INTEGER FAST LAPPED ORTHOGONAL TRANSFORM BASED ON DIRECT-LIFTING OF DCTS FOR LOSSLESS-TO-LOSSY IMAGE CODING

Taizo Suzuki and Masaaki Ikehara

Department of Electronics and Electrical Engineering, Keio University
Yokohama, Kanagawa 223-8522, Japan
Email: {suzuki, ikehara}@tkhm.elec.keio.ac.jp

ABSTRACT

Integer lapped orthogonal transforms (LOTs) are vital technologies for the unification of lossless and lossy image coding, called lossless-to-lossy image coding. In this paper, we present an efficient realization of integer fast LOT (FLOT) based on direct-lifting of discrete cosine transforms (DCTs) which are type-II, III and IV. Although the conventional integer FLOTs suffer from degradation of coding performance due to much rounding error generated by cascading lifting structures, this paper presents a realization of a simpler, faster and more efficient transform with only some adders, 1-bit shifters and direct use of DCTs for lifting coefficients. As result, the proposed method is validated in lossless-to-lossy image coding.

Index Terms— Direct-lifting, discrete cosine transform (DCT), fast lapped orthogonal transform (FLOT), lossless-to-lossy image coding

1. INTRODUCTION

With the rapid development of hardware such as PCs and mobile phones etc. and the continual expansion of broadband, lossless-to-lossy image coding, which is the unification of lossy and lossless image coding, is required to obtain higher quality and compression ratio. Since JPEG [1], which is the international standard in image coding, separately uses discrete cosine transform (DCT) [2] and differential pulse code modulation (DPCM) for lossy and lossless image coding, respectively, it cannot achieve lossless-to-lossy image coding. In lossless-to-lossy image coding, lossy compression is obtained by interrupting the scalable lossless data which is coded by integer transform. Although 5/3-tap discrete wavelet transform (5/3-DWT) by lifting structures [3] with integer coefficients and rounding operations is used for lossless compression in JPEG2000 [4] which is the next generation standard, its lossy performance is undesirable. Also, a hierarchical lapped biorthogonal transform (HLBT) in JPEG XR [5], which is the another next generation standard, has achieved lossless-to-lossy image coding, but it does not have sufficient coding performance.

On the other hand, DCT in JPEG is known as one of the fastest and the most efficient transforms for image coding. HLBT in JPEG XR is also a DCT-based lapped transform. As such DCT-based lapped transform, fast lapped orthogonal transform (FLOT) [6] is well-known. FLOT is constructed by cascading DCT type-II (DCT-II), type-III (DCT-III), type-IV (DCT-IV), rotation matrices with $\pi/4$ angles, ± 1 operations, a delay matrix and permutation matrices. Although it is easy that FLOT is applied to lossless-to-lossy image coding by lifting factorization of rotation matrices, called integer FLOT, the obtained integer-to-integer transform is unsuitable due to much rounding error by many rounding operations.

In this paper, we present an efficient realization of integer FLOT based on direct-lifting of DCTs. The proposed FLOT is a simpler, faster and more efficient transform with only some adders, 1-bit shifters and direct use of DCTs for lifting coefficients. Finally, our method is validated in lossless-to-lossy image coding simulation.

Notations: $\mathbf{I}$, $\mathbf{J}$, $\mathbf{M}^T$ and $\mathbf{D}$ are an identity matrix, a reversal identity matrix, transpose of a matrix $\mathbf{M}$ and a diagonal matrix with alternating ± 1 entries (i.e., $\mathbf{D} = \mathrm{diag}\{1, -1, 1, -1, \cdots\}$), respectively.

This work was supported in part by a Grant-in-Aid for the Japan Society for the Promotion of Science (JSPS) Fellows and a Grant-in-Aid for the Global Center of Excellence (G-COE) for High-Level Global Cooperation for Leading-Edge Platform on Access Spaces from the Ministry of Education, Culture, Sport, Science, and Technology in Japan.

