

PAPER

Dual-DCT-Lifting-Based Lapped Transform with Improved Reversible Symmetric Extension

Taizo SUZUKI^{†a)}, *Member* and Masaaki IKEHARA^{††}, *Fellow*

SUMMARY We present a lifting-based lapped transform (L-LT) and a reversible symmetric extension (RSE) in the boundary processing for more effective lossy-to-lossless image coding of data with various qualities from only one piece of lossless compressed data. The proposed dual-DCT-lifting-based LT (D²L-LT) parallel processes two *identical* LTs and consists of 1-D and 2-D DCT-liftings which allow the direct use of a DCT matrix in each lifting coefficient. Since the DCT-lifting can utilize any existing DCT software or hardware, it has great potential for elegant implementations that are dependent on the architecture and DCT algorithm used. In addition, we present an improved RSE (IRSE) that works by recalculating the boundary processing and solves the boundary problem that the DCT-lifting-based L-LT (DL-LT) has. We show that D²L-LT with IRSE mostly outperforms conventional L-LTs in lossy-to-lossless image coding.

key words: discrete cosine transform, lapped transform, lifting structure, lossy-to-lossless image coding, reversible symmetric extension

1. Introduction

Discrete cosine transform (DCT) [1] is classified into several types and they have many fast implementations. Image and video compression (image coding) standards such as the JPEG and H.26x series [2]–[5] use M -channel ($M = 2^k$, $k \in \mathbb{N}$, $k \geq 2$) type-II DCT (DCT-II) with a high energy compaction capability and its inverse transform, type-III DCT (DCT-III). However, the use of DCT-II and III generates unpleasant *blocking artifacts* in images reconstructed from low-bit compressions because they do not consider the continuity between adjacent $M \times M$ blocks.

To solve the blocking artifact problem, many M -channel ($M \times 2M$) lapped transforms (LTs) have been presented. The frequency-domain LT (FDLT) in [6] composed of a DCT-II and a post-filtering on the block boundaries. The modulated LT (MLT) in [6], that is used in audio coding standards such as MP3 and AAC [7], [8], composed of a type-IV DCT (DCT-IV) and window functions as the pre-filtering part. Moreover, the time-domain LT (TDLT) in [9], that overcomes the blocking artifact problem while keeping the traditional DCT-based frameworks, composed of a DCT-II and a pre-filtering on the block boundaries. Hence, these LTs can be simply derived from a DCT with a pre- or post-filtering. Although the M -channel ($M \times KM$, $K \in \mathbb{N}$, $K \geq 2$) generalized lapped orthogonal and biorthogonal transform

(GenLOT and GLBT) without DCT constraints [10], [11] achieve good coding, the complexity reduction by DCT algorithms cannot be contributed unlike the DCT-based LTs.

The lifting structure [12]–[14] has yielded several integer-to-integer transforms that map integer input signals to integer output signals for lossy-to-lossless image coding such that data with various qualities can be obtained from only one piece of lossless compressed data. In particular, several lifting-based DCTs (L-DCTs) have been researched [15]–[18] in part because the DCT is the most-used transform in image coding and L-DCTs enable lossy-to-lossless image coding while preserving compatibility with DCT-based frameworks. Obviously, L-DCTs cause the blocking artifact problem in low-bit compressions.

JPEG XR [19] uses a lifting-based LT (L-LT) [20] to overcome the blocking artifact problem and achieve lossy-to-lossless image coding. The L-LT in JPEG XR performs better than L-DCTs because LT prevents blocking artifacts generated by DCT algorithm in low-bit compressions. Nevertheless, the block size is too small, i.e., $M = 4$, to elicit sufficient resolution in the frequency domain. A simple expansion of the block size degrades the transform performance because of rounding errors in each lifting step. Wang et al. proposed an L-LT in [21] based on triangular elementary reversible matrices (TERMs) [22]. However, it cannot directly use any existing DCT software or hardware for low complexity. We proposed an L-LT in [23] using parallel processing of two *different* LTs and a DCT-lifting [17] which is a lifting structure with a DCT matrix in each lifting coefficient. The DCT-lifting not only elicits high transform performance by merging many rounding operations, but can also utilize any existing DCT software or hardware; as a result, the DCT-lifting-based LT (DL-LT) has great potential for elegant implementations that are dependent on the architecture and DCT algorithm used.

On the other hand, the LT requires a certain signal extension for the boundary processing unlike DCT. In particular, a symmetric extension [24] is usually used to avoid boundary errors and send no extra signals to the decoder. Although it is often simplified as a symmetric non-expansive convolution [25], we will call it symmetric extension. A reversible symmetric extension (RSE) [26] has been proposed as a smooth boundary processing that extends the symmetric extension even further, because the conventional symmetric extensions cannot be directly utilized in the L-LTs due to the rounding errors in each lifting step. However, the DL-LT cannot use any conventional smooth boundary processing,

Manuscript received March 11, 2016.

Manuscript revised November 1, 2016.

[†]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 Electronics and Electrical Engineering, Keio University, Yokohama-shi, 223-8522 Japan.

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

DOI: 10.1587/transfun.E100.A.1109

Table 1 Advantages of each LT: (A) lossy-to-lossless image coding, (B) DCT-lifting, (C) smooth boundary solution.

Transform	(A)	(B)	(C)
LT	N/A	N/A	✓
L-LT	✓	N/A	✓
DL-LT	✓	✓	N/A
D ² L-LT	✓	✓	✓

including the RSE in [26], because of the anomalous structure it uses to improve the transform performance. The L-LT in JPEG XR solves the boundary problem by using a *pseudo* symmetric extension (PSE), whereas the DL-LT has to adopt a periodic extension (PE) that suffers the boundary error.

We present a novel DL-LT and RSE for lossy-to-lossless image coding. The DL-LT has great potential for elegant implementations as described above and the RSE yields an effective boundary solution for the DL-LT. Table 1 shows advantages of each LT. In Sect. 2, we review and define an LT. In Sect. 3, we introduce a novel DL-LT that parallel processes two *identical* LTs, unlike the DL-LT presented in [23][†]. In Sect. 4, this novel DL-LT is then improved by using 2-D blockwise implementations derived from a separable 2-D transform [9]. The conventional DL-LT can only use a 1-D DCT-lifting, whereas the proposed DL-LT can use both 1-D and 2-D DCT-liftings based on block-liftings in [28] and [29], and hence, it is called a dual-DL-LT (D²L-LT). The 2-D DCT-lifting operates more effectively than the 1-D DCT-lifting by making productive use of the 2-D space of an image. In Sect. 5, we present an improved RSE (IRSE) by recalculating the boundary processing and solve the boundary problem. It is achieved thanks to parallel processing of two identical LTs. In Sect. 6, we show that D²L-LT with IRSE mostly outperforms the conventional L-LTs at lossy-to-lossless image coding. Sect. 7 concludes this paper.

Notation: \mathbf{I}_α , \mathbf{J}_α , \mathbf{D}_α , $\mathcal{D}\{\dots\}$, \otimes , and \cdot^T denote an $\alpha \times \alpha$ identity matrix, an $\alpha \times \alpha$ reversal identity matrix, an $\alpha \times \alpha$ diagonal matrix with $(-1)^\beta$ in the (β, β) -element, a (block) diagonal matrix, a Kronecker product, and transpose of a matrix, respectively ($\alpha, \beta \in \mathbb{N}$, $\alpha \geq 2$, $0 \leq \beta \leq \alpha - 1$). Let N be $N = M/2$. Superscripts x , y , and xy in the matrices mean to operate horizontally, vertically, and horizontally and vertically, respectively. For example,

$$\mathbf{A}^x \mathbf{x} = \mathbf{x} \mathbf{A}^T \quad (1)$$

$$\mathbf{A}^y \mathbf{x} = \mathbf{A} \mathbf{x} \quad (2)$$

$$\mathbf{A}^{xy} \mathbf{x} = \mathbf{A}^x \mathbf{A}^y \mathbf{x} = \mathbf{A} \mathbf{x} \mathbf{A}^T, \quad (3)$$

where \mathbf{A} and \mathbf{x} are a matrix and a 2-D input signal block, respectively. Superscript w means a 2-D transform of the matrix, which is expressed by a Kronecker product, $\mathbf{A}^w = \mathbf{A}^x \otimes \mathbf{A}^y$. Table 2 shows summary of the special abbreviations in this study.

