INTEGER FAST LAPPED BIORTHOGONAL TRANSFORM VIA APPLICATIONS OF DCT MATRICES AND DYADIC-VALUED FACTORS FOR LIFTING COEFFICIENT BLOCKS

Taizo Suzuki and Hiroyuki Kudo

Faculty of Engineering, Information and Systems, University of Tsukuba
Tsukuba, Ibaraki, 305-8573 Japan
Email: {taizo, kudo}@cs.tsukuba.ac.jp

ABSTRACT

This paper presents a realization of integer fast lapped biorthogonal transform (FLBT) via applications of discrete cosine transform (DCT) matrices and dyadic-valued factors for lifting coefficient blocks. It is obtained by using the block-lifting factorization as our previous work and easy matrix manipulations. The proposed FLBT has higher coding performance and fewer rounding operations than the conventional methods. The practicality of the proposed FLBT is validated through lossy-to-lossless image compression (coding) simulation which unifies lossy and lossless image coding.

Index Terms— Block-lifting structure, discrete cosine transform (DCT), dyadic-valued factor, fast lapped biorthogonal transform (FLBT), lossy-to-lossless image coding

1. INTRODUCTION

The discrete cosine transform (DCT) and discrete sine transform (DST) [1] have found wide signal processing applications, especially image/video compression (coding) standards such as JPEG and H.26x series [2, 3]. Both of DCT and DST are classified into several different types. The type-II DCT (DCT-II) and its inverse transform, type-III DCT (DCT-III), are usually adopted to the transform and inverse transform part in image/video coding standard, respectively, because the DCT-II has high energy compaction capability and DCTs have many fast implementations. However, the DCT-II generates an unpleasant artifact, i.e., blocking artifact, for a reconstructed image in low-bitrate compression. The M-channel ($M = 2^k$, $k \in \mathbb{N}$, $k \geq 2$) lapped transform (LT) [4] has been proposed to overcome the problem. In particular, an M-channel fast lapped orthogonal transform (FLOT) reduces the blocking artifacts while keeping comparatively fast implementation, because it is composed of only the postprocessing part, which is also constructed by the $M/2$-channel DCT-III, $M/2$-channel type-IV DST (DST-IV) and some 2-channel Hadamard transforms, after the M-channel DCT-II implementation.

On another front, several integer LTs [5, 6] based on lifting structure [7] have been researched for lossy-to-lossless image coding which unifies lossy and lossless image coding. As part of this trend, a 4-channel hierarchical lapped biorthogonal transform (HLBT) [8], which was developed FLOT to achieve higher coding performance and composed of lifting structures, was adopted to the newest image coding standard JPEG XR [9]. However, the HLBT is a transform researched to seek faster implementation while keeping coding performance moderately. And then [10] has proposed an integer FLOT with bigger block size, e.g., $M = 8$ and 16, and higher coding performance than the HLBT. But, since the FLOT must be implemented by two parallel systems, it increases the complexity. Thereby a transform with more higher coding performance than the conventional methods and as little complexity as possible should be innovated for the further future in image/video coding.

This paper designs an integer fast lapped biorthogonal transform (FLBT) with big block size and higher coding performance/less complexity than the conventional methods. In this regard, the operations such as multipliers and adders are increased if the block size is just simply extended, and it is undesired. The proposed FLBT focuses on the block-lifting structure in our previous work [11, 12]. By using the structures and easy matrix manipulations, the FLBT can be fast implemented because its most of lifting coefficients are composed of DCT matrices and dyadic-valued factors.

Notations: $\mathbf{I}$, $\mathbf{J}$, $\mathbf{D}$ and $\mathbf{M}^T$ are an identity matrix, a reversal identity matrix, a diagonal matrix with $(-1)^k$ ($k \in \mathbb{N}$) in the (k, k)-element and a transpose of the matrix $\mathbf{M}$, respectively.

