Integer fast lapped transforms based on direct-lifting of DCTs for lossy-to-lossless image coding

doi: 10.1186/1687-5281-2013-65

EURASIP Journal on Image and Video Processing
a SpringerOpen Journal

RESEARCH Open Access

Integer fast lapped transforms based on direct-lifting of DCTs for lossy-to-lossless image coding

Taizo Suzuki[1*] and Masaaki Ikehara[2]

Abstract

The discrete cosine transforms (DCTs) have found wide applications in image/video compression (image coding). DCT-based lapped transforms (LTs), called fast LTs (FLTs), overcome blocking artifacts generated at low bit rate image coding by DCT while keeping fast implementation. This paper presents a realization of more effective integer FLT (IntFLT) for lossy-to-lossless image coding, which is unified lossless and lossy image coding, than the conventional IntFLTs. It is composed of few operations and direct application of DCTs to lifting blocks, called direct-lifting of DCTs. Since the direct-lifting can reuse any existing software/hardware for DCTs, the proposed IntFLTs have a great potential for fast implementation which is dependent on the architecture design and DCT algorithms. Furthermore, the proposed IntFLTs do not need any side information unlike integer DCT (IntDCT) based on direct-lifting as our previous work. Moreover, they can be easily extended to larger size which is recently required as in DCT for the standard H.26x series. As a result, the proposed method shows better lossy-to-lossless image coding than the conventional IntFLTs.

1 Introduction

The most popular image/video compression (image coding) standards, JPEG [1,2] and H.26x series [3,4], employ discrete cosine transform (DCT) [5] at their transformation stages. DCT can be basically classified into types I to IV (DCT-I to IV) and has numerous fast implementations [6-10] and applications for signal processing. In them, DCT-II, so-called DCT, has excellent energy compaction capability and DCT-III is its inverse transform, so-called inverse DCT (IDCT). However, DCT generates annoying blocking artifacts at low bit rates because the DCT bases are short and create discontinuities at block boundaries due to non-overlapping. To overcome this drawback, lapped transforms (LTs), which are classified into lapped orthogonal transform (LOT) and lapped biorthogonal transform (LBT), have received much attention. DCT-based fast LTs (FLTs), which are classified into fast LOT (FLOT) and fast LBT (FLBT), are well-known as fast and effective transform for image coding [11]. FLTs are constructed by cascading DCT-II, DCT-III, DCT-IV, rotation matrices with $\pi/4$ angles, ± 1 operations, scaling factors, a delay matrix, and permutation matrices. To improve the coding performance and reduce the complexity more, LiftLT with VLSI-friendly implementation has been proposed by Tran [12][a]. However, the LTs cannot be applied to the lossless mode.

On the other hand, JPEG achieves the lossless mode by using differential pulse code modulation (DPCM) in place of DCT. JPEG 2000 [13] employs 9/7-tap and 5/3-tap discrete wavelet transforms (9/7-DWT and 5/3-DWT) for lossy and lossless modes, respectively [14]. They mean that JPEG and JPEG 2000 do not have compatibility between the lossy and lossless mode. Of course, lossless transform such as 5/3-DWT is applicable to lossy-to-lossless image coding. However, its lossy performance is not good compared with 9/7-DWT because each transform is suitable only in each mode. The next standard JPEG XR [15] has solved this problem by achieving lossy-to-lossless image coding which is unified lossy and lossless image coding. JPEG XR employs only hierarchical lapped transform (HLT) for both of lossy and lossless modes [16]. The HLT is composed of lifting structures [17-19] with rounding operations and achieves integer-to-integer

*Correspondence: taizo@cs.tsukuba.ac.jp
[1]Faculty of Engineering, Information and Systems, University of Tsukuba, Tsukuba 305-8573, Japan
Full list of author information is available at the end of the article

transform, whereas it does not have enough coding performance, especially for images with many high frequency components. Various lifting-based filter banks (L-FBs) [20-28], which contain integer DCTs (IntDCTs) [29-35], have been researched to improve coding performance. However, these except for IntDCTs are not practical due to the complexity.

This paper presents a realization of integer FLT (IntFLT), which is constructed by lifting structures with rounding operations, for lossy-to-lossless image coding. Although FLT can be easily applied to lossy-to-lossless image coding by simple lifting factorizations of rotation matrices and scaling factors, the obtained integer transform is unsuitable due to large rounding error because of many rounding operations. The conventional IntFLTs also have many operations, whereas the proposed IntFLTs have simple implementations with few operations and direct application of DCTs to lifting blocks, called direct-lifting of DCTs. The direct-lifting can reuse any existing software/hardware for DCTs[b]. As a result, although the proposed IntFLTs are apparently sacrificing the complexity to achieve the lossless mode compared with LiftLT, they have a great potential for fast implementation which is dependent on the architecture design and DCT algorithms. Furthermore, the proposed IntFLTs do not need any side information unlike IntDCT based on direct-lifting as our previous work [35]. Moreover, they can be easily extended to larger size which is recently required as in DCT for H.26x series. Such IntFLT already proposed in [36] cannot achieve enough coding performance due to the orthogonality. This paper introduces IntFLT without such a restriction. Finally, the proposed method shows better lossy-to-lossless image coding than the conventional IntFLTs.

1.1 Notations

Several special matrices with reserved symbols are as follows: **I**, **J**, **0**, and **D** are an identity matrix, a reversal identity matrix, a null matrix, and a diagonal matrix with alternating ± 1 entries (i.e., $\mathrm{diag}\{1,-1,1,-1,\cdots\}$), respectively. Also, $\cdot^T$ and $\cdot^{-1}$ are transpose and inverse of a matrix, respectively.

