TWO-DIMENSIONAL NON-SEPARABLE BLOCK-LIFTING-BASED M-CHANNEL BIORTHOGONAL FILTER BANKS

Taizo Suzuki and Hiroyuki Kudo

Faculty of Engineering, Information and Systems, University of Tsukuba
Tsukuba, Ibaraki, 305-8573 Japan
Email: {taizo, kudo}@cs.tsukuba.ac.jp

ABSTRACT

We propose a two-dimensional (2D) non-separable block-lifting structure (NSBL) that is easily formulated from the one-dimensional (1D) separable block-lifting structure (SBL) and 2D non-separable lifting structure (NSL). The NSBL can be regarded as an extension of the NSL because a two-channel NSBL is completely equivalent to a NSL. We apply the NSBL to M-channel ($M = 2^n$, $n \in \mathbb{N}$) biorthogonal filter banks (BOFBs). The NSBL-based BOFBs (NSBL-BOFBs) outperform SBL-based BOFBs (SBL-BOFBs) at lossy-to-lossless coding, whose image quality is scalable from lossless data to high compressed lossy data, because their rounding error is reduced by merging many rounding operations, i.e., the number of the NSBL is the almost half that of the SBL.

Index Terms— Biorthogonal filter bank (BOFB), lossy-to-lossless image coding, non-separable block-lifting structure (NSBL).

1. INTRODUCTION

Filter banks (FBs) [1] have been widely researched as a way to efficiently compress various signals. The polyphase matrices of M-channel ($M = 2^n$, $n \in \mathbb{N}$) FBs are presented as

$$\begin{bmatrix} H_0(z) & H_1(z) & \cdots & H_{M-1}(z) \end{bmatrix}^T = \mathbf{E}(z^M)\mathbf{e}(z)^T$$
$$\begin{bmatrix} F_0(z) & F_1(z) & \cdots & F_{M-1}(z) \end{bmatrix} = \mathbf{e}(z)\mathbf{R}(z^M)$$

where $\mathbf{e}(z) = [1, z^{-1}, \cdots, z^{-(M-1)}]$, and $H_i(z)$, $F_i(z)$, z, and $\cdot^T$ denote an analysis filter, a synthesis filter, a delay element, and matrix transposition, respectively. If $\mathbf{E}(z)$ is invertible, the inverse of $\mathbf{E}(z)$ can be chosen as a synthesis polyphase matrix $\mathbf{R}(z)$, and such FBs are called perfect reconstruction FBs (PRFBs). Especially, PRFBs without paraunitariness are called biorthogonal FBs (BOFBs). The JPEG series and H.26x series of global standards use various classes of PRFBs, including the discrete cosine/sine transform (DCT and DST) [2], discrete wavelet transform (DWT) [3], and lapped transform (LT) [4]. However, there is a growing need for better FBs in order to alleviate the burden on servers and free up communication bandwidth.

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

Table 1. Classification of lifting structures.

	SL [5]	SBL [9]	NSL [10]	NSBL
Block-Lifting	—	✓	—	✓
Non-Separable	—	—	✓	✓

Lossy-to-lossless image coding, which merges two or more pieces of data into one piece of data of the same piece of content, i.e., "one source multi-use" image coding, has attracted attention from researchers as a possible way to meet this need. Reversible transforms that map integers to integers, called integer-to-integer transforms, are important tools for lossy-to-lossless image coding. Sweldens presented a lifting structure [5] with which to achieve integer-to-integer transforms, and this structure has been applied to many FBs [6–8]. Although JPEG XR has scalability ranging from lossless to lossy as a result of using a lifting-based LT (L-LT), its coding performance is not sufficient especially for images with high-frequency components (texture).

The one-dimensional (1D) separable block-lifting structure (SBL) of BOFB was proposed in [9] for the purpose of designing lifting-based FBs (L-FBs) with higher coding performance. Usually, the design parameters and structure of L-FBs are constrained when factorizing the original FB into lifting structures, whereas the SBL-based BOFBs (SBL-BOFBs) presented in [9] do not constrain them except in the initial block. The SBL is better at lossy-to-lossless image coding because it uses fewer rounding operations in comparison with the standard 1D separable lifting structure (SL). Furthermore, the two-dimensional (2D) non-separable lifting structure (NSL) for DWTs, NSL-based DWTs (NSL-DWTs), proposed in [10] performs even better at coding because it uses fewer rounding operations than the SL.

