

PAPER

M -Channel Fast Hartley Transform Based Integer DCT for Lossy-to-Lossless Image Coding

Taizo SUZUKI^{†a)}, Member and Hirotomo ASO^{††}, Fellow

SUMMARY This paper presents an M -channel ($M = 2^n$ ($n \in \mathbb{N}$)) integer discrete cosine transforms (IntDCTs) based on fast Hartley transform (FHT) for lossy-to-lossless image coding which has image quality scalability from lossy data to lossless data. Many IntDCTs with lifting structures have already been presented to achieve lossy-to-lossless image coding. Recently, an IntDCT based on *direct-lifting* of DCT/IDCT, which means direct use of DCT and inverse DCT (IDCT) to lifting blocks, has been proposed. Although the IntDCT shows more efficient coding performance than any conventional IntDCT, it entails many computational costs due to an extra information that is a key point to realize its *direct-lifting* structure. On the other hand, the almost conventional IntDCTs without an extra information cannot be easily expanded to a larger size than the standard size $M = 8$, or the conventional IntDCT should be improved for efficient coding performance even if it realizes an arbitrary size. The proposed IntDCT does not need any extra information, can be applied to size $M = 2^n$ for arbitrary n , and shows better coding performance than the conventional IntDCTs without any extra information by applying the *direct-lifting* to the pre- and post-processing block of DCT. Moreover, the proposed IntDCT is implemented with a half of the computational cost of the IntDCT based on *direct-lifting* of DCT/IDCT even though it shows the best coding performance.

key words: *direct-lifting, fast Hartley transform (FHT), integer discrete cosine transform (IntDCT), lossy-to-lossless image coding*

1. Introduction

Discrete cosine transform (DCT) [1] has several types, and DCT type-II (DCT-II) and -III (DCT-III) are commonly called DCT and inverse DCT (IDCT). They are applied to lossy image/video compression (coding) such as the international standard JPEG [2] and H.26x family [3], because DCT has excellent energy compaction capability, many fast implementations [4]–[7] and numerous applications on JPEG devices. Since it is compatible with the lossy JPEG, DCT is especially promising.

Recently, with the rapid development of multimedia devices such as media tablets and smartphones, and the continual expansion of broadband, lossy-to-lossless image coding, which has image quality scalability from lossy data to lossless data, is demanded to respond to the various needs. To achieve it, many integer DCTs (IntDCTs) based on lifting structures [8]–[10] have been proposed [11]–[13]. We also

have proposed a *direct-lifting* which is a so effective structure for lossy-to-lossless image coding [14]. The IntDCT, however, requires an extra information because both of a transform and its inverse transform are required, i.e., both of DCT and IDCT must be simultaneously implemented. As its solution, DCT is separated to the pre- and post-processing block as the first stage. If at least one of the pre-/post-processing blocks is a symmetric orthogonal matrix, the DCT may perform efficient lossy-to-lossless image coding due to application of a *direct-lifting*. Although our previous work in [15] has such structure using a *direct-lifting* of Walsh-Hadamard transform (WHT) as the pre-processing block, it should be improved for more efficient coding performance because there are many rounding operations in the post-processing block. Also, the conventional IntDCTs except for [13], [14] cannot be easily expanded to a larger size $M = 2^n$ ($n \in \mathbb{N}$) than the standard size $M = 8$. Such arbitrary size is needed for the super high resolution standards as 4K Digital Cinema (4096×2160) and 8K Super Hi-Vision (7680×4320). Nevertheless, many rounding operations newly generated cause worse coding performance due to their rounding error when the IntDCT has a larger size than $M = 8$.

On the other hand, faster DCT algorithms without multipliers are widely researched for real time encode/decode implementations [15]–[17]. We also have proposed a multiplierless fast DCT based on fast Hartley transform (FHT) [18]. The structure has an FHT and the residue orthogonal matrix with $(M - 2)/2$ rotation matrices as the pre- and post-processing blocks, respectively. It is easy that the DCT obtains an arbitrary size $M = 2^n$.

In this paper, we adopt *direct-liftings* for both of the pre- and post-processing blocks of DCT based on FHT to achieve effective lossy-to-lossless image coding. We can implement it without any complex calculation because *direct-lifting* can be easily adopted for a combination of two symmetric orthogonal matrices. Both FHT as the pre-processing block and the residue orthogonal matrix as the post-processing block are symmetric orthogonal matrices. Also, two-dimensional (2-D) separable block transform is considered to achieve a more effective implementing. The resulting simple structure avoids a propagation of rounding error generated by cascading rounding operations. As a result, the proposed IntDCT does not need any extra information, can be applied to size $M = 2^n$ for arbitrary n , and shows better coding performance than the conventional IntDCT without any extra information. Moreover, the proposed Int-

Manuscript received July 4, 2012.

Manuscript revised October 19, 2012.

[†]The author is with the Faculty of Engineering, Information and Systems, University of Tsukuba, Tsukuba-shi, 305-8573 Japan.

^{††}The author is with the Department of Electrical and Electronic Engineering, College of Engineering, Nihon University, Koriyama-shi, 963-8642 Japan.

a) E-mail: taizo@cs.tsukuba.ac.jp

DOI: 10.1587/transfun.E96.A.762

Table 1 The comparison of IntDCTs.