2. REVIEW

2.1. M-Channel FLOT and FLBT

An M-channel FLOT can be constructed in polyphase structure from components with well-known fast-computable algorithms such as DCT and DST. One of the most elegant solutions is the FLOT whose polyphase matrix is [4]

$$\mathbf{E}(z) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathcal{S}_{IV}\mathcal{C}_{III} \end{bmatrix} \mathbf{W}\mathbf{\Lambda}(z)\mathbf{W} \begin{bmatrix} \mathcal{C}_{II} & \mathbf{0} \\ \mathbf{0} & \mathcal{C}_{IV} \end{bmatrix} \mathbf{W}\tilde{\mathbf{I}}$$

where

$$\mathbf{W} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{I} & -\mathbf{I} \end{bmatrix}, \ \mathbf{\Lambda}(z) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & z^{-1}\mathbf{I} \end{bmatrix}, \ \tilde{\mathbf{I}} = \begin{bmatrix} \mathbf{0} & \mathbf{J} \\ \mathbf{I} & \mathbf{0} \end{bmatrix},$$

and $\mathcal{C}_{II}$, $\mathcal{C}_{III}$, $\mathcal{C}_{IV}$ and $\mathcal{S}_{IV}$ are DCT-II, DCT-III, type-IV DCT (DCT-IV) and type-IV DST (DST-IV) matrices with $(M/2) \times (M/2)$ size, respectively. The (m, n)-element of N-channel ($N = 2^k$, $k \in \mathbb{N}$, $k \geq 1$) DCT-II and DCT-IV matrices are defined as

$$[\mathcal{C}_{II}]_{m,n} = \sqrt{\frac{2}{N}} c_m \cos\left(\frac{m\,(n+1/2)\,\pi}{N}\right)$$

$$[\mathcal{C}_{IV}]_{m,n} = \sqrt{\frac{2}{N}} \cos\left(\frac{(m+1/2)\,(n+1/2)\,\pi}{N}\right)$$

where $c_m = 1/\sqrt{2}$ ($m = 0$) or 1 ($m \neq 0$), $\mathcal{C}_{III} = \mathcal{C}_{II}^{-1} = \mathcal{C}_{II}^T$, $\mathcal{C}_{IV}^{-1} = \mathcal{C}_{IV}^T = \mathcal{C}_{IV}$ and $\mathcal{S}_{IV} = \mathbf{D}\,\mathcal{C}_{IV}\mathbf{J}$ are established, respectively. Also, the HLBT [8] adopts the scaling factor s. In this paper,

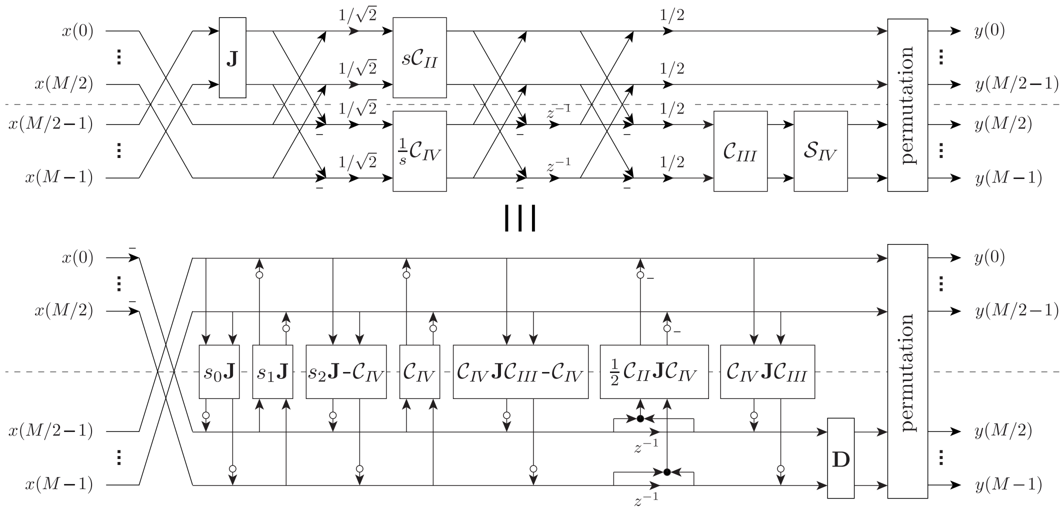

Fig. 1. Lattice structure of the FLBT and its block-lifting structure (white and black circles mean rounding operations and adders, respectively): (top) lattice structure, (bottom) block-lifting structure.

