Fast Multiplierless Approximations of the DCT With the Lifting Scheme

Jie Liang, *Student Member, IEEE,* and Trac D. Tran, *Member, IEEE*

***Abstract*—In this paper, we present the design, implementation, and application of several families of fast multiplierless approximations of the discrete cosine transform (DCT) with the lifting scheme called the binDCT. These binDCT families are derived from Chen's and Loeffler's plane rotation-based factorizations of the DCT matrix, respectively, and the design approach can also be applied to a DCT of arbitrary size. Two design approaches are presented. In the first method, an optimization program is defined, and the multiplierless transform is obtained by approximating its solution with dyadic values. In the second method, a general lifting-based scaled DCT structure is obtained, and the analytical values of all lifting parameters are derived, enabling dyadic approximations with different accuracies. Therefore, the binDCT can be tuned to cover the gap between the Walsh–Hadamard transform and the DCT. The corresponding two-dimensional (2-D) binDCT allows a 16-bit implementation, enables lossless compression, and maintains satisfactory compatibility with the floating-point DCT. The performance of the binDCT in JPEG, H.263+, and lossless compression is also demonstrated.**

***Index Terms*—binDCT, DCT, integer DCT, lifting scheme, lossless compression, multiplierless, scaled DCT.**

I. Introduction

THE discrete cosine transform (DCT) [1], [2] is a robust approximation of the optimal Karhunen-Loève transform (KLT) for a first-order Markov source with large correlation coefficient. It has satisfactory performance in terms of energy compaction capability, and many fast DCT algorithms with efficient hardware and software implementations have been proposed. The DCT has found wide applications in image/video processing and other fields. It has become the heart of many international standards such as JPEG, H.26x, and the MPEG family [3]–[5].

There are mainly four types of the DCT, and they are labeled I–IV [2]. Among them, the DCT-II is the most useful. Many different fast algorithms for the DCT computation have been developed for image and video applications. Some of them take advantage of the relationships between the DCT and various existing fast transforms, including the FFT [1], [6]–[8], the Walsh-Hadamard transform (WHT) [9], [10], and the discrete Hartley transform (DHT) [11]. Some algorithms are based on the sparse factorizations of the DCT matrix [12]–[17], and many of them are recursive [12], [14], [16], [17]. Besides one-dimensional (1-D) algorithms, two-dimensional (2-D) DCT algorithms have also been investigated extensively [6], [18]–[21], generally leading to less computational complexity than the row-column application of the 1-D methods. However, the implementation of the direct 2-D DCT requires much more effort than that of the separable 2-D DCT.

The theoretical lower bound on the number of multiplications required for the 1-D eight-point DCT has been proven to be 11 [22], [23]. In this sense, the method proposed by Loeffler *et al.* [15], with 11 multiplications and 29 additions, is the most efficient solution. However, in image and video processing, quantization is often required to compress the data. In these circumstances, significant algorithmic savings can be achieved if some operations of the DCT are incorporated into the quantization step. This leads to a class of fast 1-D and 2-D DCTs that are generally referred to as the *scaled* DCT [5], [8], [21], [23]–[25]. For example, the Arai's method needs only five multiplications [3], [8].

All of the aforementioned fast algorithms still need floating-point multiplications, which are slow in both hardware and software implementations. To achieve faster implementation, coefficients in many algorithms such as [7], [8], [16], and [17] can be scaled and approximated by integers such that floating-point multiplications can be replaced by integer multiplications [3], [26]–[28]. The resulting algorithms are much faster than the original versions and, therefore, have wide practical applications.

Another approach for integer DCT is presented in [29] by searching integer orthogonal transforms with the same symmetry and similar energy compaction capability to the DCT. The new transform can be implemented with integer multiplications and additions. However, the overall complexity of this integer DCT is not satisfactory, compared with other fast integer algorithms, such as [8].

The fixed-point multiplications required by these fast algorithms generally need 32-bit data bus, which is costly in VLSI implementation and hand-held devices where the CPU capability, bus width, and battery power are limited. Therefore, designing good approximations of the DCT that can be implemented with narrower bus width and simpler arithmetic operations, such as shift and addition, is a challenging topic.

Another disadvantage associated with most algorithms that employ floating or fixed-point multiplications is the difficulty in applying them to lossless compression, due to the finite-length representations and the corresponding roundoff errors. Several efficient structures have been proposed that have the property

Manuscript received February 21, 2001; revised September 21, 2001. This work was supported in part by the National Science Foundation under Grant CCR-0093262 and by FastVDO Inc. The associate editor coordinating the review of this paper and approving it for publication was Dr. Alle-Jan van der Veen.

The authors are with the Department of Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: jieliang@jhu.edu; trac@jhu.edu).

Publisher Item Identifier S 1053-587X(01)10495-2.

1053–587X/01$10.00 © 2001 IEEE

of perfect reconstruction with minimum bit expansion. For example, a *ladder network* was introduced in [30]. More systematic results were summarized in [31] and [32] with the name *lifting scheme*. The lifting structure enables flexible and fast biorthogonal transform, and it also allows lossless transform, making it a powerful building tool for wavelet transforms.

It has been proven that any orthogonal filterbank can be decomposed into delay elements and plane rotations by lattice factorizations [33]. It is easy to show that any plane rotation can be represented by lifting steps. Therefore, it follows that the DCT—a simple orthogonal filterbank—can be constructed from the lifting scheme if we start from any plane rotation-based factorization of the DCT matrix, such as those in [12]–[15], and represent each plane rotation by its lifting implementation. The new transform will enjoy the properties of both the DCT and the lifting scheme.

The earliest application of this idea appeared in [30], where a four-point DCT was implemented in terms of the ladder network. In this method, floating-point multiplications were used in the ladder (lifting) steps, and floor operations were applied subsequently to obtain integer transform coefficients. The inputs can be perfectly reconstructed in this way. The idea was extended in [34] to obtain an eight-point lossless DCT by examining the relationship between the DCT matrix and the general reversible (lossless) transform. Integer results were still obtained through the combination of floating-point multiplications and floor operations. Recently, a lossless lapped orthogonal transform (LOT) was obtained with the same idea [35]. However, since fast implementation was not the main concern in [30], [34], and [35], the resulting structures were not optimal in terms of simplicity.

In this paper, we propose and describe the design of fast invertible block transforms that can replace the DCT in future wireless and portable computing applications. The new transform, which is called the binDCT, has the following properties.

1) Both the forward and the inverse transforms can be implemented using only binary shift and addition operations.
2) The idea of the scaled DCT is employed to reduce the complexity of the binDCT.
3) The binDCT inherits all desirable DCT characteristics such as high coding gain, no DC leakage, symmetric basis functions, and recursive construction.
4) The binDCT also inherits all lifting properties such as fast implementations, invertible integer-to-integer mapping, in-place computation, and low dynamic range.

This lifting scheme-based fast multiplierless approximation of the DCT was first proposed in [36] and was generalized in [37]. Several preliminary results were also reported in [38] and [39]. A similar method was later obtained in [40] in which the WHT-based DCT factorization [2], [9], [10] is used, which is not as elegant as that of [12], [15]. Besides, the result in [40] is not a scaled DCT. Hence, the performance of this method is not as good as that of the binDCT, given the same level of complexity.

The paper is organized as follows. Section II will briefly introduce the plane rotation-based DCT factorizations and their relationships with the lifting scheme. In Section III, we define some criteria for measuring the transform performance. Section IV presents the general solution and the design of the binDCT via the optimization approach. The systematic, analytical design of the binDCT and design examples will be presented in Sections V and VI. Important design and implementation issues are discussed in Section VII, whereas the applications of the binDCT in JPEG, H.263+, and lossless compression are demonstrated in Section VIII. Finally, Section IX contains the conclusion.

II. Plane Rotation-Based DCT Factorizations and the Lifting Scheme

A. Plane Rotation-Based DCT Factorizations

Chen *et al.* proposed a recursive algorithm to factor any N-point DCT-II with $N = 2^m, m \geq 2$ into plane rotations and butterflies [12], [13]. The factorization has a very regular structure and is six times as fast as the DFT-based fast DCT algorithm [1]. The method was generalized by Wang to all versions of DCT, DST, the discrete W transform, as well as the DFT with the size of power of 2 [14]. Similar results were also reported in [41].

In this paper, we will concentrate on the four-point, eight-point and 16-point transforms since they are the most useful ones in practical applications. Block transforms of other sizes can be designed in a similar fashion. The factorization of the eight-point DCT in [12]–[14] is given in Fig. 1(a), where the result in the dashed box is the scaled four-point DCT. It contains series of butterflies and five plane rotations, which can be implemented with a total of 13 multiplications and 29 additions [14], [15]. Note that a scaling factor of $1/2$ should be applied at the end to obtain the true DCT coefficients.

A more elegant factorization for eight-point and 16-point DCT was proposed by Loeffler *et al.* [15], as shown in Fig. 1(b). It also contains the scaled four-point DCT. This method only needs 11 multiplications and 29 additions, achieving the multiplication lower bound as proven in [22] and [23]. One of its variations is adopted by the Independent JPEG Group in its popular JPEG implementation [42]. Note that this factorization requires a uniform scaling factor of $1/\sqrt{8}$ at the end of the flow graph to obtain the true DCT coefficients. In the 2-D transform, this scaling factor becomes 1/8, which can be easily implemented by a shift operation.

B. Lifting Scheme and the Plane Rotation

Fig. 2(a) illustrates the decomposition of a plane rotation into three lifting steps [30], [32]. This can be written in matrix form as

$$\begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} = \begin{bmatrix} 1 & p \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ u & 1 \end{bmatrix} \begin{bmatrix} 1 & p \\ 0 & 1 \end{bmatrix} \tag{1}$$

where $p = (\cos(\alpha) - 1)/\sin(\alpha)$, and $u = \sin(\alpha)$.

