the separable DWT wavelets.

TABLE IV
QUANTIZED FILTER COEFFICIENTS FOR ORTHOGONAL DESIGNS

	Lattice-12		Lattice-14		Lattice-6	
n	$h_0[n]$	$g_0[n]$	$h_0[n]$	$g_0[n]$	$h_0[n]$	$g_0[n]$
1	-2048	-2048	262144	-262144	-256	-256
2	15872	-1024	-311296	-327680	640	-640
3	41600	151552	1805312	778240	2738	9250
4	-41024	56576	-1946368	1140736	1739	14541
5	53584	-265856	-6666928	-4372480	-120	3400
6	671148	1252512	17872272	406912	-48	-1360
7	894613	3428256	45399007	23770208	–	–
8	242570	2319488	34534531	31370144	–	–
9	-183464	-188160	733239	12720384	–	–
10	-69056	-583680	-8386514	-5235840	–	–
11	19840	-25600	2047904	-2600448	–	–
12	2560	51200	1495936	2181120	–	–
13	–	–	-428032	-245760	–	–
14	–	–	-360448	196608	–	–
N_u	$57.65/2^{26}$	$15.32/2^{26}$	$70.58/2^{32}$	$102.04/2^{32}$	$-19.74/2^{16}$	$-3.71/2^{16}$
N_q	$58/2^{26}$	$15/2^{26}$	$71/2^{32}$	$102/2^{32}$	$-20/2^{16}$	$-4/2^{16}$

TABLE V
QUANTIZED FILTER COEFFICIENTS FOR BIORTHOGONAL DESIGNS

	Lifting-5/3		Lifting-9/3	
n	$h_0[n]$	$h_1[n]$	$h_0[n]$	$h_1[n]$
1	3	0	-3	0
2	-4	-24	4	0
3	-22	32	23	0
4	-12	-8	-32	-192
5	3	0	-164	256
6	–	–	-112	-64
7	–	–	16	0
8	–	–	16	0
9	–	–	-4	0
K_u	$-45.25/2^{10}$	$-22.62/2^{10}$	$-362.03/2^{16}$	$-181.01/2^{16}$
K_q	$-45/2^{10}$	$-23/2^{10}$	$-362/2^{16}$	$-181/2^{16}$

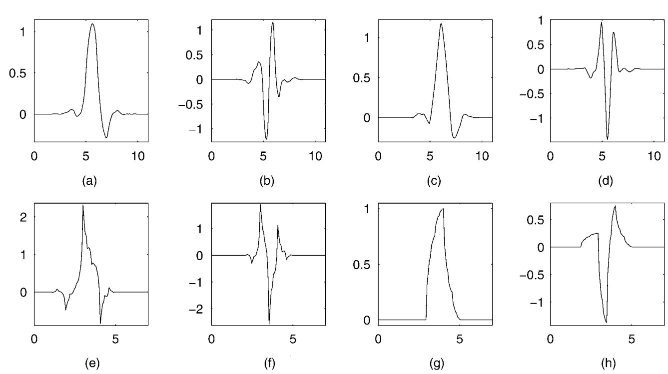

Fig. 7. Lattice-12: (a) $\phi_h(t)$; (b) $\psi_h(t)$; (c) $\phi_g(t)$; (d) $\psi_g(t)$. Lifting-5/3: (e) $\phi_h(t)$; (f) $\psi_h(t)$; (g) $\phi_g(t)$; (h) $\psi_g(t)$.

A. Smoothness in Wavelet and Scaling Functions

Since most natural images have smooth transitions, the smoothness of the resulting scaling and wavelet bases is a desirable property for every image processing application. The precise relationship between regularity and smoothness is unknown, however generally a higher number of vanishing moments results in smoother bases. The rational-coefficient designs presented here generate functions that exhibit smoothness and are very close approximations with low complexity. In Fig. 7, the Lattice-12 and Lifting-5/3 wavelet and scaling functions (for analysis stage) are plotted. The Lattice-12 designs are orthogonal, so the synthesis scaling and wavelet functions are the time-reversals of analysis functions. For Lifting-5/3, the imaginary-tree scaling and wavelet functions are the time-reversals of the real-tree functions. Table VIII (D) lists the dc leakage of all the filters discussed in this paper. It can be seen that all of them are very good at attenuating dc.

