Integer Lapped Transforms and Their Applications to Image Coding

W. C. Fong, *Member, IEEE*, S. C. Chan, Arumugam Nallanathan, *Member, IEEE*, and K. L. Ho, *Member, IEEE*

Abstract—This paper proposes new integer approximations of the lapped transforms, called the integer lapped transforms (ILT), and studies their applications to image coding. The ILT are derived from a set of orthogonal sinusoidal transforms having short integer coefficients, which can be implemented with simple integer arithmetic. By employing the same scaling constants in these integer sinusoidal transforms, integer versions of the lapped orthogonal transform (LOT), the lapped biorthogonal transform (LBT), and the hierarchical lapped biorthogonal transform (HLBT) are developed. The ILTs with 5-b integer coefficients are found to have similar coding gain (within 0.06 dB) and image coding performances as their real-valued counterparts. Furthermore, by representing these integer coefficients as sum of powers-of-two coefficients (SOPOT), multiplier-less lapped transforms with very low implementation complexity are obtained. In particular, the implementation of the eight-channel multiplier-less integer LOT (ILOT), LBT (ILBT), and HLBT (IHLBT) require 90 additions and 44 shifts, 98 additions and 59 shifts, and 70 additions and 38 shifts, respectively.

I. INTRODUCTION

PERFECT reconstruction (PR) critically decimated quadrature mirror filter (QMF) banks have many important applications in speech, audio and image processing. The theory and design of M-channel PR maximally decimated uniform filter bank has been extensively studied [7]. The cosine modulated filter banks (CMFB) [9] and the lapped transforms [2], [4], [6] are two efficient classes of filter banks with low implementation complexity and good performance. The lapped orthogonal transform (LOT) is an M-channel orthogonal linear-phase PR filter bank with filter length $2M$. It has slightly higher arithmetic complexity than the DCT but the coding gain is significantly higher with much less blocking artifacts. Other generalizations of the LOT include the biorthogonal generalized lapped transform (GLT) in [4], [6], the lapped biorthogonal transform (LBT) and the hierarchical lapped biorthogonal transform (HLBT) in [14]. These biorthogonal lapped transforms have higher coding gain and reduced blocking artifacts than the LOT. Moreover, the HLBT, when cascaded in a tree structure, can achieve a performance comparable to the embedded zero-tree wavelet coding with very low implementation complexity [14].

Recently, there has been an increasing interest in designing filter banks with low implementation complexity. Approaches based on the sum of powers-of-two (SOPOT) coefficients [3] or integer coefficients [12] have been proposed. By representing the filter coefficients in SOPOT or canonical signed digits representation, coefficient multiplications can be implemented with simple shifts and additions, giving rise to multiplier-less realization. In [12], a subspace approach was proposed to design CMFB with integer-coefficient prototype filter. Since integer-coefficient filter banks require only integer arithmetic (additions and possibly multiplications), the implementation can be greatly simplified. Also, if sufficient word length is used to represent the intermediate data, round off error can be completely eliminated. In practice, rounding has to be performed at various stages of the computation so that the intermediate data can be accommodated in reasonable wordlength, say in 16 or 32 b. Usually, the wordlength of the filter coefficients is chosen to be as small as possible so that only a few or no rounding of the intermediate integer values are required for a certain internal wordlength for storing the intermediate data.

In this paper, integer versions of the various lapped transforms, called the integer lapped transforms (ILT), are proposed and evaluated for image coding applications. The proposed ILT are derived from a set of orthogonal sinusoidal transforms having integer coefficients with small wordlength. An integer wordlength of 5 b is chosen in this work so that no rounding or a few rounding is necessary for 32-b and 16-b internal wordlength, respectively. By properly designing these integer sinusoidal transforms, integer versions of the lapped orthogonal transform (LOT), the lapped biorthogonal transform (LBT), and the hierarchical lapped biorthogonal transform (HLBT) are developed. These ILTs have similar coding gain and image coding performances as their real-valued counterparts. Furthermore, by representing the integer coefficients as sum of powers-of-two coefficients (SOPOT), multiplier-less ILT with very low implementation complexity are obtained.

The paper is organized as follows. Section II reviews the theory and design of integer sinusoidal transforms, which was originally proposed by Cham [1]. Section III is devoted to the theory, design and implementation of the Integer Lapped Orthogonal Transform (ILOT). The results so obtained are furthermore developed in Sections IV and V to obtain the Integer Lapped Biorthogonal Transform (ILBT) and the Integer Hierarchical Lapped Biorthogonal Transform (IHLBT), respectively. Their performances in image coding are then evaluated in Section VI. Finally, the results are summarized in Section VII, the conclusion.

II. INTEGER SINUSOIDAL TRANSFORMS

According to Wang [10], there are four types of DCTs and DSTs (type I to IV). DCT-II and DCT-III are the commonly used discrete cosine transform (DCT) and its inverse transform (IDCT). In this paper, integer versions of the DCT-II (DST-II) and the DCT-IV (DST-IV) are developed to construct the integer lapped transforms (ILT). The definitions of the real-valued DCT-II, DCT-IV, DST-II, and DST-IV are summarized as follows.