2. REVIEW

2.1. Discrete Cosine Transform (DCT)

DCT is basically classified into four types [2]. Among them, type-II, type-III and type-IV (DCT-II, DCT-III and DCT-IV) are famous and often used for image coding such as JPEG [1] and JPEG XR [5] etc. and many devices and fast algorithms for DCT have been developed. The m-column and n-row element of M-channel DCT-II, DCT-III and DCT-IV matrices $\mathbf{C}_{II}$, $\mathbf{C}_{III}$ and $\mathbf{C}_{IV}$ are defined as

$$[\mathbf{C}_{II}]_{m,n} = \sqrt{\frac{2}{M}} c_m \cos\left(\frac{m\,(n+1/2)\,\pi}{M}\right)$$
$$[\mathbf{C}_{III}]_{m,n} = \sqrt{\frac{2}{M}} c_n \cos\left(\frac{(m+1/2)\,n\pi}{M}\right)$$
$$[\mathbf{C}_{IV}]_{m,n} = \sqrt{\frac{2}{M}} \cos\left(\frac{(m+1/2)\,(n+1/2)\,\pi}{M}\right)$$

where $\mathbf{C}_{II}^{-1} = \mathbf{C}_{II}^{T} = \mathbf{C}_{III}, \mathbf{C}_{IV}^{-1} = \mathbf{C}_{IV}^{T} = \mathbf{C}_{IV}$,

$$c_m = \begin{cases} \frac{1}{\sqrt{2}} & (m = 0) \\ 1 & (m \neq 0) \end{cases}, \quad c_n = \begin{cases} \frac{1}{\sqrt{2}} & (n = 0) \\ 1 & (n \neq 0) \end{cases}$$

and $0 \leq m, n \leq M-1$, respectively. For simplicity, let us define as that $M = 2^n$ ($n \in \mathbb{N}$). Also, when "cos" is replaced by "sin", it is called discrete sine transform (DST).

2.2. Fast Lapped Orthogonal Transform (FLOT)

An M-channel fast lapped transform can be constructed in polyphase structure from components with well-known fast-computable algorithms such as DCT and DST. One of the most elegant solution is

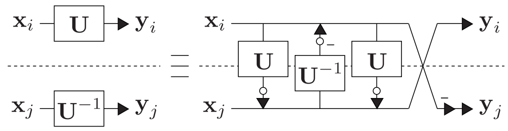

Fig. 1. A direct-lifting (white circles represent rounding operations).

the type-II FLOT whose polyphase matrix is [6]

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

where

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

and $\mathbf{S}_{IV}$ is type-IV (DST-IV) in DST. Since the following relationship between DST-IV and DCT-IV matrix can be easily established: $\mathbf{S}_{IV} = \mathbf{D}\mathbf{C}_{IV}\mathbf{J}$, (1) can be represented by

$$\mathbf{E}(z) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}\mathbf{C}_{IV}\mathbf{J}\mathbf{C}_{III} \end{bmatrix} \mathbf{W}\mathbf{\Lambda}(z)\mathbf{W} \begin{bmatrix} \mathbf{C}_{II} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{IV} \end{bmatrix} \mathbf{W}\tilde{\mathbf{I}}. \qquad (2)$$

The FLOT with this polyphase matrix is implemented in the top half in Fig. 2.

2.3. Direct-Lifting Structure

The lifting structure is cascading elementary matrices which are identity matrices with one single nonzero off-diagonal element [3]. In [7], we have presented the direct-lifting. It is a breakthrough structure because a combination of every block and its inverse one can be directly applied to the lifting coefficient as shown in Fig. 1 where $\mathbf{x}_i$, $\mathbf{x}_j$, $\mathbf{y}_i$ and $\mathbf{y}_j$ are the input signals and their output signals, respectively. It is also expressed by

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

where $\mathbf{U}$ is an arbitrary nonsingular matrix.

3. FLOT BASED ON DIRECT-LIFTING OF DCTS

In this section, a realization of FLOT for lossless-to-lossy image coding is presented. The system is implemented by the parallel process of two different type FLOTs, direct-lifting of DCTs in the process, some adders and some 1-bit shifters. It means that the proposed FLOT is a simple, fast and efficient transform for lossless-to-lossy image coding.