B. Hilbert Performance and Directional 2-D Wavelets

Fig. 8 shows a frequency response of the four-level decomposition using Lattice-12 filters, where the real and imaginary coefficients are combined to form complex coefficients $(|\Psi_h(\omega)+j\Psi_g(\omega)|)$. We have found that the Hilbert performance of all of the proposed designs is almost as good as the original ones. Measurements of Hilbert energy and Hilbert PSNR, as shown in Table VIII (B, C), illustrate that Lattice-12 design outperforms all discussed filter sets.

A direct consequence of good Hilbert transform pair is better directional selectivity in the 2-D domain [5]. In a separable 2-D wavelet, the directional wavelets for filters $\{h_0[n], h_1[n]\}$ are defined as

$$\begin{aligned}\psi_h^H(x,y) &= \psi_h(y)\phi_h(x) \quad \text{Horizontal edges}\\ \psi_h^V(x,y) &= \phi_h(y)\psi_h(x) \quad \text{Vertical edges}\\ \psi_h^D(x,y) &= \psi_h(y)\psi_h(x) \quad \text{Diagonal edges.}\end{aligned}$$

The directional bases $\{\psi_g^H(x,y), \psi_g^V(x,y), \psi_g^D(x,y)\}$ for $\{g_0[n], g_1[n]\}$ are defined similarly. An application that demonstrates the power of directional properties of the proposed designs is the directional feature extraction, as shown in Fig. 9. For this test, Lifting-9/3 was used along with the *zone-plate* image, which has frequency content from dc to $\pm\pi$. The traditional separable 2-D DWT only captures edges along horizontal, vertical, and diagonal directions (shown in the top row). Edges along several other directions can be more efficiently captured by adding and subtracting $\{\psi_h^H(x,y), \psi_h^V(x,y)\}$ with $\{\psi_g^H(x,y), \psi_g^V(x,y)\}$. These angles are given by

$$\begin{aligned}\psi_h^H(x,y) + \psi_g^H(x,y) &\quad 15^\circ\text{edges}\\ \psi_h^V(x,y) + \psi_g^V(x,y) &\quad 75^\circ\text{edges}\\ \psi_h^V(x,y) - \psi_g^V(x,y) &\quad 105^\circ\text{edges}\\ \psi_h^H(x,y) - \psi_g^H(x,y) &\quad 165^\circ\text{edges.}\end{aligned}$$

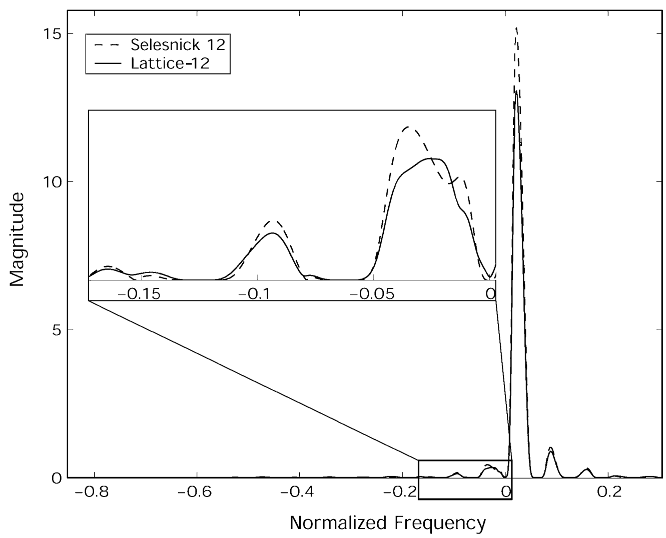

Fig. 8. Plot of $|\Psi_h(\omega) + j\Psi_g(\omega)|$ for Selesnick-12 and Lattice-12.