DCT

$$[\boldsymbol{C}_M^{II}] = \sqrt{2/M}\{\varepsilon_k \cos(k(n+1/2)\pi/M)\},$$
$$k, n = 0, 1, \ldots, M-1. \tag{2.1}$$

$$[\boldsymbol{C}_M^{IV}] = \sqrt{2/M}\{\cos((k+1/2)(n+1/2)\pi/M)\},$$
$$k, n = 0, 1, \ldots, M-1. \tag{2.2}$$

DST

$$[\boldsymbol{S}_M^{II}] = \sqrt{2/M}\{\varepsilon_k \sin(k(n-1/2)\pi/M)\},$$
$$k, n = 1, \ldots, M. \tag{2.3}$$

$$[\boldsymbol{S}_M^{IV}] = \sqrt{2/M}\{\sin((k+1/2)(n+1/2)\pi/M)\},$$
$$k, n = 0, 1, \ldots, M-1. \tag{2.4}$$

where

$$\varepsilon_i = \begin{cases} 1/\sqrt{2}, & i = 0 \text{ or } M \\ 1, & \text{otherwise.} \end{cases}$$

The subscript and the superscript of $\boldsymbol{C}$ and $\boldsymbol{S}$ denote, respectively, the length and the type of the transformation. The type-II integer cosine transform (ICT), $\boldsymbol{T}_M^{C-II}$, proposed in [1] has the following matrix representation:

$$\boldsymbol{P}_M^{C-II} = \boldsymbol{K}_M^{C-II} \cdot \boldsymbol{T}_M^{C-II} \tag{2.5}$$

where $\boldsymbol{K}_M^{C-II}$ is a real diagonal matrix and $\boldsymbol{P}_M^{C-II}$ is an orthogonal matrix approximating DCT-II. Again, the superscript and subscript denote, respectively, the type and the length of the transformation. For example, $C-II$ means type-II discrete cosine transform and so on. Using the concept of dyadic symmetry, Cham [1] proposed the following construction for order-8 ICT

$$\boldsymbol{T}_8^{C-II} = \begin{bmatrix} k & k & k & k & k & k & k & k \\ a & b & c & d & -d & -c & -b & -a \\ e & f & -f & -e & -e & -f & f & e \\ b & -d & -a & -c & c & a & d & -b \\ k & -k & -k & k & k & -k & -k & k \\ c & -a & d & b & -b & -d & a & -c \\ f & -e & e & -f & -f & e & -e & f \\ d & -c & b & -a & a & -b & c & -d \end{bmatrix} \tag{2.6}$$

where a, b, c, d, e, f, and k are integer parameters. It can be verified that the rows of $\boldsymbol{T}_8^{C-II}$ are orthogonal to each other if

$$a \cdot b = a \cdot c + b \cdot d + c \cdot d. \tag{2.7}$$

In addition, a, b, c, d, e, and f are chosen to satisfy the conditions

$$a \geq b \geq c \geq d \geq 0, \quad \text{and} \quad e \geq f \geq 0 \tag{2.8}$$

so that the basis functions of the ICT will more closely resemble those of the DCT-II. In [1], the constants a, b, c, d, e, f, and k are chosen to be 8-b integers and the (i, i)th entry of the normalization matrix $\boldsymbol{K}_M^{C-II}$, k_i, is used to normalize the ith row of $\boldsymbol{T}_M^{C-II}$ so that $\boldsymbol{P}_M^{C-II}$ is orthogonal. If T_i is the L_2-norm of the ith row of $\boldsymbol{T}_M^{C-II}$, then k_i is given by

$$k_i = |1/T_i|. \tag{2.9}$$

An exhaustive search is then performed to maximize the coding gain for common AR(1) process. The multiplications with k_is are usually absorbed into the quantization process to reduce the arithmetic complexity required. The inverse of the ICT can be obtained from (2.5) as follows:

$$\left(\boldsymbol{P}_M^{C-II}\right)^{-1} = \left(\boldsymbol{P}_M^{C-II}\right)^T = \left(\boldsymbol{K}_M^{C-II} \cdot \boldsymbol{T}_M^{C-II}\right)^T$$
$$= \left(\boldsymbol{T}_M^{C-II}\right)^T \left(\boldsymbol{K}_M^{C-II}\right)^T. \tag{2.10}$$

Similarly, the scaling $(\boldsymbol{K}_M^{C-II})^T$ can be absorbed into the dequantization process to reduce the arithmetic complexity. Using these integer sinusoidal transforms, it is possible to construct lapped transforms with integer coefficients, as we shall see in the following section.

III. INTEGER LAPPED ORTHOGONAL TRANSFORM (ILOT)

A. Theory and Design of ILOT

As mentioned earlier, the LOT is an orthogonal linear phase M-channel PR filter bank with length $2M$. Its $(M \times 2M)$ transform matrix is given by [2]

$$\boldsymbol{P}_M^{LOT} = \boldsymbol{P}_M \begin{bmatrix} \boldsymbol{I}_{M/2} & \boldsymbol{0}_{N/2} \\ \boldsymbol{0}_{M/2} & \left(\boldsymbol{C}_{M/2}^{II}\boldsymbol{S}_{M/2}^{IV}\right)^T \end{bmatrix}$$
$$\cdot \begin{bmatrix} \boldsymbol{D}_e - \boldsymbol{D}_o & \boldsymbol{J}_{M/2}(\boldsymbol{D}_e - \boldsymbol{D}_o) \\ \boldsymbol{D}_e - \boldsymbol{D}_o & -\boldsymbol{J}_{M/2}(\boldsymbol{D}_e - \boldsymbol{D}_o) \end{bmatrix} \tag{3.1}$$

where $\boldsymbol{D}_e$ and $\boldsymbol{D}_o$ are $(M/2) \times (M)$ matrices containing the even and odd basis functions of DCT-II, respectively. $\boldsymbol{P}_M$ is a permutation matrix which permutes the kth and the $(k+M/2)$th rows to the $2k$th and the $(2k+1)$th $(k = 0, \ldots, M/2 - 1)$ rows, respectively. Fig. 1 shows the flow graph of an eight-channel LOT. It is natural to consider replacing the eight-point DCT in the LOT by the ICT $\boldsymbol{P}_8^{C-II}$ to construct an integer LOT (ILOT). Unfortunately, the scaling matrix, $\boldsymbol{K}_8^{C-II}$, is real-valued. Therefore, it is necessary to perform real multiplications immediately after $\boldsymbol{T}_8^{C-II}$ before subsequent additions in the LOT can be performed. A method to avoid this problem is to design an integer cosine transform with all the scaling constants being identical. In this case, the scaling constant can be moved to the end of the transformation and absorbed in the quantization process. The remaining problem is to find similar integer transforms for the four-point DCT-II and the four-point DST-IV. Using the same concept as in [1], we propose the order-4 type-II integer cosine transform as follows:

