SIMPLIFIED DCT-LIFTING-BASED REVERSIBLE LAPPED TRANSFORMS USING PARALLEL PROCESSING OF TWO SAME TYPE LAPPED TRANSFORMS

Taizo Suzuki

Faculty of EIS, University of Tsukuba
Tsukuba, Ibaraki, 305-8573 Japan
Email: taizo@cs.tsukuba.ac.jp

Masaaki Ikehara

EEE Dept., Keio University
Yokohama, Kanagawa, 223-8522 Japan
Email: ikehara@tkhm.elec.keio.ac.jp

ABSTRACT

We present a realization of reversible lapped transforms (RevLTs) with simplified implementations, which are constructed by DCT and DST matrices, adders, and bit-shifters, for lossy-to-lossless image coding in this paper. Each DCT or DST matrix is directly used to each lifting coefficient block and it is called DCT-lifting structure. The structure is obtained by considering parallel processing of two 'same' type LTs and using DCT-lifting factorizations as our previous work. Furthermore, the Hadamard transform and scaling parts in the RevLTs are effectively implemented by extending 2D non-separable lifting structures derived from lifting-based lapped transform (L-LT) used for JPEG XR, the newest image coding standard. As a result, the proposed RevLTs achieve not only simplified implementations with any block size, but also comparable lossy-to-lossless image coding performance to the conventional RevLTs.

Index Terms— DCT, DST, lapped transform (LT), lifting structure, lossy-to-lossless image coding.

1. INTRODUCTION

JPEG and H.26x series are DCT [1]-based image/video compression (image coding) standards [2–5]. Since type-II DCT (DCT-II) has high energy compaction capability, it is often applied to the transform part in image coding. Type-III DCT (DCT-III) is the inverse transform of DCT-II. Such facts have provided many fast implementations for DCTs [6–9]. However, the DCTs generate an unpleasant *blocking artifact* for the reconstructed signals in low-bitrate coding due to the discontinuity between blocks. M-channel ($M = 2^k$, $k \in \mathbb{N}$) lapped transform (LT) [10] has overcame such a problem. Let LT in this paper be DCT-based LT well-known as a fast LT.

Several integer-to-integer transforms (integer transforms), mainly constructed by lifting structures [11], have been researched to achieve lossy-to-lossless image coding which has the scalability from lossless to lossy data. A 4×8 lifting-based lapped transform (L-LT) [12] used for JPEG XR [13], the newest image coding standard, is one of the most popular integer transforms. It has a very simple 2D non-separable lifting structures with only several adders and bit-shifters, and achieves low-complexity lossy-to-lossless image coding. Unfortunately, its filter size is too small to perform sufficient coding. In [14] and [15], we have proposed reversible LTs (RevLTs) to improve the coding performance with larger filter size than the 4×8 L-LT. The conventional RevLTs were based on DCT-lifting structures, which are directly applied DCT matrices to lifting coefficient blocks, i.e., direct-lifting structures of DCTs [16]. They can also reuse any existing software/hardware for DCTs as the lifting coefficient blocks. The RevLTs in [14] had very complicated lifting coefficient blocks in exchange for few lifting steps. Although the another RevLTs in [15] obtained by considering parallel processing of two 'different' type LTs have simple lifting coefficients, they have complicated implementations in exchange for removing floating-point multipliers.

In this paper, we present a realization of the RevLTs with simplified implementations, which are constructed by DCT and DST matrices, adders, and bit-shifters, for lossy-to-lossless image coding than the conventional RevLTs. By considering parallel processing of two 'same' type LTs unlike the RevLTs in [15] and using DCT-lifting factorizations in [14, 17], more parts are implemented by only simple DCT-lifting coefficients. Furthermore, the Hadamard transform and scaling parts in the RevLTs are effectively implemented by extending 2D non-separable lifting structures derived from L-LT in [12]. As a result, the proposed RevLTs achieve not only simplified implementations with any block size, but also comparable lossy-to-lossless image coding performance to the conventional RevLTs.

Notations: $\mathbf{I}$, $\mathbf{J}$, $\mathrm{diag}\{\cdots\}$, $\cdot^T$, and $\otimes$ are an identity matrix, a reversal identity matrix, a block diagonal matrix, matrix transposition, and a Kronecker product, respectively.