It can be shown that any $M \times M$ orthogonal matrix can be expressed as the product of $M \times (M-1)/2$ plane rotations [43]. Similarly, any real invertible matrix can be completely characterized by $M \times (M-1)$ plane rotations and M scaling factors, according to the singular value decomposition (SVD) of the matrix. From these, it can be proven that any invertible FIR filterbank can be decomposed into lifting steps [32].

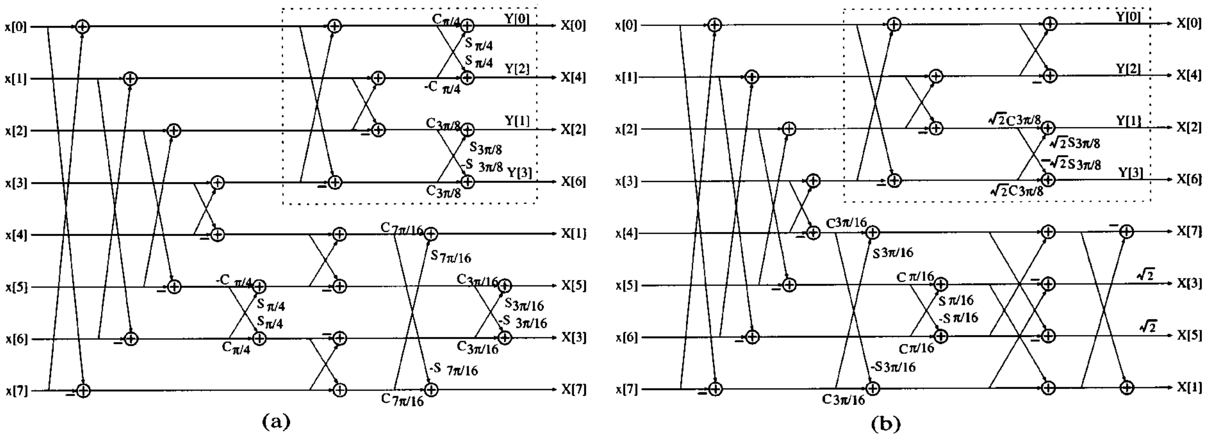


Fig. 1. Signal flow graphs of the eight-point DCT. (a) Chen's factorization. (b) Loeffler's factorization.

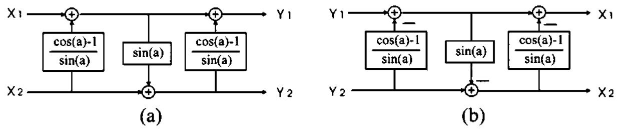


Fig. 2. Representation of a plane rotation by three lifting steps. (a) Forward rotation. (b) Inverse rotation.

Each lifting step is a biorthogonal transform, and its inverse also has a simple lifting structure, i.e.,

$$\begin{bmatrix}1 & x\\ 0 & 1\end{bmatrix}^{-1}=\begin{bmatrix}1 & -x\\ 0 & 1\end{bmatrix},\ \begin{bmatrix}1 & 0\\ x & 1\end{bmatrix}^{-1}=\begin{bmatrix}1 & 0\\ -x & 1\end{bmatrix}. \tag{2}$$

As a result, the inverse of the plane rotation can be represented by lifting steps as

$$\begin{bmatrix}\cos(\alpha) & -\sin(\alpha)\\ \sin(\alpha) & \cos(\alpha)\end{bmatrix}^{-1}=\begin{bmatrix}1 & -p\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ -u & 1\end{bmatrix}\begin{bmatrix}1 & -p\\ 0 & 1\end{bmatrix} \tag{3}$$

as shown in Fig. 2(b). This means that to invert a lifting step, we simply need to subtract out what was added in at the forward transform. Hence, the original signal can still be perfectly reconstructed even if the floating-point multiplication results in the lifting steps are rounded to integers, as long as the same procedure is applied to both the forward and inverse transforms. This is the basis for many lifting-based lossless transforms [34]. Another advantage of the lifting step over the butterfly is that it enables in-place computation, i.e., no buffer is required, which is a desired property in the VLSI implementations.

However, floating-point multiplications are still needed in the above approach. To obtain fast implementation, we can approximate the floating-point lifting coefficients by hardware-friendly dyadic values (i.e., rationals in the format of $k/2^m; k, m$ are integers), which can be implemented by only shift and addition operations. In doing so, we can achieve various fast approximations of the original transform, which we name the binDCT. The multiplication elimination also enables the binDCT to be implemented with a narrower data bus than other algorithms. Since perfect reconstruction is guaranteed by the lifting structure itself, the remaining problem is to select the dyadic lifting parameters such that the binDCT can achieve similar coding performance as the DCT.

TABLE I
CODING GAINS OF SOME COMMONLY USED TRANSFORMS (IN DECIBELS)

Type	4-pt DCT	8-pt WHT	8-pt DCT	8-pt KLT	16-pt DCT	16-pt KLT
C_g	7.5701	7.9461	8.8259	8.8462	9.4555	9.4781

III. Performance Measures

This section defines some criteria used in measuring and evaluating the performance of our proposed fast transforms.

A. Coding Gain

Coding gain is one of the most important factors to be considered for a transform used in compression applications. A transform with higher coding gain compacts more energy into a fewer number of coefficients. As a result, higher objective performances such as PSNR would be achieved after quantization. Since the coding gain of the DCT approximates the optimal KLT closely, it is desired that the binDCT has similar coding gain to that of the original DCT. The biorthogonal coding gain C_g is defined as [44], [45]

$$C_g \triangleq 10\log_{10}\frac{\sigma_x^2}{\left(\prod_{i=0}^{M-1}\sigma_{x_i}^2 \parallel f_i \parallel^2\right)^{\frac{1}{M}}} \tag{4}$$

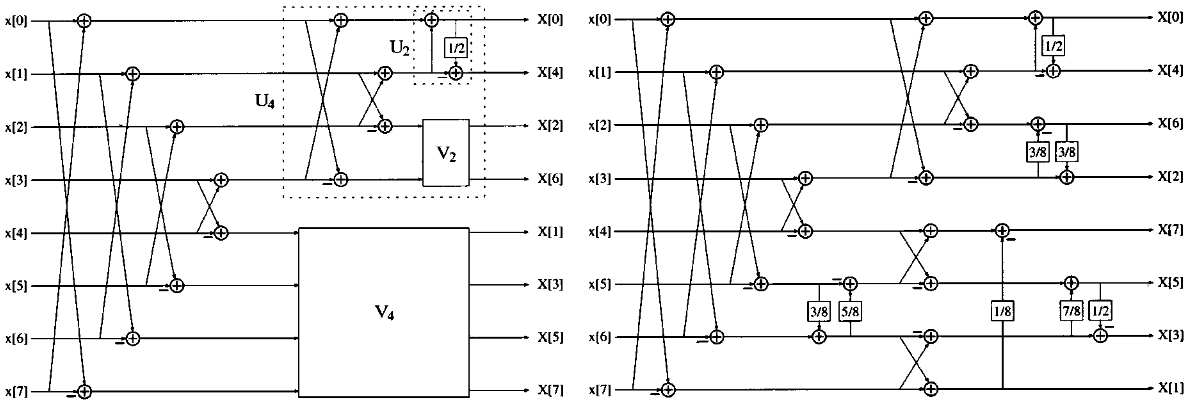
where

M	number of subbands;
σ_x^2	variance of the input;
$\sigma_{x_i}^2$	variance of the ith subband;
$\parallel f_i \parallel^2$	norm of the ith synthesis basis function.

The coding gains of some commonly used transforms are tabulated in Table I, with the assumption that the input signal is a first-order Gaussian–Markov process with zero-mean, unit variance, and correlation coefficient $\rho = 0.95$ (a good approximation for natural images). Note that the coding gain of the DCT is very close to that of the optimal KLT.

B. Mean Square Error (MSE)

To maintain the compatibility between the binDCT and the true DCT outputs, the MSE between the DCT and the binDCT



(a) (b)

Fig. 3. (a) General structure of the recursive DCT. (b) A binDCT Example: 30 additions, 12 shifts, 8.77 dB coding gain.

coefficients should be minimized. With reasonable assumptions of the input signal, the MSE can be explicitly calculated as follows [46].

Assume that $\mathbf{U}_M$ is the true $M \times M$ DCT matrix, and $\mathbf{U}'_M$ is its approximation. Then, for a given input column vector $\mathbf{x}$, the error between the 1-D M-point DCT coefficients and the approximated transform coefficients is

$$\mathbf{e} = \mathbf{U}_M\mathbf{x} - \mathbf{U}'_M\mathbf{x} = (\mathbf{U}_M - \mathbf{U}'_M)\mathbf{x} \triangleq \mathbf{D}\mathbf{x}. \tag{5}$$

From the above equation, the MSE between the approximated DCT and the original DCT can be given by

$$\begin{aligned}\epsilon &\triangleq \frac{1}{M}E[\mathbf{e^T e}] = \frac{1}{M}E[\mathbf{x^T D^T D x}]\\ &= \frac{1}{M}E[\text{Trace}\{\mathbf{D x x^T D^T}\}] = \frac{1}{M}\text{Trace}\{\mathbf{D R_{xx} D^T}\}\end{aligned} \tag{6}$$

where $\mathbf{R_{xx}} \triangleq \mathbf{E}[\mathbf{x x^T}]$ is the autocorrelation matrix of the input signal. Hence, if we model the input signal as a first-order Gaussian–Markov process, the matrix $\mathbf{R_{xx}}$ can be easily calculated, and the MSE can be derived deterministically.