[†]Although we have proposed a similar DL-LT that parallel processes two identical LTs in [27] unlike the new DL-LT in this study, the RSE in this study is inapplicable to the DL-LT in [27].

Table 2 Summary of special abbreviations in this study.

LT	Lapped Transform
L-LT	Lifting-based LT
DL-LT	DCT-Lifting-based LT
D ² L-LT	Dual-DCT-Lifting-based LT
PSE	Pseudo Symmetric Extension
PE	Periodic Extension
RSE	Reversible Symmetric Extension
IRSE	Improved RSE

2. Lapped Transform

The $M \times 2M$ LT used in this study is based on the lattice structure of TDLT [9], a fast-computable structure. The structure is shown in Fig. 1, where \mathbf{x} and \mathbf{y} are an input signal vector and its output signal vector, respectively, and its polyphase matrix is expressed as^{††}

$$\mathbf{E}(z) = \underbrace{\mathbf{P}_1 \begin{bmatrix} \mathbf{C}_{II} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{IV} \end{bmatrix} \mathbf{P}_0 \widetilde{\mathbf{W}} \Lambda(z)}_{\text{postprocessing}} \cdot \underbrace{\widetilde{\mathbf{W}} \mathbf{P}_0 \begin{bmatrix} \mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{III} \mathbf{C}_{IV} \mathcal{V} \end{bmatrix} \mathbf{P}_0 \widetilde{\mathbf{W}}}_{\text{preprocessing}}, \quad (4)$$

where

$$\mathbf{P}_0 = \begin{bmatrix} \mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_N \end{bmatrix} \quad (5)$$

$$\Lambda(z) = \begin{bmatrix} \mathbf{0} & \mathbf{I}_N \\ z^{-1} \mathbf{I}_N & \mathbf{0} \end{bmatrix} \quad (6)$$

$$\widetilde{\mathbf{W}} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I}_N & \mathbf{J}_N \\ \mathbf{J}_N & -\mathbf{I}_N \end{bmatrix}, \quad (7)$$

z^{-1} is a delay element, the (m, n) -element of an $M \times M$ permutation matrix \mathbf{P}_1 is

$$[\mathbf{P}_1]_{m,n} = \begin{cases} 1 & (n \leq N - 1 \text{ and } m = 2n) \\ 1 & (n \geq N \text{ and } m = 2n - M + 1), \\ 0 & (\text{otherwise}) \end{cases}, \quad (8)$$

and \mathcal{V} is an arbitrary $N \times N$ nonsingular matrix to improve the coding performance, respectively. If $\mathcal{V} = \mathbf{I}_N$, it is called a lapped orthogonal transform (LOT). \mathbf{C}_{II} , \mathbf{C}_{III} , and \mathbf{C}_{IV} are $N \times N$ DCT-II, DCT-III, and DCT-IV matrices, and the (m, n) -elements of \mathbf{C}_{II} and \mathbf{C}_{IV} are defined as

$$[\mathbf{C}_{II}]_{m,n} = \sqrt{\frac{2}{N}} c_m \cos\left(\frac{m(n+1/2)\pi}{N}\right) \quad (9)$$

$$[\mathbf{C}_{IV}]_{m,n} = \sqrt{\frac{2}{N}} \cos\left(\frac{(m+1/2)(n+1/2)\pi}{N}\right), \quad (10)$$

where $c_m = 1/\sqrt{2}$ ($m = 0$) or 1 ($m \neq 0$). Moreover, $\mathbf{C}_{IV}^{-1} =$

^{††} $\mathbf{C}_{III} \mathbf{S} \mathbf{C}_{IV}$ in [9] is replaced by $\mathbf{C}_{III} \mathbf{C}_{IV} \mathcal{V}$. In the simplest form in [9], \mathbf{S} is a scale factor.

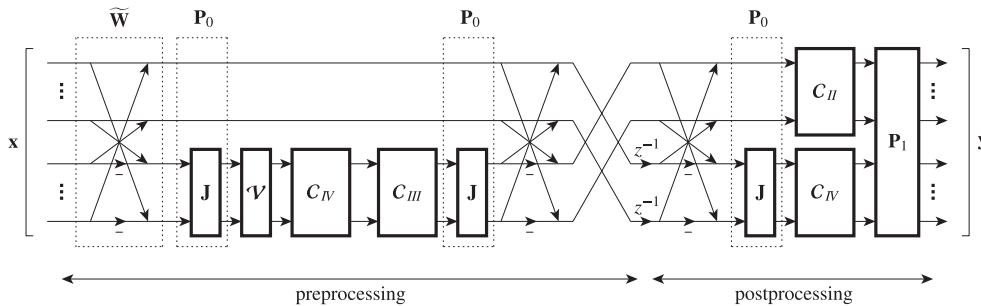


Fig. 1 Lattice structure of LT.

$C_{IV}^T = C_{IV}$, $C_{III} = C_{II}^{-1} = C_{II}^T$, and the postprocessing in Eq. (4) is an $M \times M$ DCT-II.

3. DCT-Lifting-Based Lapped Transform that Parallel Processes Two Identical Lapped Transforms

This section presents a DL-LT that parallel processes two identical LTs. This DL-LT is unlike the DL-LT in [23] that parallel processes two different LTs.

3.1 1-D DCT-Lifting of C_{II} and C_{III}

This subsection presents the 1-D DCT-lifting [17], which is a lifting structure with a DCT matrix in each lifting coefficient, of C_{II} and C_{III} in the proposed LT. Eq. (4) is represented by

$$\mathbf{E}(z) = \underbrace{\mathbf{P}_1 \mathbf{D}_M \begin{bmatrix} C_{II} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{IV} \end{bmatrix} \mathbf{W} \Lambda(z)}_{\text{postprocessing}} \cdot \underbrace{\mathbf{W} \begin{bmatrix} \mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & C_{III} C_{IV} \mathcal{V} \end{bmatrix} \mathbf{W} \mathbf{P}_0}_{\text{preprocessing}}, \quad (11)$$

where

$$\mathbf{W} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I}_N & \mathbf{I}_N \\ \mathbf{I}_N & -\mathbf{I}_N \end{bmatrix}, \quad (12)$$

$C_{II} = \mathbf{D}_N C_{II} \mathbf{J}_N$, and type-IV DST (DST-IV) $\mathbf{S}_{IV} = \mathbf{D}_N C_{IV} \mathbf{J}_N$. We can obtain the following 1-D DCT-lifting from Eq. (11).

$$\begin{aligned} & \begin{bmatrix} C_{II} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_N \end{bmatrix} \mathbf{W} \Lambda(z) \mathbf{W} \begin{bmatrix} \mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & C_{III} \end{bmatrix} \\ &= - \begin{bmatrix} \mathbf{I}_N & -C_{II} \\ \mathbf{0} & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \mathbf{I}_N & \mathbf{0} \\ \frac{1}{2} C_{III} & \mathbf{I}_N \end{bmatrix} \Lambda(z) \\ & \cdot \begin{bmatrix} \mathbf{I}_N & \frac{1}{2} C_{III} \\ \mathbf{0} & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \mathbf{I}_N & \mathbf{0} \\ -C_{II} & \mathbf{I}_N \end{bmatrix} \end{aligned} \quad (13)$$

The sign inversions such as \mathbf{D}_M in Eq. (11) and $-$ in Eq. (13) can be removed because they have no impact on the transfer function. Consequently, Eq. (4) can be represented by

$$\begin{aligned} \mathbf{E}(z) &= \mathbf{P}_1 \underbrace{\begin{bmatrix} \mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{IV} \end{bmatrix} \begin{bmatrix} \mathbf{I}_N & -C_{II} \\ \mathbf{0} & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \mathbf{I}_N & \mathbf{0} \\ \frac{1}{2} C_{III} & \mathbf{I}_N \end{bmatrix} \Lambda(z)}_{\text{postprocessing}} \\ & \cdot \underbrace{\begin{bmatrix} \mathbf{I}_N & \frac{1}{2} C_{III} \\ \mathbf{0} & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \mathbf{I}_N & \mathbf{0} \\ -C_{II} & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & C_{IV} \mathcal{V} \end{bmatrix} \mathbf{W} \mathbf{P}_0}_{\text{preprocessing}}. \end{aligned} \quad (14)$$

Most of this structure includes very simple operations with DCT matrices, adders, bit-shifters, and permutations, but it is not a complete lifting structure.