2. REVIEW AND DEFINITION

2.1. Lapped Transform (LT)

M-channel LT in this paper is based on polyphase structure from components with well-known fast-computable algorithms and a scaling coefficient s like L-LT in [12]. One of the most elegant solutions is shown in Fig. 1 and expressed as [14, 15]

$$\mathbf{E}(z) = \mathbf{P}\,\mathrm{diag}\{\mathbf{I}, \boldsymbol{\mathcal{S}}_{IV}\boldsymbol{\mathcal{C}}_{III}\}\mathbf{W}\boldsymbol{\Lambda}(z)\mathbf{W}\,\mathrm{diag}\{\boldsymbol{\mathcal{C}}_{II}, \boldsymbol{\mathcal{C}}_{IV}\}\mathbf{S}\mathbf{W}\tilde{\mathbf{I}} \tag{1}$$

where

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

z is a delay element, and the (m, n)-element of permutation matrix $\mathbf{P}$ as

$$[\mathbf{P}]_{m,n} = \begin{cases} 1 & (n \le M/2 - 1 \text{ and } m = 2n)\\ 1 & (n \ge M/2 \text{ and } m = 2n - M + 1)\\ 0 & (\text{otherwise})\end{cases},$$

This work was supported by JSPS Grant-in-Aid for Young Scientists (B) Grant Number 25820152.

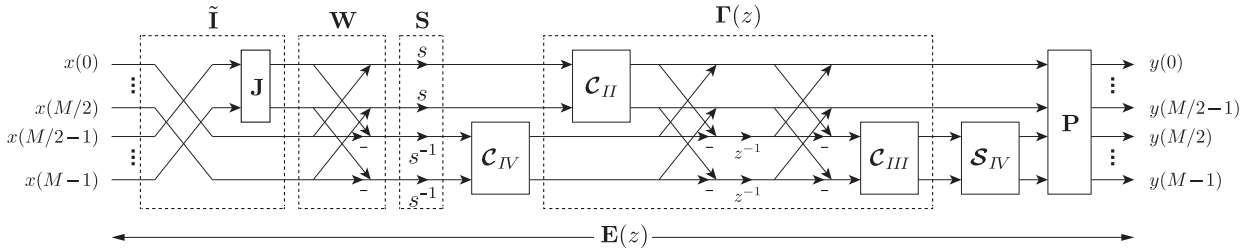

Fig. 1. Lattice structure of M-channel LT.

respectively. Also, $\boldsymbol{\mathcal{C}}_{II}$, $\boldsymbol{\mathcal{C}}_{III}$, $\boldsymbol{\mathcal{C}}_{IV}$, and $\boldsymbol{\mathcal{S}}_{IV}$ are DCT-II, DCT-III, type-IV DCT (DCT-IV), and type-IV DST (DST-IV) matrices with $N \times N$ ($N = M/2$) size. The (m, n)-elements of $\boldsymbol{\mathcal{C}}_{II}$ and $\boldsymbol{\mathcal{C}}_{IV}$ are defined as

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

where $c_m = 1/\sqrt{2}$ ($m = 0$) or 1 ($m \neq 0$), $\boldsymbol{\mathcal{C}}_{IV}^{-1} = \boldsymbol{\mathcal{C}}_{IV}^{T} = \boldsymbol{\mathcal{C}}_{IV}$, and $\boldsymbol{\mathcal{C}}_{III}$ and $\boldsymbol{\mathcal{S}}_{IV}$ are established as $\boldsymbol{\mathcal{C}}_{III} = \boldsymbol{\mathcal{C}}_{II}^{-1} = \boldsymbol{\mathcal{C}}_{II}^{T}$ and $\boldsymbol{\mathcal{S}}_{IV} = \mathbf{D}\,\boldsymbol{\mathcal{C}}_{IV}\mathbf{J} = \mathbf{J}\,\boldsymbol{\mathcal{C}}_{IV}\mathbf{D}$ ($[\mathbf{D}]_{m,n} = (-1)^m$ ($m = n$) or 0 ($m \neq n$)), respectively. The optimal s is experimentally-determined in each block size M, e.g., $s = 0.8981$ and 0.9360 in cases of $M = 8$ and 16, respectively. It is clear that the LTs with the scaling coefficient $s = 1$ are lapped orthogonal transforms (LOTs) with type-II lattice structure in [10].