2 Review

2.1 Fast lapped transform (FLT)

An M-channel ($M = 2^k$, $k \in \mathbb{N}$) FLT can be constructed in polyphase structure from components with well-known fast-computable algorithms. One of the most elegant solution is the type-II FLOT. The polyphase matrix $\mathbf{E}(z)$ is expressed as [11]

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

where

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

z^{-1} is a delay, and $\mathbf{C}_{II}$, $\mathbf{C}_{III}$, $\mathbf{C}_{IV}$, and $\mathbf{S}_{IV}$ are DCT-II, DCT-III, DCT-IV, and type-IV discrete sine transform (DST-IV) matrices whose (m,n)-elements are presented by

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

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

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

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

where $c_i = 1/\sqrt{2}$ ($i = 0$) or 1 ($i \neq 0$). Also, $\mathbf{C}_{II}^{-1} = \mathbf{C}_{II}^T = \mathbf{C}_{III}$, $\mathbf{C}_{IV}^{-1} = \mathbf{C}_{IV}^T = \mathbf{C}_{IV}$, and $\mathbf{S}_{IV}^{-1} = \mathbf{S}_{IV}^T = \mathbf{S}_{IV}$. Since the following relationship between DST-IV and DCT-IV matrices can be established: $\mathbf{S}_{IV} = \mathbf{D}\mathbf{C}_{IV}\mathbf{J}$, Equation 1 can be easily represented by

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

On the other hand, the HLT for JPEG XR is based on FLOT with scaling factors [16]. By inspiring it, the FLT in this paper is defined by

$$\mathbf{E}(z) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}\mathbf{C}_{IV}\mathbf{J}\mathbf{C}_{III} \end{bmatrix} \mathbf{W}\mathbf{\Lambda}(z)\mathbf{W} \begin{bmatrix} s_0\mathbf{C}_{II} & \mathbf{0} \\ \mathbf{0} & s_1\mathbf{C}_{IV} \end{bmatrix} \mathbf{W}\tilde{\mathbf{I}}\mathbf{J} \tag{3}$$

where $s_1 = s_0^{-1}$ which is the restriction for lifting factorization. This is called FLBT in this paper. Since FLOT in Equation 2 is understandably equal to FLBT in Equation 3 with $s_0 = s_1 = 1$, we use this equation (3) as a representative expression of FLT. The FLT with this polyphase matrix is implemented as shown in the top half in Figure 1.

2.2 Direct-lifting structure

In [35], we have presented direct-lifting which is a class of block-lifting [25] known as a more effective lifting structure for lossy-to-lossless image coding than standard lifting structure [17-19]. The block-lifting reduces rounding error by merging many rounding operations. The direct-lifting is a key technology to produce novel IntFLTs. To achieve the lifting, we suppose a processing of two individual $M \times 1$ signals $\mathbf{x}_i$ and $\mathbf{x}_j$ by an $M \times M$ arbitrary nonsingular matrix $\mathbf{T}$ and its inverse transform matrix $\mathbf{T}^{-1}$, respectively, as shown at the left side of Figure 2. The

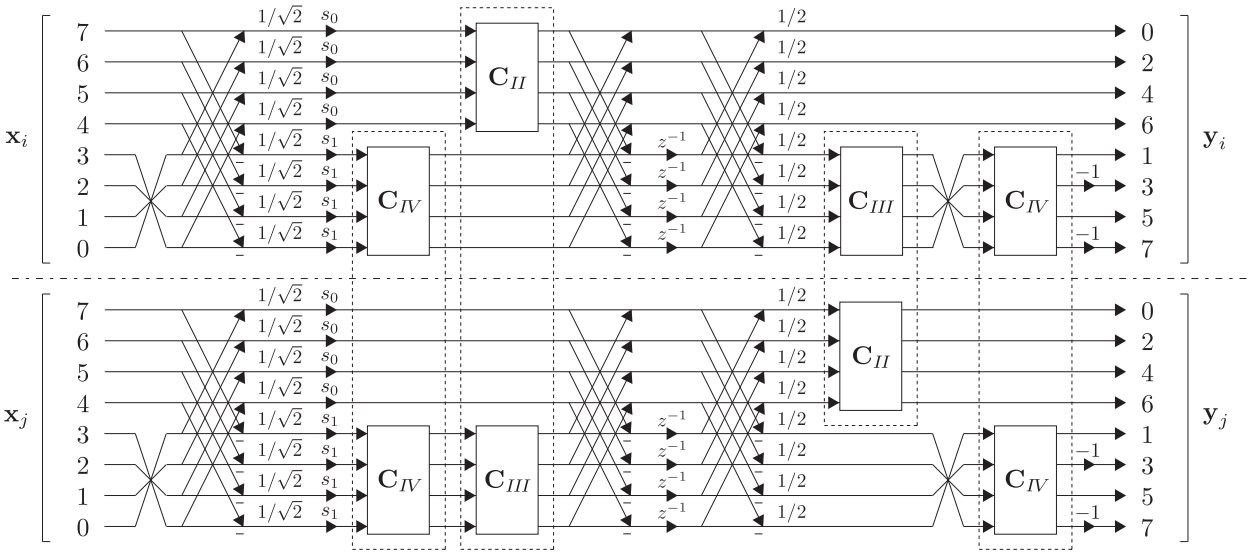

Figure 1 Parallel process of two different type M-channel FLTs (drawn for $M = 8$). (Top) FLT in Equation 3; (bottom) FLT in Equation 6.

input signals $\mathbf{x}_i$ and $\mathbf{x}_j$ are simultaneously transformed to the output signals $\mathbf{y}_i$ and $\mathbf{y}_j$ by $\mathbf{T}$ and $\mathbf{T}^{-1}$ as

$$\begin{bmatrix} \mathbf{y}_i \\ \mathbf{y}_j \end{bmatrix} = \begin{bmatrix} \mathbf{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_j \end{bmatrix}.$$

This block diagonal matrix $\mathrm{diag}\{\mathbf{T}, \mathbf{T}^{-1}\}$ can be factorized into complete block-liftings such as

$$\begin{bmatrix} \mathbf{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^{-1} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{T} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{T}^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{T} & \mathbf{I} \end{bmatrix}. \tag{4}$$

Thus, the parallel block system of $\mathbf{T}$ and $\mathbf{T}^{-1}$ can be efficiently implemented by the block-liftings as shown at the right side of Figure 2. This is a breakthrough structure because any block $\mathbf{T}$ and its inverse one $\mathbf{T}^{-1}$ can be directly applied to the block-lifting coefficients without breaking their forms. Although any existing software/hardware for DCT cannot be directly reused for the conventional IntDCTs, we can admit any of them as the lifting blocks when $\mathbf{T} = \mathbf{C}_{II}$.