C. DC Leakage

Another desired property of an image transform is that the bandpass and highpass subbands should have no DC leakage, i.e., the constant input should be completely captured by the DC subband. In wavelet theory, this means that these high-frequency subbands should have at least one vanishing moment [45]. The zero dc leakage not only improves the coding efficiency but also prevents the annoying checkerboard artifact that can occur if high-frequency bands are severely quantized [45]. The DCT is a good example of image transforms with zero DC leakage.

IV. General Solution and the Optimization Approach

The hardware-unfriendly components of the DCT factorization are the plane rotations. A trivial way to obtain a multiplierless approximation of the DCT is to replace each rotation by three lifting steps as shown in Fig. 2(a) and then approximate the lifting coefficients by hardware-friendly dyadic rationals. However, in image and video processing, simplicity is always desired to make the transform as fast as possible. This section presents the general solution of approximating the DCT with more efficient lifting scheme.

From a filterbank standpoint, the M-point DCT is the most basic M-channel linear-phase paraunitary filter bank (LPPUFB). All M linear-phase filters have the same length M. If M is even, and if the symmetric filters are permuted to the top, the DCT matrix can be written as

$$\mathbf{U}_M = \frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{U}_{\frac{M}{2}} & \mathbf{0}\\ \mathbf{0} & \mathbf{V}_{\frac{M}{2}}\end{bmatrix}\begin{bmatrix}\mathbf{I}_{\frac{M}{2}} & \mathbf{J}_{\frac{M}{2}}\\ \mathbf{J}_{\frac{M}{2}} & -\mathbf{I}_{\frac{M}{2}}\end{bmatrix} \tag{7}$$

where $\mathbf{I}_{M/2}$ is the $(M/2) \times (M/2)$ identity matrix, and $\mathbf{J}_{M/2}$ is the counter-identity matrix or reversal matrix. If M is a power of 2, the matrix $\mathbf{U}_{M/2}$ in (7) can be factorized recursively, i.e.,

$$\mathbf{U}_{\frac{M}{2}} = \frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{U}_{\frac{M}{4}} & \mathbf{0}\\ \mathbf{0} & \mathbf{V}_{\frac{M}{4}}\end{bmatrix}\begin{bmatrix}\mathbf{I}_{\frac{M}{4}} & \mathbf{J}_{\frac{M}{4}}\\ \mathbf{J}_{\frac{M}{4}} & -\mathbf{I}_{\frac{M}{4}}\end{bmatrix}. \tag{8}$$

Barring an input reversal, the matrices $\mathbf{V}_n$s in (7) and (8) are n-point DCT-IV, and their closed-form factorization is available in [12], [14], leading to a recursive factorization of the DCT-II.

The result in (7) actually covers all M-channel M-tap linear phase filterbanks if $\mathbf{U}_{M/2}$ and $\mathbf{V}_{M/2}$ are chosen to be any invertible matrices. In this paper, we consider the general structure for the eight-point binDCT, as given in Fig. 3(a), where $\mathbf{U}_2$ is fixed to be the unnormalized Haar to guarantee the zero DC leakage property.

An optimization program is constructed in which we represent the matrices $\mathbf{V}_4$ and $\mathbf{V}_2$ by suitable number of lifting steps and butterflies and search for the optimal lifting coefficients that maximize the coding gain. We start from the factorizations given in Fig. 1 and replace each rotation by three lifting steps and then reduce the number of lifting steps gradually to obtain more efficient binDCTs.

The searched optimal results are approximated by dyadic values since they can be implemented by only shifts and additions. For example, $3/8x$ can be implemented by two shifts and one addition, as it can be written as $x/4 + x/8$, where the divisions by 4 and 8 can be performed by right shifts. Similarly, $7x/16$ should be implemented as $x/2 - x/16$. One such result is shown in Fig. 3(b), whose coding gain is quite close to that of the DCT.

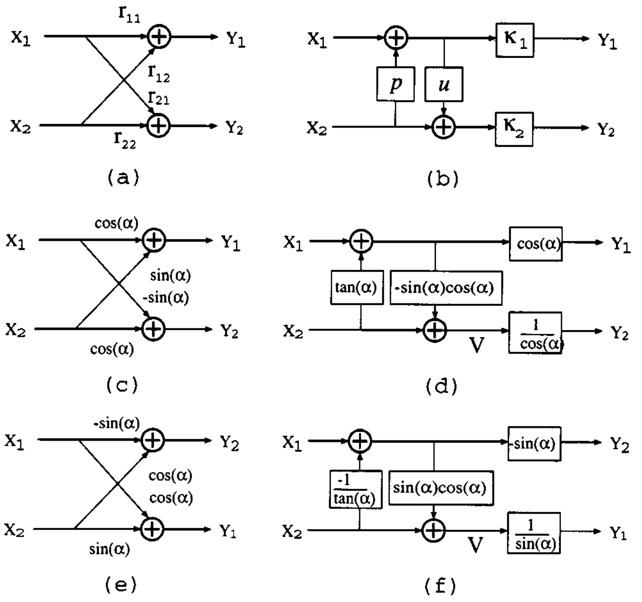


Fig. 4. (a) General butterfly. (b) Scaled lifting structure. (c) Plane rotation. (d) Scaled lifting structure for (c). (e) Permuted plane rotation. (f) Scaled lifting structure for (e).

It should be noted that the binDCT is also a kind of scaled DCT. This is not a major problem in direct application of these transforms. However, when the compatibility between the binDCT transform and the true DCT transform is desired, it is necessary to consider the scaling relationship between the binDCT and the DCT. In this case, the systematic design method given in the next section becomes necessary since it can provide the analytical values of the scaling factors. Besides, different tradeoff between the complexity and the performance of the binDCT can be easily achieved.

V. Systematic Design of the binDCT

A. Scaled Lifting Structure

A plane rotation can be represented by three lifting steps, as shown in Fig. 2, if pure lifting structure is desired. However, the example in the last section reveals that we can also construct a scaled DCT with only two lifting steps for the rotation angles at the end of the signal flow.

This simplified lifting structure can be generalized as in Fig. 4(a) and (b), where a general butterfly (not necessarily an orthogonal plane rotation) is represented by two lifting steps and two scaling factors. The two scaling factors can be absorbed in the quantization stage; thus only two lifting steps are left in the transform, making it more efficient than the conventional representation. Due to the analogy between this idea and that of the scaled DCT [3], [5], [8], we refer to this as the *scaled lifting structure*.

The solutions for the lifting parameters in the scaled lifting structure can be derived as follows. From the flow graphs in Fig. 4(a), we can obtain the following relationship:

$$\begin{aligned} Y_1 &= r_{11}X_1 + r_{12}X_2 \\ Y_2 &= r_{21}X_1 + r_{22}X_2. \end{aligned} \tag{9}$$

Similarly, the outputs of the scaled lifting structure as given in Fig. 4(b) can be rewritten as

$$\begin{aligned} Y_1 &= \kappa_1(X_1 + pX_2) = \kappa_1 X_1 + \kappa_1 p X_2 \\ Y_2 &= \kappa_2(u(X_1 + pX_2) + X_2) = \kappa_2 u X_1 + \kappa_2(1 + pu)X_2. \end{aligned} \tag{10}$$

By equalizing the coefficients of X_1 and X_2 in (9) and (10), the four unknowns can be uniquely determined as

$$\begin{aligned} p &= \frac{r_{12}}{r_{11}} \\ u &= \frac{r_{11}r_{21}}{r_{11}r_{22} - r_{21}r_{12}} \\ \kappa_1 &= r_{11} \\ \kappa_1 &= \frac{r_{11}r_{22} - r_{21}r_{12}}{r_{11}} \end{aligned} \tag{11}$$

where we need $r_{11} \neq 0$ and $r_{11}r_{22} - r_{21}r_{12} \neq 0$.

This analytical solution is the starting point for obtaining binDCTs with different complexities and performances.

B. Sensitivity Analysis and the Permuted Scaled Lifting Structure

This section analyzes the effect of finite-length approximations of the lifting parameters on the performance of the binDCT. A permuted version of the scaled lifting structure will be proposed to improve coding performance in certain circumstances.

In Fig. 4(c), we redraw the familiar rotation angle depicted in Fig. 1. The solution of the corresponding scaled lifting structure can be obtained by (11), as shown in Fig. 4(d).

The signal at the point V in Fig. 4(d) can be expressed as

$$\begin{aligned} V &= ux_1 + (1 + pu)x_2 \\ &= -\sin(\alpha)\cos(\alpha)X_1 + \cos^2(\alpha)X_2. \end{aligned} \tag{12}$$

Equation (12) shows that for plane rotations as shown in Fig. 4(c), the values of $1 + pu$, i.e., $\cos^2(\alpha)$, would be very small if the rotation angle is close to $k\pi + \pi/2$, where k is any integer. For example, $\cos^2(7\pi/16) = 0.038\,06$. Therefore, a large relative error for $1 + pu$ could result when the lifting parameters p and u are truncated or rounded, leading to a drastic change in the frequency response of the result. Another problem in this case is that the lifting parameter $\tan(\alpha)$ would be much greater than 1. This increases the dynamic range of the intermediate result and is not desired in both software and hardware implementations.