3.2 1-D DCT-Lifting of C_{IV} or \mathbf{S}_{IV}

This subsection describes the C_{IV} and \mathbf{S}_{IV} parts in Eq. (14). We consider two identical LTs as shown in Fig. 2, where \mathbf{x} , $\hat{\mathbf{x}}$, \mathbf{y} , and $\hat{\mathbf{y}}$ are an input signal vector, an input signal vector that is not \mathbf{x} , the output signal vector of \mathbf{x} , and the output signal vector of $\hat{\mathbf{x}}$, respectively. Consider a 1-D simultaneous implementation of $\mathcal{D}\{\mathbf{I}_N, \mathbf{T}\}$ on different horizontal or vertical lines, where \mathbf{T} is an $N \times N$ symmetric orthogonal matrix such that $\mathbf{T}^{-1} = \mathbf{T}^T = \mathbf{T}$:

$$\begin{bmatrix} \mathbf{y} \\ \hat{\mathbf{y}} \end{bmatrix} = \begin{bmatrix} \mathcal{D}\{\mathbf{I}_N, \mathbf{T}\} & \mathbf{0} \\ \mathbf{0} & \mathcal{D}\{\mathbf{I}_N, \mathbf{T}\} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \hat{\mathbf{x}} \end{bmatrix}, \quad (15)$$

where $\mathcal{D}\{\mathbf{T}, \mathbf{T}\}$ is factorized into block-liftings as [23]

$$\begin{bmatrix} \mathbf{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{T} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_N \\ -\mathbf{I}_N & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I}_N & \mathbf{0} \\ \mathbf{T} & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \mathbf{I}_N & -\mathbf{T} \\ \mathbf{0} & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \mathbf{I}_N & \mathbf{0} \\ \mathbf{T} & \mathbf{I}_N \end{bmatrix}. \quad (16)$$

Since C_{IV} and \mathbf{S}_{IV} are $N \times N$ symmetric orthogonal matrices, the block-lifting in Eq. (16) can be applied to each combination of C_{IV} and \mathbf{S}_{IV} framed by dash-dotted lines in Fig. 2.

3.3 1-D Block-Lifting of \mathbf{W}

The \mathbf{W} in Eq. (14) can be easily factorized into block-lifting structures as

$$\mathbf{W} = \begin{bmatrix} \mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_N \end{bmatrix} \begin{bmatrix} \mathbf{I}_N & \mathbf{0} \\ (1 - \sqrt{2})\mathbf{I}_N & \mathbf{I}_N \end{bmatrix}$$

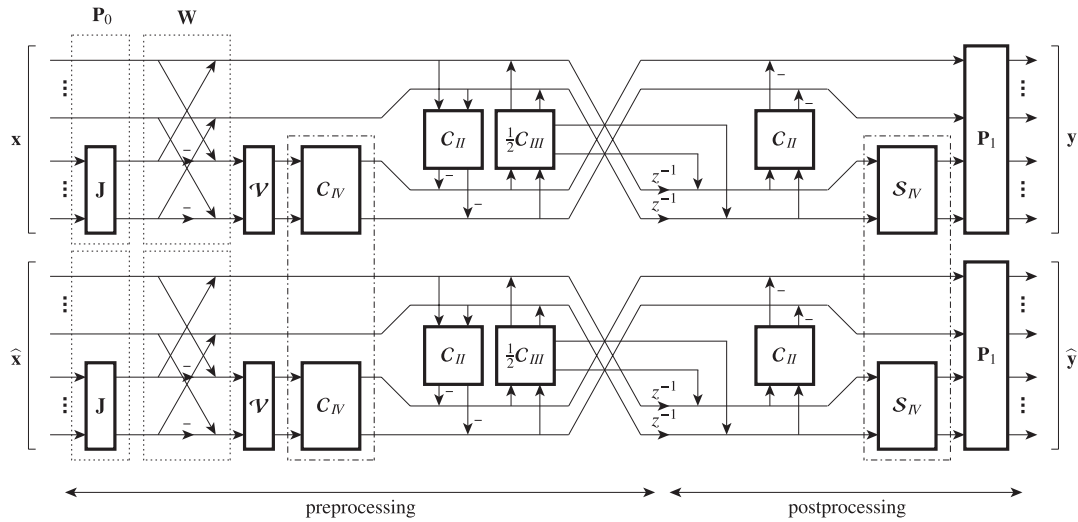


Fig. 2 Represented structure of LT by parallel processing of two identical LTs.

$$\begin{bmatrix} \mathbf{I}_N & \frac{1}{\sqrt{2}}\mathbf{I}_N \\ \mathbf{0} & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \mathbf{I}_N & \mathbf{0} \\ (1-\sqrt{2})\mathbf{I}_N & \mathbf{I}_N \end{bmatrix}. \quad (17)$$

However, the floating-point multipliers $(1 - \sqrt{2})$ and $1/\sqrt{2}$ cause the structure to have high complexity. The problem will be solved by using a 2-D blockwise implementation, as described in Sect. 4.4.

3.4 1-D Implementation of \mathbf{P}_0 and \mathcal{V}

The permutation matrix \mathbf{P}_0 in Eq. (14) is already reversible because it only permutes signals. Also, we can easily define \mathcal{V} in Eq. (14) as an upper triangle matrix with 1s in all diagonal coefficients,

$$\mathcal{V} = \begin{bmatrix} 1 & \times & \cdots & \times \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \times \\ 0 & \cdots & 0 & 1 \end{bmatrix}, \quad (18)$$

where \times indicates any real (floating-point) value determined to improve the coding performance. It is clear that the \mathcal{V} already has the complete lifting structure. LT with an upper triangular matrix \mathcal{V} shows better coding gain than LOT with $\mathcal{V} = \mathbf{I}$, e.g., the coding gains of 16×32 LOT and LT are 9.76 dB and 9.89 dB, respectively. For low complexity, the floating-point lifting coefficients are approximated by rational (dyadic) coefficients $\gamma/2^\delta$ ($\gamma, \delta \in \mathbb{N}$). We allocated a 6-bit word length ($\delta = 6$) to each coefficient in this study. The coding gain of the approximated LT is almost equivalent to that of the original LT.

4. Dual-DCT-Lifting-Based Lapped Transform

This section describes a 2-D blockwise implementation of a separable 2-D transform [9] of the 1-D LT presented above.

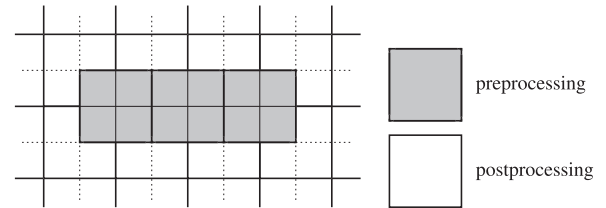


Fig. 3 2-D blockwise implementation of LT.

4.1 Separable 2-D Transform

When an $M \times M$ image \mathbf{X} is 2-D transformed by an $M \times M$ transform $\mathbf{T} = \prod_{k=K}^0 \mathbf{T}_k$ ($K \in \mathbb{N}$), the transformed image \mathbf{Y} can be expressed as

$$\mathbf{Y} = \mathbf{T}_K \cdots \mathbf{T}_1 \mathbf{T}_0 \mathbf{X} \mathbf{T}_0^T \mathbf{T}_1^T \cdots \mathbf{T}_K^T = \mathbf{T}_K^{xy} \cdots \mathbf{T}_1^{xy} \mathbf{T}_0^{xy} \mathbf{X}. \quad (19)$$

It means that \mathbf{X} is 2-D transformed with \mathbf{T}_{k+1} after the 2-D transform by \mathbf{T}_k . Using this fact, the D^2L -LT can be implemented by cascading the 2-D transform of the pre/postprocessing blocks in Fig. 2. Note that the 2-D blockwise implementation of LT is overlapped via the preprocessing and postprocessing as shown in Fig. 3. DL-LT in [23] cannot use a separable 2-D transform due to its parallel processing of two different LTs, whereas D^2L -LT can do so, thanks to its use of two identical LTs. Consequently, it is easy to make a 2-D blockwise implementation in each $M \times M$ block as in [20].