3 IntFLTs based on direct-lifting of DCTs

This section presents a realization of IntFLT for lossy-to-lossless image coding. The IntFLTs have simple implementations with few operations and direct-lifting of DCTs.

3.1 Direct-lifting of DCTs

FLT in Equation 3 is transferred into the another type so that direct-lifting (4) can be applied. First, Equation 3 is rewritten as

$$\mathbf{E}(z) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}\mathbf{C}_{IV}\mathbf{J}\mathbf{C}_{III} \end{bmatrix} \mathbf{W}\mathbf{\Lambda}(z)\mathbf{W} \begin{bmatrix} \mathbf{C}_{II} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{II} \end{bmatrix} \times \begin{bmatrix} s_0\mathbf{I} & \mathbf{0} \\ \mathbf{0} & s_1\mathbf{C}_{III}\mathbf{C}_{IV} \end{bmatrix} \mathbf{W}\tilde{\mathbf{I}}\mathbf{J} \tag{5}$$

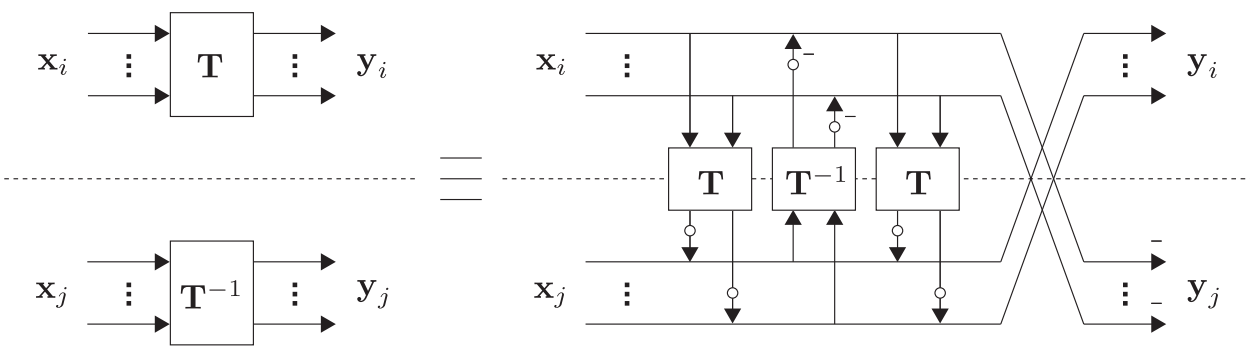

Figure 2 Direct-lifting structure (white circles represent rounding operations).

where $\mathbf{C}_{II}\mathbf{C}_{III} = \mathbf{I}$. By moving $\mathrm{diag}\{\mathbf{C}_{II}, \mathbf{C}_{II}\}$ to the postprocessing part, Equation 5 is rewritten as

$$\begin{aligned}\mathbf{E}(z) &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}\mathbf{C}_{IV}\mathbf{J}\mathbf{C}_{III} \end{bmatrix} \begin{bmatrix} \mathbf{C}_{II} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{II} \end{bmatrix} \mathbf{W}\mathbf{\Lambda}(z)\mathbf{W} \\ &\quad \times \begin{bmatrix} s_0\mathbf{I} & \mathbf{0} \\ \mathbf{0} & s_1\mathbf{C}_{III}\mathbf{C}_{IV} \end{bmatrix} \mathbf{W}\tilde{\mathbf{I}}\mathbf{J} \\ &= \begin{bmatrix} \mathbf{C}_{II} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}\mathbf{C}_{IV}\mathbf{J} \end{bmatrix} \mathbf{W}\mathbf{\Lambda}(z)\mathbf{W} \begin{bmatrix} s_0\mathbf{I} & \mathbf{0} \\ \mathbf{0} & s_1\mathbf{C}_{III}\mathbf{C}_{IV} \end{bmatrix} \mathbf{W}\tilde{\mathbf{I}}\mathbf{J} \triangleq \tilde{\mathbf{E}}(z) \end{aligned} \tag{6}$$

where $\mathbf{C}_{III}\mathbf{C}_{II} = \mathbf{I}$ and $\tilde{\mathbf{E}}(z)$ are used to distinguish from the original $\mathbf{E}(z)$ in Equation 3. Of course, $\tilde{\mathbf{E}}(z)$ is the same transfer function as $\mathbf{E}(z)$. The FLT with this polyphase matrix $\tilde{\mathbf{E}}(z)$ is implemented as shown at the bottom half in Figure 1.

Next, as already mentioned, we consider the parallel process of two different type FLTs in Equations 3 and 6 as follows:

$$\begin{bmatrix} \mathbf{y}_i \\ \mathbf{y}_j \end{bmatrix} = \begin{bmatrix} \mathbf{E}(z) & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{E}}(z) \end{bmatrix} \begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_j \end{bmatrix}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are individual input signals along process direction, and $\mathbf{y}_i$ and $\mathbf{y}_j$ are their output signals as shown in Figure 1. It means that when a row (column) signals are processed by Equation 3, other row (column) signals are processed by Equation 6. However, each DCT matrix in both FLT is processed by direct-lifting of each combination of DCT-II/DCT-III, DCT-III/DCT-II, and DCT-IV/DCT-IV as shown in dashed line box in Figure 1. For example, the combination of DCT-II/DCT-III is factorized as

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

by substituting it into Equation 4.