3.1. Application of Direct-Lifting of DCTs

First, we present another class of FLOT. DCT-II matrix $\mathbf{C}_{II}$ in (2) is moved to the last building block. Since

$$\mathbf{W}\mathbf{\Lambda}(z)\mathbf{W} \begin{bmatrix} \mathbf{C}_{II} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{II} \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{II} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{II} \end{bmatrix} \mathbf{W}\mathbf{\Lambda}(z)\mathbf{W},$$

(2) is rewritten as

$$\mathbf{E}(z) = \begin{bmatrix} \mathbf{C}_{II} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}\mathbf{C}_{IV}\mathbf{J} \end{bmatrix} \mathbf{W}\mathbf{\Lambda}(z)\mathbf{W} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{III}\mathbf{C}_{IV} \end{bmatrix} \mathbf{W}\tilde{\mathbf{I}} \qquad (3)$$

where $\mathbf{C}_{II}\mathbf{C}_{III} = \mathbf{C}_{III}\mathbf{C}_{II} = \mathbf{I}$. The FLOT with this polyphase matrix are implemented in the bottom half in Fig. 2. The filter properties of (3) completely accord with them of (2) because it is obtained by only transposition of $\mathbf{C}_{II}$.

Next, as already mentioned, we consider the parallel process of two different type FLOTs in (2) and (3). It means that when a row (column) signals are processed by (2), other row (column) signals are processed by (3). However, each DCT matrix in both FLOT is processed by direct-lifting between each system which is each combination of DCT-II in (2) and DCT-III in (3), DCT-III in (2) and DCT-II in (3) and DCT-IV in (2) and DCT-IV in (3). These combinations are shown in dashed line area in Fig. 2 and factorized into direct-liftings as follows:

$$\begin{bmatrix} \mathbf{C}_{II} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{III} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{C}_{II} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{C}_{III} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{C}_{II} & \mathbf{I} \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{C}_{III} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{II} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{C}_{III} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{C}_{II} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{C}_{III} & \mathbf{I} \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{C}_{IV} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{IV} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{C}_{IV} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{C}_{IV} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{C}_{IV} & \mathbf{I} \end{bmatrix}$$

3.2. More Multiplierless Structure

In above subsection, DCT parts were implemented with lifting structures. This subsection produces the lifting factorizations of $\mathbf{W}$s. Since $\mathbf{W}$ is constructed by rotation matrices with $\pi/4$ angles, it can be simply factorized into lifting structures. But such case generates many multipliers. To eliminate multipliers, the following matrices are used in place of $\mathbf{W}$s.

$$\mathbf{W}_1 = \begin{bmatrix} \frac{1}{\sqrt{2}}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \sqrt{2}\mathbf{I} \end{bmatrix} \mathbf{W} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \frac{1}{2}\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{I} & \mathbf{I} \end{bmatrix}$$

$$\mathbf{W}_2 = \mathbf{W} \begin{bmatrix} \sqrt{2}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\sqrt{2}}\mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{I} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\frac{1}{2}\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \end{bmatrix}$$

$$\mathbf{W}_3 = \begin{bmatrix} \sqrt{2}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\sqrt{2}}\mathbf{I} \end{bmatrix} \mathbf{W} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \frac{1}{2}\mathbf{I} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \end{bmatrix}$$

$$\mathbf{W}_4 = \mathbf{W} \begin{bmatrix} \frac{1}{\sqrt{2}}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \sqrt{2}\mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\frac{1}{2}\mathbf{I} & \mathbf{I} \end{bmatrix}$$