$$\boldsymbol{T}_4^{C-II} = \begin{bmatrix} \ell & \ell & \ell & \ell \\ a_1 & b_1 & -b_1 & -a_1 \\ \ell & -\ell & -\ell & \ell \\ b_1 & -a_1 & a_1 & -b_1 \end{bmatrix} \tag{3.2}$$
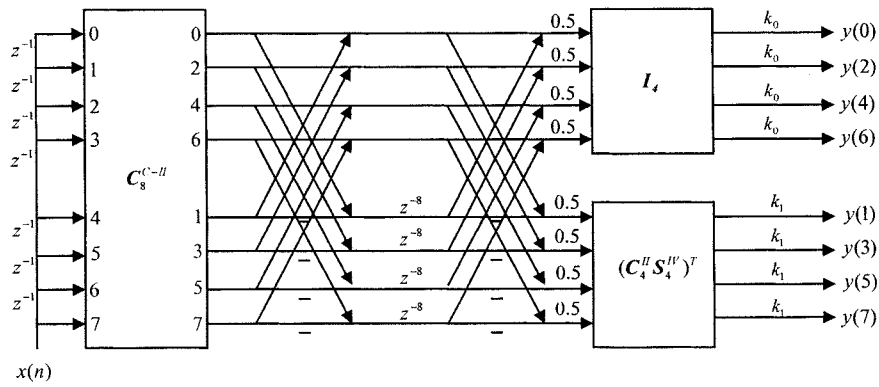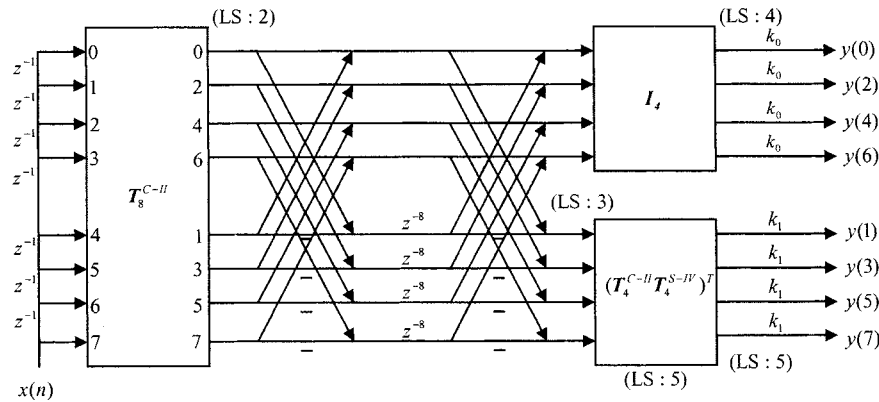
Fig. 1.   Flow graph of eight-channel LOT.

Fig. 2.   Flow graph of eight-channel ILOT (LS: left shift).

where a_1, b_1, and ℓ are integers to be determined. Like T_8^{C-II}, a_1 and b_1 have to satisfy

$$a_1 \geq b_1 \geq 0. \tag{3.3}$$

Similarly, the order-4 type-IV integer sine transform is defined as follows:

$$T_4^{S-IV} = \begin{bmatrix} a_2 & b_2 & c_2 & d_2 \\ b_2 & d_2 & a_2 & -c_2 \\ c_2 & a_2 & -d_2 & b_2 \\ d_2 & -c_2 & b_2 & -a_2 \end{bmatrix}. \tag{3.4}$$

Again, the rows of T_4^{S-IV} will be orthogonal to each other if

$$c_2 \cdot d_2 = a_2 \cdot b_2 + b_2 \cdot d_2 + c_2 \cdot a_2. \tag{3.5}$$

Moreover, to approximate the basis functions of DST-IV, the following condition is imposed

$$a_2 \geq b_2 \geq c_2 \geq d_2 \geq 0. \tag{3.6}$$

For simplicity, the scaling constants for the matrix product, $(T_4^{C-II} T_4^{S-IV})^T$, are also designed to be identical. Finally, the following transform matrix for the order-8 ILOT is obtained

$$P_8^{ILOT} = P_M \begin{bmatrix} k_0 I_4 & 0_4 \\ 0_4 & k_1 I_4 \end{bmatrix} \begin{bmatrix} I_4 & 0_4 \\ 0_4 & (T_4^{C-II} T_4^{S-IV})^T \end{bmatrix}$$

$$\cdot \begin{bmatrix} \tilde{D}_e - \tilde{D}_o & J_4(\tilde{D}_e - \tilde{D}_o) \\ \tilde{D}_e - \tilde{D}_o & -J_4(\tilde{D}_e - \tilde{D}_o) \end{bmatrix} \tag{3.7}$$

where $\tilde{D}_e$ and $\tilde{D}_o$ are (4×8) matrices containing the even and odd basis functions of T_8^{C-II}, respectively. The scaling constant for T_8^{C-II} is k_0. For the odd-indexed outputs, the scaling constants of T_8^{C-II} and $(T_4^{C-II} T_4^{S-IV})^T$ are merged together to form k_1. These multiplications can readily be absorbed into the quantization process. Fig. 2 shows the flow graph of the integer LOT with eight channels. This ILOT is parameterized by a set of integers and is referred to as ILOT$(a, b, c, d, e, f, k, a_1, b_1, \ell, a_2, b_2, c_2, d_2)$.