3.2 Lifting structure of rotation matrix with $\pi/4$ angle

Since $\mathbf{W}$ in Equations 3 and 6 and Figure 1 includes the scaling factor $1/\sqrt{2}$, we factorize this into lifting structure. In [14], $\mathbf{W}$ is simply factorized as

$$\mathbf{W} = \begin{bmatrix} \mathbf{I} & w_0\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ w_1\mathbf{I} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & w_0\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \end{bmatrix}$$

where $w_0 = 1 - \sqrt{2}$ and $w_1 = 1/\sqrt{2}$. But this factorization includes many floating-point multipliers. To eliminate as much multipliers as possible, the following scaled matrices are used in place of pure $\mathbf{W}$.

$$\begin{aligned} \begin{bmatrix} \frac{1}{\sqrt{2}}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \sqrt{2}\mathbf{I} \end{bmatrix} \mathbf{W} &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \frac{1}{2}\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{I} & \mathbf{I} \end{bmatrix} \triangleq \mathbf{W}_1 \\ \mathbf{W} \begin{bmatrix} \sqrt{2}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\sqrt{2}}\mathbf{I} \end{bmatrix} &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{I} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\frac{1}{2}\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \end{bmatrix} \triangleq \mathbf{W}_2 \\ \begin{bmatrix} \sqrt{2}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\sqrt{2}}\mathbf{I} \end{bmatrix} \mathbf{W} &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \frac{1}{2}\mathbf{I} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \end{bmatrix} \triangleq \mathbf{W}_3 \\ \mathbf{W} \begin{bmatrix} \frac{1}{\sqrt{2}}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \sqrt{2}\mathbf{I} \end{bmatrix} &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\frac{1}{2}\mathbf{I} & \mathbf{I} \end{bmatrix} \triangleq \mathbf{W}_4 \end{aligned}$$

Note that a lifting structure with coefficient 1/2 and rounding operation can be replaced by one adder and one bit-shifter [37], i.e., multiplierless operations. With

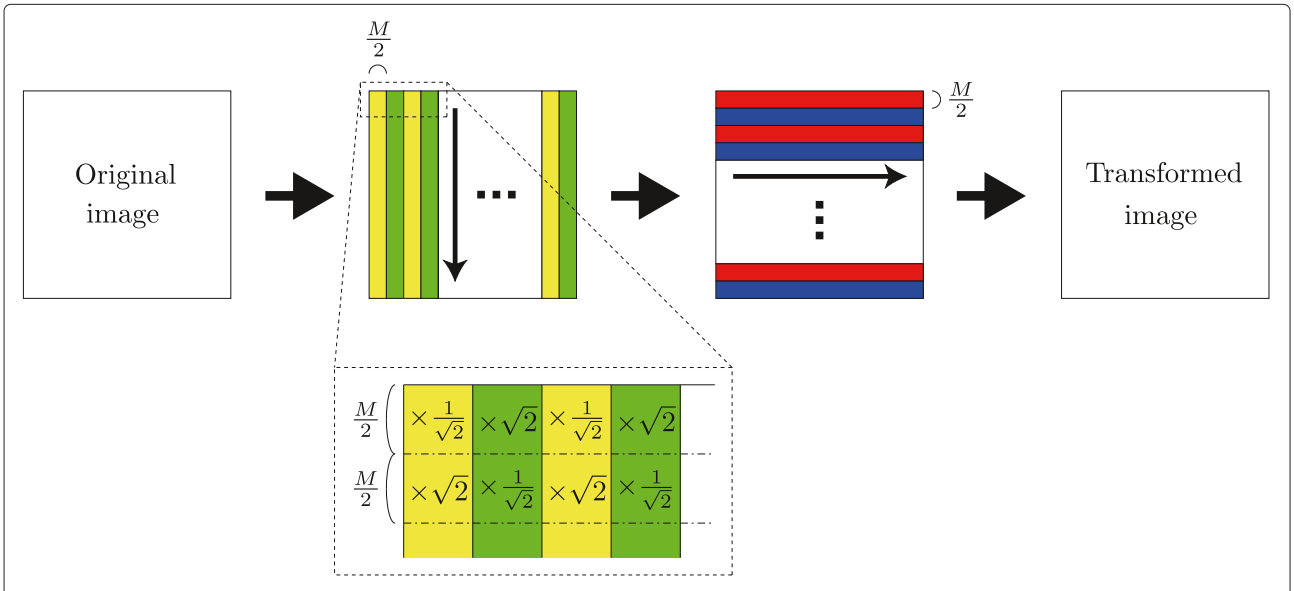

Figure 3 Process of the two-dimensional transform of an image by the proposed IntFLTs.

these matrices, Equations 3 and (6) are represented as follows:

$$\begin{bmatrix} \frac{1}{\sqrt{2}}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \sqrt{2}\mathbf{I} \end{bmatrix} \mathbf{E}(z) = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}\mathbf{C}_{IV}\mathbf{J}\mathbf{C}_{III} \end{bmatrix} \mathbf{W}_1 \Lambda(z) \mathbf{W}_2 \times \begin{bmatrix} s_0\mathbf{C}_{II} & \mathbf{0} \\ \mathbf{0} & s_1\mathbf{C}_{IV} \end{bmatrix} \mathbf{W}_1 \tilde{\mathbf{I}}\mathbf{J} \quad (7)$$

$$\begin{bmatrix} \sqrt{2}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\sqrt{2}}\mathbf{I} \end{bmatrix} \tilde{\mathbf{E}}(z) = \begin{bmatrix} \mathbf{C}_{II} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}\mathbf{C}_{IV}\mathbf{J} \end{bmatrix} \mathbf{W}_3 \mathbf{\Lambda}(z) \mathbf{W}_2 \times \begin{bmatrix} s_0\mathbf{I} & \mathbf{0} \\ \mathbf{0} & s_1\mathbf{C}_{III}\mathbf{C}_{IV} \end{bmatrix} \mathbf{W}_1 \tilde{\mathbf{I}}\mathbf{J} \quad (8)$$