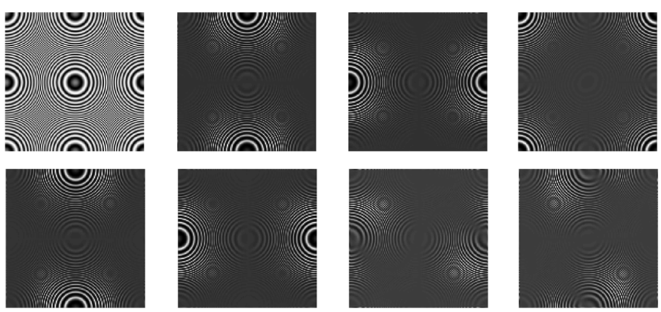

Fig. 9. Directional feature extraction for the *zoneplate* image. Clockwise from top-left: original image, $\psi_h^H(x,y), \psi_h^V(x,y), \psi_h^D(x,y), \psi_h^H(x,y) - \psi_g^H(x,y), \psi_h^V(x,y) - \psi_g^V(x,y), \psi_h^V(x,y) + \psi_g^V(x,y), \psi_h^H(x,y) + \psi_g^H(x,y)$.

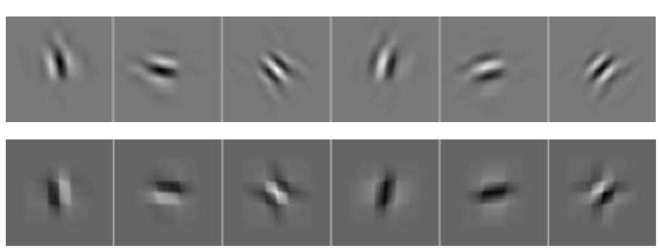

Fig. 10. Directional wavelets from Lattice-12 (top) and Lifting-5/3 (bottom). From left to right, edges at angles: 15°, 75°, 45°, 165°, 105°, 135°.

Another example is illustrated in Fig. 10, where the six directional wavelets for Lattice-12 and Lifting-5/3 are plotted. The filter sets succeed in isolating different orientations, without the checkerboard effects (checkerboarding appears in separable 2-D DWT wavelets because it mixes the $+45^\circ$ and -45° orientations [6]).

C. Denoising Performance

The dual-tree complex wavelet transform is a powerful tool for image and video denoising due to its near shift and rotation invariance [21], [20]. Fig. 11 shows[2] PSNR versus threshold point plot for the Stonehenge image. For this experiment, the thresholds were applied globally across all the wavelet coefficients. It is more effective to apply the thresholding nonlinearity to the magnitude of the transform (as opposed to denoising real and imaginary parts separately), since the magnitudes are less sensitive to aliasing distortion. Table VI presents a comparison of denoising performance for two different values of noise standard deviations, σ_N of 15 and 25.

[2]Most of our denoising experiments were based on the Matlab code available at http://taco.poly.edu/WaveletSoftware/index.html

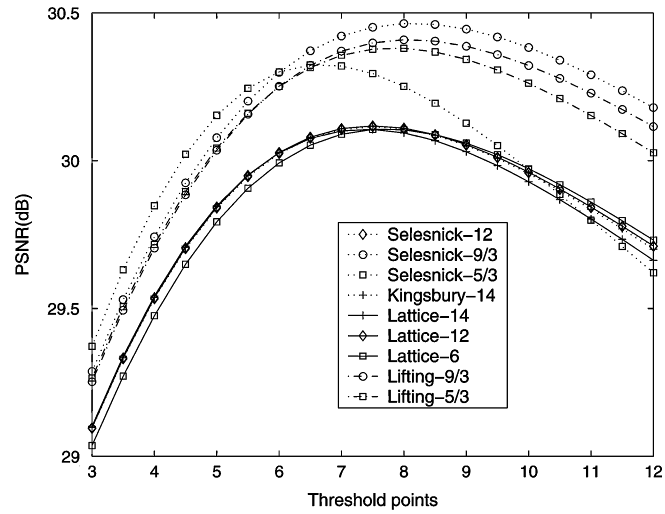

Fig. 11. PSNR versus threshold points plot for Stonehenge. The original image with an additive white Gaussian (AWG) noise of $\sigma_N = 15$ (noisy $\mathrm{PSNR} = 27.16$ dB) was denoised using CWT filters. The best achievable PSNR with Daubechies-9/7 2-D separable DWT was 29.13 dB.