	Efficient coding performance	Arbitrary size $M = 2^n$	No side information
[11]	✓		✓
[12]			✓
[13]	✓	✓	✓
[14]	✓(Best)	✓	✓
[15]	✓		✓
Prop.	✓(2nd best)	✓	✓

DCT is implemented with a half of the computational cost of the IntDCT based on *direct-lifting* of DCT/IDCT [14] even though it shows the best coding performance. Table 1 summarizes the comparison of IntDCTs.

Notations: \mathbf{I} , \mathbf{J} , $\{\cdot\}_{[M]}$, $\{\cdot\}^T$ and $\text{diag}\{\mathbf{M}, \mathbf{N}\}$ are an identity matrix, a reversal identity matrix, $N \times N$ matrix, transpose of a matrix and a block diagonal matrix, respectively.

2. Review

2.1 Discrete Cosine Transform (DCT)

Not only is DCT [1] often used for the international standard JPEG [2] and H.26x family [3], but also numerous devices for DCT have been developed. The (m, n) -element of M -channel DCT-II matrix \mathcal{C} , DCT-III matrix \mathcal{D} and DCT-IV matrix \mathcal{E} are defined as

$$\begin{aligned}
 [\mathcal{C}]_{m,n} &= \sqrt{\frac{2}{M}} c_m \cos\left(\frac{m(n+1/2)\pi}{M}\right) \\
 [\mathcal{D}]_{m,n} &= \sqrt{\frac{2}{M}} c_n \cos\left(\frac{(m+1/2)n\pi}{M}\right) \\
 [\mathcal{E}]_{m,n} &= \sqrt{\frac{2}{M}} \cos\left(\frac{(m+1/2)(n+1/2)\pi}{M}\right)
 \end{aligned}$$

where $\mathcal{D} = \mathcal{C}^{-1} = \mathcal{C}^T$, $\mathcal{E}^{-1} = \mathcal{E}^T = \mathcal{E}$, $0 \leq m, n \leq M - 1$, $c_k = 1/\sqrt{2}$ ($k = 0$) or 1 ($k \neq 0$), respectively. For simplicity, let M be defined as $M = 2^n$ ($n \in \mathbb{N}$). Also, many fast implementations of DCT have been widely researched [4]–[7]. For example, Chen’s fast implementation of M -channel DCT $\mathcal{C}_{[M]}$ in [4] is the most popular structure as

$$\mathcal{C}_{[M]} = \frac{1}{\sqrt{2}} \mathbf{P}_{\mathcal{C}_{[M]}} \begin{bmatrix} \mathcal{C}_{[M/2]} & \mathbf{0} \\ \mathbf{0} & \mathcal{E}_{[M/2]} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{J} \\ \mathbf{J} & -\mathbf{I} \end{bmatrix} \quad (1)$$

where $\mathbf{P}_{\mathcal{C}_{[M]}}$ is an $M \times M$ permutation matrix. In case of $M = 8$, it is simplified by 4 rotation matrices with $\pi/4$ angles and 4-channel DCT-II/IV as shown in Fig. 1(a). Note that any coding system by DCT is limited to operation in only lossy image coding because distortion of the decoded image is unavoidable with these lossy algorithms.

2.2 Fast Hartley Transform (FHT)

The FHT is a tool for the frequency analysis, design and implementation of digital signal processing algorithms and systems like DCT. It is strictly symmetrical concerning the

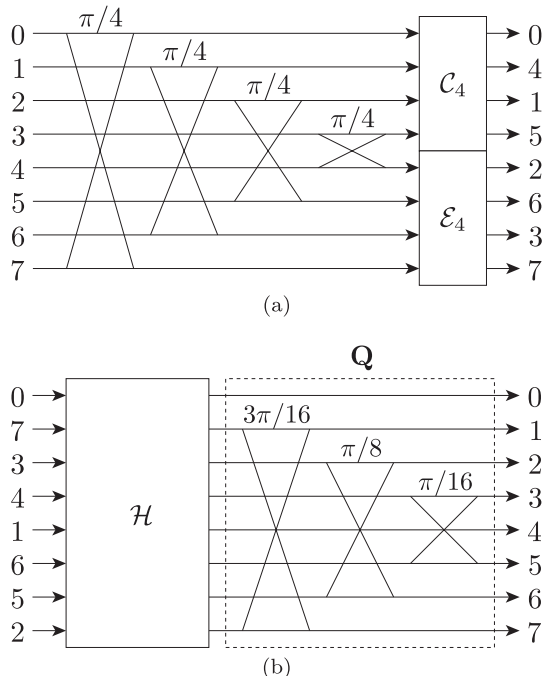


Fig. 1 8-channel DCT \mathcal{C} : (a) Chen’s DCT in [4], (b) FHT based DCT in [18].

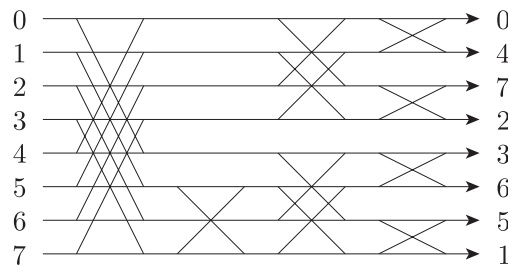


Fig. 2 8-channel FHT $\mathbf{P}\mathcal{H}$.

transformation and its inverse, i.e., a symmetric orthogonal matrix. The (m, n) -element of M -channel FHT matrix \mathcal{H} is defined as [19]

$$[\mathcal{H}]_{m,n} = \frac{1}{\sqrt{M}} \text{cas}\left(\frac{2mn\pi}{M}\right)$$

where $\text{cas } \theta = \cos \theta + \sin \theta$ and $\mathcal{H}^{-1} = \mathcal{H}^T = \mathcal{H}$. Since \mathcal{H} is not according to the frequency band order, let \mathcal{H} with the correct order be $\mathbf{P}\mathcal{H}$ where \mathbf{P} is a permutation matrix. In case of $M = 8$, it is simplified by 13 rotation matrices with $\pi/4$ angles as shown in Fig. 2.