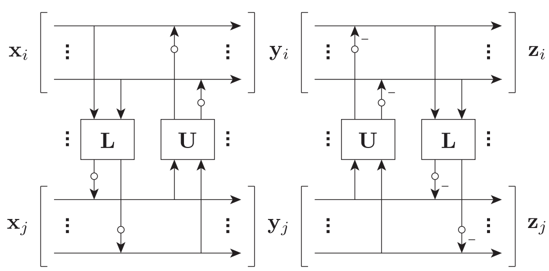

Fig. 2. Block-lifting structures.

the FLBT inspired by the HLBT as

$$\mathbf{E}(z) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathcal{S}_{IV}\mathcal{C}_{III} \end{bmatrix} \mathbf{W}\boldsymbol{\Lambda}(z)\mathbf{W} \begin{bmatrix} s\mathcal{C}_{II} & \mathbf{0} \\ \mathbf{0} & \frac{1}{s}\mathcal{C}_{IV} \end{bmatrix} \mathbf{W}\tilde{\mathbf{I}} \qquad (1)$$

is used and shown in the top of Fig. 1.

2.2. Block-Lifting Structure

The block-lifting structure [12], shown in Fig. 2, is to improve the basic lifting structure [7] for an effective implementation of lossy-to-lossless image coding. The structure achieves a higher compression ratio due to fewer rounding operations. In Fig. 2, the analysis input signal vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, the analysis output and synthesis input signal vectors $\mathbf{y}_i$ and $\mathbf{y}_j$, the synthesis output signal vectors $\mathbf{z}_i$ and $\mathbf{z}_j$, and the lifting coefficient blocks $\mathbf{L}$ and $\mathbf{U}$ are presented by

$$\begin{aligned} \mathbf{y}_j &= \mathbf{x}_j + \mathrm{round}\{\mathbf{L}\mathbf{x}_i\}, & \mathbf{y}_i &= \mathbf{x}_i + \mathrm{round}\{\mathbf{U}\mathbf{y}_j\} \\ \mathbf{z}_i &= \mathbf{y}_i - \mathrm{round}\{\mathbf{U}\mathbf{y}_j\} = \mathbf{x}_i, & \mathbf{z}_j &= \mathbf{y}_j - \mathrm{round}\{\mathbf{L}\mathbf{y}_i\} = \mathbf{x}_j. \end{aligned}$$

In this case, the matrices and its inverse matrices are expressed by

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{L} & \mathbf{I} \end{bmatrix}, \quad \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{L} & \mathbf{I} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{L} & \mathbf{I} \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{I} & \mathbf{U} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \begin{bmatrix} \mathbf{I} & \mathbf{U} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{I} & -\mathbf{U} \\ \mathbf{0} & \mathbf{I} \end{bmatrix},$$

respectively.

3. THE PROPOSED FLBT

In this section, we introduce an integer FLBT composed of DCT matrices and dyadic-valued factors for lifting coefficient blocks.

Theorem: The resulting structure is presented as

$$\begin{aligned} \mathbf{E}(z) = &\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathcal{C}_{IV}\mathbf{J}\mathcal{C}_{III} & \mathbf{I} \end{bmatrix} \boldsymbol{\Lambda}(z) \begin{bmatrix} \mathbf{I} & \frac{-1-z^{-1}}{2}\mathcal{C}_{II}\mathbf{J}\mathcal{C}_{IV} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \\ &\times \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathcal{C}_{IV}\mathbf{J}\mathcal{C}_{III} - \mathcal{C}_{IV} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathcal{C}_{IV} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ s_2\mathbf{J} - \mathcal{C}_{IV} & \mathbf{I} \end{bmatrix} \\ &\times \begin{bmatrix} \mathbf{I} & s_1\mathbf{J} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ s_0\mathbf{J} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix} \end{aligned}$$

where $s_0 = (\sqrt{2} - s)/s$, $s_1 = -s/\sqrt{2}$ and $s_2 = (\sqrt{2}s - 1)/s^2$, respectively, as shown in the bottom of Fig. 1. Since s_0, s_1 and s_2, however, are non-dyadic-valued (floating-point) factors, they are approximated to dyadic-values as $\alpha/2^{\beta}$ $(\alpha, \beta \in \mathbb{N})$ for faster implementation as shown in Table 1.