Figure 3 shows the process that an image is two-dimensionally transformed by the proposed IntFLT. The ith $(1 \leq ((i+1) \bmod M) \leq M/2)$ row signals and the jth $((M/2+1) \leq ((j+1) \bmod M) \leq M)$ row signals, i.e., the yellow and green areas in Figure 3, are processed by FLTs in Equations 7 and 8, respectively. Here, note that the one-dimensionally transformed output signals are scaled by $1/\sqrt{2}$ and $\sqrt{2}$ as compared with the output signals transformed by normal FLTs as shown in the dashed line box in Figure 3. By considering these scales $1/\sqrt{2}$ and $\sqrt{2}$ for the next column process, Equations 3 and 6 are represented again as follows:

$$\mathbf{E}(z) \begin{bmatrix} \sqrt{2}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\sqrt{2}}\mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}\mathbf{C}_{IV}\mathbf{J}\mathbf{C}_{III} \end{bmatrix} \mathbf{W}_2 \Lambda(z) \mathbf{W}_1 \times \begin{bmatrix} s_0\mathbf{C}_{II} & \mathbf{0} \\ \mathbf{0} & s_1\mathbf{C}_{IV} \end{bmatrix} \mathbf{W}_2 \tilde{\mathbf{I}}\mathbf{J} \quad (9)$$

Table 1 Comparisons of coding gain of the ideal FLTs and the proposed IntFLTs (dB)

	Ideal FLTs	Proposed FLTs
8 × 16 FLOT	9.2189	9.2189
8 × 16 FLBT	9.4475	9.4475
16 × 32 FLOT	9.7593	9.7593
16 × 32 FLBT	9.8455	9.8455

$$\tilde{\mathbf{E}}(z) \begin{bmatrix} \frac{1}{\sqrt{2}}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \sqrt{2}\mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{II} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}\mathbf{C}_{IV}\mathbf{J} \end{bmatrix} \mathbf{W}_4 \Lambda(z) \mathbf{W}_3 \times \begin{bmatrix} s_0\mathbf{I} & \mathbf{0} \\ \mathbf{0} & s_1\mathbf{C}_{III}\mathbf{C}_{IV} \end{bmatrix} \mathbf{W}_4 \tilde{\mathbf{I}}\mathbf{J}. \quad (10)$$

Similarly, the ith column signals and the jth column signals, i.e., the red and blue areas in Figure 3, are processed by FLTs in Equations 9 and 10, respectively. Consequently, the scales are changed temporarily for fast implementation and restored after two-dimensional transform.

3.3 Lifting structure of scaling part

In this subsection, we present lifting structures of each scaling part $\mathrm{diag}\{s_0\mathbf{I}, s_1\mathbf{I}\}$ including in Equations 7 to 10. According to Equation 4, we define a simple realization of integer transform in the scaling part as follows:

$$\begin{bmatrix} s_0\mathbf{I} & \mathbf{0} \\ \mathbf{0} & s_1\mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ s_0\mathbf{I} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -s_1\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ s_0\mathbf{I} & \mathbf{I} \end{bmatrix}$$

where $s_1 = s_0^{-1}$. The lifting coefficients s_0 and s_1 in the scaling part are empirically determined.

Table 2 Comparison of lossless image coding (LBR (bpp))

Test	5/3-DWT	HLT	Conventional FLTs					Proposed FLTs		
Images	[14]	[16]	(A)	(B)	(C)	(D)	(E)	(F)	(G)	(H)
Baboon	6.25	6.23	6.24	6.24	6.23	6.24	6.23	*6.22*	*6.22*	*6.22*
Barbara	4.97	4.96	5.00	4.95	4.95	4.93	4.95	4.85	4.90	*4.83*
Boat	5.19	5.20	5.22	5.22	5.19	5.21	5.19	5.16	5.16	*5.15*
Elaine	5.26	5.27	5.30	5.26	5.27	5.25	5.28	5.21	5.25	*5.20*
Finger	5.88	5.89	5.91	5.79	5.85	5.78	5.89	*5.75*	5.84	*5.75*
Finger2	5.64	5.62	5.65	5.57	5.57	5.56	5.63	*5.51*	5.55	*5.51*
Goldhill	*5.08*	5.12	5.21	5.20	5.17	5.19	5.18	5.15	5.15	5.14
Grass	6.09	6.09	6.11	6.09	6.08	6.08	6.10	*6.07*	6.08	*6.07*
Lena	*4.58*	4.64	4.74	4.77	4.71	4.75	4.71	4.69	4.66	4.67
Pepper	*4.96*	5.00	5.03	5.06	4.99	5.04	4.99	5.00	*4.96*	4.98
Avg.	5.39	5.40	5.44	5.42	5.40	5.40	5.42	5.36	5.38	*5.35*

A, the conventional 8 × 16 FLOT; B, the conventional 16 × 32 FLOT; C, the conventional 8 × 16 FLBT; D, the conventional 16 × 32 FLBT; E, the proposed 8 × 16 FLOT; F, the proposed 16 × 32 FLOT; G, the proposed 8 × 16 FLBT; and H, the proposed 16 × 32 FLBT.

4 Results

4.1 Coding gain

This paper designed 8 × 16 and 16 × 32 IntFLTs. First, the comparison of coding gain of the ideal FLTs and the proposed IntFLTs is shown.