Analyzing the example given in the last section reveals that the output sequence of some rotation angles are permuted. This implies that a permutation of the output, as shown in Fig. 4(e), might lead to a much more robust scaled lifting structure. Since the coefficients are permuted accordingly, the new transform is equivalent to the previous one. The general expression in (11) is still valid for this case, and the corresponding scaled lifting parameters are given in Fig. 4(f). The signal at V in Fig. 4(f) is now given by

$$V = \sin(\alpha)\cos(\alpha)X_1 + \sin^2(\alpha)X_2. \tag{13}$$

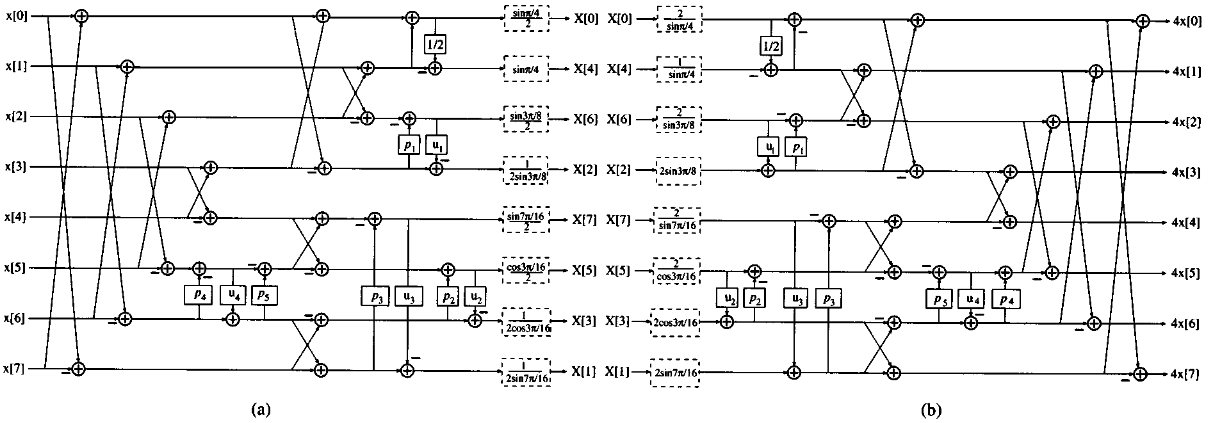


Fig. 5. General structure of the binDCT family based on Chen's factorization. (a) Forward transform. (b) Inverse transform.

TABLE II
SEVERAL CONFIGURATIONS OF BINDCT BASED ON CHEN'S FACTORIZATION

	Floating-point	binDCT-C1	C2	C3	C4	C5	C6	C7	C8	C9
p_1	0.4142135623	13/32	7/16	13/32	7/16	3/8	1/2	1/2	1	0
u_1	0.3535533905	11/32	3/8	11/32	3/8	3/8	3/8	1/2	1/2	0
p_2	0.6681786379	11/16	5/8	11/16	5/8	7/8	7/8	1	1	0
u_2	0.4619397662	15/32	7/16	15/32	7/16	1/2	1/2	1/2	1/2	0
p_3	0.1989123673	3/16	3/16	3/16	3/16	3/16	3/16	1/4	0	0
u_3	0.1913417161	3/16	3/16	3/16	3/16	3/16	1/4	1/4	0	0
p_4	0.4142135623	13/32	13/32	7/16	7/16	7/16	7/16	1/2	0	0
u_4	0.7071067811	11/16	11/16	11/16	11/16	11/16	3/4	3/4	1/2	0
p_5	0.4142135623	13/32	13/32	3/8	3/8	3/8	3/8	1/2	1/2	0
Shifts	-	23	21	21	19	17	14	9	5	1
Adds	-	42	39	40	37	36	33	28	24	18
MSE	-	$1.1e-5$	$5.7e-5$	$3.4e-5$	$8.5e-5$	$4.2e-4$	$5.8e-4$	$2.3e-3$	$4.0e-2$	$2.9e-2$
$C_g(8)$ (dB)	-	8.8251	8.8240	8.8233	8.8220	8.8159	8.8033	8.7686	8.4083	7.9204
$C_g(4)$ (dB)	-	7.5697	7.5697	7.5697	7.5697	7.5566	7.5493	7.5485	7.1744	7.1503

Note that the coefficient of X_2 at V changes from $\cos^2(\alpha)$ to $\sin^2(\alpha)$, which is more robust to truncation errors than (12) for rotation angles close to $k\pi+\pi/2$. This explains the optimization results given in the last section. Besides, the augment of the dynamic range in Fig. 4(d) is also avoided now, as the first lifting parameter becomes $1/\tan(\alpha)$, instead of $\tan(\alpha)$.

In general, when the scaled lifting structure is used to obtain finite-length approximation of the transform with high coding gain and minimal dynamic range, the original structure in Fig. 4(d) should be used if $\cos^2(\alpha) > \sin^2(\alpha)$, and the permuted version in Fig. 4(f) should be adopted if $\cos^2(\alpha) < \sin^2(\alpha)$. When $\cos^2(\alpha) = \sin^2(\alpha)$, both formats reduce to the unnormalized Haar transform.

VI. EIGHT-POINT BINDCT FAMILIES

A. Eight-Point binDCT Type C

The above analysis leads to the general structure of the binDCT based on Chen's factorization, which is denoted as the binDCT type C and shown in Fig. 5. Note that some sign manipulations are involved here to make all the scaling factors positive. The intermediate rotation with angle of $\pi/4$ in $\mathbf{V}_4$ is implemented by three lifting steps, and the permuted version of the scaled lifting structure is used for the angles of $3\pi/8$ and $7\pi/16$.

The rotation of $\pi/4$ between $X[0]$ and $X[4]$ is also implemented by the scaled lifting structure, instead of a butterfly. The purpose is to achieve one vanishing moment and to make all subbands experience the same number of butterflies during the forward and inverse transforms. Since the multiplication of two butterflies introduces a scaling factor of 2, the combination of the forward and inverse transforms thus generates a uniform scaling factor of 4 for all subbands, which becomes 16 for the 2-D transform. This can be compensated by a simple shift operation. The scaling factors in the dashed boxes will be absorbed in the quantization stage. They are bypassed in lossless compression or when the compatibility with the true DCT is not required.

The property of the lifting structure allows us to adjust the lifting parameters without losing perfect reconstruction of the signals. Therefore, from the analytical expressions given in (1) and (11), we can obtain their proper dyadic approximations. This is more flexible than the previous optimization-based design method.

Table II lists the analytical values of all the lifting parameters and some configurations of this binDCT family, where the dyadic values are obtained by truncating or rounding the corresponding analytical values with different accuracies. $C_g(8)$ and $C_g(4)$ are the coding gains of these eight-point binDCTs and the four-point DCTs embedded in them. Fig. 6 compares the

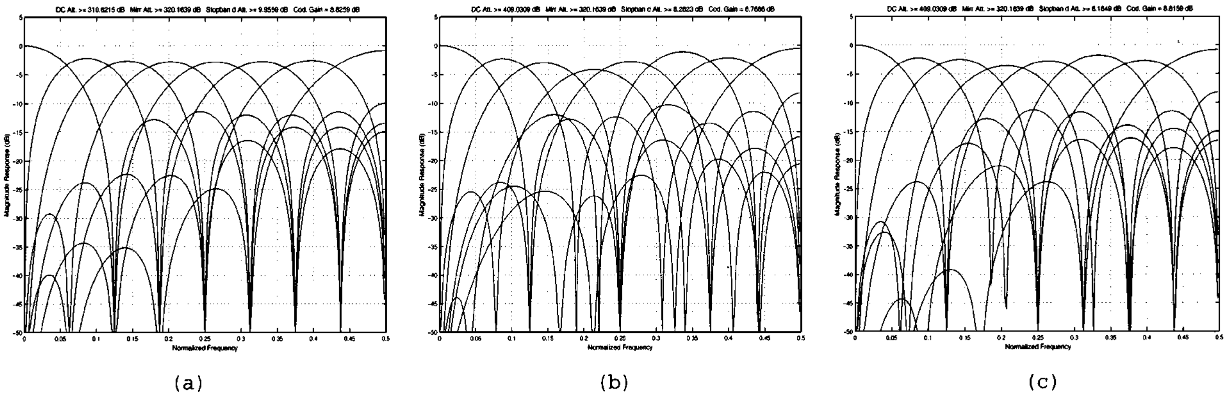


Fig. 6. Frequency responses of (a) True DCT. (b) binDCT-C7: nine shifts and 28 adds. (c) binDCT-C5: 17 shifts and 36 adds.

TABLE III
BINDCT-C7 COEFFICIENTS

binDCT-C7 Forward Transform Matrix								binDCT-C7 Inverse Transform Matrix							
1	1	1	1	1	1	1	1	1/2	1	1	1	1	1/2	1/2	1/4
15/16	101/128	35/64	1/4	-1/4	-35/64	-101/128	-15/16	1/2	13/16	1/2	1/8	-1	-11/16	-3/4	-35/64
3/4	1/2	-1/2	-3/4	-3/4	-1/2	1/2	3/4	1/2	21/32	-1/2	-23/16	-1	-3/32	3/4	101/128
1/2	3/32	-11/16	-1/2	1/2	11/16	-3/32	-1/2	1/2	1/4	-1	-1	1	1/2	-1/2	-15/16
1/2	-1/2	-1/2	1/2	1/2	-1/2	-1/2	1/2	1/2	-1/4	-1	1	1	-1/2	-1/2	15/16
1	-23/16	-1/8	1	-1	1/8	23/16	-1	1/2	-21/32	-1/2	23/16	-1	3/32	3/4	-101/128
1/2	-1	1	-1/2	-1/2	1	-1	1/2	1/2	-13/16	1/2	-1/8	-1	11/16	-3/4	35/64
1/4	-21/32	13/16	-1	1	-13/16	21/32	-1/4	1/2	-1	1	-1	1	-1/2	1/2	-1/4

frequency responses of the true DCT and several binDCT configurations.