2.2. DCT-Lifting Structure

The block-lifting structure [18], a special class of standard lifting structure [11], achieves an effective lossy-to-lossless image coding by merging many rounding operations. In Fig. 2, the analysis input signal vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, the analysis output and synthesis input signal vectors $\mathbf{y}_i$ and $\mathbf{y}_j$, the synthesis output signal vectors $\mathbf{z}_i$ and $\mathbf{z}_j$, and the lifting coefficient blocks $\mathbf{T}_L$ and $\mathbf{T}_U$ are presented by

$$\begin{aligned}
\mathbf{y}_j &= \mathbf{x}_j + \mathrm{round}\{\mathbf{T}_L\mathbf{x}_i\} \\
\mathbf{y}_i &= \mathbf{x}_i + \mathrm{round}\{\mathbf{T}_U\mathbf{y}_j\} \\
\mathbf{z}_i &= \mathbf{y}_i - \mathrm{round}\{\mathbf{T}_U\mathbf{y}_j\} = \mathbf{x}_i \\
\mathbf{z}_j &= \mathbf{y}_j - \mathrm{round}\{\mathbf{T}_L\mathbf{y}_i\} = \mathbf{x}_j.
\end{aligned}$$

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

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

When the lifting coefficient block is directly expressed by a DCT or DST matrix, e.g., $\mathbf{T}_L = \boldsymbol{\mathcal{C}}_{II}$, the structure is termed DCT-lifting structure, a special class of direct-lifting structure [16], in this paper. The structure allows any DCT algorithm and software/hardware in the block.

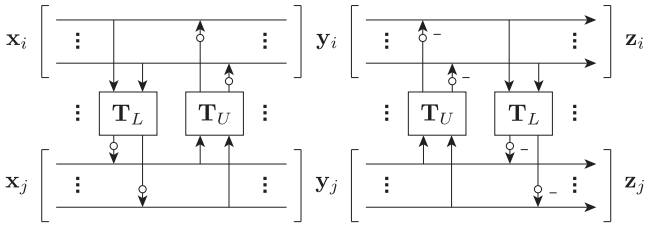

Fig. 2. Block-lifting structures (white circles mean rounding operations).

3. SIMPLIFIED DCT-LIFTING-BASED REVLT

We introduce a realization of RevLTs with simplified implementations in this section. When an image $\mathbf{X}$ is 2D-transformed to $\mathbf{Y}$ by an 1D transform $\mathbf{T} = \mathbf{T}_{K-1}\cdots\mathbf{T}_1\mathbf{T}_0$, the 2D-transformed image $\mathbf{Y}$ is expressed by

$$\mathbf{Y} = \mathbf{T}_{K-1}\cdots\mathbf{T}_1\mathbf{T}_0\mathbf{X}\mathbf{T}_0^T\mathbf{T}_1^T\cdots\mathbf{T}_{K-1}^T.$$

It means that $\mathbf{X}$ is 2D-transformed by $\mathbf{T}_{k+1}$ ($0 \leq k \leq K-2$, $K \in \mathbb{N}$) after 2D transform by $\mathbf{T}_k$. With the fact, the proposed RevLTs are obtained by each 2D transform of $\tilde{\mathbf{I}}$, $\mathbf{W}$, $\mathbf{S}$, $\mathbf{P}$, and the residual part in Eq. (1). The parts $\tilde{\mathbf{I}}$ and $\mathbf{P}$ are not discussed any more because they just only reorder signals.