The coding gain is one of the most important factors to be considered 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 biorthogonal coding gain is defined as [38]

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

where σ_x^2 is the variance of the input signal, $\sigma_{x_k}^2$ is the variance of the kth subbands and $\|f_k\|^2$ is the norm of the kth synthesis filter. Although the coding gain does not completely dominate all image coding results due to rounding error, it is clear that all of coding gain are not lost as shown in Table 1.

For comparison, the coding gain of LiftLT [12] is 9.5378 (dB) which is higher than the proposed 8 × 16 IntFLTs because this is optimized for lossy coding.

Table 3 Comparison of lossy image coding (PSNR (dB))

	Comp.	LiftLT	9/7-DWT	HLT	Conventional FLTs				Proposed FLTs			
	Ratio	[12]	[14]	[16]	(A)	(B)	(C)	(D)	(E)	(F)	(G)	(H)
Baboon	1 : 32	22.72	22.90	22.38	22.86	22.88	22.88	22.89	22.87	*22.91*	22.89	*22.91*
	1 : 16	24.98	25.11	24.67	25.11	25.17	25.14	25.15	25.13	*25.19*	25.16	*25.19*
	1 : 8	28.31	*28.55*	28.10	28.49	28.45	28.48	28.42	*28.55*	28.53	28.52	28.51
Barbara	1 : 32	28.01	27.56	27.01	27.80	28.72	28.02	28.84	27.83	28.77	28.03	*28.90*
	1 : 16	32.02	31.49	30.85	31.70	32.55	32.02	32.65	31.76	32.67	32.08	*32.80*
	1 : 8	37.07	36.28	36.00	36.33	36.65	36.60	36.66	36.59	37.13	36.84	*37.19*
Boat	1 : 32	29.20	*29.43*	28.80	28.93	28.98	29.21	29.08	28.97	29.04	29.23	29.13
	1 : 16	32.36	*32.45*	32.02	32.05	32.00	32.26	32.08	32.13	32.13	32.33	32.22
	1 : 8	*35.70*	35.50	35.21	35.20	35.07	35.23	35.04	35.44	35.43	35.46	35.40
Elaine	1 : 32	31.83	*31.99*	31.54	31.28	31.29	31.55	31.42	31.32	31.35	31.61	31.50
	1 : 16	32.86	*32.98*	32.20	32.39	32.63	32.53	32.74	32.49	32.78	32.62	32.87
	1 : 8	35.00	34.77	34.24	34.46	34.97	34.44	34.94	34.67	*35.32*	34.69	35.26
Finger	1 : 32	23.42	23.69	22.95	23.56	23.94	23.71	23.99	23.57	23.97	23.72	*24.02*
	1 : 16	26.62	26.92	26.31	26.76	27.27	26.92	27.29	26.79	27.32	26.94	*27.34*
	1 : 8	30.67	30.50	30.12	30.57	31.29	30.76	31.30	30.64	*31.44*	30.82	*31.44*
Finger2	1 : 32	24.36	24.63	23.27	24.40	24.72	24.62	24.79	24.42	24.76	24.63	*24.82*
	1 : 16	27.73	28.04	27.09	27.84	28.19	28.09	28.26	27.86	28.25	28.12	*28.32*
	1 : 8	31.83	31.85	31.39	31.80	32.24	32.10	32.23	31.89	*32.42*	32.19	32.39
Goldhill	1 : 32	29.72	*30.06*	29.62	29.39	29.53	29.66	29.62	29.43	29.59	29.69	29.68
	1 : 16	32.35	*32.37*	32.02	31.97	32.06	32.15	32.10	32.05	32.20	32.22	32.25
	1 : 8	*35.57*	35.36	35.17	35.03	34.98	35.06	34.93	35.26	35.36	35.29	35.30
Grass	1 : 32	24.44	24.63	24.26	24.58	24.62	24.66	24.63	24.61	24.66	*24.68*	24.65
	1 : 16	26.65	26.73	26.37	26.76	26.87	26.81	26.87	26.78	*26.92*	26.84	*26.92*
	1 : 8	29.67	29.62	29.39	29.74	29.90	29.77	29.86	29.80	*30.00*	29.81	29.98
Lena	1 : 32	32.80	*33.46*	32.76	32.27	32.44	32.83	32.69	32.37	32.55	32.89	32.83
	1 : 16	36.04	*36.32*	35.90	35.42	35.42	35.92	35.65	35.59	35.71	36.08	35.96
	1 : 8	*39.02*	38.83	38.62	37.95	37.57	38.09	37.64	38.47	38.53	38.55	38.50
Pepper	1 : 32	32.35	*32.89*	32.52	31.62	31.73	32.33	31.99	31.68	31.84	32.38	32.13
	1 : 16	34.69	*35.13*	34.71	34.13	33.78	34.58	34.09	34.29	34.02	34.73	34.34
	1 : 8	*37.07*	36.79	36.39	36.18	35.94	36.21	35.91	36.50	36.44	36.53	36.41

A, the conventional 8 × 16 FLOT; B, the conventional 16 × 32 FLOT; C, the conventional 8 × 16 FLBT; D, the conventional 16 × 32 FLOT; E, the proposed 8 × 16 FLOT; F, the proposed 16 × 32 FLOT; G, the proposed 8 × 16 FLBT; and H, the proposed 16 × 32 FLBT.

4.2 Lossy-to-lossless image coding

Lossy-to-lossless image coding results by the designed IntFLTs are shown in this subsection. As targets for comparison, LiftLT [12], 5/3-DWT and 9/7-DWT for JPEG 2000 [14], HLT for JPEG XR [16], and the conventional 8×16 and 16×32 IntFLTs were applied. The conventional 8×16 and 16×32 IntFLTs are based on simple three-step lifting factorizations of rotation matrices and scaling factors [14]. The periodic extension was used for image boundaries except for DWTs and HLT. To evaluate transform performance fairly, a very common wavelet-based zerotree coder SPIHT [39] was adopted for all[c]. Moreover, we used 8-bit gray scale test images with 512×512 size such as Barbara.