4.2 2-D DCT-Lifting of C_{II} and C_{III}

We proposed a 2-D block-lifting in [29]. Let \mathcal{B}_k be a set of lower and upper DCT-lifting matrices in Eq. (14), i.e.,

$$\mathcal{B}_k = \begin{bmatrix} \mathbf{I}_N & \mathcal{U}_k \\ \mathbf{0} & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \mathbf{I}_N & \mathbf{0} \\ \mathcal{L}_k & \mathbf{I}_N \end{bmatrix} \quad \text{for } k = 0 \text{ or } 1, \quad (20)$$

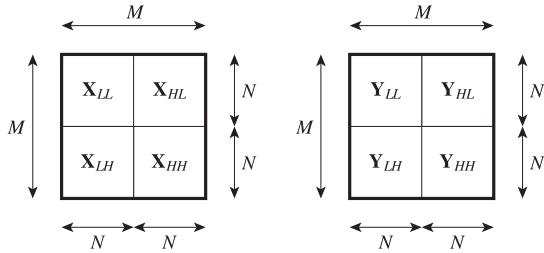


Fig. 4 $M \times M$ input and output blocks.

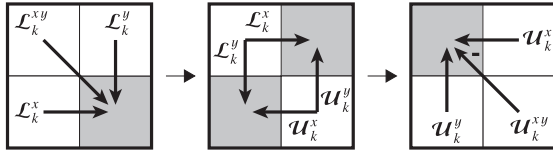


Fig. 5 2-D block-lifting derived from a combination of lower and upper block-lifting matrices (gray blocks mean changed blocks and rounding operations are omitted).

where $\mathbf{U}_0 = \mathcal{L}_1 = -\mathbf{C}_{II}$ and $\mathbf{U}_1 = \mathcal{L}_0 = \frac{1}{2}\mathbf{C}_{III}$. For the $M \times M$ image in Fig. 4, a 2-D DCT-lifting of \mathcal{B}_k can be implemented by (see Fig. 5)

$$\begin{bmatrix} \mathbf{Y}_{LL} \\ \mathbf{Y}_{HL} \\ \mathbf{Y}_{LH} \\ \mathbf{Y}_{HH} \end{bmatrix} = \mathcal{B}_k^w \begin{bmatrix} \mathbf{X}_{LL} \\ \mathbf{X}_{HL} \\ \mathbf{X}_{LH} \\ \mathbf{X}_{HH} \end{bmatrix}, \quad (21)$$

where

$$\mathcal{B}_k^w = \begin{bmatrix} \mathbf{I}_N & \mathbf{U}_k^x & \mathbf{U}_k^y & -\mathbf{U}_k^{xy} \\ \mathbf{0} & \mathbf{I}_N & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_N \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_N & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathcal{L}_k^x & \mathbf{I}_N & \mathbf{0} & \mathbf{U}_k^y \\ \mathcal{L}_k^y & \mathbf{0} & \mathbf{I}_N & \mathbf{U}_k^x \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_N \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_N & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_N & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_N & \mathbf{0} \\ \mathcal{L}_k^{xy} & \mathcal{L}_k^y & \mathcal{L}_k^x & \mathbf{I}_N \end{bmatrix}, \quad (22)$$

\mathbf{X}_{LL} , \mathbf{X}_{HL} , \mathbf{X}_{LH} , and \mathbf{X}_{HH} are the top-left, top-right, bottom-left, and bottom-right $N \times N$ blocks of an $M \times M$ image, and \mathbf{Y}_{LL} , \mathbf{Y}_{HL} , \mathbf{Y}_{LH} , and \mathbf{Y}_{HH} are their respective output blocks. The 2-D DCT-lifting reduces rounding errors compared with the 1-D DCT-lifting.

4.3 2-D Blockwise Implementation of 1-D DCT-Lifting of \mathbf{C}_{IV} or \mathcal{S}_{IV}

This subsection describes a 2-D blockwise implementation of the \mathbf{C}_{IV} or \mathcal{S}_{IV} part:

$$\begin{bmatrix} \mathbf{Y}_{LL} \\ \mathbf{Y}_{HL} \\ \mathbf{Y}_{LH} \\ \mathbf{Y}_{HH} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_N & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^x & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{T}^y & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{T}^{xy} \end{bmatrix} \begin{bmatrix} \mathbf{X}_{LL} \\ \mathbf{X}_{HL} \\ \mathbf{X}_{LH} \\ \mathbf{X}_{HH} \end{bmatrix}. \quad (23)$$

It can be simplified as

$$\begin{bmatrix} \mathbf{Y}_{HL} \\ \mathbf{Y}_{LH} \\ \mathbf{Y}_{HH} \end{bmatrix} = \begin{bmatrix} \mathbf{T}^x & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{T}^x \end{bmatrix} \begin{bmatrix} \mathbf{I}_N & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}^y & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{T}^y \end{bmatrix} \begin{bmatrix} \mathbf{X}_{HL} \\ \mathbf{X}_{LH} \\ \mathbf{X}_{HH} \end{bmatrix}, \quad (24)$$

where $\mathbf{Y}_{LL} = \mathbf{X}_{LL}$. Here, \mathbf{X}_{HL} and \mathbf{X}_{HH} are horizontally implemented after the vertical implementations of \mathbf{X}_{LH} and \mathbf{X}_{HH} . $\mathcal{D}\{\mathbf{T}^x, \mathbf{T}^x\}$ and $\mathcal{D}\{\mathbf{T}^y, \mathbf{T}^y\}$ in Eq. (24) can be factorized into 1-D DCT-liftings as in Eq. (16). Unfortunately, unlike the DCT-lifting of \mathbf{C}_{II} and \mathbf{C}_{III} , the 1-D DCT-liftings of \mathbf{C}_{IV} and \mathcal{S}_{IV} cannot be replaced by 2-D DCT-liftings. The 2-D blockwise implementations of the 1-D DCT-liftings of \mathbf{C}_{IV} and \mathcal{S}_{IV} are shown in Fig. 6. We must take care of the implementation directions, i.e., horizontally or vertically. This structure also has simple operations like those of the structure in Sect. 4.2.

4.4 2-D Block-Lifting of \mathbf{W}

The \mathbf{W} in Eq. (14) can be implemented by extending the non-separable 2-D transform of the Hadamard transform that is really used in JPEG XR [30][†] as follows:

$$\begin{bmatrix} \mathbf{Y}_{LL} \\ \mathbf{Y}_{HL} \\ \mathbf{Y}_{LH} \\ \mathbf{Y}_{HH} \end{bmatrix} = \mathbf{W}^w \begin{bmatrix} \mathbf{X}_{LL} \\ \mathbf{X}_{HL} \\ \mathbf{X}_{LH} \\ \mathbf{X}_{HH} \end{bmatrix}, \quad (25)$$

where

$$\mathbf{W}^w = \mathbf{W}^x \otimes \mathbf{W}^y = \frac{1}{2} \begin{bmatrix} \mathbf{I}_N & \mathbf{I}_N & \mathbf{I}_N & \mathbf{I}_N \\ \mathbf{I}_N & -\mathbf{I}_N & \mathbf{I}_N & -\mathbf{I}_N \\ \mathbf{I}_N & \mathbf{I}_N & -\mathbf{I}_N & -\mathbf{I}_N \\ \mathbf{I}_N & -\mathbf{I}_N & -\mathbf{I}_N & \mathbf{I}_N \end{bmatrix}. \quad (26)$$

The \mathbf{W}^w is factorized into lifting structures (see Fig. 7),

$$\mathbf{W}^w = \begin{bmatrix} \mathbf{I}_N & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_N & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \mathbf{I}_N & \mathbf{0} & \mathbf{0} & -\mathbf{I}_N \\ \mathbf{0} & \mathbf{I}_N & \mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_N \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_N & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_N & \mathbf{0} & \mathbf{0} \\ \frac{1}{2}\mathbf{I}_N & -\frac{1}{2}\mathbf{I}_N & \mathbf{I}_N & \mathbf{0} \\ \frac{1}{2}\mathbf{I}_N & -\frac{1}{2}\mathbf{I}_N & \mathbf{0} & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \mathbf{I}_N & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_N & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I}_N \\ \mathbf{0} & \mathbf{0} & -\mathbf{I}_N & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_N & \mathbf{0} & \mathbf{0} & \mathbf{I}_N \\ \mathbf{0} & \mathbf{I}_N & -\mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_N \end{bmatrix}. \quad (27)$$

[†]Note that this simple transform is different from the Hadamard transform described in [20].