2.3 Direct-Lifting Structure

Lifting structure, also known as a ladder structure, is an implementation method of wavelet transforms originally proposed [8]–[10] by Sweldens. It is a special type of lattice structure: a cascading construction using only elementary matrices, that is, identity matrices with a single nonzero off-diagonal element. Figure 3(a) shows a standard lifting struc-

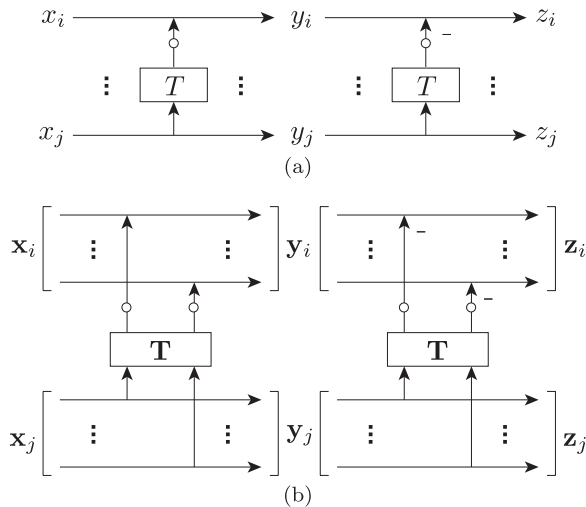


Fig. 3 A lifting structure (white circles mean rounding operations): (a) a standard lifting, (b) a block-lifting.

ture. In this case, the lifting matrix and its inverse matrix are as follows:

$$\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & -T \\ 0 & 1 \end{bmatrix}.$$

Then, they are represented by

$$\begin{aligned} y_i &= x_i + \text{round}\{Tx_j\} & z_i &= y_i - \text{round}\{Ty_j\} = x_i \\ y_j &= x_j & z_j &= y_j = x_j \end{aligned}$$

where T and $\text{round}\{\cdot\}$ are a lifting coefficient and a rounding operation, respectively. The lifting structures with rounding operation can achieve integer-to-integer transform for lossy-to-lossless image coding.

To obtain more efficient transform performance than a transform by standard lifting, *block-lifting* as a class of lifting structure has been proposed [20]. Instead of a lifting coefficient T , we adopt a lifting matrix (block) \mathbf{T} . Such *block-lifting* can be expressed as follows:

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

and

$$\begin{aligned} \mathbf{y}_i &= \mathbf{x}_i + \text{round}\{\mathbf{T}\mathbf{x}_j\} & \mathbf{z}_i &= \mathbf{y}_i - \text{round}\{\mathbf{T}\mathbf{y}_j\} = \mathbf{x}_i \\ \mathbf{y}_j &= \mathbf{x}_j & \mathbf{z}_j &= \mathbf{y}_j = \mathbf{x}_j \end{aligned}$$

where \mathbf{x}_i , \mathbf{x}_j , \mathbf{y}_i , \mathbf{y}_j , \mathbf{z}_i and \mathbf{z}_j are the input and output signal matrices, as shown in Fig. 3(b). Similar to standard lifting, it is clear that the reconstructed signals \mathbf{z}_i and \mathbf{z}_j are exactly the same value as \mathbf{x}_i and \mathbf{x}_j . The *block-lifting* would be efficient for lossy-to-lossless image coding due to having fewer number of rounding operations[†].

Moreover, we have proposed *direct-lifting* as a class of *block-lifting* [14]. First, we process the two individual signals \mathbf{x}_i and \mathbf{s}_i by a matrix \mathbf{M} and its inverse matrix \mathbf{N} , i.e.,

$$\begin{bmatrix} \mathbf{y}_i \\ \mathbf{t}_i \end{bmatrix} = \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{x}_i \\ \mathbf{s}_i \end{bmatrix}$$

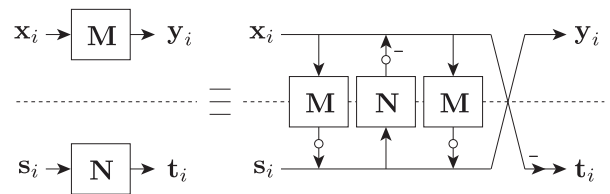


Fig. 4 A direct-lifting structure (white circles mean rounding operations).

where $\mathbf{MN} = \mathbf{NM} = \mathbf{I}$, and \mathbf{y}_i and \mathbf{t}_i are the output signals of \mathbf{x}_i and \mathbf{s}_i , respectively, as shown on the left side of Fig. 4. Then, the matrix $\text{diag}\{\mathbf{M}, \mathbf{N}\}$ can be redefined to lifting structures as

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{N} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{M} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{N} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{M} & \mathbf{I} \end{bmatrix}.$$

This is called *direct-lifting* because two matrices \mathbf{M} and \mathbf{N} are directly used to lifting blocks. To obtain lossless data, a rounding operation is applied to each lifting step because it achieves an integer-to-integer transform without any distortion in the quantization part in image coding flow. Commonly, an extra information is used to implement *direct-lifting* because both of a transform \mathbf{M} and its inverse transform \mathbf{N} must be implemented simultaneously [14]. But the extra information is not required if \mathbf{M} and \mathbf{N} have the symmetric orthogonality, i.e., $\mathbf{N} = \mathbf{M}^{-1} = \mathbf{M}^T = \mathbf{M}$.