Proof: As the preparation for lifting factorization, (1) is factorized into several matrices as

$$\mathbf{E}(z) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} \boldsymbol{\Phi}\boldsymbol{\Gamma}(z)\boldsymbol{\Xi}\boldsymbol{\Phi}\mathbf{S}\tilde{\mathbf{W}}\mathbf{J} \qquad (2)$$

where

$$\boldsymbol{\Phi} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathcal{C}_{IV}\mathbf{J} \end{bmatrix}, \quad \boldsymbol{\Gamma}(z) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathcal{C}_{III} \end{bmatrix} \mathbf{W}\boldsymbol{\Lambda}(z)\mathbf{W} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathcal{C}_{II} \end{bmatrix}$$

$$\boldsymbol{\Xi} = \begin{bmatrix} \mathcal{C}_{II} & \mathbf{0} \\ \mathbf{0} & \mathcal{C}_{III} \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} s\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \frac{1}{s}\mathbf{I} \end{bmatrix} \text{ and } \tilde{\mathbf{W}} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I} & \mathbf{J} \\ \mathbf{J} & -\mathbf{I} \end{bmatrix},$$

Table 1. Lifting coefficients of the scaling part.

Channel M		Lifting Coefficients Float.	Dyad.
8	s_0	0.5747	147/256
	s_1	-0.6351	-163/256
	s_2	0.3349	43/128
16	s_0	0.5109	33/64
	s_1	-0.6619	-85/128
	s_2	0.3695	47/128

respectively. Also, the following equations are used to factorize into lifting structures.

$$\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{L}_0 & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{L}_1 & \mathbf{I}\end{bmatrix}=\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{L}_0+\mathbf{L}_1 & \mathbf{I}\end{bmatrix} \quad (3)$$

$$\begin{bmatrix}\mathbf{I} & \mathbf{U}_0\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{U}_1\\ \mathbf{0} & \mathbf{I}\end{bmatrix}=\begin{bmatrix}\mathbf{I} & \mathbf{U}_0+\mathbf{U}_1\\ \mathbf{0} & \mathbf{I}\end{bmatrix} \quad (4)$$

$$\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{N}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{L}_2 & \mathbf{I}\end{bmatrix}=\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{N}\mathbf{L}_2 & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{N}\end{bmatrix} \quad (5)$$

$$\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{N}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{U}_2\\ \mathbf{0} & \mathbf{I}\end{bmatrix}=\begin{bmatrix}\mathbf{I} & \mathbf{U}_2\mathbf{N}^{-1}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{N}\end{bmatrix} \quad (6)$$

where $\mathbf{N}$ is an arbitrary nonsingular matrix, and $\mathbf{L}_k$ and $\mathbf{U}_k$ ($k = 0, 1, 2$) are arbitrary matrices, respectively. According to [11, 12], the block-lifting factorizations of $\mathbf{\Gamma}(z)$ and $\mathbf{\Xi}$ in (2) are produced, respectively, as

$$\mathbf{\Gamma}(z)=\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathcal{C}_{III} & \mathbf{I}\end{bmatrix}\mathbf{\Lambda}(z)\begin{bmatrix}\mathbf{I} & \frac{1-z^{-1}}{2}\mathcal{C}_{II}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ -\mathcal{C}_{III} & \mathbf{I}\end{bmatrix}$$
$$\mathbf{\Xi}=\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathcal{C}_{III} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & -\mathcal{C}_{II}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathcal{C}_{III} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{0} & \mathbf{I}\\ -\mathbf{I} & \mathbf{0}\end{bmatrix}$$

and the merged structure of $\mathbf{\Gamma}(z)$ and $\mathbf{\Xi}$ is simplified as

$$\mathbf{\Gamma}(z)\mathbf{\Xi}=\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathcal{C}_{III} & \mathbf{I}\end{bmatrix}\mathbf{\Lambda}(z)\begin{bmatrix}\mathbf{I} & \frac{-1-z^{-1}}{2}\mathcal{C}_{II}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathcal{C}_{III} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{0} & \mathbf{I}\\ -\mathbf{I} & \mathbf{0}\end{bmatrix}.$$