The configurations in Table II have different tradeoffs between the complexity and the performance. The configuration with 23 shifts has a coding gain of 8.8251 dB, which almost equals to the 8.8259 dB of the original DCT. Even the nine-shift version has a satisfactory coding gain of 8.7686 dB. In binDCT-C9, where all lifting parameters are set to 0, the coding gain is still 7.9204 dB, which is very close to that of the WHT. Note that in measuring the MSE according to (6), we use the floating-point values of the scaling factors, which are always combined with the quantization steps and rounded to integers in practical implementations. Therefore, the actual MSE might be slightly different from the ones in Table II.

As an example, Table III tabulates the forward and inverse transform matrices of the binDCT-C7, without including the final scaling factors. The embedded four-point DCTs are given in Table IV.

B. Eight-Point binDCT Type L

The aforementioned design method can also be applied to the Loeffler's factorization of the eight-point DCT [15]. We denote this type of binDCT as the binDCT type L. The general structure is given in Fig. 7(a). The top four subbands are exactly the same as the binDCT type C. Since the other two rotations are not at the end of the flow graph, we represent them with the standard three lifting steps. The final butterfly to obtain $X[7]$ and $X[1]$ is also implemented as two liftings to maintain the same number of butterflies for each subband, leading to a uniform scaling factor after the inverse binDCT transform.

The analytical values of the lifting parameters in Fig. 7(a) can be easily calculated, and the results are summarized in Table V, together with some binDCT configurations. The coding gain $C_g(4)$ of the embedded four-point DCTs are also listed. The frequency response of the binDCT-L3 is presented in Fig. 7(b). The relationship between the performance and the complexity of this type of the binDCT is very similar to that of the binDCT type C. However, its scaling matrix is more integer friendly than the binDCT type C.

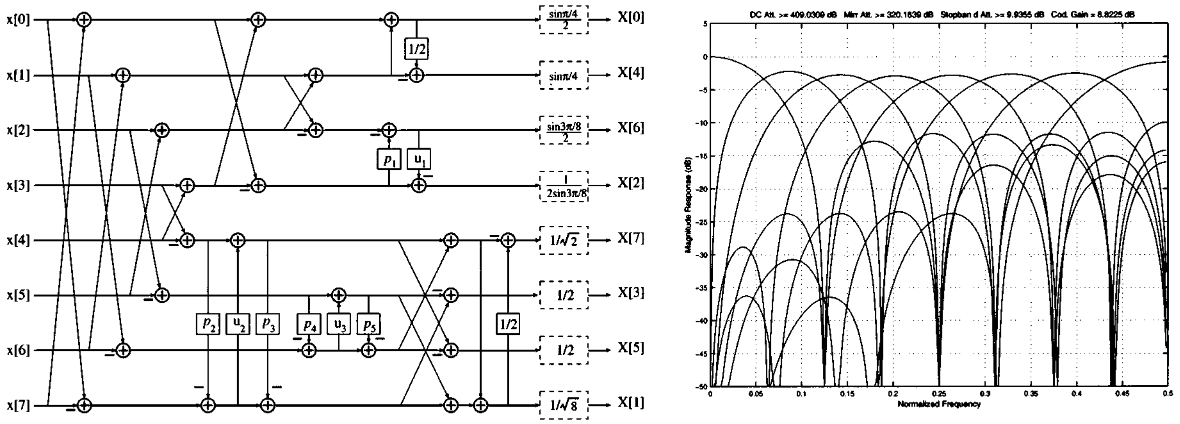
TABLE IV
FOUR-POINT BINDCT EMBEDDED IN THE BINDCT-C7

Forward Transform Matrix				Inverse Transform Matrix			
1	1	1	1	1/2	1	1	1/2
3/4	1/2	-1/2	-3/4	1/2	1/2	-1	-3/4
1/2	-1/2	-1/2	1/2	1/2	-1/2	-1	3/4
1/2	-1	1	-1/2	1/2	-1	1	-1/2

VII. DISCUSSIONS

A. Performance Comparison of the Two Types of Scaled Lifting Structures

In this section, we use the highpass subband of the binDCT-C5 in Table II to demonstrate the necessity of the permuted scaled lifting structure discussed earlier. In Fig. 8(a), the frequency response of the binDCT is obtained when the rotation with angle $7\pi/16$ is implemented with the normal scaled lifting structure. The analytical values of the lifting parameters are $p = 5.027\,339\,492$ and $u = -0.191\,341\,72$, and they can be approximated as $5(3/128) = 5.023\,437\,5$ and $-3/16 = -0.1875$. The result in Fig. 8(b) is obtained when the output $X[7]$ and $X[1]$ are permuted and both p_3 and u_3 are chosen as $3/16$, which require fewer number of arithmetic operations. As shown in Fig. 8, for this type of rotation angle, the frequency response of the output is distorted dramatically if the outputs are not permuted, even though each

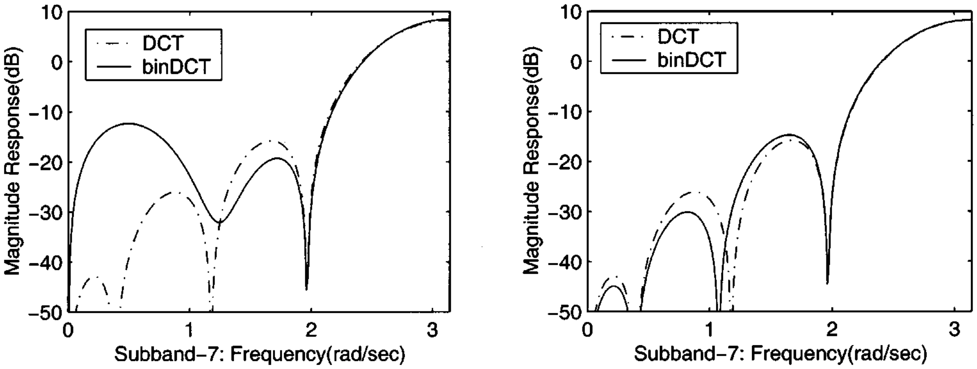


(a) (b)

Fig. 7. (a) binDCT family based on Loeffler's factorization. (b) Frequency responses of binDCT-L3: 16 shifts and 34 adds.

TABLE V
FAMILY OF EIGHT-POINT BINDCTS BASED ON LOEFFLER'S FACTORIZATION

	Floating-point	binDCT-L1	L2	L3	L4	L5	L6	L7	L8	L9
p_1	0.4142135623	13/32	13/32	7/16	3/8	1/2	1/2	1/2	1	0
u_1	0.3535533905	11/32	11/32	3/8	1/4	1/2	1/2	1/2	1/2	0
p_2	0.3033466836	19/64	5/16	1/4	1/4	1/4	0	0	0	0
u_2	0.5555702330	9/16	9/16	9/16	1/2	1/2	1/2	1/2	1/2	0
p_3	0.3033466836	19/64	5/16	5/16	1/4	1/4	1/4	0	0	0
p_4	0.0984914033	3/32	3/32	1/8	1/8	1/8	0	0	0	0
u_3	0.1950903220	3/16	3/16	3/16	3/16	1/4	1/4	0	0	0
p_5	0.0984914033	3/32	3/32	3/32	3/32	1/8	0	0	0	0
Shifts	-	22	20	16	13	10	7	5	4	2
Adds	-	40	38	34	31	28	25	23	23	20
MSE	-	$8.2e-6$	$1.1e-5$	$4.0e-5$	$3.6e-4$	$6.9e-4$	$2.2e-3$	$6.3e-3$	$1.3e-2$	$3.2e-2$
$C_g(8)$ (dB)	-	8.8257	8.8242	8.8225	8.8027	8.7716	8.7132	8.5464	8.3416	7.8219
$C_g(4)$ (dB)	-	7.5697	7.5697	7.5697	7.5600	7.5485	7.5485	7.5485	7.1744	7.1503



(a) (b)

Fig. 8. Frequency response of the seventh subband in the binDCT-C5. (a) $X[1]$ and $X[7]$ are not permuted. (b) $X[1]$ and $X[7]$ are permuted.

lifting parameter approximates its analytical value with very high accuracy. On the contrary, the frequency response of the permuted version agrees very well with the true DCT and, therefore, leads to higher coding gain and smaller MSE.

B. Relationship With the WHT

It is interesting to note that in the Chen's factorization of the eight-point DCT, if we remove the intermediate rotation of $\pi/4$, replace all the other rotations by butterflies, and insert a permutation as shown by the dashed box in Fig. 9(a), the factorization would reduce to the Walsh–Hadamard transform, which can be turned into a special binDCT. Hence, the proposed binDCT family can bridge the gap between the WHT and the DCT by increasing the resolution of the approximation. The Loeffler's factorization can also be reduced to the WHT by deleting two of its rotations and adding one more butterfly, as shown in Fig. 9(b).