3.1. 2D Non-Separable Lifting Factorization of Hadamard Transform Part

The first Hadamard transform part $\mathbf{W}$ in Eq. (1) is effectively implemented by extending a 2D non-separable transform in [12]. Consider an image that has been 2D-transformed by $\mathbf{W}_{2D}$, which is the 2D transform of $\mathbf{W}$, as

$$\begin{bmatrix}\mathbf{Y}_{LL}^T & \mathbf{Y}_{HL}^T & \mathbf{Y}_{LH}^T & \mathbf{Y}_{HH}^T\end{bmatrix}^T = \mathbf{W}_{2D}\begin{bmatrix}\mathbf{X}_{LL}^T & \mathbf{X}_{HL}^T & \mathbf{X}_{LH}^T & \mathbf{X}_{HH}^T\end{bmatrix}^T$$

where $\mathbf{X}_{\times\times}$s are the top-left, top-right, bottom-left, and bottom-right $N \times N$ blocks of an $M \times M$ image, and $\mathbf{Y}_{\times\times}$s are their respective output blocks. We extend the 4×4 2D non-separable transform in [12] to $2M \times 2M$ 2D non-separable transform as

$$\mathbf{W}_{2D} = \mathbf{W} \otimes \mathbf{W} = \frac{1}{2}\begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{I} & \mathbf{I} \\ \mathbf{I} & -\mathbf{I} & \mathbf{I} & -\mathbf{I} \\ \mathbf{I} & \mathbf{I} & -\mathbf{I} & -\mathbf{I} \\ \mathbf{I} & -\mathbf{I} & -\mathbf{I} & \mathbf{I} \end{bmatrix},$$

and its lifting structures are expressed by

$$\mathbf{W}_{2D} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{I} & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}$$
$$\times \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \frac{1}{2}\mathbf{I} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{I} \end{bmatrix}$$
$$\times \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\frac{1}{2}\mathbf{I} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}.$$

The lifting structure has very simple operations with only $5M$ adders, M bit-shifters, and permutations. It is easier processing than the implementation for the Hadamard transform part in [15].

3.2. 2D Non-Separable Lifting Factorization of Scaling Part

In the same way as Section 3.1, $\mathbf{S}_{2D}$, which is the 2D transform of the scaling part $\mathbf{S}$, is also implemented by using a 2D non-separable lifting structure as [12]

$$\begin{bmatrix} \mathbf{Y}_{LL}^T & \mathbf{Y}_{HL}^T & \mathbf{Y}_{LH}^T & \mathbf{Y}_{HH}^T \end{bmatrix}^T = \mathbf{S}_{2D} \begin{bmatrix} \mathbf{X}_{LL}^T & \mathbf{X}_{HL}^T & \mathbf{X}_{LH}^T & \mathbf{X}_{HH}^T \end{bmatrix}^T$$

where

$$\mathbf{S}_{2D} = \mathbf{S} \otimes \mathbf{S} = \mathrm{diag}\{s^2\mathbf{I}, \mathbf{I}, \mathbf{I}, s^{-2}\mathbf{I}\}$$
$$= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ s^2\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}$$
$$\times \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & -s^{-2}\mathbf{I} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ s^2\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}.$$

Since s^2 and s^{-2} are floating-point coefficients which cause high complexity, they are experimentally approximated to dyadic coefficients as $\alpha/2^\beta$ ($\alpha, \beta \in \mathbb{N}$) to delete multipliers, e.g., $\mathrm{diag}\{s^2, s^{-2}\} = \mathrm{diag}\{413/2^9, 317/2^8\}$ and $\mathrm{diag}\{449/2^9, 292/2^8\}$ in cases of $M = 8$ and 16, respectively.

3.3. Simplified DCT-Lifting Factorizations of Residual Part

Let $\mathbf{\Gamma}(z)$ be

$$\mathbf{\Gamma}(z) \triangleq \mathrm{diag}\{\mathbf{I}, \boldsymbol{\mathcal{C}}_{III}\}\mathbf{W}\mathbf{\Lambda}(z)\mathbf{W}\,\mathrm{diag}\{\boldsymbol{\mathcal{C}}_{II}, \mathbf{I}\} \quad (2)$$

which is a part of the residual part. After $\mathbf{W}\mathbf{\Lambda}(z)\mathbf{W}$ in Eq. (2) is factorized into simple lifting structures as