Here, we propose a 2D non-separable block-lifting structure (NSBL) that is easily formulated from the SBL and NSL methods. The NSBL can be regarded as an extension of the NSL because a NSBL with $M = 2$ is completely equivalent

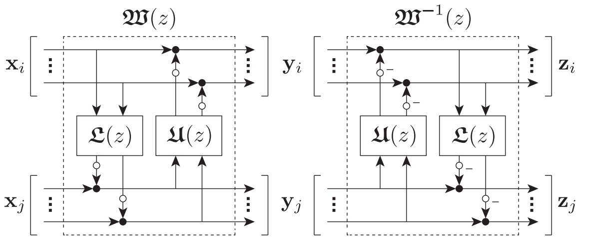

Fig. 1. Block-lifting structures (black and white circles mean adders and rounding operations, respectively).

to a NSL. We apply the NSBL to M-channel BOFBs, NSBL-based BOFBs (NSBL-BOFBs), and show that the BOFBs perform better at lossy-to-lossless coding than the conventional SBL-BOFBs do because their rounding error is reduced by merging many rounding operations.

Notations: A classification of lifting structures is shown in Table. 1. $\mathbf{I}_m$, $\mathbf{0}$, $\det(\cdot)$, and $\mathrm{diag}(\cdots)$ denote an $m \times m$ identity matrix, null matrix, determinant of a matrix, and (block) diagonal matrix, respectively. $\mathbf{I}_m$ is simply expressed by $\mathbf{I}$ if its size is clear. Indexes x, y, w, and $2d$ in the matrices mean to operate horizontally, vertically, horizontally or vertically, and horizontally and vertically, respectively.

2. REVIEW AND DEFINITIONS

2.1. Block-Lifting Structure

We proposed the block-lifting structure [9], which is a special class of standard lifting structure [5], as shown in Fig. 1. It is good for lossy-to-lossless image coding because it reduces the rounding error by merging many rounding operations. In Fig. 1, the analysis input signal vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, the analysis output (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 $\mathfrak{L}(z)$ and $\mathfrak{U}(z)$ are related as follows:

$$\mathbf{y}_j = \mathbf{x}_j + \mathscr{R}(\mathfrak{L}(z)\mathbf{x}_i), \qquad \mathbf{y}_i = \mathbf{x}_i + \mathscr{R}(\mathfrak{U}(z)\mathbf{y}_j)$$
$$\mathbf{z}_i = \mathbf{y}_i - \mathscr{R}(\mathfrak{U}(z)\mathbf{y}_j) = \mathbf{x}_i, \; \mathbf{z}_j = \mathbf{y}_j - \mathscr{R}(\mathfrak{L}(z)\mathbf{y}_i) = \mathbf{x}_j$$

where $\mathscr{R}(\cdot)$ denotes a rounding operation. In these cases, the matrices and their inverse matrices are expressed by

$$\begin{bmatrix}\mathbf{y}_i\\ \mathbf{y}_j\end{bmatrix} = \mathfrak{W}(z)\begin{bmatrix}\mathbf{x}_i\\ \mathbf{x}_j\end{bmatrix}, \quad \begin{bmatrix}\mathbf{z}_i\\ \mathbf{z}_j\end{bmatrix} = \mathfrak{W}^{-1}(z)\begin{bmatrix}\mathbf{y}_i\\ \mathbf{y}_j\end{bmatrix} = \begin{bmatrix}\mathbf{x}_i\\ \mathbf{x}_j\end{bmatrix}$$

where $\mathfrak{W}(z) = \mathfrak{B}_U\mathfrak{B}_L$, $\mathfrak{W}^{-1}(z) = \mathfrak{B}_L^{-1}\mathfrak{B}_U^{-1}$,

$$\mathfrak{B}_U = \begin{bmatrix}\mathbf{I} & \mathfrak{U}(z)\\ \mathbf{0} & \mathbf{I}\end{bmatrix}, \quad \mathfrak{B}_U^{-1} = \begin{bmatrix}\mathbf{I} & -\mathfrak{U}(z)\\ \mathbf{0} & \mathbf{I}\end{bmatrix}$$
$$\mathfrak{B}_L = \begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathfrak{L}(z) & \mathbf{I}\end{bmatrix}, \quad \mathfrak{B}_L^{-1} = \begin{bmatrix}\mathbf{I} & \mathbf{0}\\ -\mathfrak{L}(z) & \mathbf{I}\end{bmatrix}.$$

Note that the rounding operations are actually implemented even if the lifting matrix expression omits the notation of them. The block-lifting structure for a 1D implementation is called "SBL" to distinguish it from the "NSBL" proposed in this paper. When $M = 2$, they will also be called "SL" and "NSL".