For comparison, the lifting-based approximation of the DCT in [40] requires 45 additions and 18 shifts. Its coding gain is only 8.692 dB, which is even lower than that of the binDCT-C7 and binDCT-L6, which need only 28 additions and nine shifts as well as 25 additions and seven shifts, respectively. The reason is that in the WHT-based DCT factorization, the WHT is totally

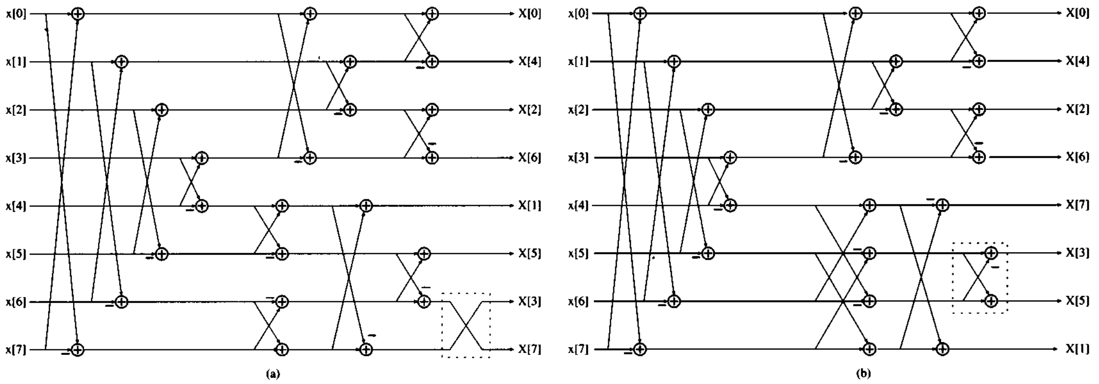


Fig. 9. WHT derived from (a) Chen's factorization of the DCT and (b) Loeffler's factorization of the DCT.

separated from the rotation angles, whereas it is embedded in Chen's and in Loeffler's methods. Besides, the idea of the scaled DCT is not employed in [40].

C. Dynamic Range Analysis

The elimination of the floating-point and fixed-point multiplications enables the binDCT to be implemented with narrower data bus than other fast algorithms. The dynamic range of the binDCT is analyzed in this section using the method as shown in [47].

Assume the original input data are eight-bit signed integers, ranging from -128 to 127, as processed in the JPEG standard [3]. To check the dynamic range of the binDCT, we examine the signs of the binDCT coefficients and find out the set of input data that would generate the maximum or minimum outputs in different binDCT subbands. For example, the signs of the second subband in Table III are

$$\{+ + + + - - - -\}$$

and therefore, the input

$$\{127, 127, 127, 127, -128, -128, -128, -128\}$$

would give the maximum output of this subband, and

$$\{-128, -128, -128, -128, 127, 127, 127, 127\}$$

would lead to its minimum output. The maximum or minimum output of each subband can then be calculated by feeding in those worst-case inputs.

As all lifting parameters in the binDCT are less than unity, they can be implemented with addition and right-shift operations, which can minimize the intermediate dynamic range. In this case, it can be verified that the absolute value of the worst intermediate result in each subband is less than that of its final output. Besides, since the absolute sum of the first row of the binDCT matrix is much greater than that of other rows, the dynamic range of the binDCT is thus determined by the DC subband. With the input range of $[-128, 127]$, the binDCT DC outputs would be within $[-1024, 1016]$. Feeding this into the second pass of the binDCT, the DC outputs of the 2-D binDCT would be within -8192 and 8128, which only need 14 bits to represent. Thus, the binDCT can be well fitted into a 16-bit architecture. This also allows 16-bit implementations of the DCT in video coding applications such as MPEG and H.26x, where the inputs are between -256 and 255 after motion estimation, which only requires one more bit than the JPEG case. Note that we can further reduce the dynamic range to 13 bits if we distribute half of the final down-scaling factors of the inverse transform to the forward side.

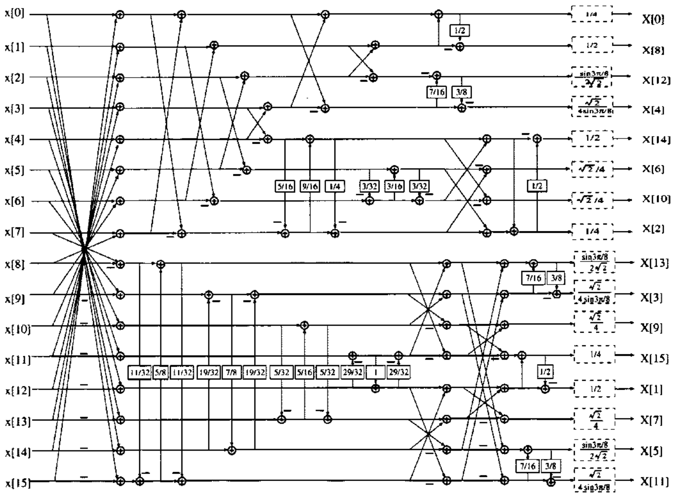


Fig. 10. Sixteen-point binDCT based on Loeffler's factorization: 51 shifts, 106 additions. The coding gain is 9.4499 dB.

It can be verified that the binDCT type L has the same dynamic range as the binDCT type C. That is, it only needs at most 14 bits to perform the 2-D binDCT if the inputs are within -128 and 127. The capability of high-performance implementation of the binDCT with 16-bit simple arithmetic operations makes it very promising for low-cost handheld devices.

D. binDCT of Other Sizes

The same analytical design approach can be applied to generate binDCT of arbitrary size. Any rotation-based fast factorization of the DCT can be employed to reduce the complexity of the binDCT. In this section, a 16-point binDCT will be presented.

An elegant factorization of the 16-point DCT was proposed by Loeffler *et al.* [15], which needs 31 multiplications and 81 additions. Although the lower bound for the number of multiplications of 16-point DCT is 26 [22], the Loeffler's 16-point fac-

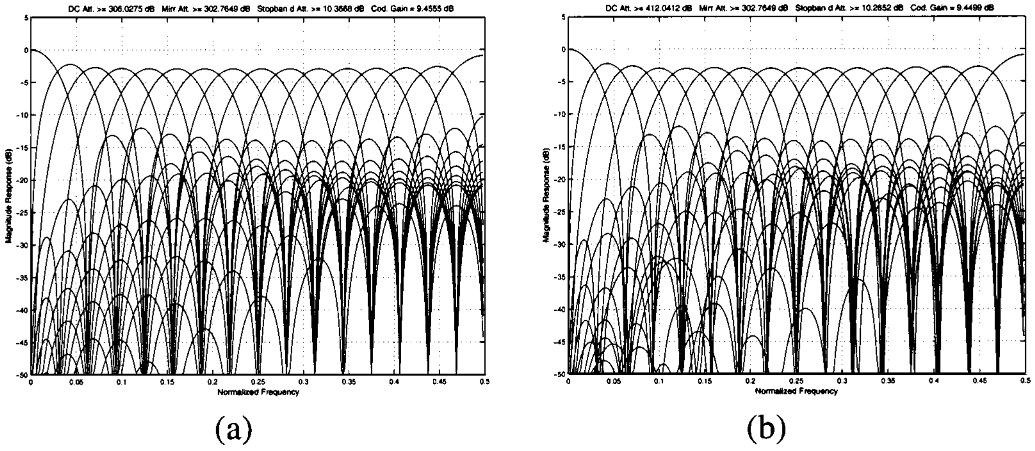


Fig. 11. (a) Frequency response of the 16-point DCT. (b) Frequency response of the 16-point binDCT example—Coding gain: 9.4499 dB.

torization is, thus far, one of the most efficient solutions. Unfortunately, this factorization cannot be generalized to larger sizes.

With our proposed design method, a family of 16-point binDCTs can be easily obtained from this factorization. The general structure and an example is given in Fig. 10. Except the scaling factors, the even part of the 16-point binDCT is identical to the eight-point binDCT type L. The output order is slightly different from that in [15], as the permuted lifting structure is used for some rotations in the final stage. In addition, the negative signs associated with $X[1]$ and $X[11]$ in [15] have been absorbed in the lifting steps. This example requires 51 shifts and 106 additions. Its coding gain is 9.4499 dB, which is very close to the 9.4555 dB coding gain of the true 16-point DCT. The MSE of this approximation is $8.4952E-5$. Its frequency response is depicted in Fig. 11, together with that of the true 16-point DCT.

VIII. Experimental Results

In this section, we demonstrate the applications of the proposed binDCT in JPEG, H.263+, and lossless compression. Comparison with JPEG2000 is not made since it is based on wavelet transform.

A. Performance of the binDCT in JPEG

The proposed families of eight-point binDCTs have been implemented according to the framework of the JPEG standard, based on the source code from the Independent JPEG Group (IJG) [42]. Three versions of DCT implementation are provided in the IJG's code. The floating version is based on the Arai's scaled DCT algorithm with five floating multiplications and 29 additions [3], [8]. The slow integer version is a variation of the Loeffler's algorithm with 12 fixed-point multiplications and 32 additions, and the fast integer version is the Arai's algorithm with five fixed-point multiplications. To apply the binDCT, we replace the DCT part by the proposed binDCT, and the JPEG quantization matrix is modified to incorporate the 2-D binDCT scaling factors.

Fig. 12 compares the PSNR results of the reconstructed Lena image with IJG's floating DCT, IJG's fast integer DCT, and the binDCT-C4. It is observed that the performance of the binDCT is very close to that of the floating DCT in most cases. In particular, when the quality factor is below 95, the difference between the binDCT-C4 and the floating DCT is less than 0.1 dB. Experiments also showed that even the degradation of the binDCT-C7 is less than 0.5 dB. When the quality factor is above 90, the degradations of both fast DCTs become obvious due to the roundoff errors introduced by the scaling factors. However, the result of the binDCT is still reasonable. For example, when the quality factor is 100, the binDCT result is 10.3 dB better than that of the IJG's fastest integer DCT.