3. M-Channel FHT Based Integer DCT (IntDCT)

3.1 FHT Based DCT

A fast FHT based DCT [18] is definitely different from the most popular Chen's structure in (1). The structure is separated to a permutation matrix \mathbf{P}^T , FHT \mathcal{H} as the pre-processing block, the residue orthogonal matrix \mathbf{Q} as the post-processing block and a diagonal matrix \mathbf{D} with either of +1 or -1 values as follows:

$$\mathbf{C} = \mathbf{D}\mathbf{Q}\mathcal{H}\mathbf{P}^T \quad (2)$$

where

$$\mathbf{D} = \begin{bmatrix} \mathbf{I}_{[M/2+1]} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_{[M/2-1]} \end{bmatrix}$$

$$\mathbf{Q} = \begin{bmatrix} 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}\mathbf{C}\mathbf{J} & \mathbf{0} & \mathbf{J}\mathbf{S} \\ \mathbf{0} & \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}\mathbf{J} & \mathbf{0} & -\mathbf{C} \end{bmatrix} \quad \begin{cases} [\mathbf{C}]_{k,k} = \cos\left(\frac{(k+1)\pi}{2M}\right) \\ [\mathbf{S}]_{k,k} = \sin\left(\frac{(k+1)\pi}{2M}\right) \end{cases}$$

for $0 \leq k \leq M/2 - 2$ as shown in Fig. 1(b). Note that this structure can be easily expanded to a larger size than $M = 8$. However, if this DCT is factorized into the lifting part and the scaling part according to [18], it is inapplicable to lossy-to-lossless image coding due to M multipliers as the generated scaling part.

[†]When the number of rounding operations increases, the quantization error simultaneously increased affects the subband energy compaction [13].

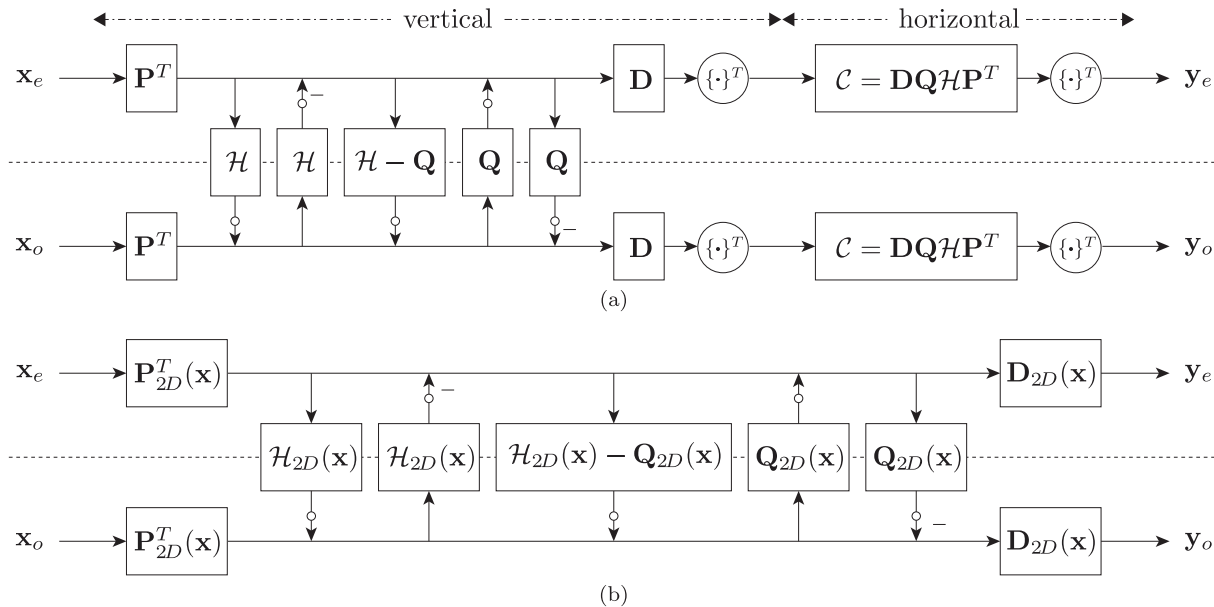


Fig. 5 *M*-channel FHT based IntDCT (white circles mean rounding operations): (a) by using 1-D transform in each of vertical and horizontal directions, (b) by making full use of 2-D separable block transform.

3.2 Direct-Lifting of Pre-/Post-Processing Blocks of DCT

To achieve lossy-to-lossless image coding without any extra information, we apply *direct-lifting* factorization to FHT and the residue orthogonal matrix as the pre- and post-processing blocks of DCT, respectively.