By using (3)-(6) and the block-lifting factorization in [11], (2) is represented by

$$\mathbf{E}(z)=\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{D}\end{bmatrix}\tilde{\mathbf{\Gamma}}(z)\tilde{\mathbf{\Phi}}\begin{bmatrix}\mathbf{J} & \mathbf{0}\\ \mathbf{0} & \mathbf{J}\end{bmatrix}\mathbf{S}\tilde{\mathbf{W}}\mathbf{J} \quad (7)$$

where

$$\tilde{\mathbf{\Gamma}}(z)=\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathcal{C}_{IV}\mathbf{J}\mathcal{C}_{III} & \mathbf{I}\end{bmatrix}\mathbf{\Lambda}(z)\begin{bmatrix}\mathbf{I} & \frac{-1-z^{-1}}{2}\mathcal{C}_{II}\mathbf{J}\mathcal{C}_{IV}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\times\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathcal{C}_{IV}\mathbf{J}\mathcal{C}_{III} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{0} & \mathbf{I}\\ -\mathbf{I} & \mathbf{0}\end{bmatrix}$$
$$\tilde{\mathbf{\Phi}}=\begin{bmatrix}\mathcal{C}_{IV} & \mathbf{0}\\ \mathbf{0} & \mathcal{C}_{IV}\end{bmatrix}=\begin{bmatrix}\mathbf{0} & -\mathbf{I}\\ \mathbf{I} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ -\mathcal{C}_{IV} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathcal{C}_{IV}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ -\mathcal{C}_{IV} & \mathbf{I}\end{bmatrix}.$$

The merged structure of $\tilde{\mathbf{\Gamma}}(z)$ and $\tilde{\mathbf{\Phi}}$ are simplified as

$$\tilde{\mathbf{\Gamma}}(z)\tilde{\mathbf{\Phi}}=\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathcal{C}_{IV}\mathbf{J}\mathcal{C}_{III} & \mathbf{I}\end{bmatrix}\mathbf{\Lambda}(z)\begin{bmatrix}\mathbf{I} & \frac{-1-z^{-1}}{2}\mathcal{C}_{II}\mathbf{J}\mathcal{C}_{IV}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\times\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathcal{C}_{IV}\mathbf{J}\mathcal{C}_{III}-\mathcal{C}_{IV} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathcal{C}_{IV}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ -\mathcal{C}_{IV} & \mathbf{I}\end{bmatrix} \quad (8)$$

Table 2. Coding gain (CG [dB]).

Channel M	Conv. FLOTs [10] Float.	Dyad.	Prop. FLBTs Float.	Dyad.
8	9.219	9.219	9.447	9.447
16	9.759	9.759	9.845	9.845

by using (3)-(6) again. Also, it is clear that

$$\begin{bmatrix}\mathbf{J} & \mathbf{0}\\ \mathbf{0} & \mathbf{J}\end{bmatrix}\mathbf{S}\tilde{\mathbf{W}}\mathbf{J}=\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ s_2\mathbf{J} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & s_1\mathbf{J}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ s_0\mathbf{J} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{0} & \mathbf{I}\\ -\mathbf{I} & \mathbf{0}\end{bmatrix}. \quad (9)$$

Consequently, the block-lifting factorization of FLBT in (1) is achieved by substituting (8) and (9) to (7). □

4. EXPERIMENTAL RESULTS

This paper designed the 8×16 and 16×32 FLBTs.[1] In this section, the proposed FLBTs are compared with the HLBT for JPEG XR [8] and the conventional FLOTs [10] by the coding gain, the number of rounding operations and the lossy-to-lossless simulation results.