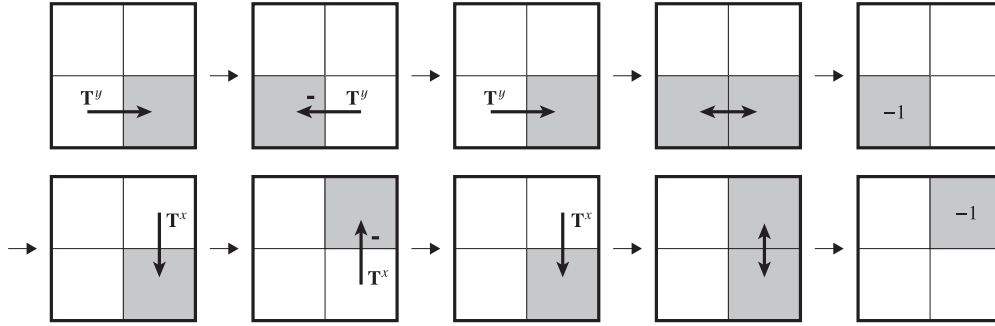


Fig. 6 2-D blockwise implementation of 1-D block-lifting of T^x and T^y (gray blocks mean changed blocks and rounding operations are omitted).

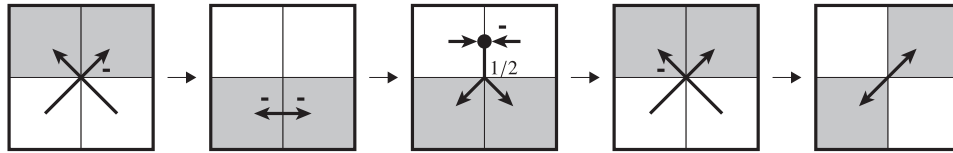


Fig. 7 2-D block-lifting of W (gray blocks mean changed blocks and rounding operations are omitted).

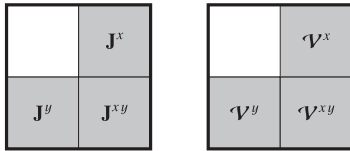


Fig. 8 2-D blockwise implementation of P_0 and V (gray blocks mean changed blocks).

This structure also has very simple operations with only adders, bit-shifters, and permutations.

4.5 2-D Blockwise Implementation of P_0 and V

A 2-D blockwise implementation P_0^w of an $M \times M$ block is expressed as

$$\begin{bmatrix} \mathbf{Y}_{LL} \\ \mathbf{Y}_{HL} \\ \mathbf{Y}_{LH} \\ \mathbf{Y}_{HH} \end{bmatrix} = \mathbf{P}_0^w \begin{bmatrix} \mathbf{X}_{LL} \\ \mathbf{X}_{HL} \\ \mathbf{X}_{LH} \\ \mathbf{X}_{HH} \end{bmatrix}, \quad (28)$$

where $\mathbf{P}_0^w = \mathbf{P}_0^x \otimes \mathbf{P}_0^y = \mathcal{D}\{\mathbf{I}_N, \mathbf{J}_N^x, \mathbf{J}_N^y, \mathbf{J}_N^{xy}\}$, as shown at the left side of Fig. 8. Similarly, a 2-D blockwise implementation of the V part is shown at the right side of Fig. 8. The V already has the complete lifting structure as defined in Eq. (18).

5. Improved Reversible Symmetric Extension

Here, we devise an IRSE for D²L-LT by recalculating the boundary processing derived from the RSE we proposed in [26].

5.1 Boundary Processing of 1-D Signals

Consider a boundary processing of 1-D signals in Eq. (14)

as shown at the left of Fig. 9. If the symmetric input signals, $[(\mathbf{J}\mathbf{x})^T, \mathbf{x}^T]^T$ are input to the boundaries and rounding operations are excluded, the processing can be expressed as

$$\mathfrak{P} \begin{bmatrix} \mathbf{J}\mathbf{x} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}}\mathbf{J}\mathbf{x} \\ -\sqrt{2}\mathbf{C}_{II}\mathbf{J}\mathbf{x} \end{bmatrix}, \quad (29)$$

where \mathfrak{P} is the preprocessing part in Eq. (14),

$$\mathfrak{P} = \begin{bmatrix} \mathbf{I}_N & \frac{1}{2}\mathbf{C}_{III} \\ \mathbf{0} & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \mathbf{I}_N & \mathbf{0} \\ -\mathbf{C}_{II} & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{IV}\mathbf{V} \end{bmatrix} \mathbf{W}\mathbf{P}_0. \quad (30)$$

The left of Fig. 9 with the symmetric extension is redrawn at the right of Fig. 9. The DCT-II matrix \mathbf{C}_{II} can be easily factorized into lifting structures by using a lifting factorization due to $\det(\mathbf{C}_{II}) = 1$. In this study, we used the L-DCT in [18] to factorize \mathbf{C}_{II} in Eq. (29) into lifting structures. On the other hand, although a combination of $\sqrt{2}$ and $1/\sqrt{2}$ can be easily factorized into a lifting factorization, we will not use the lifting structures with the floating-point lifting coefficients. The floating-point scaling coefficients $\sqrt{2}$ and $1/\sqrt{2}$ are instead with dyadic coefficients by considering a 2-D implementation, as described in the next subsection.

5.2 Boundary Processing of 2-D Signals

If a boundary processing of 2-D signals (image) is considered with symmetric extended input signals, a special processing is implemented in the gray area in Fig. 10. The boundary processing of $\mathbf{X}_{\infty s}$, whose size is $N \times N$, in the figure are expressed using several separable 2-D transforms:

$$\mathbf{X}_{tl} : 2\mathbf{C}_{II}\mathbf{J}_N\mathbf{X}_{tl}\mathbf{J}_N\mathbf{C}_{II}^T \quad (31)$$