TABLE VI
DENOISING RESULTS FROM THE DWT FILTER SETS ARE COMPARED WITH THE NOISE-ADDED VERSION (Δ dB IS REPORTED)

	Lena		Barbara	
σ_N	15	25	15	25
Daubechies-9/7	1.399	0.804	0.971	0.658
Daubechies-5/3	1.300	0.795	0.782	0.626
Selesnick-12	2.959	3.416	2.172	2.683
Selesnick-9/3	3.131	3.503	2.130	2.638
Selesnick-5/3	2.981	3.559	2.085	2.637
Kingsbury-14	2.971	3.420	2.215	2.724
Lattice-14	2.966	3.430	2.231	2.745
Lattice-12	2.964	3.423	2.178	2.691
Lattice-6	2.855	3.354	1.994	2.541
Lifting-9/3	3.082	3.478	2.091	2.611
Lifting-5/3	3.045	3.477	2.065	2.587
Noise-Added (dB)	27.51	23.01	27.52	23.04

TABLE VII
HARDWARE COMPLEXITY PER DWT COEFFICIENT IN TERMS OF ADDITIONS (a) AND SHIFTS (s) FOR 1-DIMENSIONAL, 1-LEVEL DWT

	$\{H_0(z), H_1(z)\}$	$\{G_0(z), G_1(z)\}$
Daub-9/7	$5(a)/4(s)$	–
Daub-5/3	$2(a)/1(s)$	–
Lattice-14	$17(a)/14(s)$	$13(a)/8(s)$
Lattice-12	$13(a)/11(s)$	$11(a)/10(s)$
Lattice-6	$7(a)/7(s)$	$9(a)/8(s)$
Lifting-9/3	$4(a)/4(s)$	$4(a)/4(s)$
Lifting-5/3	$3(a)/2(s)$	$3(a)/2(s)$

D. Hardware Complexity

Table VII summarizes the complexity of all approximations. The complexity of a reasonable approximation of the

TABLE VIII
FILTER PROPERTY COMPARISONS OF THE PROPOSED DESIGNS

	A. Coding Gain (dB)				*B*. Hilbert Energy	*C*. Hilbert PSNR (dB)	*D*. DC Leakage (dB)	
Filters	$H_0(z)$	$F_0(z)$	$G_0(z)$	$P_0(z)$			$H_0(z)$	$G_0(z)$
Selesnick-12	9.622	9.622	9.622	9.622	1.30	46.641	∞	∞
Selesnick 9/3	9.204	8.782	9.204	8.782	5.06	35.286	∞	∞
Selesnick 5/3	9.293	6.959	9.293	6.959	5.40	29.740	∞	∞
Kingsbury-14	9.675	9.675	9.675	9.675	1.12	48.939	∞	∞
Lattice-14T	9.649	9.649	9.649	9.649	1.74	40.059	96.551	96.551
Lattice-14	9.649	9.649	9.648	9.648	2.04	43.808	95.700	105.65
Lattice-12	9.614	9.614	9.629	9.629	0.96	49.291	103.19	111.21
Lattice-6	9.109	9.109	9.113	9.113	0.95	44.759	88.720	89.710
Lifting-9/3	9.227	8.520	9.227	8.520	3.93	37.586	∞	∞
Lifting-5/3	9.175	7.691	9.175	7.691	5.42	33.203	∞	∞

Daubechies-9/7 in the lifting domain is also illustrated. The Lifting-5/3 design requires only 6 additions and four shifts per wavelet coefficient for both real and imaginary DWT trees.

VII. CONCLUSION

In summary, we have presented the following key original contributions in this paper.