First, the proposed IntFLTs and the conventional methods are applied to lossless image coding. The comparison of lossless bit rate (LBR)

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

is shown in Table 2.

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, is shown in Table 3.

Even though the proposed and conventional IntFLTs have same transfer function, the proposed IntFLTs perform better coding than the conventional IntFLTs, especially lossy image coding results show excellent performance. We consider that this is mainly due to the reduction of rounding operations as shown in Table 4 and no large lifting coefficients[d]. Moreover, note that the proposed IntFLTs have a more effective implementation than the conventional IntFLTs due to the construction with few operations and direct-lifting of DCTs. The direct-lifting can reuse any existing software/hardware for DCTs. On the other hand, LiftLT and 9/7-DWT perform often good lossy image coding because they were designed for the lossy mode. However, it cannot preserve the high frequency components in the images as shown in Figure 4, whereas the proposed IntFLTs, especially the proposed 16×32 IntFLT, can preserve them.

Table 4 Comparison of number of rounding operations in each one-dimensional transform of M × 1 signals

	Conventional FLTs	Proposed FLTs
8×16 FLOT	72	*36*
8×16 FLBT	84	*48*
16×32 FLOT	240	*90*
16×32 FLBT	264	*114*

5 Conclusions

This paper presented integer fast lapped transforms (IntFLTs) for effective lossy-to-lossless image coding, which were constructed by few operations and direct-lifting of discrete cosine transforms (DCTs). Due to merging, many rounding operations and keeping small lifting coefficients by use of direct-lifting, the proposed IntFLTs performed better coding than the conventional IntFLTs in lossy-to-lossless image coding. Also, the proposed IntFLTs can preserve the high frequency components in the images. Since the direct-lifting can reuse any existing software/hardware for DCTs, the proposed IntFLTs have a great potential for fast implementation which is dependent on the architecture design and DCT algorithms. Furthermore, the proposed IntFLTs do not need any side information

Figure 4 Comparison of particular area of reconstructed image Barbara when bit rate is 0.25 (bpp). (Left to right) 9/7-DWT, LiftLT, and 16×32 FLBT.

unlike IntDCT based on direct-lifting as our previous work.

Endnotes

[a]"The conventional IntFLTs" do not include LiftLT in this paper.

[b]Any other lifting-based DCTs cannot reuse all existing software/hardware for DCTs.

[c]The block transform coefficients through 2^k-channel ($k \in \mathbb{N}$) FLTs are rearranged to a k-level wavelet-like multi-resolution representation, and they are applied to the zerotree coder [40], e.g., 3-level wavelet-like multi-resolution representation when $M = 8$.

[d]The IntFLT referred by [20] has less rounding operations. However, it performs undesirable coding due to large lifting coefficients. For example, although the 8×16 FLOT has only five rounding operations in each 4×4 DCT, its application of lossless image coding shows 5.06 (bpp) for Barbara.

Competing interests
The authors declare that they have no competing interests.

Acknowledgements
The authors would like to thank the anonymous reviewers for providing many constructive suggestions that significantly improve the presentation of this paper. This work was supported by JSPS Grant-in-Aid for Young Scientists (B) grant number 25820152.

Author details
[1]Faculty of Engineering, Information and Systems, University of Tsukuba, Tsukuba 305-8573, Japan. [2]Department of Electronics and Electrical Engineering, Keio University, Yokohama 223-8522, Japan.