$$\mathbf{X}_{br} : \frac{1}{2}\mathbf{X}_{br} \quad (32)$$

$$\mathbf{X}_{tr} : -\mathbf{C}_{II}\mathbf{J}_N\mathbf{X}_{tr} \quad (33)$$

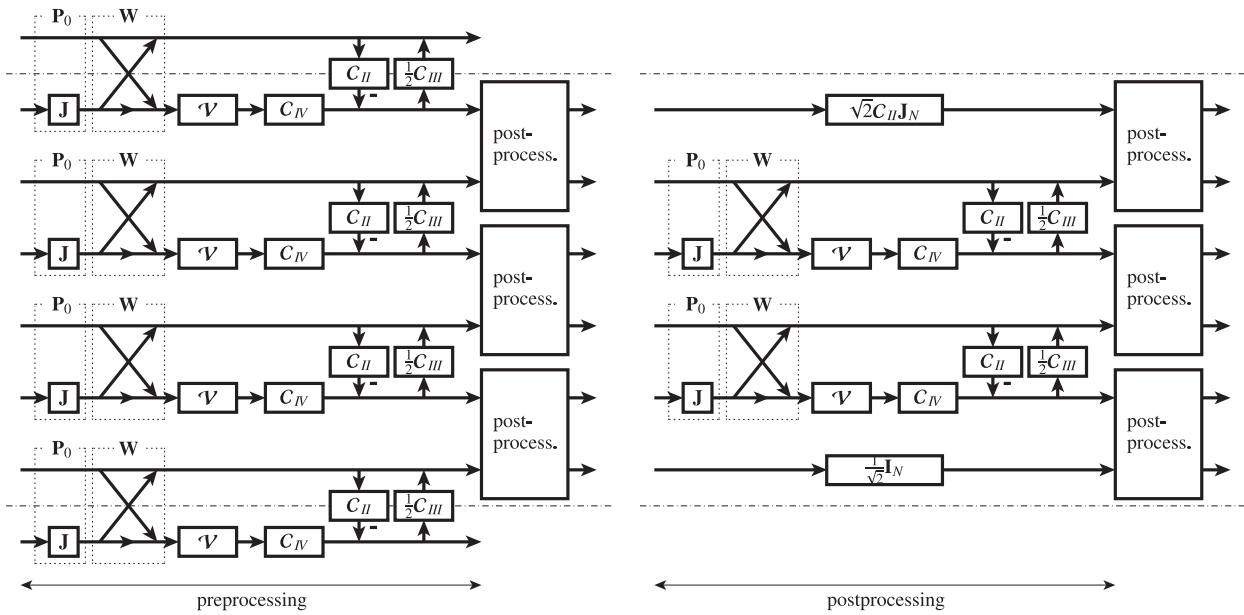


Fig. 9 Boundary processing in 1-D signals (dash-dotted lines mean signal boundaries): (left) appearance of signal extension; (right) non-expansive convolution derived from symmetric extension.

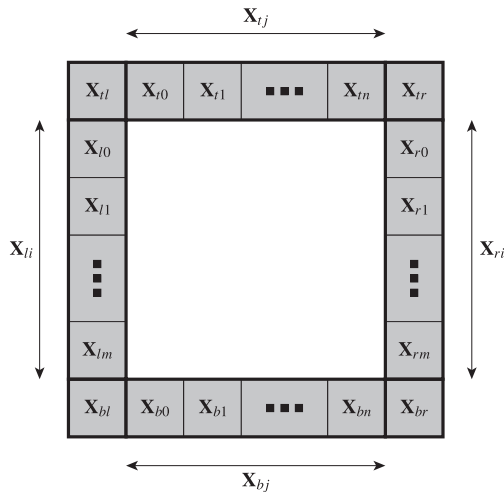


Fig. 10 Boundary processing in 2-D signals (gray area means boundary processing area).

$$\mathbf{X}_{bl} : -\mathbf{X}_{bl} \mathbf{J}_N \mathbf{C}_{II}^T \quad (34)$$

$$\mathbf{X}_{tj} : \sqrt{2} \mathbf{C}_{II} \mathbf{J}_N \mathbf{X}_{tj} \mathfrak{P}^T \quad (35)$$

$$\mathbf{X}_{bj} : \frac{1}{\sqrt{2}} \mathbf{X}_{bj} \mathfrak{P}^T \quad (36)$$

$$\mathbf{X}_{li} : \sqrt{2} \mathfrak{P} \mathbf{X}_{li} \mathbf{C}_{II}^T \quad (37)$$

$$\mathbf{X}_{ri} : \frac{1}{\sqrt{2}} \mathfrak{P} \mathbf{X}_{ri} \quad (38)$$

for $i = 0, 1, \dots, m$ and $j = 0, 1, \dots, n$. The scaling coefficients $\sqrt{2}$ and $1/\sqrt{2}$ appearing in Eqs. (35)–(38) are merged into \mathbf{W} in each \mathfrak{P} in order to eliminate the floating-point coefficients:

$$\sqrt{2} \mathbf{W} = \begin{bmatrix} \mathbf{I}_N & \mathbf{I}_N \\ \mathbf{I}_N & -\mathbf{I}_N \end{bmatrix} \triangleq \widehat{\mathbf{W}} \quad (39)$$

$$\frac{1}{\sqrt{2}} \mathbf{W} = \frac{1}{2} \widehat{\mathbf{W}}. \quad (40)$$

Since $\widehat{\mathbf{W}}^{-1} = \frac{1}{2} \widehat{\mathbf{W}}$, each combination of $\widehat{\mathbf{W}}$ and $\frac{1}{2} \widehat{\mathbf{W}}$ in Eqs. (35)–(38) can be factorized into block-liftings without multipliers:

$$\begin{bmatrix} \widehat{\mathbf{W}} & \mathbf{0} \\ \mathbf{0} & \frac{1}{2} \widehat{\mathbf{W}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_N \\ -\mathbf{I}_N & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I}_N & \mathbf{0} \\ \widehat{\mathbf{W}} & \mathbf{I}_N \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_N & -\frac{1}{2} \widehat{\mathbf{W}} \\ \mathbf{0} & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} \mathbf{I}_N & \mathbf{0} \\ \widehat{\mathbf{W}} & \mathbf{I}_N \end{bmatrix}. \quad (41)$$

Each residual DCT part of \mathfrak{P} is directly implemented by 1-D DCT-liftings. Furthermore, the combinations of 2 and $1/2$ in Eqs. (31) and (32) can be simply factorized into lifting structures without multipliers:

$$\begin{bmatrix} 2 & 0 \\ 0 & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & -\frac{1}{2} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}. \quad (42)$$

6. Experimental Results

We designed 16×32 D²L-LT and compared it with conventional methods, such as the L-LT in JPEG XR [20] and DL-LT [23] with the same transfer function as the D²L-LT, by using the following lossless bitrate (LBR) [bpp] and peak signal-to-noise ratio (PSNR) [dB] in lossy-to-lossless image coding:

$$\text{LBR [bpp]} = \frac{\text{Total number of bits [bit]}}{\text{Total number of pixels [pixel]}} \quad (43)$$