4.1. Coding Gain and Number of Rounding Operations

The coding gain is one of the most important and popular factors for a transform in compression applications. A transform with higher coding gain compacts more energy into a fewer number of coefficients. As a result, higher objective performance such as PSNR would be achieved after quantization. The coding gain (CG) is defined as [13]

$$\text{CG [dB]}=10\log_{10}\frac{\sigma_x^2}{\prod_{k=0}^{M-1}\sigma_{x_k}^2\parallel f_k\parallel^2}$$

where σ_x^2 is the variance of the input signal, $\sigma_{x_k}^2$ is the variance of the k-th subbands and $\parallel f_k \parallel^2$ is the norm of the k-th synthesis filter. Table 2 shows the comparisons of coding gain of the conventional FLOTs [10] and the proposed FLBTs with floating-point and dyadic-valued factors for the lifting coefficients in the scaling part.[2] It is clear that the most of coding gain are not lost even if the lifting coefficients are approximated to dyadic-valued factors. Moreover, the proposed FLBTs have higher coding gains than the conventional FLOTs.

We also compare the number of rounding operations because a transform with fewer rounding operations achieves more effective lossy-to-lossless coding due to reducing rounding error. Actually, the number is reduced from $9M/2$ to $5M/2$, i.e., the proposed FLBTs have fewer rounding operations than the conventional FLOTs [10].

4.2. Lossy-to-Lossless Image Coding

Finally, the proposed FLBTs are validated in lossy-to-lossless image coding. To evaluate transform performance fairly, a very common

[1] Of course, the FLBTs with bigger block size can be easily designed.

[2] The coding gain of the proposed method cannot be accurately compared with one of the HLBT for JPEG XR because the dyadic-valued HLBT is implemented by non-separable transform. In the floating-point case, its coding gain is 8.447[dB].

Table 3. Comparison of lossless image coding (LBR [bpp]).

Test Images	HLBT [8]	Conv. FLOTs [10] 8×16	Conv. FLOTs [10] 16×32	Prop. FLBTs 8×16	Prop. FLBTs 16×32
Barbara	4.96	4.95	4.85	4.86	**4.80**
Boat	5.20	5.19	5.16	5.14	**5.13**
Finger	5.89	5.89	5.75	5.82	**5.73**
Goldhill	**5.12**	5.18	5.15	**5.12**	**5.12**
Lena	4.64	4.71	4.69	**4.62**	4.64
Pepper	5.00	4.99	5.00	**4.93**	4.97

Table 4. Comparison of lossy image coding (PSNR [dB]).

Bitrate [bpp]	HLBT [8]	Conv. FLOTs [10] 8×16	Conv. FLOTs [10] 16×32	Prop. FLBTs 8×16	Prop. FLBTs 16×32
Barbara					
1.00	36.00	36.59	37.13	37.13	**37.43**
0.50	30.85	31.76	32.67	32.14	**32.87**
0.25	27.01	27.83	28.77	28.05	**28.90**
Boat					
1.00	35.21	35.44	35.43	**35.63**	35.55
0.50	32.02	32.13	32.13	**32.39**	32.20
0.25	28.80	28.97	29.04	**29.21**	29.08
Finger					
1.00	30.12	30.64	31.44	30.92	**31.52**
0.50	26.31	26.79	27.32	27.00	**27.36**
0.25	22.95	23.57	23.97	23.77	**24.01**
Goldhill					
1.00	35.17	35.26	35.36	35.50	**35.53**
0.50	32.02	32.05	32.20	32.36	**32.41**
0.25	29.62	29.43	29.59	29.80	**29.81**
Lena					
1.00	38.62	38.47	38.53	**39.01**	38.90
0.50	35.90	35.59	35.71	**36.30**	36.25
0.25	32.76	32.37	32.55	**33.08**	33.00
Pepper					
1.00	36.39	36.50	36.44	**36.77**	36.68
0.50	34.71	34.29	34.02	**34.88**	34.48
0.25	**32.52**	31.68	31.84	**32.52**	32.29

wavelet-based coder SPIHT [14] was adopted for all after the transformed coefficients were rearranged from subband mode to multiresolution mode similar to wavelet transform. Also, the periodic extension was used for image boundaries. Moreover, we used 8-bit gray scale test images with 512×512 size such as *Barbara*. After the proposed FLBTs, the HLBT and the conventional FLOTs are applied to lossless image coding, the lossy compressed data can be achieved by interrupting the obtained lossless bitstream if it is required. The lossy-to-lossless image coding results are compared by lossless bit rate (LBR) and peak signal-to-noise ratio (PSNR) as