- Several performance measures were reported which will be helpful in evaluating future dual-tree CWT filter pairs.
- Using the lifting scheme and lattice structure, several fast new designs and implementations of the dual-tree CWT wavelet bases were presented. Since all lifting and lattice parameters were chosen to be dyadic rationals, a very efficient implementation with only shift and addition operations is possible. Many of our new dual-tree wavelet filter designs have the lowest level of computational complexity ever reported in the current literature.
- We show that filters that are time reversals of each other are also related in lattice and lifting domain. These interesting theorems are further validated by observing lattice/lifting coefficients of the proposed designs. The observations are vital in simplifying the design process as well as the final implementation of many of our proposed filter pairs.
- Our new rational-coefficient designs have been evaluated with various performance measures, including denoising and directional feature extraction. The results confirm that rational designs outperform the 2-D separable Daubechies-9/7 in both the applications. There is little difference, if any, between the previously published irrational-coefficient dual-tree solutions and the rational-coefficient dual-tree designs presented in this paper. In other words, the rational coefficient constraints do not impose a severe penalty on the dual-tree's most desirable properties such as time-invariance, directionality, wavelet smoothness, and energy compaction.

When compared to the Daubechies' filters, the performance of CWT filters in image compression is not as interesting as the applications discussed here. The interested reader is, however, referred to [22] and [23], where results for JPEG-2000 image compression for the dual-tree CWT were reported.

For future work, we will continue to explore the close relationships between the wavelet coefficients from the real and imaginary tree of the dual-tree CWT. This exploitation can reduce the level of redundancy that currently exists in a dual-tree wavelet system. For example, can we obtain the imaginary dual-tree wavelet coefficients from a set of approximated wavelet coefficients from the real tree (typical in compression applications)? How does quantization affect vital dual-tree properties such as directionality and shift invariance? Another problem under investigation is an efficient method to recover a subset of wavelet coefficients of one tree from the other. The answers to these questions will not only benefit practical applications, but they will also improve our understanding of complex wavelets.

REFERENCES

[1] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1992.
[2] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, rev. ed. Wellesley, MA: Wellesley-Cambridge Press, 1998.
[3] I. W. Selesnick, "Hilbert transform pairs of wavelet bases," *IEEE Signal Process. Lett.*, vol. 8, pp. 170–173, Jun. 2001.
[4] N. G. Kingsbury, "The dual-tree complex wavelet transform: A new technique for shift invariance and directional filters," in *Proc. IEEE DSP Workshop*, Aug. 1998, Bryce, Canyon, paper no. 86.
[5] N. G. Kingsbury, "A dual-tree complex wavelet transform with improved orthogonality and symmetry properties," in *Proc. IEEE ICIP*, Vancouver, Canada, Sep. 2000, vol. 2, pp. 375–378.
[6] I. W. Selesnick, R. G. Baraniuk, and N. C. Kingsbury, "The dual-tree complex wavelet transform," *IEEE Signal Process. Mag.*, vol. 22, no. 6, pp. 123–151, Nov. 2005.
[7] I. W. Selesnick, "The design of approximate Hilbert transform pairs of wavelet bases," *IEEE Trans. Signal Process.*, vol. 50, no. 5, pp. 1144–1152, May 2002.
[8] N. G. Kingsbury, "Design of Q-shift complex wavelets for image processing using frequency domain energy minimisation," *Proc. IEEE ICIP*, pp. 1013–1016, Sep. 2003.
[9] P. P. Vaidyanathan and P.-Q. Hoang, "Lattice structures for optimal design and robust implementation of two-channel perfect-reconstruction QMF banks," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, pp. 81–94, Jan. 1988.
[10] W. Sweldens, , A. F. Laine and M. Unser, Eds., "The lifting scheme: A new philosophy in biorthogonal wavelet constructions," in *Wavelet Applications in Signal and Image Processing III*. New York: SPIE, 1995, vol. 2569, pp. 68–79.
[11] W. Sweldens, R. Calderbank, I. Daubechies, and B.-L. Yeo, "Wavelet transforms that map integers to integers," *Appl. Comput. Harmon. Anal.*, vol. 5, no. 3, pp. 332–369, 1998.
[12] I. Daubechies and W. Sweldens, "Factoring wavelet transforms in lifting step," Bell Laboratories, Murray Hill, NJ, Tech. Rep., 1996.
[13] J. Liang and T. D. Tran, "Fast multiplierless approximations of the DCT with the lifting scheme," *IEEE Trans. Signal Process.*, vol. 49, pp. 3032–3044, Dec. 2001.
[14] Y. J. Chen, S. Oraintara, T. D. Tran, K. Amaratunga, and T. Q. Nguyen, "Multiplierless approximation of transforms with adder constraint," *IEEE Signal Process. Lett.*, vol. 9, pp. 344–347, Nov. 2002.
[15] G. M. Greuel, G. Pfister, and H. Schönemann, "Singular reference manual," Algebraic Geometry Group, Univ. of Kaiserslautern, Germany, Tech. Rep., 2004 [Online]. Available: http://www.singular.uni-kl.de/