We now proceed to the design of the ILOT. An important issue in transform and subband coding is the efficiency of the transform or filter bank employed. The coding gain is frequently used as an effective measure of transform efficiency. For orthogonal transform, the coding gain, G_T, is given by

$$G_T = \frac{\frac{1}{M} \sum_{i=0}^{M-1} \sigma_i^2}{\left(\prod_{i=0}^{M-1} \sigma_i^2 \right)} \tag{3.8}$$

where σ_i^2 is the variance of the output at the ith analysis filter. For subband coding, the general coding gain formula is given by [11]

$$G_{SBC} = \frac{1}{M \left(\prod_{k=0}^{M-1} A_k B_k \right)^{1/M}} \tag{3.9}$$

where

$$A_k = \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} h_k(j) h_k(i) \rho^{|i-j|} \quad \text{and} \quad B_k = \frac{1}{M} \sum_{j=0}^{N-1} g_k^2(j).$$
(3.10)

$h_k(n)$ and $g_k(n)$ are the impulse responses of the kth analysis and synthesis filters of length N. Here, the input is assumed to be a first-order auto-regressive process with correlation coefficient ρ.

To reduce the wordlength for the integer implementation, the kernel for T_8^{C-II}, T_4^{C-II} and T_4^{S-IV} are chosen to be 5-b integers. Exhaustive search is performed to optimize the coding gain of this ILOT. The optimal solution was found to be ILOT(24, 20, 12, 6, 23, 7, 17, 17, 7, 13, 3, 6, 10, 12) with a coding gain of 9.16 dB for AR(1) process with correlation coefficient of 0.95. This is close to the coding gain of the LOT, which is 9.22 dB. They also have similar frequency responses [13], which is not shown here due to page limitation. The diagonal scaling values k_0 and k_1 are 0.0104 and 0.000 023 53, respectively. The multiplier-less implementation of this ILOT will be described in the next section.

B. Multiplier-Less ILOT

It can be seen from Section III-A that the arithmetic complexity of the ILOT is mainly due to the matrix multiplications of T_8^{C-II}, T_4^{C-II} and T_4^{S-IV}. In the LOT, fast algorithms for C_8^{II} and $(C_4^{II} S_4^{IV})^T$ are available [5], [15], which can be implemented either in fixed-point or floating-point arithmetic. Due to the use of integer arithmetic, such fast algorithms are in general not available. Fortunately, because of the small word-length of the integer coefficients, the arithmetic computations required can be significantly reduced by exploiting the actual values and the sum of powers-of-two representation of the coefficients. As a result, multiplier-less implementation of the ILOT becomes feasible. First of all, let us consider the multiplication of $(T_4^{S-IV})^T = T_4^{S-IV}$. It can be seen that $a_2 = 3$, $b_2 = 6$, and $d_2 = 12$ are multiples of 3, and $c_2 = 10$ can be rewritten as $c_2 = 6 + 4$. By proper arranging the operation, we can implement $X = T_4^{S-IV} x$, where $X = [X(0), X(1), X(2), X(3)]^T$ and $x = [x(0), x(1), x(2), x(3)]^T$, as follows:

$$X(0) = 3((x(0) + 2x(1)) + 2x(2) + 4x(3)) + 4x(2)$$
$$X(1) = 3(2(x(0) + 2x(1)) + x(2) - 2x(3)) - 4x(3)$$
$$X(2) = 3(2x(0) + x(1) - 2(2x(2) - x(3))) + 4x(0)$$
$$X(3) = 3(4x(0) - 2x(1) + (2x(2) - x(3))) - 4x(1).$$
(3.11)

Careful examination reveals that (3.11) can be implemented in 18 additions and 14 shifts. The implementation of T_8^{C-II} and T_4^{C-II} are slightly simpler due to the dyadic symmetry. In fact, they can be further decomposed as follows:

$$T_8^{C-II} = P_8 \begin{bmatrix} T_4^{C-II} & 0 \\ 0 & T_4^{C-IV} \end{bmatrix} \begin{bmatrix} I_4 & J_4 \\ I_4 & -J_4 \end{bmatrix}$$
(3.12)

where

$$P_8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$T_4^{C-IV} = \begin{bmatrix} d & c & b & a \\ -c & -a & -d & b \\ b & d & -a & c \\ -a & b & -c & d \end{bmatrix}.$$

$$T_4^{C-II} = P_4 \begin{bmatrix} k T_2^{C-II} & 0 \\ 0 & T_2^{C-IV} \end{bmatrix} \begin{bmatrix} I_2 & J_2 \\ I_2 & -J_2 \end{bmatrix}$$
(3.13)

where

$$P_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad T_2^{C-IV} = \begin{bmatrix} f & e \\ -e & f \end{bmatrix}$$

and

$$T_2^{C-II} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Here, J_N is the $(N \times N)$ exchange matrix. The flowgraphs for decomposing T_4^{C-II} and T_8^{C-II} are shown in Fig. 3(a) and (b), respectively. Next, we shall consider the multiplication of T_4^{C-IV}. Using similar rearrangement as in (3.11), one can implement $X = T_4^{C-IV} x$, where $X = [X(0), X(1), X(2), X(3)]^T$ and $x = [x(0), x(1), x(2), x(3)]^T$, as follows:

$$X(0) = 6((x(0) + 2x(1)) + 4x(2) + 4x(3)) - 4x(2)$$
$$X(1) = 6(-2(x(0) + 2x(1)) - x(2) + 4x(3)) - 4x(3)$$
$$X(2) = 6(4x(0) + x(1) - 2(2x(2) - x(3))) - 4x(0)$$
$$X(3) = 6(-4x(0) + x(1) - (2x(2) - x(3))) - 4x(1)$$
(3.14)

which requires 18 additions and 16 shifts. To perform the matrix multiplication for T_2^{C-IV}, we can decompose $e = 23$ and $f = 7$ into SOPOT coefficients and implement the matrix multiplication $[X(0), X(1)]^T = T_2^{C-IV}[x(0), x(1)]^T$ in six additions and four shifts as follows:

$$X(0) = 16x(1) + (8 - 1)(x(0) + x(1))$$
$$X(1) = (8 - 1)(-x(0) + x(1)) - 16x(0).$$
(3.15)