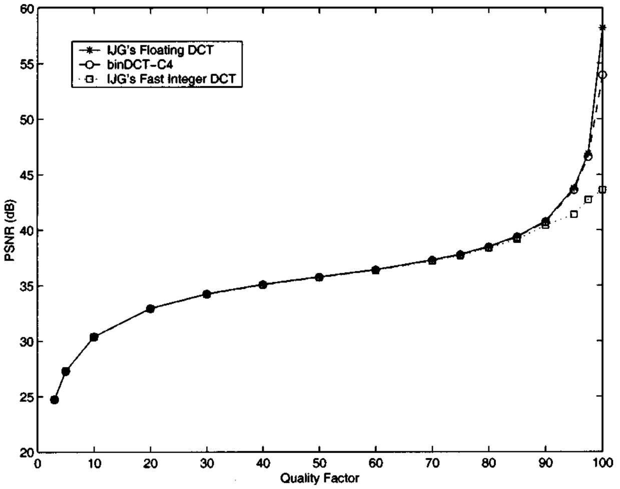


Fig. 12. Comparison of IJG's floating DCT, IJG's fast integer DCT, and binDCT-C4.

In terms of the compression ratio, the compressed file size with binDCT-C7 is about 1–3% smaller than that with the floating DCT, whereas the compressed size with binDCT-C4 is slightly larger than the latter, but the difference is less than 0.5% in most cases.

Table VI compares the compatibility of different fast DCT algorithms with respect to the floating DCT, for which the image Lena is compressed with the floating DCT and decompressed with different fast inverses. It can be seen that the differences among the binDCT-C4, binDCT-L3, and the IJG's fast integer DCT are negligible in most cases. However, the binDCT-L3 has better performance when the quantization step is very small as

TABLE VI
PSNR(dB) OF THE RECONSTRUCTED IMAGE WITH DIFFERENT INVERSE DCT ALGORITHMS

Quality Factor	IJG Int. DCT	IJG Fast Int. DCT	BinDCT C4	BinDCT L3
100	58.85	45.02	44.38	50.12
90	40.79	40.53	40.52	40.66
80	38.51	38.39	38.39	38.44
60	36.43	36.36	36.38	36.38
40	35.11	35.06	35.06	35.06
20	32.95	32.92	32.92	33.31
10	30.40	30.39	30.38	30.37
5	27.33	27.32	27.30	27.30

TABLE VII
EXECUTING TIMES OF DIFFERENT DCT'S FOR A 8 × 8 IMAGE BLOCK

Algorithms	Time ($\times 10^{-6} Sec.$)
IJG Floating DCT	119.05
IJG Int. DCT	4.10
IJG Fast Int. DCT	2.39
binDCT-C1	2.45
binDCT-C4	2.09
binDCT-C7	2.06

the scaling factors of the binDCT-L has smaller roundoff errors than the binDCT-C.

The average executing times of different DCT algorithms for an 8×8 image block are summarized in Table VII, which amounts to repeating the 1-D transform 16 times. These results were measured on a PC with Linux operation system and Pentium-III 550 MHz CPU. It can be seen from the table that the floating DCT is much slower than the other methods. Among the fast algorithms, the binDCT-C4 and the binDCT-C7 are 13–14% faster than the integer Arai's algorithm, which is one of the fastest DCT implementations. However, the binDCT would lose its speed advantage gradually as the complexity increases. For example, the binDCT-C1 is slightly slower than the integer Arai's algorithm.

More significant improvement can be expected if the algorithm is run on low-end CPUs, where the fixed-point multiplication may take many more instruction cycles to process than shift and addition operations. The binDCT can be expected to have tremendous advantage in low-cost hardware implementation in terms of size, speed, and power consumption—all are critical considerations for many hand-held devices.

B. Performance of the binDCT in H.263+

The binDCT has also been implemented in the video coding standard H.263+, based on a public domain H.263+ software [48]. The DCT in the encoder of the selected H.263+ implementation is based on Chen's factorization with floating-point multiplications, and the DCT in the decoder is the scaled version of this method with fixed-point multiplications. In H.263+, a uniform quantization step is applied to all the DCT coefficients of a block. In the binDCT-based version, the quantization step is modified by the 2-D binDCT scaling matrix to maintain compatibility with the standard. In this part, the binDCT-based H.263+ is compared with the original H.263+ software, and some luminance PSNR results of the reconstructed sequence are shown in Fig. 13 for the 400-frame QCIF test video sequence Foreman.

Four scenarios of the configuration of the encoder and the decoder are compared in Fig. 13(a) and (b), with the default quantization steps (40 for I frames and 26 for P frames). The average PSNRs of the reference H.263+ implementation is 30.55 dB. If the binDCT-C4 is used in both the encoder and the decoder, the average PSNR drops to 30.46 dB. However, the compression ratio is improved to 102.67:1 from 101.03:1. If the floating DCT is used by the encoder and the binDCT-C4 is used in the decoder, the average PSNR is 30.43 dB. On the contrary, when the sequence is encoded by the binDCT-C4 and decoded by the default DCT, the average PSNR is 30.39 dB. These results show that the compatibility of the binDCT with other DCT implementations is satisfactory.

In Fig. 13(c), a quantization step of 4 is used for all frames, where the PSNRs given by the binDCT-C4 and binDCT-C7 are 0.79 dB and 0.61 dB higher than that of the reference H.263+, with about 2.5% increase in the file size. In summary, the overall performance of the binDCT-C4-based H.263+ is very similar to the reference H.263+.

C. Performance of the binDCT in Lossless Compression

As previously mentioned, lossless compression can be easily achieved with the lifting-based binDCT by bypassing the scaling factors. To improve the compression ratio, we replace all butterflies in the binDCT by lifting steps, as shown in Fig. 14, which can further reduce the dynamic range of the transform. For instance, the DC coefficient in the modified structure is the average of all inputs, instead of their summation as in the original structure.

The lossless binDCT has been implemented with two coding methods: Huffman and SPIHT [49]. To use Huffman coding, a new Huffman table is obtained by modifying the one in the JPEG standard since the statistical distribution of the binDCT coefficients is different from that of the original DCT coefficients when the scaling factors are bypassed. In the SPIHT method, we rearrange the binDCT coefficients according to the pattern of the wavelet transform coefficients before applying zerotree processing [50].

The binDCT-based lossless transforms are compared with two advanced context model-based prediction methods: the HP LOCO-I [51] and CALIC [52]. The results are summarized in Table VIII, showing that the overall compression ratio of the binDCT-based method is not as good as these methods. However, the proposed binDCT is much simpler, and it provides a unified framework for both lossy and lossless compression.

IX. CONCLUSION

We present the design and application of the binDCT, which is a fast multiplierless approximation of the DCT with the lifting scheme. All the lifting parameters in our design are chosen to be dyadic rationals, enabling fast implementations with only shift and addition operations. Several binDCT families are derived from Chen's and Loeffler's plane rotation-based factorizations of the DCT matrix, respectively, and the design method can be applied to DCT of arbitrary size. Different tradeoffs between the complexity and the performance can be easily achieved by the binDCT. The new transform has

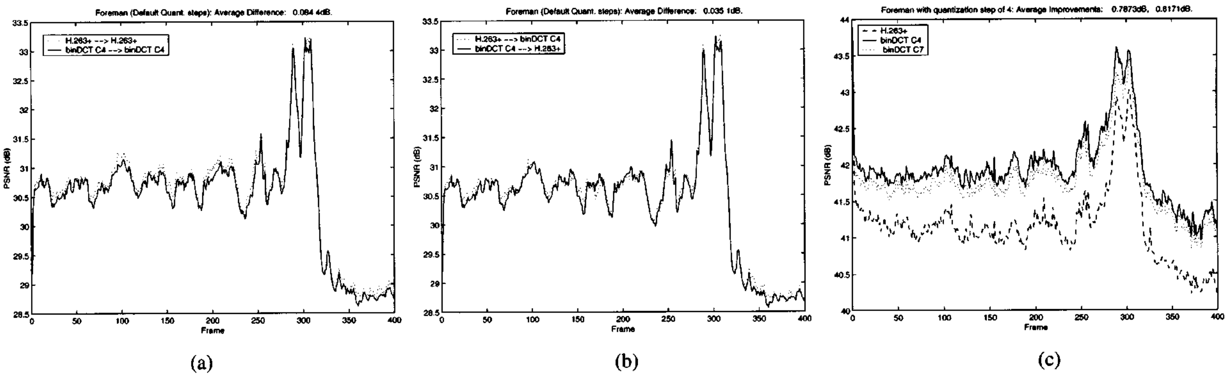


Fig. 13. (a) Comparison between the reference H.263+ and the binDCT-based H.263+. (b) PSNR results with different DCTs in the encoder and the decoder. (c) PSNR results with a quantization step of 4 for all frames.

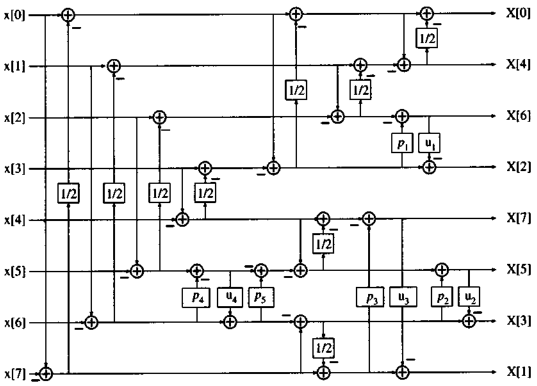


Fig. 14. Lossless binDCT from Chen's factorization with minimized dynamic range.

TABLE VIII
LOSSLESS CODING RESULTS (in BITS PER PIXEL)

Image	binDCT-C4 + Huffman	binDCT-C4 + SPIHT	HP LOCO-I	CALIC
Balloon	3.78	3.58	2.90	2.78
Zelda	4.44	4.33	3.89	3.69
Hotel	5.20	5.07	4.38	4.18
Barbara	5.22	5.11	4.69	4.31
Board	4.34	4.24	3.68	3.51
Girl	4.60	4.50	3.93	3.72
Gold	5.20	5.04	4.48	4.35
Boats	4.67	4.56	3.93	3.78
Average	4.68	4.55	3.99	3.79

been implemented in JPEG, H.263+, and lossless compression with satisfactory performance. Moreover, the binDCT can be implemented with 16-bit data bus, making it very suitable for fast, low-cost, low-power, yet high-performance multimedia computing and communication applications.