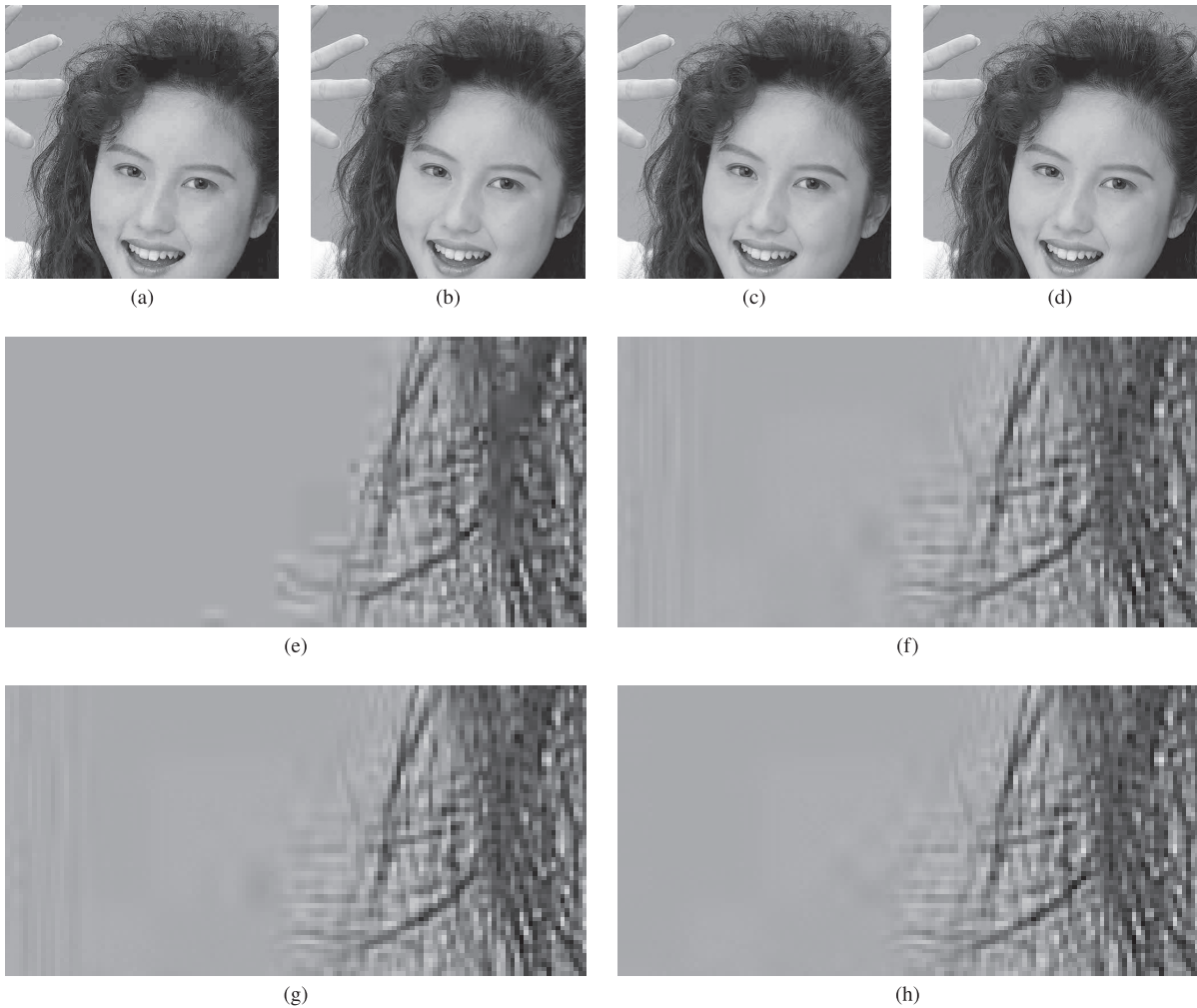


Fig. 11 Comparison of whole and particular areas of a reconstructed image *Woman* for a bitrate of 0.25 bpp: (a-d) whole areas, (e-h) particular areas which left sides indicate image boundaries, (a,e) 4×8 L-LT in [20] with PSE, (b,f) 16×32 DL-LT in [23] with PE, (c,g) 16×32 D^2 L-LT with PE, (d,h) 16×32 D^2 L-LT with IRSE.

$$\text{PSNR [dB]} = 10 \log_{10} \left(\frac{255^2}{\text{MSE}} \right), \quad (44)$$

where MSE is the mean squared error. To evaluate the transform performance fairly, we employed 2-level decomposition for the L-LT in JPEG XR, where only core transform (L-DCT) was used in the second stage in accordance with the specification, whereas we employed 1-level decompositions for 16×32 DL-LT and D^2 L-LT. We used PSE, PE, and IRSE for the boundary processing, eight 512×512 8-bit grayscale images such as *Barbara* in [31], and six 1024×1024 8-bit clipped grayscale images such as *Bike* in [32] and *Bridge* in [33]. A very common wavelet-based coder SPIHT [34] was used to encode the transformed images after the transformed coefficients were ordered to a tree of wavelet coefficients as in [35].

Tables 3 and 4 respectively show lossless and lossy image coding results. Although the D^2 L-LT with PE did not show the obvious advantage compared with the DL-LT in [23] and the L-LT in [20] showed the best perfor-

Table 3 Lossless image coding results (LBR [bpp]).

Test Images	L-LT [20]	DL-LT [23]	D^2 L-LT
	PSE	PE	IRSE (PE)
<i>Baboon</i>	6.24	6.22	6.19 (6.21)
<i>Barbara</i>	4.95	4.80	4.74 (4.77)
<i>Boat</i>	5.22	5.12	5.09 (5.10)
<i>Elaine</i>	5.28	5.17	5.13 (5.16)
<i>Finger</i>	5.90	5.75	5.74 (5.74)
<i>Goldhill</i>	5.10	5.13	5.08 (5.12)
<i>Lena</i>	4.62	4.64	4.57 (4.61)
<i>Pepper</i>	5.00	4.95	4.92 (4.93)
<i>Bike</i>	5.10	5.16	5.12 (5.14)
<i>Cafe</i>	6.00	6.11	6.08 (6.10)
<i>Woman</i>	4.79	4.86	4.81 (4.83)
<i>Bridge</i>	4.31	4.34	4.28 (4.29)
<i>Cathedral</i>	4.18	4.26	4.19 (4.21)
<i>Deer</i>	5.16	5.02	5.01 (5.01)

mance in some lossless image coding results, the D^2 L-LT with IRSE mostly outperformed other L-LTs. Moreover, the perceptual visual quality at the boundaries was clearly im-

Table 4 Lossy image coding results (PSNR [dB]).

Test Image	Bitrate [bpp]	L-LT [20] PSE	DL-LT [23] PE	D ² L-LT IRSE (PE)
<i>Baboon</i>	0.25	22.42	22.95	23.02 (22.96)
	0.50	24.67	25.21	25.30 (25.22)
	1.00	28.12	28.52	28.65 (28.53)
<i>Barbara</i>	0.25	26.96	28.96	29.15 (28.95)
	0.50	30.92	32.96	33.21 (32.96)
	1.00	36.13	37.49	37.72 (37.52)
<i>Boat</i>	0.25	28.84	29.29	29.38 (29.29)
	0.50	32.05	32.38	32.49 (32.38)
	1.00	35.26	35.63	35.74 (35.66)
<i>Elaine</i>	0.25	31.52	31.73	32.02 (31.72)
	0.50	32.31	33.06	33.23 (33.07)
	1.00	34.22	35.48	35.66 (35.53)
<i>Finger</i>	0.25	23.09	24.00	24.03 (24.00)
	0.50	26.43	27.38	27.42 (27.38)
	1.00	30.20	31.55	31.62 (31.56)
<i>Goldhill</i>	0.25	29.66	29.72	30.17 (29.72)
	0.50	32.20	32.38	32.65 (32.40)
	1.00	35.31	35.49	35.72 (35.54)
<i>Lena</i>	0.25	32.96	33.02	33.43 (33.02)
	0.50	36.10	36.22	36.56 (36.21)
	1.00	38.80	38.85	39.17 (38.98)
<i>Pepper</i>	0.25	32.54	32.38	32.55 (32.38)
	0.50	34.80	34.60	34.75 (34.64)
	1.00	36.49	36.71	36.91 (36.83)
<i>Bike</i>	0.25	26.35	27.08	27.20 (27.08)
	0.50	30.76	31.11	31.27 (31.11)
	1.00	35.29	35.36	35.56 (35.41)
<i>Cafe</i>	0.25	20.94	21.54	21.58 (21.54)
	0.50	24.24	24.52	24.58 (24.52)
	1.00	28.82	28.73	28.83 (28.73)
<i>Woman</i>	0.25	30.04	30.21	30.49 (30.21)
	0.50	33.10	33.25	33.45 (33.25)
	1.00	36.90	36.97	37.20 (37.04)
<i>Bridge</i>	0.25	34.38	34.90	35.05 (34.88)
	0.50	36.92	37.43	37.56 (37.43)
	1.00	39.86	40.27	40.50 (40.42)
<i>Cathedral</i>	0.25	33.17	33.88	34.10 (33.87)
	0.50	36.94	37.67	37.86 (37.66)
	1.00	40.72	41.00	41.19 (41.12)
<i>Deer</i>	0.25	32.89	33.12	33.12 (33.12)
	0.50	33.48	34.07	34.09 (34.09)
	1.00	34.70	35.66	35.75 (35.75)

proved as shown at Fig. 11. The experimental results indicate that the DCT-lifting, especially 2-D DCT-lifting, suppressed the rounding error to preserve transform performance by merging many rounding operations and the IRSE solved the boundary problem that conventional DL-LT has.

7. Conclusion

We presented a DL-LT composed of 1-D and 2-D DCT-liftings by considering parallel processing of two identical LTs and improved the conventional RSE by recalculating the boundary processing for more effective lossy-to-lossless image coding. These developments are called D²L-LT and IRSE, respectively. The DCT-lifting suppresses the rounding error to preserve transform performance and allows for direct use of a DCT matrix in each lifting coefficient. The 2-D DCT-lifting operates more effectively than the 1-D DCT-

lifting by making productive use of the 2-D space of an image. Furthermore, we solved the boundary problem that conventional DL-LT has by formulating IRSE. Finally, we showed that D²L-LT with IRSE mostly outperformed conventional L-LTs in lossy-to-lossless image coding.