2.2. SBL-BOFBs

We factorized the BOFBs composed of building blocks with McMillan degree γ_k, where $\gamma_k = M/2$ for simplicity, into the SBL in [9] as

$$\mathbf{E}(z) = \prod_{k=K-1}^{1}\{\mathbf{E}_k(z)\}\mathbf{G}_0 \tag{1}$$

where

$$\mathbf{E}_k(z) = \begin{bmatrix}\mathbf{I} & \mathbf{0}\\ -\mathbf{L}_k & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{U}_k\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\mathbf{\Lambda}(z)\begin{bmatrix}\mathbf{I} & -\mathbf{U}_k\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{L}_k & \mathbf{I}\end{bmatrix},$$

$\mathbf{\Lambda}(z) = \mathrm{diag}\{\mathbf{I}, z^{-1}\mathbf{I}\}$, and $\mathbf{L}_k$ and $\mathbf{U}_k$ are arbitrary $M/2 \times M/2$ matrices if paraunitariness is not required. In addition, the first block $\mathbf{G}_0$ is constrained to be $\det(\mathbf{G}_0) = \pm n$ ($n \in \mathbb{N}$) for the purpose of making a lifting factorization. To improve coding performance, Eq. (1) can be rewritten as

$$\mathbf{E}(z) = \mathbf{W}_K(z)\prod_{k=K-1}^{1}\{\mathbf{\Lambda}(z)\mathbf{W}_k(z)\}\mathbf{G}_0, \tag{2}$$

where

$$\mathbf{W}_k(z) = \begin{bmatrix}\mathbf{I} & \hat{\mathbf{U}}_k(z)\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \hat{\mathbf{L}}_k(z) & \mathbf{I}\end{bmatrix}, \; \mathbf{W}_K(z) = \begin{bmatrix}\mathbf{I} & \mathbf{0}\\ -\mathbf{L}_{K-1} & \mathbf{I}\end{bmatrix},$$

$\hat{\mathbf{U}}_k(z) = (z^{-1} - 1)\mathbf{U}_k$, and $\hat{\mathbf{L}}_k(z) = \mathbf{L}_1$ ($k = 1$) or $\mathbf{L}_k - \mathbf{L}_{k-1}$ (otherwise). In comparison with the SBL-BOFBs in Eq. (1), the SBL-BOFBs in Eq. (2) are more effective at lossy-to-lossless image coding because they reduce the rounding error by merging more rounding operations.

2.3. NSL-DWTs

9/7-tap and 5/3-tap SL-based DWTs (9/7-SL-DWT and 5/3-SL-DWT) [3] are used in the JPEG 2000 lossy and lossless modes, respectively. Let $\mathbf{E}_W(z)$ be a polyphase matrix of SL-DWTs, expressed as

$$\mathbf{E}_W(z) = \mathrm{diag}\{s, s^{-1}\}\prod_{k=N-1}^{0}\mathbf{w}_k(z) \tag{3}$$

where

$$\mathbf{w}_k(z) = \begin{bmatrix}1 & u_k(z)\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ l_k(z) & 1\end{bmatrix}.$$

If an image is 2D-transformed by the NSL-DWT polyphase matrix $\mathbf{E}^{2d}_{W,k}(z_{2d})$ in Eq. (3), one can write

$$\begin{bmatrix} Y^T_{LL} & Y^T_{HL} & Y^T_{LH} & Y^T_{HH} \end{bmatrix}^T = \mathbf{E}^{2d}_{W,k}(z_{2d}) \begin{bmatrix} X^T_{LL} & X^T_{HL} & X^T_{LH} & X^T_{HH} \end{bmatrix}^T$$

where X_{LL}, X_{HL}, X_{LH}, and X_{HH} are the top-left, top-right, bottom-left, and bottom-right pixels in 2×2 blocks composing the image, Y_{LL}, Y_{HL}, Y_{LH}, and Y_{HH} are their output pixels, and

$$\mathbf{E}^{2d}_{W,k}(z_{2d}) = \mathrm{diag}\{s^2, 1, 1, s^{-2}\} \prod_{k=N-1}^{0} \mathbf{w}^{2d}_k(z_{2d}).$$