Note that a combination of an $1/2$ multiplier and a rounding operation can be replaced by an 1-bit shifter [8]. Using these matrices, (2) and (3) are represented as follows:

$$\begin{aligned} \mathbf{E}_1(z) &\triangleq \begin{bmatrix} \frac{1}{\sqrt{2}}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \sqrt{2}\mathbf{I} \end{bmatrix} \mathbf{E}(z) \\ &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}\mathbf{C}_{IV}\mathbf{J}\mathbf{C}_{III} \end{bmatrix} \mathbf{W}_1\mathbf{\Lambda}(z)\mathbf{W}_2 \begin{bmatrix} \mathbf{C}_{II} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{IV} \end{bmatrix} \mathbf{W}_1\tilde{\mathbf{I}} \end{aligned} \qquad (4)$$

$$\begin{aligned} \mathbf{E}_2(z) &\triangleq \begin{bmatrix} \sqrt{2}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\sqrt{2}}\mathbf{I} \end{bmatrix} \mathbf{E}(z) \\ &= \begin{bmatrix} \mathbf{C}_{II} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}\mathbf{C}_{IV}\mathbf{J} \end{bmatrix} \mathbf{W}_3\mathbf{\Lambda}(z)\mathbf{W}_2 \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{III}\mathbf{C}_{IV} \end{bmatrix} \mathbf{W}_1\tilde{\mathbf{I}} \end{aligned} \qquad (5)$$

Fig. 3 shows the process of the two-dimensional transform of an image by the proposed FLOTs. Let M ($M = 2^k$, $k \in \mathbb{N}$) be block size, the i-th ($1 \leq ((i+1) \bmod M) \leq M/2$) row signals and the j-th ($(M/2+1) \leq ((j+1) \bmod M) \leq M$) row signals,

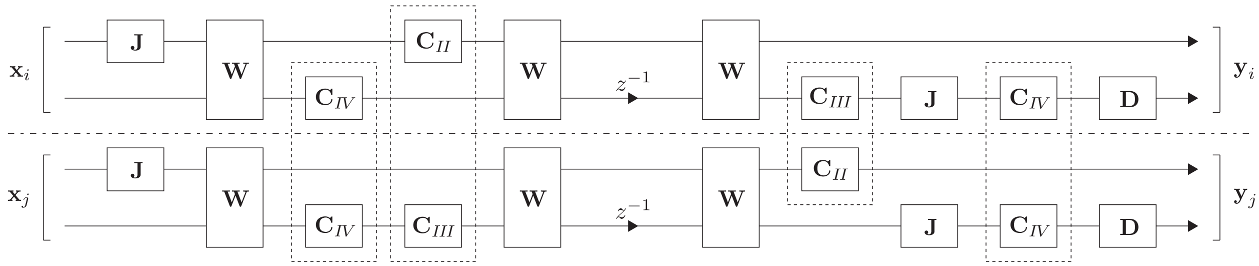

Fig. 2. The parallel process of two different type FLOTs: (top) FLOT in (2), (bottom) FLOT in (3).

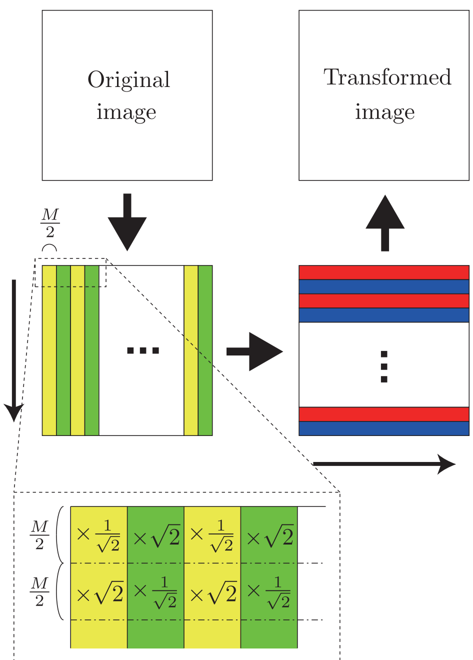

Fig. 3. The process of the two-dimensional transform of an image by the proposed FLOTs.