Generally, it is careful that the *direct-lifting* requires an extra information because both transform **M** and its inverse transform **N** (**N** ≠ **M**) must be implemented. However, we can easily come up with a simple realization for lossy-to-lossless image coding without any extra information. First, an *hM* × *wM* (*h, w* ∈ ℕ) image **X** is segmented to *M* × *M* block **x_k** (0 ≤ *k* ≤ *hw* - 1). The segmented block **x_k** is sequentially transformed by *hw* DCTs as follows:

$$[\mathbf{y}_0^T, \mathbf{y}_1^T, \dots, \mathbf{y}_{hw-1}^T]^T = \text{diag}\{C, C, \dots, C\} \cdot [\mathbf{x}_0^T, \mathbf{x}_1^T, \dots, \mathbf{x}_{hw-1}^T]^T$$

We process each combination of the two adjacent signal blocks **x_e** and **x_o** (*e* and *o* are an even and an odd value, respectively) by DCT, i.e.,

$$\begin{aligned} \begin{bmatrix} \mathbf{y}_e \\ \mathbf{y}_o \end{bmatrix} &= \begin{bmatrix} C & \mathbf{0} \\ \mathbf{0} & C \end{bmatrix} \begin{bmatrix} \mathbf{x}_e \\ \mathbf{x}_o \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} \end{bmatrix} \begin{bmatrix} \mathcal{H} & \mathbf{0} \\ \mathbf{0} & \mathcal{H} \end{bmatrix} \begin{bmatrix} \mathbf{P}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^T \end{bmatrix} \begin{bmatrix} \mathbf{x}_e \\ \mathbf{x}_o \end{bmatrix} \end{aligned} \quad (3)$$

where **y_e** and **y_o** are the output signals of **x_e** and **x_o**, respectively. Then, a combination of the pre- and post-processing blocks in (3) is factorized into five step *direct-liftings* as follows:

$$\begin{aligned} \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} \end{bmatrix} \begin{bmatrix} \mathcal{H} & \mathbf{0} \\ \mathbf{0} & \mathcal{H} \end{bmatrix} &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{Q} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{Q} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \\ &\times \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathcal{H} - \mathbf{Q} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathcal{H} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathcal{H} & \mathbf{I} \end{bmatrix} \end{aligned} \quad (4)$$

(4) is shown in Fig. 5(a). Consequently, the two adjacent blocks separated from an image can be simultaneously implemented by (4), and it does not require any extra information. Since **P^T** and **D** have no necessity to be factorized into lifting structures, we have just realized a novel IntDCT by (3) and (4). As mentioned in Sect. 2.3, although an extra information is often used to implement *direct-lifting*, the proposed structure does not require it due to the symmetric orthogonalities of **H** and **Q**. The proof of (4) is presented in the Appendix.

3.3 Making Full Use of Two-Dimensional (2-D) Separable Block Transform

A DCT matrix **C** is commonly applied into an *M* × *M* input signal **x** in column- and row-wise (vertical and horizontal direction) separately. The output signal **y** is expressed by

$$\mathbf{y} = (\mathbf{C}(\mathbf{C}\mathbf{x})^T)^T = \mathbf{C}\mathbf{x}\mathbf{C}^T. \quad (5)$$

We call it a 2-D separable block transform [14]. Since the DCT can be factorized as **C** = **DQHPT** in (2), (5) is represented by

$$\mathbf{y} = \mathbf{D}\mathbf{Q}\mathbf{H}\mathbf{P}^T \mathbf{x} (\mathbf{D}\mathbf{Q}\mathbf{H}\mathbf{P}^T)^T = \mathbf{D}\mathbf{Q}\mathbf{H}\mathbf{P}^T \mathbf{x} \mathbf{P}\mathbf{H}^T \mathbf{Q}^T \mathbf{D}^T.$$

This equation means the 2-D separable block transforms are implemented according to the following order as shown in

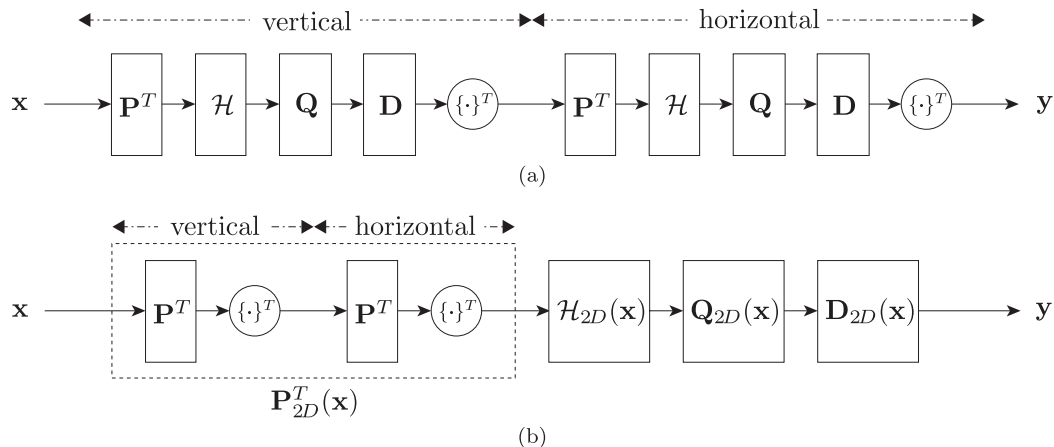


Fig. 6 A 2-D separable block transform: (a) a standard transform, (b) an ordered transform to each block.

Fig. 6(b).