The 2D implementation $\mathbf{w}^{2d}_k(z_{2d})$ in $\mathbf{E}^{2d}_W(z_{2d})$ is represented as [10]

$$\begin{aligned} \mathbf{w}^{2d}_k(z_{2d}) = &\begin{bmatrix} 1 & \begin{bmatrix} u^x_k(z_x) & u^y_k(z_y) & -u^{2d}_k(z_{2d}) \end{bmatrix} \\ \mathbf{0} & \mathbf{I}_3 \end{bmatrix} \\ &\times \begin{bmatrix} 1 & \begin{bmatrix} l^x_k(z_x)^T & l^y_k(z_y)^T \end{bmatrix} & 0 \\ \mathbf{0} & \mathbf{I}_2 & \mathbf{0} \\ 0 & \begin{bmatrix} u^y_k(z_y)^T & u^x_k(z_x)^T \end{bmatrix} & 1 \end{bmatrix}^T \\ &\times \begin{bmatrix} \mathbf{I}_3 & \mathbf{0} \\ \begin{bmatrix} l^{2d}_k(z_{2d}) & l^y_k(z_y) & l^x_k(z_x) \end{bmatrix} & 1 \end{bmatrix}. \end{aligned} \tag{4}$$

The NSL is also more effective at lossy-to-lossless image coding than the SL is because it uses fewer rounding operations.

3. NSBL-BOFBS

3.1. Derivation of NSBL

We introduce NSBL in this subsection.

Theorem: Consider an image that has been 2D-transformed by the set of lower and upper block-lifting matrices in Fig. 1 as follows (Fig. 2):

$$\begin{bmatrix} \mathbf{Y}^T_{LL} & \mathbf{Y}^T_{HL} & \mathbf{Y}^T_{LH} & \mathbf{Y}^T_{HH} \end{bmatrix}^T = \mathfrak{W}^{2d}(z_{2d}) \begin{bmatrix} \mathbf{X}^T_{LL} & \mathbf{X}^T_{HL} & \mathbf{X}^T_{LH} & \mathbf{X}^T_{HH} \end{bmatrix}^T \tag{5}$$

where

$$\mathfrak{W}^{2d}(z_{2d}) = \begin{bmatrix} \mathfrak{W}^x(z_x) & \mathbf{0} \\ \mathbf{0} & \mathfrak{W}^x(z_x) \end{bmatrix} \mathbf{P} \begin{bmatrix} \mathfrak{W}^y(z_y) & \mathbf{0} \\ \mathbf{0} & \mathfrak{W}^y(z_y) \end{bmatrix} \mathbf{P}$$

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

and $\mathfrak{W}^w(z_w) = \mathfrak{B}^w_U(z_w)\mathfrak{B}^w_L(z_w)$. Also, $\mathbf{X}_{LL}$, $\mathbf{X}_{HL}$, $\mathbf{X}_{LH}$, and $\mathbf{X}_{HH}$ are the top-left, top-right, bottom-left, and bottom-right $M/2 \times M/2$ 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. $\mathfrak{W}^{2d}(z_{2d})$ in Eq. (5) can be factorized into three NSBL matrices, as follows (Fig. 3):

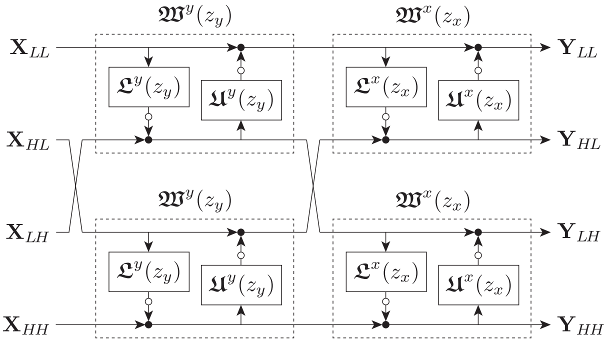

Fig. 2. 2D implementation of SBL (black and white circles mean adders and rounding operations, respectively).