Therefore, T_8^{C-II} can be implemented with 40 additions and 22 shifts (two additions and two shifts for multiplication with $k = 17$). On the other hand, the multiplication of $(T_4^{C-II})^T$ can be implemented by transposing both sides of (3.13) as follows:

$$(T_4^{C-II})^T = \begin{bmatrix} \ell T_2^{C-II} & 0 \\ 0 & T_2^{C-IV} \end{bmatrix} \begin{bmatrix} I_2 & I_2 \\ J_2 & -J_2 \end{bmatrix} P_4^T.$$
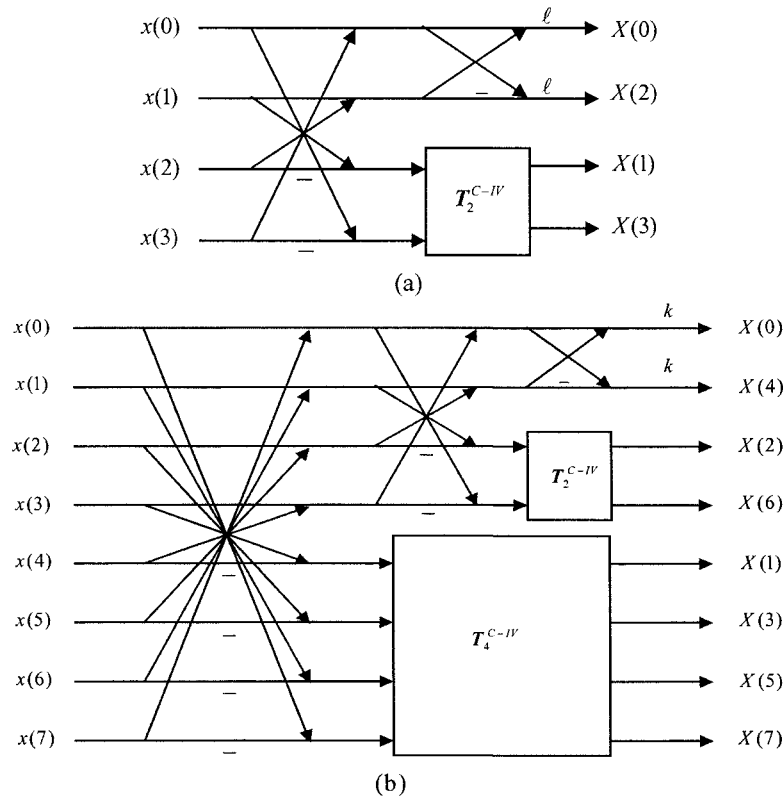(3.16)

Fig. 3.   Flow graphs for computing (a) T_4^{C-II} and (b) T_8^{C-II}.

Since the multiplication with $\ell = 13$ requires four additions and four shifts, the arithmetic complexity required for $(T_4^{C-II})^T$ is 16 additions and eight shifts. Combining all these results, the arithmetic complexity of the eight-point ILOT is 90 additions and 44 shifts.

IV. INTEGER LAPPED BIORTHOGONAL TRANSFORM (ILBT)

A. Theory and Design of Eight-Point ILBT

The LBT is a biorthogonal linear-phase M-channel perfect reconstruction filter banks with length $2M$. It and the GLT are biorthogonal generalizations of the LOT. The advantages of these biorthogonal filter banks are that they can be designed to reduce blocking artifacts and improve the coding gain. The polyphase matrix of the GLT is defined as follows [4]:

$$E(z) = \tfrac{1}{2} P^T \begin{bmatrix} U_{00} & 0_{M/2} \\ 0_{M/2} & U_{11} \end{bmatrix} \begin{bmatrix} I_{M/2} & 0_{M/2} \\ 0_{M/2} & \left(C_{M/2}^{II} S_{M/2}^{IV}\right)^T \end{bmatrix}$$

$$\cdot \begin{bmatrix} I_{M/2} & I_{M/2} \\ I_{M/2} & -I_{M/2} \end{bmatrix} \begin{bmatrix} I_{M/2} & 0_{M/2} \\ 0_{M/2} & z^{-1} I_{M/2} \end{bmatrix} R_1 \quad (4.1)$$

where U_{00} and U_{11} are block diagonal nonsingular matrices

$$R_1 = P_M^T \, diag\{B_2 \cdots B_2\} D C_M^{II} J_M$$

$$B_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \qquad \text{and } D \text{ is a diagonal matrix.}$$

GLT with greater overlap can similarly be defined. In the LBT

$$U_{00} = U_{11} = I_{M/2}, \qquad D = diag(1, m, 1, \ldots 1)$$

and

$$m = \sqrt{2} \quad (4.2)$$

which was first reported in [14]. The signal flow graph for the LBT is shown in Fig. 4. By approximating the constant m as the rational number 17/12 and using the previous results for the ILOT, the integer eight-channel length-16 LBT (ILBT) in Fig. 5 is obtained. Its arithmetic complexity is 98 additions and 59 shifts. The frequency responses of the LBT and its integer version are shown in Fig. 6. It can be seen that they are similar to each other. In the next section, we shall consider the construction of the eight-channel integer HLBT from an four-channel ILBT.