- A 2-D separable block transform by \mathbf{P}^T is implemented.
- A 2-D separable block transform by \mathcal{H} is implemented after (a) is implemented.
- A 2-D separable block transform by \mathbf{Q} is implemented after (b) is implemented.
- A 2-D separable block transform by \mathbf{D} is implemented after (c) is implemented.

Also, let the 2-D separable block transforms of \mathbf{x} by \mathbf{P}^T , \mathcal{H} , \mathbf{Q} and \mathbf{D} be defined as

$$\begin{aligned} \mathbf{P}^T \mathbf{x} \mathbf{P} &\triangleq \mathbf{P}_{2D}^T(\mathbf{x}), & \mathcal{H} \mathbf{x} \mathcal{H}^T &\triangleq \mathcal{H}_{2D}(\mathbf{x}), \\ \mathbf{Q} \mathbf{x} \mathbf{Q}^T &\triangleq \mathbf{Q}_{2D}(\mathbf{x}) & \text{and} & \mathbf{D} \mathbf{x} \mathbf{D}^T &\triangleq \mathbf{D}_{2D}(\mathbf{x}), \end{aligned}$$

respectively. Therefore, (4) are represented by

$$\begin{aligned} &\begin{bmatrix} \mathbf{Q}_{2D}(\mathbf{x}) & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{2D}(\mathbf{x}) \end{bmatrix} \begin{bmatrix} \mathcal{H}_{2D}(\mathbf{x}) & \mathbf{0} \\ \mathbf{0} & \mathcal{H}_{2D}(\mathbf{x}) \end{bmatrix} = \\ &\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{Q}_{2D}(\mathbf{x}) & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{Q}_{2D}(\mathbf{x}) \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathcal{H}_{2D}(\mathbf{x}) - \mathbf{Q}_{2D}(\mathbf{x}) & \mathbf{I} \end{bmatrix} \\ &\quad \times \begin{bmatrix} \mathbf{I} & -\mathcal{H}_{2D}(\mathbf{x}) \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathcal{H}_{2D}(\mathbf{x}) & \mathbf{I} \end{bmatrix} \end{aligned}$$

as shown in Fig. 5(b). Such lifting matrices are more effective for lossy-to-lossless image coding than lifting matrices without considering the 2-D separable block transform, because the number of rounding operations per an one-dimensional (1-D) transform of $M \times 1$ signals is reduced from $(5M - 4)/2$ to $(5M - 4)/4$ in each block and rounding error is also reduced much more. Table 2 shows the number of rounding operations in each IntDCT. It is clear that the number of rounding operations of the proposed structure is less than the previous works except for [12].

4. Experimental Results

The resulting FHT based IntDCT and the conventional IntDCTs in [11]–[15] ($M = 8$) are applied to lossy-to-lossless

Table 2 The number of rounding operations in each M -channel IntDCT. ($N_{\lfloor M/2 \rfloor}$ indicates the number of rounding operations in $M/2$ -channel IntDCT in [13].)

Size M	Conv. IntDCT					Prop. IntDCT
	[11]	[12]	[13]	[14]	[15]	
8	21	8	23	12	21	9
M	–	–	$\frac{9}{4}M + N_{\lfloor M/2 \rfloor}$	$\frac{3}{2}M$	–	$\frac{5}{4}M - 1$

image coding. Also, since the conventional methods in [11], [12], [15] are inapplicable to a larger size DCT than $M = 8$, only the proposed IntDCT and the conventional IntDCTs in [13], [14] ($M = 16$) are applied to lossy-to-lossless image coding as an example of larger sizes. An integer-to-integer transform can be obtained by using a rounding operation at each lifting step. As described above, only the conventional IntDCT in [14] requires a side information, and the others do not require one. To evaluate transform performance fairly, after the images were transformed, they were encoded by the set partitioning in hierarchical trees (SPIHT) progressive image transmission algorithm with PSNR scalability [21]. Test images are several 512×512 grayscale standard images such as *Barbara* and *Lena*, respectively. In lossy image coding, we compare their peak signal-to-noise ratios (PSNRs):

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

where MSE is the mean squared error, for three standard images as shown in Table 3. In lossless image coding, we compare their lossless bit rates (LBRs) [bpp]:

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

as shown in Table 4.

Although our previous work in [14] shows the best performance with an extra information, the proposed IntDCT is implemented with a half of the computational cost of [14], since the proposed one does not require any extra information. It is, however, clear that the proposed IntDCT is better

Table 3 Lossy image coding results (PSNR [dB]).