i.e., the yellow and green areas in Fig. 3, are processed by FLOTs in (4) and (5), respectively. Here, note that the scales in the one-dimensionally-transformed output signals are risen to $\times 1/\sqrt{2}$ and $\times\sqrt{2}$ as compared with the output signals transformed by normal FLOTs as shown in the dashed line area in Fig. 3. In this regard, considering these scales $1/\sqrt{2}$ and $\sqrt{2}$ for the next column process, (2) and (3) are represented again as follows:

$$\begin{aligned}\mathbf{E}_3(z) &\triangleq \mathbf{E}(z)\begin{bmatrix}\sqrt{2}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \frac{1}{\sqrt{2}}\mathbf{I}\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{D}\mathbf{C}_{IV}\mathbf{J}\mathbf{C}_{III}\end{bmatrix}\mathbf{W}_2\mathbf{\Lambda}(z)\mathbf{W}_1\begin{bmatrix}\mathbf{C}_{II} & \mathbf{0}\\ \mathbf{0} & \mathbf{C}_{IV}\end{bmatrix}\mathbf{W}_2\tilde{\mathbf{I}}\end{aligned} \tag{6}$$

Table 1. Comparison of lossless image coding (LBR [bpp]).

Test images	HLBT [5]	Conv. FLOTs 8×16	Conv. FLOTs 16×32	Prop. FLOTs 8×16	Prop. FLOTs 16×32
Barbara	4.95	5.00	4.95	4.95	**4.85**
Boat	5.21	5.22	5.22	5.19	**5.16**
Finger	5.89	5.91	5.79	5.89	**5.75**

$$\begin{aligned}\mathbf{E}_4(z) &\triangleq \mathbf{E}(z)\begin{bmatrix}\frac{1}{\sqrt{2}}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \sqrt{2}\mathbf{I}\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{C}_{II} & \mathbf{0}\\ \mathbf{0} & \mathbf{D}\mathbf{C}_{IV}\mathbf{J}\end{bmatrix}\mathbf{W}_4\mathbf{\Lambda}(z)\mathbf{W}_3\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{C}_{III}\mathbf{C}_{IV}\end{bmatrix}\mathbf{W}_4\tilde{\mathbf{I}}\end{aligned} \tag{7}$$

Similarly, the i-th column signals and the j-th column signals, i.e., the red and blue areas in Fig. 3, are processed by FLOTs in (6) and (7), respectively.

4. LOSSLESS-TO-LOSSY IMAGE CODING RESULTS

In this paper, the proposed 8×16 and 16×32 FLOTs are validated in lossless-to-lossy image coding. As targets for comparison, HLBT in JPEG XR [5] and 8×16 and 16×32 FLOTs, which are applied the simple lifting factorizations of rotation matrices, were chosen. To evaluate transform performance fairly, a very common wavelet-based coder SPIHT [9] was adopted for all. Also, for image boundaries, the periodic extension was used in all LOTs. Moreover, we used 512×512 gray scale test images such as *Barbara*.

First, the proposed LOTs and the conventional methods are applied to lossless image coding. The comparison of lossless bitrate (LBR):

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

in lossless image coding is shown in Table 1.

If lossy compressed data is required, it can be achieved by interrupting the obtained lossless bitstream. The comparison of peak signal-to-noise ratio (PSNR):

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

where MSE is the mean squared error in lossy image coding are shown in Table 2. Furthermore, Fig. 4 shows the enlarged images of *Barbara* which are lossy compressed images by the proposed FLOTs and the conventional methods when bitrate is 0.25[bpp].

Table 2. Comparison of lossy image coding (PSNR [dB]).

Test images	HLBT [5]	Conv. FLOTs		Prop. FLOTs	
		8×16	16×32	8×16	16×32
bitrate: 0.25 [bpp]					
Barbara	27.01	27.80	28.72	27.83	**28.77**
Boat	28.80	28.93	28.98	28.97	**29.04**
Finger	22.95	23.65	23.94	23.57	**23.97**
bitrate: 0.50 [bpp]					
Barbara	30.85	31.70	32.55	31.76	**32.67**
Boat	32.02	32.05	32.00	**32.13**	**32.13**
Finger	26.31	26.76	27.27	26.79	**27.32**
bitrate: 1.00 [bpp]					
Barbara	36.00	36.33	36.65	36.59	**37.13**
Boat	35.21	35.20	35.07	**35.44**	35.43
Finger	30.12	30.57	31.29	30.64	**31.44**

Table 3. Comparison of number of rounding operations in each one-dimensional transform of $M \times 1$ signals.