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

by interchanging the scaling coefficients $\sqrt{2}$ and $1/\sqrt{2}$ between two $\mathbf{W}$s, $\boldsymbol{\mathcal{C}}_{II}$ in Eq. (2) is moved from the right to the left as

$$\mathbf{\Gamma}(z) = \mathrm{diag}\{\boldsymbol{\mathcal{C}}_{II}, \boldsymbol{\mathcal{C}}_{III}\} \times \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \boldsymbol{\mathcal{C}}_{II} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\frac{1}{2}\boldsymbol{\mathcal{C}}_{III} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{\Lambda}(z) \begin{bmatrix} \mathbf{I} & \frac{1}{2}\boldsymbol{\mathcal{C}}_{III} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\boldsymbol{\mathcal{C}}_{II} & \mathbf{I} \end{bmatrix}. \quad (3)$$

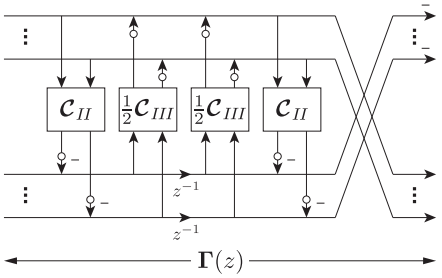

Fig. 3. 1D separable block-lifting structure of $\mathbf{\Gamma}(z)$ (white circles mean rounding operations).

The last part $\mathrm{diag}\{\boldsymbol{\mathcal{C}}_{II}, \boldsymbol{\mathcal{C}}_{III}\}$ in Eq. (3) is factorized into customized direct-lifting structures as [16]

$$\mathrm{diag}\{\boldsymbol{\mathcal{C}}_{II}, \boldsymbol{\mathcal{C}}_{III}\} = \begin{bmatrix} \mathbf{0} & -\mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\boldsymbol{\mathcal{C}}_{II} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \boldsymbol{\mathcal{C}}_{III} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\boldsymbol{\mathcal{C}}_{II} & \mathbf{I} \end{bmatrix}. \quad (4)$$

By substituting Eq. (4) into Eq. (3), $\mathbf{\Gamma}(z)$ is simplified as

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

and shown in Fig. 3. Consequently, two redundant lifting-based $\mathbf{W}$s required in [15] are grouped into the DCT-lifting structures with 1 bit-shifters. Moreover, the 1D separable block-lifting-based $\mathbf{\Gamma}(z)$ is 2D-implemented by using 2D non-separable block-lifting structures in [17] for better coding.

Next, we consider the parallel processing of two LTs in Eq. (1) as with the RevLTs in [15], where the two LTs are completely 'same' types unlike those used in [15]. It means that when a row (column) signal vector $\mathbf{x}_i$ is processed by one of two LTs, other row (column) signal vector $\mathbf{x}_j$ is processed by another one. The DCT-IV and DST-IV matrices $\boldsymbol{\mathcal{C}}_{IV}$ and $\boldsymbol{\mathcal{S}}_{IV}$ in Eq. (1) are processed by direct-lifting structures in each combination of $\boldsymbol{\mathcal{C}}_{IV}$ and $\boldsymbol{\mathcal{S}}_{IV}$ stepping over the two LTs. The direct-lifting structures are presented as [14, 15]

$$\mathrm{diag}\{\boldsymbol{\mathcal{C}}_{IV}, \boldsymbol{\mathcal{C}}_{IV}\} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \boldsymbol{\mathcal{C}}_{IV} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\boldsymbol{\mathcal{C}}_{IV} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \boldsymbol{\mathcal{C}}_{IV} & \mathbf{I} \end{bmatrix}$$
$$\mathrm{diag}\{\boldsymbol{\mathcal{S}}_{IV}, \boldsymbol{\mathcal{S}}_{IV}\} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \boldsymbol{\mathcal{S}}_{IV} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\boldsymbol{\mathcal{S}}_{IV} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \boldsymbol{\mathcal{S}}_{IV} & \mathbf{I} \end{bmatrix}.$$

The resulting lattice structure of RevLT is shown in Fig. 4. The matrices $\tilde{\mathbf{I}}$, $\mathbf{W}$, $\mathbf{S}$, $\mathbf{\Gamma}(\mathbf{z})$, and $\mathbf{P}$ are 2D-implemented as already described. Note that the RevLTs in [15] required two 'different' type LTs, whereas the proposed RevLTs require two 'same' type LTs. They produce a simpler implementation than the RevLTs in [15]. Also, they have simpler lifting coefficients than those of the RevLTs in [14].