Test image	Bit rate [bpp]	<i>M</i> = 8						<i>M</i> = 16			
		DCT [1]	Conv. IntDCT					Prop. IntDCT	Conv. IntDCT		Prop. IntDCT
		[11]	[12]	[13]	[14]	[15]		[13]	[14]		
<i>Barbara</i>	0.25	26.95	26.94	26.73	26.94	26.95	26.94	26.94	27.95	27.96	27.96
	0.50	30.70	30.67	30.40	30.67	30.68	30.68	30.67	31.79	31.84	31.83
	1.00	36.08	35.97	35.84	35.96	36.03	35.98	36.00	36.88	37.08	37.04
<i>Finger</i>	0.25	22.55	22.54	22.21	22.55	22.55	22.55	22.54	23.24	23.24	23.24
	0.50	25.57	25.55	25.36	25.55	25.56	25.56	25.56	26.57	26.58	26.58
	1.00	29.70	29.66	29.47	29.67	29.70	29.68	29.69	30.79	30.85	30.84
<i>Goldhill</i>	0.25	29.38	29.36	29.13	29.36	29.37	29.38	29.37	29.71	29.73	29.72
	0.50	31.99	31.94	31.80	31.96	31.97	31.96	31.96	32.33	32.38	32.38
	1.00	35.40	35.27	35.19	35.28	35.36	35.31	35.33	35.51	35.69	35.66
<i>Lena</i>	0.25	31.88	31.87	31.86	31.87	31.89	31.88	31.88	32.67	32.70	32.70
	0.50	35.60	35.53	35.49	35.55	35.59	35.57	35.58	36.10	36.21	36.19
	1.00	39.23	38.95	38.93	38.94	39.13	38.99	39.07	39.04	39.44	39.37
<i>Pepper</i>	0.25	31.43	31.40	31.43	31.41	31.42	31.41	31.41	32.06	32.09	32.09
	0.50	34.49	34.41	34.33	34.42	34.46	34.41	34.45	34.56	34.65	34.64
	1.00	37.04	36.83	36.71	36.82	36.97	36.85	36.92	36.77	37.03	37.00

Table 4 Lossless image coding results (LBR [bpp]).

Test image	<i>M</i> = 8					<i>M</i> = 16			
	Conv. IntDCT					Prop. IntDCT	Conv. IntDCT		Prop. IntDCT
	[11]	[12]	[13]	[14]	[15]		[13]	[14]	
<i>Airplane</i>	4.44	4.40	4.43	4.36	4.40	4.39	4.44	4.33	4.36
<i>Baboon</i>	6.27	6.28	6.27	6.27	6.27	6.27	6.22	6.21	6.21
<i>Barbara</i>	4.98	4.97	4.98	4.94	4.97	4.96	4.85	4.79	4.80
<i>Boat</i>	5.20	5.19	5.20	5.18	5.20	5.18	5.14	5.10	5.11
<i>Bridge</i>	6.00	6.00	6.00	5.99	6.00	5.99	6.00	5.99	5.99
<i>Elaine</i>	5.23	5.25	5.23	5.22	5.23	5.23	5.18	5.16	5.17
<i>Finger</i>	6.07	6.06	6.06	6.06	6.06	6.06	5.84	5.83	5.83
<i>Goldhill</i>	5.17	5.16	5.16	5.14	5.16	5.15	5.11	5.08	5.08
<i>Grass</i>	6.17	6.18	6.17	6.17	6.17	6.17	6.11	6.11	6.11
<i>Lena</i>	4.65	4.63	4.65	4.61	4.63	4.62	4.61	4.56	4.57
<i>Pepper</i>	4.96	4.96	4.96	4.94	4.96	4.94	4.94	4.91	4.91
<i>Tank</i>	5.20	5.20	5.20	5.19	5.20	5.19	5.20	5.19	5.19

than the conventional methods without any extra information in both lossy and lossless image coding, because the proposed IntDCT has a few number of rounding operations as shown in Table 2 and the simple structure avoids a propagation of rounding error generated by cascading rounding operations. Note that Chokchaitam's IntDCT in [12] with the fewest rounding operations in case of $M = 8$ cannot be performed enough due to several removed scaling factors.

5. Conclusion

This paper presented an M -channel ($M = 2^n$ ($n \in \mathbb{N}$)) fast Hartley transform (FHT) based integer discrete cosine transforms (IntDCTs) for lossy-to-lossless image coding. It was obtained by applying *direct-lifting* factorization to each of FHT and the residue orthogonal matrix as the pre- and post-processing blocks of DCT, respectively, and considering a two-dimensional (2-D) separable block transform. The resulting IntDCT shows better coding performance than the conventional IntDCT without any extra information, which causes its handling trouble, due to a few rounding operations. Moreover, the proposed IntDCT can easily achieve an arbitrary size $M = 2^n$, which is required for the super high resolution standards as 4K Digital Cinema (4096×2160)

and 8K Super Hi-Vision (7680×4320), than $M = 8$.

References

- [1] K.R. Rao and P. Yip, *Discrete Cosine Transform Algorithms*, Academic Press, 1990.
- [2] ISO/IEC 10918-1, *Information technology — Digital compression and coding of continuous-tone still images: Requirements and Guidelines*.
- [3] ISO/IEC 14496-10, *Information technology — Coding of audio-visual objects — Part 10: Advanced Video Coding*.
- [4] W.H. Chen, C.H. Smith, and S.C. Fralick, "A fast computational algorithm for the discrete cosine transform," *IEEE Trans. Commun.*, vol.CON-25, no.9, pp.1004–1009, Sept. 1977.
- [5] Z. Wang, "Fast algorithms for the discrete W transform and for the discrete Fourier transform," *IEEE Trans. Acoust. Speech Signal Process.*, vol.ASSP-32, no.4, pp.803–816, April 1984.
- [6] B.G. Lee, "A new algorithm to compute the discrete cosine transform," *IEEE Trans. Acoust. Speech Signal Process.*, vol.ASSP-32, no.6, pp.1243–1245, June 1984.
- [7] Z. Wang, "On computing the discrete Fourier and cosine transforms," *IEEE Trans. Acoust. Speech Signal Process.*, vol.ASSP-33, no.4, pp.1341–1344, April 1985.
- [8] W. Sweldens, "The lifting scheme: A new philosophy in biorthogonal wavelet constructions," *Proc. SPIE 2569*, 1995.
- [9] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Appl. Comput. Harmon. Anal.*, vol.3, no.2, pp.186–200, April 1996.