V. INTEGER HIERARCHICAL LAPPED BIORTHOGONAL TRANSFORM (IHLBT)

A. Theory and Design of IHLBT

Although the LBT has a higher coding gain and lower blocking artifacts than the LOT, it is not very suitable to be used in a hierarchical manner like the wavelet transform. This is because the high frequency basis functions will become very long producing excessive ringing artifacts if several such M-channel filter banks are cascaded together. One solution to this problem is to design lapped transforms having basis functions with different lengths. The HLBT proposed by Malvar [14] is a nonuniform filter bank, which is constructed by combining two LBTs with half the desired block size, as shown in Fig. 7(a). More specifically, the lowest coefficients of the low-order LBT are further transformed using a length-2 DCT in the second
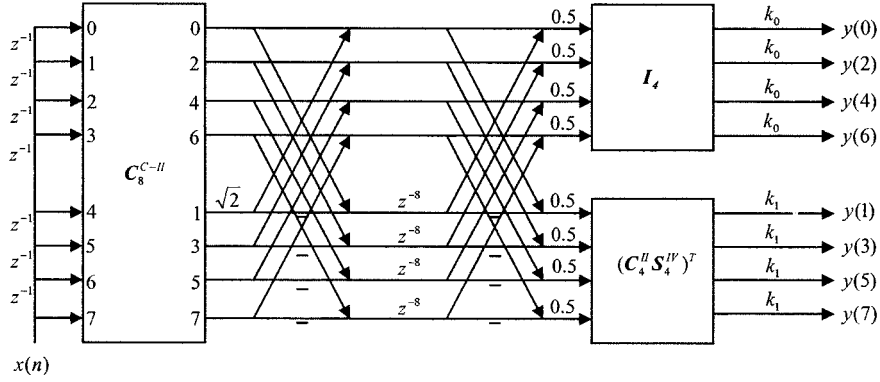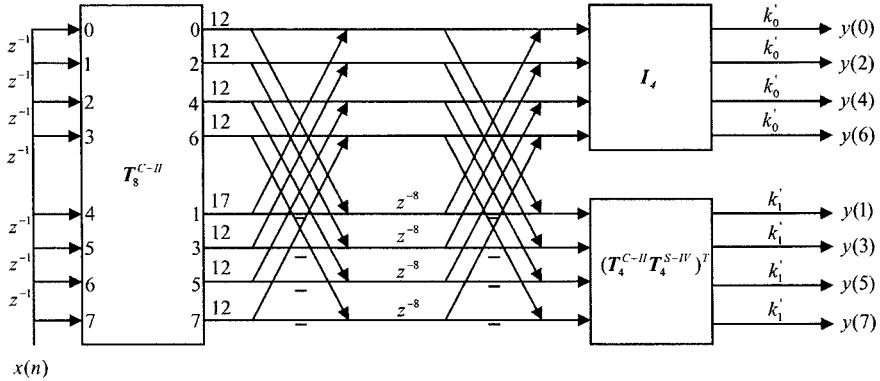
Fig. 4. Flow graph of eight-channel LBT.

Fig. 5. Flow graph of eight-channel ILBT. $k_0' = k_0/12$ and $k_1' = k_1/12$.

level. The first two basis functions and the high frequency basis functions are therefore of length $3M/2$ and M, respectively. When cascaded in a tree structure, performance comparable to the embedded zero-tree wavelet coding can be achieved with very low implementation complexity [14].

With the help of the integer sinusoidal transforms and their multiplier-less implementation, the construction of the integer HLBT (IHLBT) can be considerably simplified. More precisely, the IHLBT is obtained by replacing the four-channel LBT by the four-channel ILBT developed in Section IV. The diagonal scaling values k_0 and k_1 for the four-channel LBT are 0.019 230 7 and 0.002 027 1, respectively. The four-channel ILBT involves the implementation of the integer matrices $\boldsymbol{T}_4^{C-II}$, $\boldsymbol{T}_2^{C-II}$, and $\boldsymbol{T}_2^{S-IV}$. The implementation of $\boldsymbol{T}_4^{C-II}$ has been discussed in Section III, which requires 16 additions and eight shifts. Whereas, the implementation of $\boldsymbol{T}_2^{C-II}$ and $\boldsymbol{T}_2^{S-IV} = \begin{bmatrix} a_2 & b_2 \\ b_2 & -a_2 \end{bmatrix}$ require, respectively, two additions and four additions and four shifts.

VI. IMAGE CODING PERFORMANCE

In this section, we shall compare the image coding performance of the integer lapped transforms with their real-valued counterparts. Separable eight-channel length-16 LOT, LBT, and the HLBT are used for simulation. For a fair comparison between the systems, we follow the JPEG Baseline quantization scheme with the default Huffman code tables. The DCT is replaced by the various transforms and the quantization table is obtained by using the bit allocation algorithm proposed in [16]

without using any human visual model. The coders were tested with the "Lena" image (512 × 512 8-b gray scale) and the coding results are summarized in Table I. It can be seen that the difference in PSNR of the proposed integer lapped transforms and their real-valued counterparts is within 0.1 dB. Their visual qualities are also found to be close to each other. The arithmetic complexities and the coding gains of the various lapped transforms with $M = 8$ are summarized in Table II. The arithmetic complexities of the real-valued DCT-II and DST-IV are obtained from [15].