Acknowledgment

This work was supported by a JSPS Grant-in-Aid for Young Scientists (B), Grant Number 16K18100.

References

- [1] K.R. Rao and P. Yip, *Discrete Cosine Transform Algorithms*, Academic Press, 1990.
- [2] G.K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol.38, no.1, pp.xviii–xxxiv, Feb. 1992.
- [3] T. Sikora, "MPEG digital video-coding standards," *IEEE Signal Process. Mag.*, vol.14, no.5, pp.82–100, Sept. 1997.
- [4] T. Wiegand, G.J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol.13, no.7, pp.560–576, July 2003.
- [5] G.J. Sullivan, J.R. Ohm, W.J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol.22, no.12, pp.1649–1668, Dec. 2012.
- [6] H.S. Malvar, "Lapped transforms for efficient transform/subband coding," *IEEE Trans. Acoust., Speech, Signal Process.*, vol.38, no.6, pp.969–978, June 1990.
- [7] P. Noll, "MPEG digital audio coding," *IEEE Signal Process. Mag.*, vol.14, no.5, pp.59–81, Sept. 1997.
- [8] J. Hilpert and S. Disch, "The MPEG surround audio coding standard," *IEEE Signal Process. Mag.*, vol.26, no.1, pp.148–152, Jan. 2009.
- [9] T.D. Tran, J. Liang, and C. Tu, "Lapped transform via time-domain pre- and post-filtering," *IEEE Trans. Signal Process.*, vol.6, no.6, pp.1557–1571, June 2003.
- [10] R.L. de Queiroz, T.Q. Nguyen, and K.R. Rao, "The GenLOT: Generalized linear-phase lapped orthogonal transform," *IEEE Trans. Signal Process.*, vol.44, no.3, pp.497–507, March 1996.
- [11] T.D. Tran, R.L. de Queiroz, and T.Q. Nguyen, "The generalized lapped biorthogonal transform," *Proc. ICASSP'98*, pp.1441–1444, Seattle, WA, May 1998.
- [12] W. Sweldens, "Lifting scheme: A new philosophy in biorthogonal wavelet constructions," *Proc. SPIE*, pp.68–79, San Diego, CA, July 1995.
- [13] 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.
- [14] W. Sweldens, "The lifting scheme: A construction of second generation wavelets," *SIAM J. Math. Anal.*, vol.29, no.2, pp.511–546, 1998.
- [15] K. Komatsu and K. Sezaki, "Reversible discrete cosine transform," *Proc. ICASSP'98*, pp.1769–1772, Seattle, WA, May 1998.
- [16] J. Liang and T.D. Tran, "Fast multiplierless approximations of the DCT with the lifting scheme," *IEEE Trans. Signal Process.*, vol.49, no.12, pp.3032–3044, Dec. 2001.
- [17] 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.
- [18] T. Suzuki and H. Aso, "M-channel fast Hartley transform based integer DCT for lossy-to-lossless image coding," *IEICE Trans. Fundamentals.*, vol.E96-A, no.4, pp.762–768, April 2013.
- [19] F. Dufaux, G.J. Sullivan, and T. Ebrahimi, "The JPEG XR image coding standard," *IEEE Signal Process. Mag.*, vol.26, no.6, pp.195–

- 199, 204, Nov. 2009.
- [20] C. Tu, S. Srinivasan, G.J. Sullivan, S. Regunathan, and H.S. Malvar, "Low-complexity hierarchical lapped transform for lossy-to-lossless image coding in JPEG XR/HD Photo," Proc. SPIE, pp.70730C–70730C–12, San Diego, CA, Aug. 2008.
- [21] L. Wang, L. Jiao, J. Wu, G. Shi, and Y. Gong, "Lossy-to-lossless image compression based on multiplier-less reversible integer time domain lapped transform," Signal Process. Image Commun., vol.25, no.8, pp.622–632, Sept. 2010.
- [22] P. Hao and Q. Shi, "Matrix factorizations for reversible integer mapping," IEEE Trans. Signal Process., vol.49, no.10, pp.2314–2324, Oct. 2001.
- [23] T. Suzuki and M. Ikehara, "Integer fast lapped transforms based on direct-lifting of DCTs for lossy-to-lossless image coding," EURASIP J. Image. Video Process., vol.2013, no.65, pp.1–9, Dec. 2013.
- [24] M.J.T. Smith and S.L. Eddins, "Analysis/synthesis techniques for subband image coding," IEEE Trans. Signal Process., vol.38, no.8, pp.1446–1456, Aug. 1990.
- [25] T. Nagai, M. Ikehara, M. Kaneko, and A. Kurematsu, "Generalized unequal length lapped orthogonal transform for subband image coding," IEEE Trans. Signal Process., vol.48, no.12, pp.3365–3378, Dec. 2000.
- [26] T. Suzuki and M. Ikehara, "Reversible symmetric non-expansive convolution: An effective image boundary processing for M -channel lifting-based linear-phase filter banks," IEEE Trans. Image Process., vol.23, no.6, pp.2744–2749, June 2014.
- [27] T. Suzuki and M. Ikehara, "Simplified DCT-lifting-based reversible lapped transforms using parallel processing of two same type lapped transforms," Proc. ICIP'14, pp.5586–5590, Paris, France, Oct. 2014.
- [28] T. Suzuki, M. Ikehara, and T.Q. Nguyen, "Generalized block-lifting factorization of M -channel biorthogonal filter banks for lossy-to-lossless image coding," IEEE Trans. Image Process., vol.21, no.7, pp.3220–3228, July 2012.
- [29] T. Suzuki and H. Kudo, "2D non-separable block-lifting structure and its application to M -channel perfect reconstruction filter banks for lossy-to-lossless image coding," IEEE Trans. Image Process., vol.24, no.12, pp.4943–4951, Dec. 2015.
- [30] "T.835: Information technology - JPEG XR image coding system - Reference software," ITU [Online]. Available: <https://www.itu.int/rec/T-REC-T.835-201201-I/>
- [31] "The USC-SIPI Image Database," Univ. Southern California, Signal and Image Processing Institute [Online]. Available: <http://sipi.usc.edu/database/>
- [32] "JPEG core experiment for the evaluation of JPEG XR image coding," EPFL, Multimedia Signal Processing Group [Online]. Available: <http://mmspg.epfl.ch/iqa>
- [33] "The New Test Images - Image Compression Benchmark," Rawzor - Lossless Compression Software for Camera RAW Images [Online]. Available: http://www.imagecompression.info/test_images/
- [34] 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.
- [35] T.D. Tran, M. Ikehara, and T.Q. Nguyen, "Linear phase paraunitary filter bank with filters of different lengths and its application in image compression," IEEE Trans. Signal Process., vol.47, no.10, pp.2730–2744, Oct. 1999.



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. From 2006 to 2008, he was with Toppan Printing Co., Ltd., Japan. From 2008 to 2011, he was a Research Associate of the Global Center of Excellence (G-COE) at Keio University, Japan. From 2010 to 2011, he was a Research Fellow of the Japan Society for the Promotion of Science (JSPS) and a Visiting Scholar at the Video Processing Group, the University of California, San Diego, CA. From 2011 to 2012, he was an Assistant Professor in the Department of Electrical and Electronic Engineering, College of Engineering, Nihon University, Japan. Since 2012, he has been an Assistant Professor in the Faculty of Engineering, Information and Systems, University of Tsukuba, Japan. His current research interests are theory and design of filter banks and their application to image and video processing.



Masaaki Ikehara received the B.E., M.E., and Dr.Eng. degrees in electrical engineering from Keio University, Yokohama, Japan, in 1984, 1986, and 1989, respectively. He was Appointed Lecturer at Nagasaki University, Nagasaki, Japan, from 1989 to 1992. In 1992, he joined the Faculty of Engineering, Keio University. From 1996 to 1998, he was a visiting researcher at the University of Wisconsin, Madison, and Boston University, Boston, MA. He is currently a Full Professor with the Department of Electronics and Electrical Engineering, Keio University. His research interests are in the areas of multirate signal processing, wavelet image coding, and filter design problems.