- [10] W. Sweldens, "The lifting scheme: A construction of second generation wavelets," SIAM J. Math. Anal., vol.29, no.2, pp.511–546, 1997.
- [11] S. Fukuma, K. Ohyama, M. Iwahashi, and N. Kambayashi, "Lossless 8-point fast discrete cosine transform using lossless Hadamard transform," IEICE Technical Report, DSP99-103, Oct. 1999.
- [12] S. Chokchaitam, M. Iwahashi, and S. Jitapunkul, "A new unified lossless/lossy image compression based on a new Integer DCT," IEICE Trans. Inf. & Syst., vol.E88-D, no.2, pp.403–413, Feb. 2005.
- [13] T. Suzuki and M. Ikehara, " M -channel integer discrete cosine transform using block lifting factorization," IEICE Trans. Fundamentals (Japanese Edition), vol.J92-A, no.11, pp.888–900, Nov. 2009.
- [14] 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.
- [15] T. Suzuki and M. Ikehara, "Integer discrete cosine transform via lossless Walsh-Hadamard transform with structural regularity for low-bit-word-length," IEICE Trans. Fundamentals, vol.E93-A, no.4, pp.734–741, April 2010.
- [16] 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.
- [17] O. Nishida, M. Iwahashi, and N. Kambayashi, "A word length allocation method for multipliers of the lossless DCT," IEICE Trans. Fundamentals (Japanese Edition), vol.J86-A, no.6, pp.694–702, June 2003.
- [18] T. Suzuki, Y. Tanaka, M. Ikehara, and H. Aso, "Multiplierless fast algorithm for DCT via fast Hartley transform," Proc. ICASSP'12, pp.3469–3472, Kyoto, Japan, March 2012.
- [19] H.J. Meckelburg and D. Lipka, "Fast Hartley transform algorithm," IET Electron. Lett., vol.21, no.8, pp.311–313, April 1985.
- [20] 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.6, pp.3220–3228, 2012.
- [21] 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.

Appendix: Direct-Lifting Factorization of Pre-/Post-Processing of DCT

The proof of this derivation can be accomplished by using a *direct-lifting* and a simple matrix manipulation. First, the *direct-lifting* factorization can be easily applied to each of FHT and the residue orthogonal matrix, respectively, because FHT \mathcal{H} and the residue orthogonal matrix \mathbf{Q} are symmetric orthogonal matrices, i.e., their inverses are themselves. However, to achieve a more effective implementation than a simple application of *direct-lifting* factorization to them, the only pre-processing block $\text{diag}\{\mathcal{H}, \mathcal{H}\}$ is factorized into *direct-liftings* at the first step as follows:

$$\begin{bmatrix} \mathcal{H} & \mathbf{0} \\ \mathbf{0} & \mathcal{H} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathcal{H} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathcal{H} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathcal{H} & \mathbf{I} \end{bmatrix} \quad (\text{A}\cdot 1)$$

where $\mathcal{H}\mathcal{H} = \mathbf{I}$. Next, if the first matrix in (A·1) is multiplied by the right side of the post-processing block $\text{diag}\{\mathbf{Q}, \mathbf{Q}\}$ as

$$\begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{Q} \\ -\mathbf{Q} & \mathbf{0} \end{bmatrix},$$

this matrix is factorized into the following *direct-liftings*.

$$\begin{bmatrix} \mathbf{0} & \mathbf{Q} \\ -\mathbf{Q} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{Q} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{Q} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{Q} & \mathbf{I} \end{bmatrix} \quad (\text{A}\cdot 2)$$

Finally, (4) is obtained by connecting (A·1) and (A·2). This completes the proof.



Taizo Suzuki received the B.E., M.E. and Ph.D. degrees in electrical engineering from Keio University, Japan, in 2004, 2006 and 2010, respectively. He joined Toppan Printing Co., Ltd., Tokyo, Japan, from 2006 to 2008. After that, he was a RA of Global COE (C-12) from 2008 to 2011, a Research Fellow of the JSPS from 2010 to 2011, and a Visiting Scholar in the ECE Dept., UCSD, La Jolla, CA from 2010 to 2011, respectively. Then, he was an Assistant Professor in the EEE Dept., College of Eng., Nihon University, Koriyama, Japan from 2011 to 2012. He is currently an Assistant Professor in the Faculty of EIS, University of Tsukuba, Tsukuba, Japan. His current research interests are filter bank design and its application for image/video.



Hirotomo Aso received the B.E., M.E., and Ph.D. degrees in electrical engineering from Tohoku University, Japan, in 1968, 1970, and 1974, respectively. He was with the Department of Information Engineering, Tohoku University in 1973, and later joined the Faculty of Engineering, Nagoya University from 1979 to 1986. He was a professor in the Faculty of Engineering, Tohoku University till 2009. He is now a Professor in College of Engineering, Nihon University. He is presently engaged in research of character pattern recognition, cellular automata, concurrent program schema, parallel processing, and theory of systolic algorithms. He has received the Achievement Award (Gyoseki-sho) of IEICE in 1992. Dr. Aso is a member of IEEE, ACM, IPSJ, and JSAP.