4. EXPERIMENTAL RESULTS

The proposed 8 and 16-channel RevLTs were designed and compared with the conventional methods in lossy-to-lossless image coding simulation by the lossless bitrate (LBR) [bpp] and peak signal-

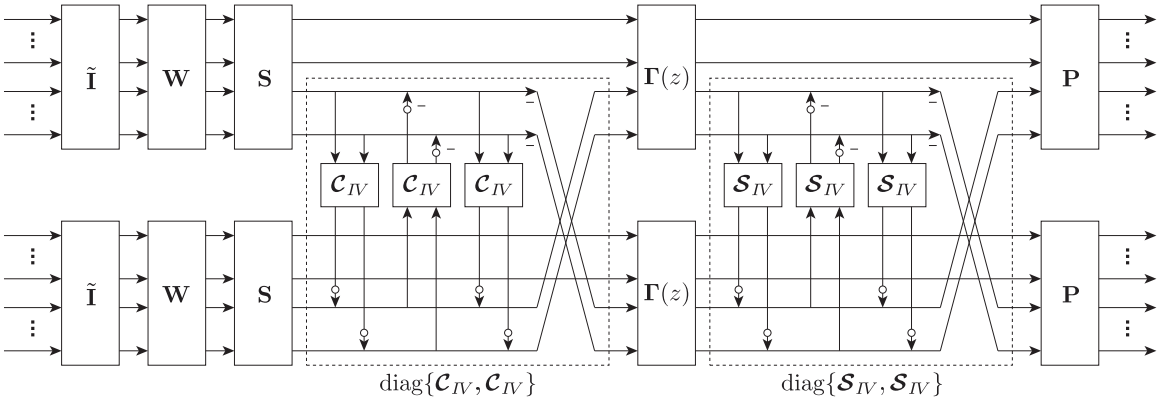

Fig. 4. Lattice structure of the proposed RevLT (white circles mean rounding operations).

Table 1. Lossless image coding results (LBR [bpp]).

Test Images	L-LT [12]	8×16 RevLTs [14]	[15]	Prop.	16×32 RevLTs [14]	[15]	Prop.
Baboon	6.23	**6.21**	**6.21**	**6.21**	**6.21**	6.22	**6.21**
Barbara	4.95	4.86	4.90	4.84	4.79	4.82	**4.78**
Boat	5.21	5.14	5.16	5.13	5.13	5.15	**5.12**
Elaine	5.26	5.23	5.26	5.23	5.19	5.20	**5.18**
Finger	5.89	5.82	5.84	5.82	**5.73**	5.75	**5.73**
Finger2	5.61	5.53	5.55	5.52	5.49	5.51	**5.48**
Goldhill	**5.10**	5.12	5.15	5.11	5.12	5.14	5.11
Grass	6.09	6.07	6.08	6.07	6.07	6.07	**6.06**
Lena	4.63	4.62	4.66	**4.61**	4.64	4.66	4.63
Pepper	4.99	**4.93**	4.96	**4.93**	4.96	4.98	4.96

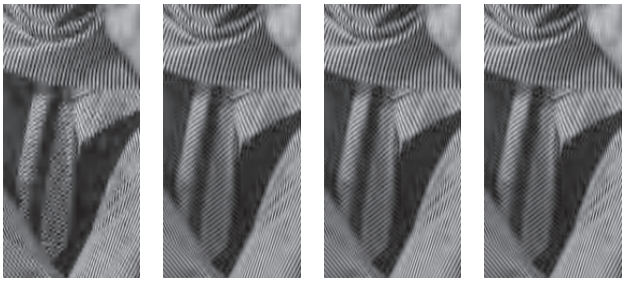

Fig. 5. Comparison of a particular area of an image *Barbara* (bitrate: 0.25[bpp]): (left-right) 4×8 L-LT, 16×32 RevLT in [14], 16×32 RevLT in [15], and 16×32 proposed RevLT.

to-noise ratio (PSNR) [dB] as follows:

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

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

where MSE is the mean squared error. To fairly evaluate the performance of transforms, we used several 512×512 8-bit grayscale test images and a wavelet-based coder SPIHT [19] to encode the transformed images as discussed in [20].