As mentioned in the introduction, rounding of the intermediate integers might be needed to avoid the growth in wordlength of the intermediate data, depending on the internal wordlength of the processing unit. For example, in the implementation of the ILOT using 32-b internal wordlength, the even outputs $y(0), y(2), y(4)$, and $y(6)$ before multiplying by the constant k_0 can be limited to 16 b by shifting them to the left by 2 b. Similarly, for the odd-indexed outputs, the results before multiplying by the constant k_1 can be limited to 16 b by shifting the results by 10 b. Therefore, after performing the row and column transformations, the output will be limited to 32 b by carrying out only one rounding operation per transformation without any multiplication. The outputs are then quantized (a division or compare operation) with a stepsize of $2^{LS} k_i k_j \Delta, i = 0, 1$ (depending on their indices), where Δ is the stepsize for the real-valued transformation and LS is the total amount of left shifts that has been performed for the given output. This differs from using fixed-point arithmetic, which usually requires a hardware multiplier together with a rounding operation so that the intermediate data after multiplication can be fitted

TABLE I
CODING RESULTS OF VARIOUS ALGORITHMS FOR IMAGE LENA [GREY SCALE (512 × 512)]

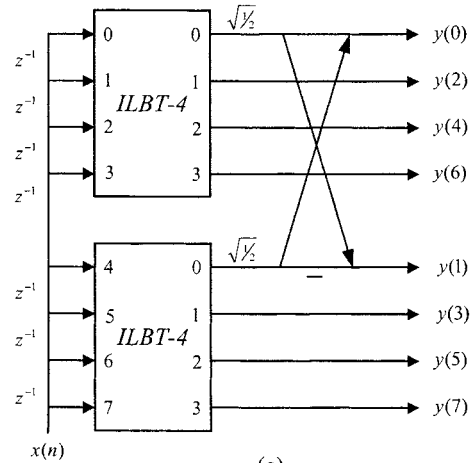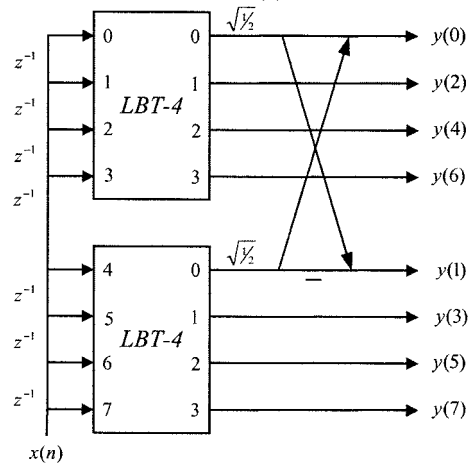| Lena	PSNR (dB)						
bpp	DCT	LOT	ILOT	GLT	IGLT	HLBT	IHLBT
0.62	36.27	36.43	36.23	36.85	36.72	36.25	36.31
0.5	35.04	35.29	35.08	35.77	35.65	35.12	35.15
0.35	32.97	33.40	33.22	33.98	33.90	33.22	33.32
0.26	30.88	31.42	31.33	31.94	31.91	31.14	31.23
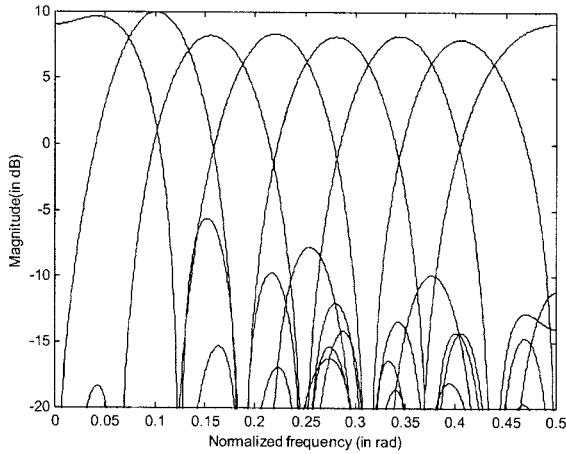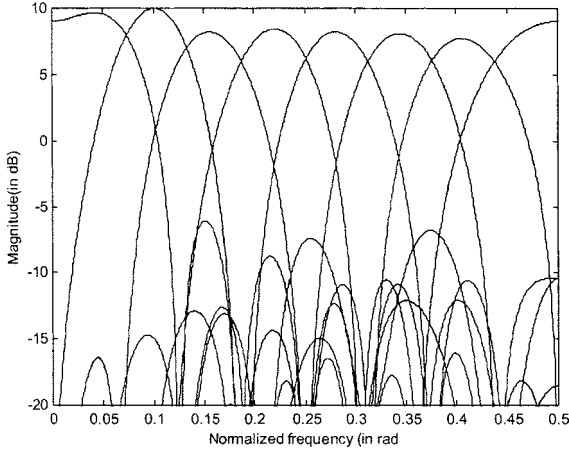
Fig. 6.   Frequency response of (a) eight-channel LBT (analysis filterbanks) and (b) eight-channel ILBT (analysis filterbanks).

Fig. 7.   Flow graphs: (a) eight-channel HLBT and (b) eight-channel IHLBT.

into the given internal wordlength. Simulation results also show that all the ILTs considered can be implemented in 16-b arithmetic without any notable differences in their PSNR and visual performances with proper rounding. For the ILOT, the outputs from T_8^{C-II} can be left shifted by 2 b (Fig. 2). Before performing T_4^{C-II} and T_4^{S-IV}, the intermediate data can be left shifted by 3 b and 5 b, respectively. Finally, the even and odd indexed results before multiplying by k_0 and k_1 can be left shifted by 4 and 5 b, respectively. Hence, the overall implementation can be accommodated in 16-b arithmetic, with slightly more rounding than using 32-b arithmetic.

VII. CONCLUSION

A new generalization of lapped transforms with integer coefficients, called the integer lapped transform (ILT), for image coding applications is presented. The ILT is derived from a set of orthogonal sinusoidal transforms having short integer coefficients, which allows them to be implemented with simple integer arithmetic. By employing the same scaling constants for these integer sinusoidal transforms, integer versions of the lapped orthogonal transform (LOT), the generalized lapped transform (GLT), and the hierarchical lapped transform (HLBT) are developed. ILT with 5-b integer coefficients was found to have similar coding gain (within 0.06 dB) and

TABLE II
CODING GAINS AND ARITHMETIC COMPLEXITIES OF VARIOUS ALGORITHMS ($M = 8$)

	DCT	ICT	LOT	ILOT	GLT	IGLT	HLBT	IHLBT
Coding Gain (dB)	8.83	8.77	9.22	9.16	9.55	9.50	9.15	9.18
Arithmetic Complexity	29 Adds 12 Mults	40 Adds 22 Shifts	66 Adds 41 Mults	90 Adds 44 Shifts	66 Adds 42 Mults	98 Adds 59 Shifts	46 Adds 14 Mults	70 Adds 38 Shifts

image coding performances as their real-valued counterparts. Furthermore, by representing these integer coefficients as sum of powers-of-two coefficients (SOPOT), multiplier-less lapped transforms with very low implementation complexity are obtained. Image coding results also confirm that their PSNR and visual performances are very close to their real-valued counterparts.