Received: 24 July 2012 Accepted: 3 December 2013
Published: 27 December 2013

References

1. GK Wallace, The JPEG still picture compression standard. IEEE Trans. Consum. Electr. **38**, 18–34 (1992)
2. W Pennebaker, J Mitchell, *JPEG, Still Image Data Compression Standard* (Van Nostrand, New York, 1993)
3. T Wiegand, GJ Sullivan, G Bjntegaard, A Luthra, Overview of the H.264/AVC video coding standard. IEEE Trans. Circuits Syst. Video Technol. **13**(7), 560–576 (2003)
4. GJ Sullivan, JR Ohm, WJ Han, T Wiegand, Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. Circuits Syst. Video Technol. **22**(12), 1649–1668 (2012)
5. KR Rao, P Yip, *Discrete Cosine Transform Algorithms* (Academic Press, Boston, 1990)
6. AK Jain, A fast Karhunen-Loeve transform for a class of random processes. IEEE Trans. Commun. **24**(9), 1023–1029 (1976)
7. WH Chen, CH Smith, SC Fralick, A fast computational algorithm for the discrete cosine transform. IEEE Trans. Commun. **25**(9), 1004–1009 (1977)
8. Z Wang, Fast algorithms for the discrete W transform and for the discrete Fourier transform. IEEE Trans. Acoust. Speech Signal Process. **ASSP-32**(4), 803–816 (1984)
9. BG Lee, A new algorithm to compute the discrete cosine transform. IEEE Trans. Acoust. Speech Signal, Process. **32**(6), 1243–1245 (1984)
10. Z Wang, On computing the discrete Fourier and cosine transforms. IEEE Trans. Acoust. Speech Signal, Process. **33**(4), 1341–1344 (1985)
11. HS Malvar, *Signal Processing with Lapped Transforms* (Artech House, Norwood, 1992)
12. TD Tran, The LiftLT: Fast lapped transforms via lifting steps. IEEE Signal Process. Lett. **7**(6), 145–148 (2000)
13. A Skodras, C Christopoulis, T Ebrahimi, The JPEG2000 still image compression standard. IEEE Signal Process. Mag. **18**(5), 36–58 (2001)
14. I Daubechies, W Sweldens, Factoring wavelet transforms into lifting steps. J. Fourier Anal. Appl. **4**(3), 245–267 (1998)
15. F Dufaux, GJ Sullivan, T Ebrahimi, The JPEG XR image coding standard. IEEE Signal Process. Mag. **26**(6), 195–199 (2009)
16. C Tu, S Srinivasan, GJ Sullivan, S Regunathan, HS Malvar, Low-complexity hierarchical lapped transform for lossy-to-lossless image coding in JPEG XR/HD photo, in *Proceedings of SPIE Application of Digital Image Processing XXXI*, vol. 7073, (San Diego, 12 Aug 2008), p. 70730
17. W Sweldens, The Lifting Scheme: A New Philosophy in Biorthogonal Wavelet Constructions, in *Proc. of SPIE Wavelet Applications in Signal and Image Processing III*, vol. 2569, (San Diego, 1 Sept 1995), pp. 68–79
18. W Sweldens, The lifting scheme: a custom-design construction of biorthogonal wavelets. Elsevier Appl. Comput. Harmon. Anal. **3**(2), 186–200 (1996)
19. W Sweldens, The lifting scheme: a construction of second generation wavelets. SIAM J. Math. Anal. **29**(2), 511–546 (1997)
20. P Hao, Q Shi, Matrix factorizations for reversible integer mapping. IEEE Trans. Signal Process. **49**(10), 2314–2324 (2001)
21. YJ Chen, KS Amaratunga, *M*-Channel lifting factorization of perfect reconstruction filter banks and reversible *M*-band wavelet transforms. IEEE Trans. Circuits Syst. II. **50**(12), 963–976 (2003)
22. YJ Chen, S Oraintara, KS Amaratunga, Dyadic-based factorizations for regular paraunitary filterbanks and *M*-band orthogonal wavelets with structural vanishing moments. IEEE Trans. Signal Process. **53**, 193–207 (2005)
23. T Suzuki, Y Tanaka, M Ikehara, Lifting-based paraunitary filterbanks for lossy/lossless image coding, in *Proc. of EURASIP EUSIPCO'06* (Florence, 4–8 Sept 2006), pp. 1–5
24. Y She, P Hao, Y Paker, Matrix factorizations for parallel integer transformation. IEEE Trans. Signal Process. **54**(12), 4675–4684 (2006)
25. S Iwamura, Y Tanaka, M Ikehara, An efficient lifting structure of biorthogonal filter banks for lossless image coding, in *Proc. of IEEE ICIP'07* (San Antonio, 16 Sept–19 Sept 2007), pp. 433–436
26. T Suzuki, M Ikehara, Lifting Factorization based on Block Parallel System of M-Channel Perfect Reconstruction Filter Banks, in *Proc. of EURASIP EUSIPCO'09* (Glasgow, 24–28 Aug 2009), pp. 1–4
27. T Suzuki, M Ikehara, *M*-channel paraunitary filter banks based on direct lifting structure of building block and its inverse transform for lossless-to-lossy image coding. IEICE Trans. Fundamentals. **E93-A**(8), 1457–1464 (2010)
28. T Suzuki, M Ikehara, TQ Nguyen, Generalized block-lifting factorization of *M*-channel biorthogonal filter banks for lossy-to-lossless image coding. IEEE Trans. Image Process. **21**(7), 3220–3228 (2012)
29. K Komatsu, K Sezaki, Reversible discrete cosine transform, in *Proc. of IEEE ICASSP'98* (Seattle, 12–15 May 1998), pp. 1769–1772
30. YJ Chen, S Oraintara, TQ Nguyen, Integer discrete cosine transform (IntDCT), in *Proc. of IEEE 2nd ICICS'99* (Singapore, 7–10 Dec 1999), pp. 1–5
31. TD Tran, The BinDCT: fast multiplierless approximation of the DCT. IEEE Signal Process. Lett. **7**(6), 141–144 (2000)
32. J Liang, TD Tran, Fast multiplierless approximations of the DCT with the lifting scheme. IEEE Trans. Signal Process. **49**(12), 3032–3044 (2001)
33. S Chokchaitam, M Iwahashi, S Jitapunkul, A new unified lossless/lossy image compression based on a new Integer DCT. IEICE Trans. Inf Syst. **E88-D**(2), 403–413 (2005)
34. T Suzuki, M Ikehara, Design of block lifting-based discrete cosine transform type-II and IV, in *Proc. of IEEE DSPWS'09* (Marco Island, 4–7 Jan 2009), pp. 480–484
35. T Suzuki, M Ikehara, Integer DCT based on direct-lifting of DCT-IDCT for lossless-to-lossy image coding. IEEE Trans. Image Process. **19**(11), 2958–2965 (2010)
36. T Suzuki, M Ikehara, Integer fast lapped orthogonal transform based on direct-lifting of DCTs for lossless-to-lossy image coding, in *Proc. of IEEE ICASSP'11* (Prague, 22–27 May 2011), pp. 1525–1528
37. T Suzuki, S Kyochi, Y Tanaka, M Ikehara, H Aso, Multiplierless lifting based FFT via fast Hartley transform, in *Proc. of IEEE ICASSP'13* (Vancouver, 26–31 May 2013), pp. 5603–5607

38. G Strang, T Nguyen, *Wavelets and Filter Banks* (Wellesley-Cambridge Press, 1996)
39. A Said, WA Pearlman, A new, fast, and efficient image codec based on set partitioning in hierarchical trees. IEEE Trans. Circuits Syst. Video Technol. **6**(3), 243–250 (1996)
40. TD Tran, TQ Nguyen, A progressive transmission image coder using linear phase uniform filterbanks as block transforms. IEEE Trans. Image Process. **8**(11), 1493–1507 (1999)

doi:10.1186/1687-5281-2013-65
Cite this article as: Suzuki and Ikehara: **Integer fast lapped transforms based on direct-lifting of DCTs for lossy-to-lossless image coding.** *EURASIP Journal on Image and Video Processing* 2013 **2013**:65.