Table 1, Table 2, and Fig. 5 show lossless and lossy image coding results. Although the RevLTs in [14] show good results by few lifting steps on the surface, they have very complicated lifting coefficient blocks as already discussed. The proposed RevLTs achieved not only simplified implementations with any block size, but also comparable coding performance to the conventional RevLTs.

Table 2. Lossy image coding results (PSNR[dB]).

Rate [bpp]	L-LT [12]	8×16 RevLTs [14]	[15]	Prop.	16×32 RevLTs [14]	[15]	Prop.
			Barbara				
0.25	27.01	28.05	28.03	28.04	**28.93**	28.91	28.92
0.50	30.85	32.14	32.08	32.12	**32.86**	32.80	**32.86**
1.00	36.00	37.04	36.85	37.03	**37.40**	37.21	**37.40**
			Finger				
0.25	22.95	23.73	23.73	23.74	**24.02**	24.01	**24.02**
0.50	26.31	26.97	26.93	26.96	27.35	27.34	**27.36**
1.00	30.12	30.87	30.82	30.88	31.51	31.45	**31.52**
			Goldhill				
0.25	29.63	29.70	29.69	**29.71**	**29.71**	29.69	**29.71**
0.50	32.02	32.29	32.22	32.29	32.31	32.25	**32.32**
1.00	35.17	35.46	35.30	**35.48**	**35.48**	35.33	**35.48**
			Lena				
0.25	32.76	**32.94**	32.89	32.91	32.87	32.84	32.85
0.50	35.90	**36.20**	36.08	36.19	36.06	35.98	36.05
1.00	38.62	38.93	38.55	**38.96**	38.80	38.54	38.86
			Pepper				
0.25	**32.53**	32.42	32.39	32.41	32.16	32.14	32.14
0.50	34.71	**34.82**	34.72	34.81	34.43	34.35	34.42
1.00	36.39	36.75	36.52	**36.80**	36.60	36.42	36.67

5. CONCLUSION

We have presented a realization of RevLTs with simplified DCT-lifting structures in this paper. The RevLTs were achieved by using two 'same' LTs unlike our previous work and extending 2D non-separable lifting structures derived from L-LT used for JPEG XR. As a result, they achieved not only simplified implementations with any block size, but also comparable lossy-to-lossless image coding performance to the conventional RevLTs.

6. REFERENCES

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

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

[3] T. Sikora, "MPEG digital video-coding standards," *IEEE Signal Process. Mag.*, vol. 14, no. 5, pp. 82–100, Sep. 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] W. H. Chen, C. H. Smith, and S. C. Fralick, "A fast computational algorithm for the discrete cosine transform," *IEEE Trans. Commun.*, vol. 25, no. 9, pp. 1004–1009, Sep. 1977.

[7] 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, Apr. 1984.

[8] B. G. Lee, "A new algorithm to compute the discrete cosine transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 6, pp. 1243–1245, June 1984.

[9] Z. Wang, "On computing the discrete Fourier and cosine transforms," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 4, pp. 1341–1344, Apr. 1985.

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

[11] W. Sweldens, "The lifting scheme: A new philosophy in biorthogonal wavelet constructions," in *Proc. of SPIE*, San Diego, CA, July 1995, vol. 2569, pp. 1–12.

[12] C. Tu, S. Srinivasan, G. J. Sullivan, S. Regunathan, and H. S. Malvar, "Low-complexity hierarchical lapped transform for lossy-to-lossless image coding in JPEG XR/HD Photo," in *Proc. of SPIE*, San Diego, CA, Aug. 2008, vol. 7073, pp. 1–12.

[13] 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.

[14] T. Suzuki and H. Kudo, "Integer fast lapped biorthogonal transform via applications of DCT matrices and dyadic-valued factors for lifting coefficient blocks," in *Proc. of ICIP'13*, Merboulne, Australia, Sep. 2013, pp. 800–804.

[15] 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.

[16] 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.

[17] T. Suzuki and H. Kudo, "Two-dimensional non-separable block-lifting-based M-channel biorthogonal filter banks," in *Proc. of EUSIPCO'14*, 2014, accepted.

[18] 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.

[19] 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.

[20] 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.