REFERENCES

[1] W. K. Cham, "Development of integer cosine transforms by the principle of dyadic symmetry," *Proc. Inst. Elect. Eng.*, vol. 1364, pp. 276–282, Aug. 1989.

[2] H. S. Malvar, *Signal Processing With Lapped Transforms*. Boston, MA: Artech House, 1992.

[3] B. R. Horng, H. Samueli, and A. N. Wilson, "The design of two-channel lattice-structure perfect-reconstruction filter banks using powers-of-two coefficients," *IEEE Trans. Circuits Syst. I*, vol. 40, pp. 497–499, July 1993.

[4] S. C. Chan, A. Nallanathan, T. S. Ng, and P. C. K. Kwok, "A class of M-channel linear-phase biorthogonal filter banks and their applications to subband coding," *IEEE Trans. Signal Processing*, vol. 47, pp. 564–571, Feb. 1999.

[5] S. C. Chan and K. L. Ho, "Fast algorithms for computing the discrete cosine transform," *IEEE Trans. Circuits Syst. II*, vol. 39, pp. 185–190, Mar. 1992.

[6] S. C. Chan, "The generalized lapped transform (GLT) for subband coding applications," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Detroit, MI, May 1995, pp. 1508–1511.

[7] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[8] T. Q. Nguyen and P. P. Vaidyanathan, "Two-channel perfect-reconstruction FIR QMF structures which yield linear-phase analysis and synthesis filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 676–690, May 1989.

[9] R. D. Koilpillai and P. P. Vaidyanathan, "Cosine-modulated FIR filter banks satisfying perfect reconstruction," *IEEE Trans. Signal Processing*, vol. 40, pp. 770–783, Apr. 1992.

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

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

[12] A. Mertins, "Discrete-coefficient linear-phase prototypes for PR cosine-modulated filter banks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 1998, pp. 1433–1436.

[13] A. Nallanathan, S. C. Chan, and T. S. Ng, "Lapped orthogonal transform with integer coefficients," in *Proc. 42nd Midwest Symposium on Circuits and Systems*, 2000, pp. 1065–1068.

[14] H. S. Malvar, "Biorthogonal and nonuniform lapped transforms for transform coding with reduced blocking and ringing artifacts," *IEEE Trans. Signal Processing*, vol. 46, pp. 1043–1053, Apr. 1998.

[15] S. C. Chan and K. L. Ho, "Direct methods for computing the discrete sinusoidal transforms," *Proc. Inst. Elect. Eng. F*, vol. 137, no. 6, pp. 433–442, Dec. 1990.

[16] W. C. Fong, S. C. Chan, and K. L. Ho, "Designing JPEG quantization matrix using rate-distortion approach and human visual system model," in *Proc. IEEE ICC'97*, pp. 1659–1663.

W. C. Fong (M'97) received the B.S. and the M.S. degrees in physics from Jinan University, Guangzhou, China, in 1986 and 1989, respectively, and the Ph.D. degree in electronic engineering from the University of Hong Kong in 1997.

Since 1997, she has been a Research Associate in the Department of Electrical and Electronic Engineering, University of Hong Kong. Her current research interests include digital image/video processing, data compression, and digital signal processing.
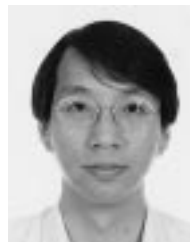
S. C. Chan received the B.Sc. and Ph.D. degrees in electrical engineering from the University of Hong Kong in 1986 and 1992, respectively.

From 1990 to 1994, he was a Lecturer in the Department of Electronic Engineering, City University of Hong Kong. He joined the Department of Electrical and Electronic Engineering, University of Hong Kong, in 1994, where he is now an Associate Professor. His main research interests are in digital signal processing and their applications to communications and data compression. He was a Visiting Researcher at Microsoft Research, Redmond, WA, and Microsoft Research, Beijing, China, in 1998 and 1999, respectively.

Arumugam Nallanathan (S'97–M'00) received the B.Sc. degree from the University of Peradeniya, Sri Lanka, in 1991, the CPGS degree from the University of Cambridge, Cambridge, U.K., in 1994, and the Ph.D. degree from the University of Hong Kong in 2000, all in electrical engineering.

From 1991 to 1992, he was an Assistant Lecturer in the Department of Electrical and Electronic Engineering, University of Peradeniya. From 1994 to 1996, he was a Lecturer in the Department of Electrical and Electronic Engineering, University of Peradeniya. Currently, he is an Assistant Professor in the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. He is a Faculty Associate of Institute for Communications Research (ICR), Singapore. His research interests include digital signal processing, wireless communications, spread spectrum techniques, coding, and modulation.

K. L. Ho (M'80) received the B.Sc.(Eng.) and M.Phil. degrees in electrical engineering from the University of Hong Kong in 1971 and 1973, respectively, and the Ph.D. degree from the University of London, London, U.K., in 1977.

In 1984, he joined the Department of Electrical and Electronic Engineering at the University of Hong Kong. His current research interests include signal processing and communications systems.

Dr. Ho is a Chartered Engineer of the Engineering Council of the United Kingdom and is a member of the IEE and the HKIE.