	Conv. FLOTs	Prop. FLOTs
8×16	90	36
16×32	240	72

Even though the proposed FLOTs and the conventional LOTs have same transfer function, all proposed FLOTs showed better coding performance than the conventional methods, especially lossy image coding results showed excellent performance. We consider that this is mainly due to the reduction of rounding operations as shown in Table 3. On the other hand, note that our FLOTs have a simple and fast implementation due to the construction with only some adders, some 1-bit shifters and direct use of DCTs for lifting coefficients. Despite such fact, the results in this section showed good performance. It means that the advantages in our methods were proved.

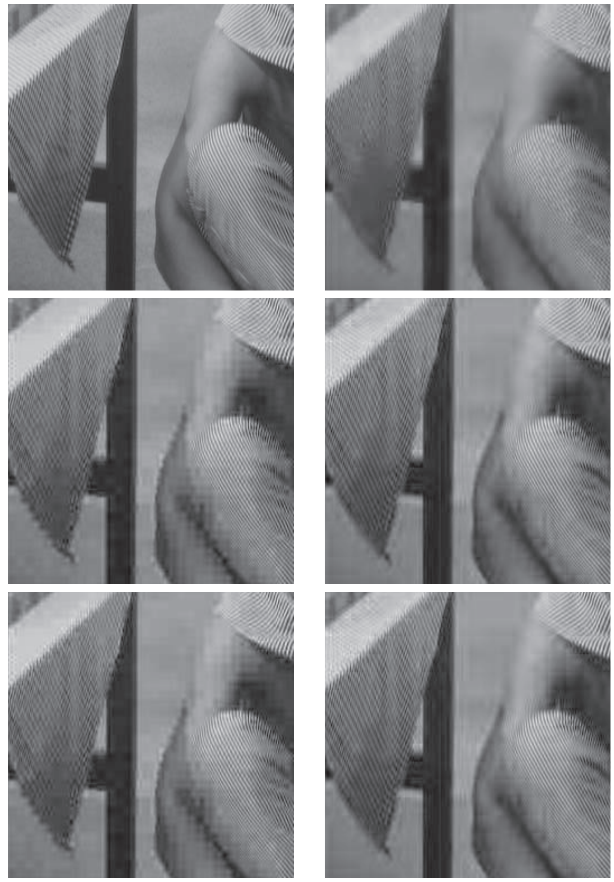

Fig. 4. Enlarged images of *Barbara* in lossy image coding (bitrate: 0.25 [bpp]): (top-left) original image, (top-right) HLBT in JPEG XR, (middle-left) conv. 8×16 FLOT, (middle-right) conv. 16×32 FLOT, (bottom-left) prop. 8×16 FLOT, (bottom-right) prop. 16×32 FLOT.

5. CONCLUSION

In this paper, we presented a fast lapped orthogonal transform (FLOT) for efficient lossless-to-lossy image coding, which is based on direct-lifting of discrete cosine transforms (DCTs) for lifting coefficients, some adders and some 1-bit shifters. Due to merging many rounding operations by use of direct-lifting, our proposals show better coding performance than the conventional methods in lossless-to-lossy image coding. However, the proposed method cannot be applied the symmetric extension for image boundaries. If the use will becomes possible, it should achieves more efficient coding performance. In addition, although has less blocking artifacts than the conventional methods, generated ringing artifacts should be improved.

6. REFERENCES

[1] ISO/IEC 10918-1, *Information technology - Digital compression and coding of continuous-tone still images Requirements and Guidelines*.

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

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

[4] ISO/IEC 15444-1, *Information technology - JPEG 2000 image coding system: Core coding system*.

[5] ISO/IEC 29199-2, *JPEG XR image coding system - Part 2: Image coding specification*.

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

[7] 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.

[8] Y. J. Chen, S. Oraintara, T. D. Tran, K. Amaratunga, and T. Q. Nguyen, "Multiplierless approximation of transforms with adder constraint," *IEEE Signal Process. Lett.*, vol. 9, no. 11, pp. 344–347, Nov. 2002.

[9] 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.