$$\text{LBR [bpp]} = \frac{\text{Total number of bits [bit]}}{\text{Total number of pixels [pixel]}}$$

$$\text{PSNR [dB]} = 10 \log_{10} \frac{255^2}{\text{MSE}}$$

where MSE is the mean squared error, respectively. The comparisons of LBR, PSNR and a particular area of the reconstructed image *Barbara* (bitrate is 0.25[bpp]) are shown in Table 3, 4 and Fig. 3, respectively.

Fig. 3. Comparison of a particular area of the reconstructed image *Barbara* (bitrate is 0.25[bpp]): (top) original image and the HLBT for JPEG XR [8], (middle) the conventional 8×16 and 16×32 FLOTs [10], (bottom) the proposed 8×16 and 16×32 FLBTs.

The proposed FLBTs show better coding performance than the conventional methods. It is considered that the reduction of rounding operations is one of the most feasible reasons for achieving such performance. On the other hand, note that the proposed FLBTs have a simple and fast implementation due to the construction with DCT matrices and dyadic-valued factors for lifting coefficient blocks.

5. CONCLUSION

In this paper, we have introduced a realization of integer fast lapped biorthogonal transform (FLBT) via applications of discrete cosine transform (DCT) matrices and dyadic-valued factors for lifting coefficient blocks. The proposed FLBTs have higher coding gain and fewer rounding operations than the conventional fast lapped orthogonal transforms (FLOTs). They could achieve more effective coding performance for lossy-to-lossless image coding than the conventional methods including a hierarchical lapped biorthogonal transform (HLBT) for the newest image coding standard JPEG XR.

6. REFERENCES

[1] K. R. Rao and P. Yip, *Discrete Cosine Transform Algorithms*, Academic Press, 1990.

[2] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electr.*, vol. 38, no. 1, pp. xviii–xxxiv, Feb. 1992.

[3] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, July 2003.

[4] H. S. Malvar, *Signal Processing with Lapped Transforms*, Norwood, MA: Artech House, 1992.

[5] T. D. Tran, "The LiftLT: fast lapped transforms via lifting steps," *IEEE Signal Process. Lett.*, vol. 7, no. 6, pp. 145–148, June 2000.

[6] W. C. Fong, S. C. Chan, A. Nallanathan, and K. L. Ho, "Integer lapped transforms and their applications to image coding," *IEEE Trans. Image Process.*, vol. 11, no. 10, pp. 1152–1159, Oct. 2002.

[7] W. Sweldens, "The lifting scheme: A new philosophy in biorthogonal wavelet constructions," in *Proc. of SPIE 2569*, San Diego, CA, July 1995.

[8] C. Tu, S. Srinivasan, G. J. Sullivan, S. Regunathan, and H. S. Malvar, "Low-complexity hierarchical lapped transform for lossy-to-lossless image coding in JPEG XR/HD Photo," in *Proc. of SPIE 7073*, San Diego, CA, Aug. 2008.

[9] F. Dufaux, G. J. Sullivan, and T. Ebrahimi, "The JPEG XR image coding standard," *IEEE Signal Process. Mag.*, vol. 26, no. 6, pp. 195–199, 204, Nov. 2009.

[10] T. Suzuki and M. Ikehara, "Integer fast lapped orthogonal transform based on direct-lifting of DCTs for lossless-to-lossy image coding," in *Proc. of ICASSP'11*, Prague, Czech Republic, May 2011, pp. 1525–1528.

[11] T. Suzuki and M. Ikehara, "Integer DCT based on direct-lifting of DCT-IDCT for lossless-to-lossy image coding," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2958–2965, Nov. 2010.

[12] T. Suzuki, M. Ikehara, and T. Q. Nguyen, "Generalized block-lifting factorization of M-channel biorthogonal filter banks for lossy-to-lossless image coding," *IEEE Trans. Image Process.*, vol. 21, no. 7, pp. 3220–3228, July 2012.

[13] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, 1996.

[14] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 243–250, June 1996.