ACKNOWLEDGMENT

The authors wish to thank the anonymous reviewers for their careful, meticulous reading of the manuscript and for providing numerous constructive suggestions, which have significantly improved the presentation of the paper.

REFERENCES

[1] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, pp. 90–93, Jan. 1974.

[2] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*. New York: Academic, 1990.

[3] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*. New York: Van Nostrand Reinhold, 1993.

[4] J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. LeGall, *MPEG Video Compression Standard*. New York: Chapman and Hall, 1997.

[5] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards: Algorithms and Architectures*. Boston, MA: Kluwer, 1997.

[6] J. Makhoul, "A fast cosine transform in one and two dimentions," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 27–34, Feb. 1980.

[7] M. Vetterli and H. Nussbaumer, "Simple FFT and DCT algorithms with reduced number of operations," *Signal Processing*, vol. 6, no. 4, pp. 267–278, Aug. 1984.

[8] Y. Arai, T. Agui, and M. Nakajima, "A fast DCT-SQ scheme for images," *Trans. IEICE*, vol. E-71, no. 11, p. 1095, Nov. 1988.

[9] D. Hein and N. Ahmed, "On a real-time Walsh-Hadamard cosine transform image processor," *IEEE Trans. Electromagn. Compat.*, vol. EMC-20, pp. 453–457, Aug. 1978.

[10] S. Venkataraman, V. Kanchan, K. R. Rao, and M. Mohanty, "Discrete transform via the Walsh-Hadamard transform," *Signal Processing*, vol. 14, no. 4, pp. 371–382, June 1988.

[11] H. Malvar, "Fast computation of the discrete cosine transform and the discrete Hartley transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 1484–1485, Oct. 1987.

[12] W. Chen, C. H. Smith, and S. C. Fralick, "A fast computational algorithm for the discrete cosine transform," *IEEE Trans. Commun.*, vol. COMM-25, pp. 1004–1009, Sept. 1977.

[13] W. Chen and C. H. Smith, "Adaptive coding of monochrome and color images," *IEEE Trans. Commun.*, vol. COMM-25, pp. 1285–1292, Nov. 1977.

[14] Z. Wang, "Fast algorithm for the discrete W transform and for the discrete Fourier transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 803–816, Aug. 1984.

[15] C. Loeffler, A. Lightenberg, and G. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications," *Proc. IEEE ICASSP*, vol. 2, pp. 988–991, Feb. 1989.

[16] B. G. Lee, "A new algoithm to compute the discrete cosine transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 1243–1245, Dec. 1984.

[17] H. S. Hou, "A fast recursive algorithm for computing the discrete cosine transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 1455–1461, Oct. 1987.

[18] A. Ligtenberg and J. O'Neill, "A single chip solution for an 8 by 8 two-dimensional DCT," *Proc. IEEE Int. Symp. Circuits Syst.*, pp. 1128–1131, May 1987.

[19] P. Duhamel and C. Guillemot, "Polynomial transform computation of the 2-D DCT," in *Proc. IEEE ICASSP*, 1990, pp. 1515–1518.

[20] S. C. Chan and K. L. Ho, "A new two-dimentional fast cosine transform algorithm," *IEEE Trans. Signal Processing*, vol. 39, pp. 481–485, Feb. 1991.

[21] E. Feig and S. Winograd, "Fast algorithms for the discrete cosine transform," *IEEE Trans. Signal Processing*, vol. 40, pp. 2174–2193, Sept. 1992.

[22] P. Duhamel and H. H'Mida, "New 2^n DCT algorithms suitable for VLSI implementation," in *Proc. ICASSP*, 1987, pp. 1805–1808.

[23] E. Feig and S. Winograd, "On the multiplicative complexity of discrete cosine transform," *IEEE Trans. Inform. Theory*, vol. 38, pp. 1387–1391, July 1992.

[24] M. Vetterli, "Fast 2-D discrete cosine transform," in *Proc. ICASSP*, Mar. 1985, pp. 1538–1541.

[25] E. Feig, "A fast scaled DCT algorithm," in *Proc. SPIE Int. Soc. Opt. Eng.*, vol. 1244, 1990, pp. 2–13.

[26] I. Yun and S. Lee, "On the fixed-point-error analysis of several fast DCT algorithms," *IEEE Trans. Circuits Syst. Vid. Technol.*, vol. 3, pp. 27–41, Feb. 1993.

[27] C. Hsu and J. Yao, "Comparative performance of fast cosine transform with fixed-point roundoff error analysis," *IEEE Trans. Signal Processing*, vol. 42, pp. 1256–1259, May 1994.

[28] Y. Jeong, I. Lee, T. Yun, G. Park, and K. Park, "A fast algorithm suitable for DCT implementation with integer multiplication," *Proc. IEEE TENCON—DSP Appl.*, pp. 784–787, 1996.

[29] W. Cham, "Development of integer cosine transforms by the principle of dyadic symmetry," in *Proc. Inst. Elect. Eng., Part 1*, vol. 136, Aug. 1989, pp. 276–282.

[30] F. Bruekers and A. Enden, "New networks for perfect inversion and perfect reconstruction," *IEEE J. Select. Areas Commun.*, vol. 10, pp. 130–137, Jan. 1992.

[31] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Appl. Comput. Harmon. Anal.*, vol. 3, no. 2, pp. 186–200, 1996.

[32] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting step," *J. Fourier Anal. Appl.*, vol. 4, no. 3, pp. 247–269, 1998.

[33] P. Vaidyanathan and P. Hoang, "Lattice structures for optimal design and robust implementation of two-band perfect reconstruction QMF banks," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 81–94, 1988.

[34] K. Komatsu and K. Sezaki, "Reversible discrete cosine transform," in *Proc. IEEE ICASSP*, vol. 3, May 1998, pp. 1769–1772.

[35] ——, "Design of lossless LOT and its performance evaluation," in *IEEE ICASSP*, vol. 4, June 2000, pp. 2119–2122.

[36] T. D. Tran, "Fast multiplierless approximation of the DCT," in *Proc. 33rd Annu. Conf. Inform. Sci. Syst.*, Mar. 1999, pp. 933–938.

[37] J. Liang and T. D. Tran, "Fast multiplierless approximation of the DCT with the lifting scheme," presented at the Proc. SPIE Apps. Dig. Imagng Process. XXIII, Aug. 2000.

[38] T. D. Tran, "A fast multiplierless block transform for image and video compression," *Proc. IEEE ICIP*, vol. 3, pp. 822–826, Oct. 1999.

[39] ——, "The binDCT: Fast multiplierless approximation of the DCT," *IEEE Signal Processing Lett.*, vol. 7, pp. 141–144, June 2000.

[40] Y. Chen, S. Oraintara, and T. Nguyen, "Integer discrete cosine transform (IntDCT)," in *Proc. 2nd Int. Conf. Inform., Commun. Signa. Process.*, Dec. 1999.

[41] N. Suehiro and M. Hatori, "Fast algorithm for the DFT and other sinusoidal transforms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 642–644, June 1986.

[42] JPEG Image Compression Software [Online]. Available: http://www.ijg.org

[43] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[44] J. Katto and Y. Yasuda, "Performance evaluation of subband coding and optimization of its filter coefficients," in *SPIE Proc. Visual Commun. Image Process.*, Boston, MA, Nov. 1991, pp. 95–106.

[45] G. Strang and T. Nguyen, *Wavelets and Filter Banks*. Wellesley, MA: Wellesley-Cambridge Press, 1997.

[46] R. Queiroz, "On unitary transform approximations," *IEEE Signal Processing Lett.*, vol. 5, pp. 46–47, Feb. 1998.

[47] X. Wan, Y. Wang, and W. Chen, "Dynamic range analysis for the implementation of fast transform," *IEEE Trans. Circiuts Syst. Vid. Technol.*, vol. 5, pp. 178–180, Apr. 1995.

[48] H.263+ Public Domain Code [Online]. Available: http://spmg.ece.ubc.ca/

[49] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circiuts Syst. Vid. Technol.*, vol. 6, pp. 243–250, June 1996.

[50] T. D. Tran and T. Q. Nguyen, "A progressive transmission image coder using linear phase filter banks as block transforms," *IEEE Trans. Image Processing*, vol. 8, pp. 1493–1507, Nov. 1999.

[51] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS," *IEEE Trans. Image Processing*, vol. 9, pp. 1309–1324, Aug. 2000.

[52] X. Wu and N. D. Memon, "Context-based, adaptive, lossless image coding," *IEEE Trans. Commun.*, vol. 45, pp. 437–444, Apr. 1997.

Jie Liang (S'99) received the B.E. and M.E. degrees from Xi'an Jiaotong University, Xi'an, China, in 1992 and 1995, respectively, and the M.E. degree from the National University of Singapore (NUS) in 1998. He has been pursuing the Ph.D. degree at the Department of Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, MD, since 1999.

He was with Hewlett-Packard Singapore and the Center for Wireless Communications, NUS, from 1997 to 1999. His current research interests include multirate signal processing and image/video compression.

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.

He joined the Department of Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, MD, in July 1998 as an Assistant Professor. His research interests are in the field of digital signal processing, particularly in multirate systems, filterbanks, 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 33rd Annual Conference on Information Sciences and Systems (CISS'99), Baltimore, MD, in March 1999. He received the NSF CAREER award in 2001.