[16] J. Katto and Y. Yasuda, "Performance evaluation of subband coding and optimization of its filter coefficients," in *Proc. SPIE Visual Commun. Image Process.*, Boston, MA, Nov. 1991, vol. 9, pp. 95–106.
[17] I. Daubechies*, Ten Lectures on Wavelets*. Philadelphia, PA: SIAM, 1992.
[18] A. D. Booth, "A signed binary multiplication technique," *Quart. J. Mech. Appl. Math.*, vol. 4, pp. 236–240, 1951.
[19] N. G. Kingsbury, "Complex wavelets for shift invariant analysis and filtering of signals," *Appl. Comput. Harmon. Anal.*, vol. 10, no. 3, pp. 234(20)–253(20), May 2001.
[20] N. G. Kingsbury, "The dual-tree complex wavelet transform: A new efficient tool for image restoration and enhancement," in *Proc. EUSIPCO*, Rhodes, Greece, Sep. 1998, pp. 319–322.
[21] I. W. Selesnick and K. Y. Li, "Video denoising using 2-D and 3-D dual-tree complex wavelet transforms," presented at the Wavelet Applications in Signal and Image Processing, San Diego, CA, Aug. 4–8, 2003.
[22] A. Abbas and T. D. Tran, "Fast approximations of the orthogonal dual-tree wavelet bases," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Mar. 2005, vol. 4, pp. 605–608.
[23] A. Abbas and T. D. Tran, "Multiplierless design of biorthogonal dual-tree complex wavelet transform using lifting scheme," in *Proc. IEEE Int. Conf. Image Processing*, Oct. 2006, pp. 1605–1608.

Adeel Abbas received the B.E. degree in electrical engineering from the National University of Sciences and Technology, Pakistan, in 2001 and the M.S. degree in electrical engineering from The Johns Hopkins University, Baltimore, MD, in 2005.

From 1999 to 2003, he served as Software Engineer at Elixir Technologies Corporation, Islamabad, Pakistan. Since September 2005, he has been with the Corporate Research Division of Thomson Multimedia, in their research laboratory, Princeton, NJ, where he is currently serving as a Member of Technical Staff. His research covers implementation aspects of image/video processing systems, with special emphasis on compression and hardware acceleration of codecs on novel architectures.

Mr. Abbas was recipient of the Best Student Paper award in the IEEE International Multi-Topic Conference (INMIC) 2001.

Trac D. Tran (S'94–M'98) received the B.S. and M.S. degrees from the Massachusetts Institute of Technology, Cambridge, in 1993 and 1994, respectively, and the Ph.D. degree from the University of Wisconsin, Madison, in 1998, all in electrical engineering.

In July 1998, he joined the Department of Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, MD, where he is currently an Associate Professor. His research interests are in the field of digital signal processing, particularly in sampling, multirate systems, filter banks, transforms, wavelets, and their applications in signal analysis, compression, processing, and communications.

Dr. Tran was the Co-Director (with Prof. J. L. Prince) of the Thirty–Third Annual Conference on Information Sciences and Systems (CISS), Baltimore, MD, in March 1999. In summer 2002, he was an ASEE/ONR Summer Faculty Research Fellow at the Naval Air Warfare Center—Weapons Division (NAWCWD), China Lake, CA. He has served as Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING as well as the IEEE TRANSACTIONS ON IMAGE PROCESSING. He currently serves as a member of the IEEE Technical Committee on Signal Processing Theory and Methods (SPTM TC). He received the NSF CAREER Award in 2001 and the William H. Huggins Excellence in Teaching Award from The Johns Hopkins University in 2007.