$$\mathfrak{W}^{2d}(z_{2d}) = \mathfrak{W}^{2d}_2(z_{2d})\mathfrak{W}^{2d}_1(z_{2d})\mathfrak{W}^{2d}_0(z_{2d}) \tag{6}$$

where

$$\mathfrak{W}^{2d}_0(z_{2d}) = \begin{bmatrix} \mathbf{I}_{3M/2} & \mathbf{0} \\ \begin{bmatrix} \mathfrak{L}^{2d}(z_{2d}) & \mathfrak{L}^y(z_y) & \mathfrak{L}^x(z_x) \end{bmatrix} & \mathbf{I} \end{bmatrix}$$

$$\mathfrak{W}^{2d}_1(z_{2d}) = \begin{bmatrix} \mathbf{I} & \begin{bmatrix} \mathfrak{L}^x(z_x)^T & \mathfrak{L}^y(z_y)^T \end{bmatrix} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_M & \mathbf{0} \\ \mathbf{0} & \begin{bmatrix} \mathfrak{U}^y(z_y)^T & \mathfrak{U}^x(z_x)^T \end{bmatrix} & \mathbf{I} \end{bmatrix}^T$$

$$\mathfrak{W}^{2d}_2(z_{2d}) = \begin{bmatrix} \mathbf{I} & \begin{bmatrix} \mathfrak{U}^x(z_x) & \mathfrak{U}^y(z_y) & -\mathfrak{U}^{2d}(z_{2d}) \end{bmatrix} \\ \mathbf{0} & \mathbf{I}_{3M/2} \end{bmatrix}.$$

It is clear that the NSBL is an extension of the NSL in [10] because the NSBL with $M = 2$ in Eq. (6) is completely equivalent to the NSL in Eq. (4).

Proof: When a matrix $\mathfrak{T} = \mathbf{T}_{n-1} \cdots \mathbf{T}_0$ ($n \in \mathbb{N}$) is applied to a 2D input signal $\mathbf{x}$ in the horizontal and vertical directions, the output signal $\mathbf{y}$ is expressed as [11]

$$\mathbf{y} = \mathfrak{T}\mathbf{x}\mathfrak{T}^T = \mathbf{T}_{n-1} \cdots \mathbf{T}_0 \mathbf{x} \mathbf{T}^T_0 \cdots \mathbf{T}^T_{n-1}. \tag{7}$$

This Eq. (7) means that the 2D implementation of $\mathbf{T}_k$ is performed after that of $\mathbf{T}_{k-1}$ ($1 \le k \le n-1$), i.e., the two SBL matrices $\mathfrak{B}^w_L(z_w)$ and $\mathfrak{B}^w_U(z_w)$ in Eq. (5) can be operated separately. The resulting representation of $\mathfrak{W}^{2d}(z_{2d})$ is

$$\mathfrak{W}^{2d}(z_{2d}) = \mathfrak{B}^{2d}_U(z_{2d})\mathfrak{B}^{2d}_L(z_{2d}) \tag{8}$$

where

$$\begin{aligned} \mathfrak{B}^{2d}_L(z_{2d}) &= \begin{bmatrix} \mathfrak{B}^x_L(z_x) & \mathbf{0} \\ \mathbf{0} & \mathfrak{B}^x_L(z_x) \end{bmatrix} \mathbf{P} \begin{bmatrix} \mathfrak{B}^y_L(z_y) & \mathbf{0} \\ \mathbf{0} & \mathfrak{B}^y_L(z_y) \end{bmatrix} \mathbf{P} \\ &= \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathfrak{L}^x(z_x) & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathfrak{L}^y(z_y) & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathfrak{L}^{2d}(z_{2d}) & \mathfrak{L}^y(z_y) & \mathfrak{L}^x(z_x) & \mathbf{I} \end{bmatrix} \end{aligned}$$

$$\mathfrak{B}_U^{2d}(z_{2d}) = \begin{bmatrix} \mathfrak{B}_U^x(z_x) & \mathbf{0} \\ \mathbf{0} & \mathfrak{B}_U^x(z_x) \end{bmatrix} \mathbf{P} \begin{bmatrix} \mathfrak{B}_U^y(z_y) & \mathbf{0} \\ \mathbf{0} & \mathfrak{B}_U^y(z_y) \end{bmatrix} \mathbf{P}$$
$$= \begin{bmatrix} \mathbf{I} & \mathfrak{U}^x(z_x) & \mathfrak{U}^y(z_y) & \mathfrak{U}^{2d}(z_{2d}) \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathfrak{U}^y(z_y) \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathfrak{U}^x(z_x) \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}.$$

Since rounding operations will have inevitably be generated in each process of the matrices, as described in Section 2.1, we separate each of $\mathfrak{B}_L^{2d}(z_{2d})$ and $\mathfrak{B}_U^{2d}(z_{2d})$ into two NSBL matrices:

$$\mathfrak{B}_L^{2d}(z_{2d}) = \mathfrak{B}_{L1}^{2d}(z_{2d})\mathfrak{B}_{L0}^{2d}(z_{2d}) \quad (9)$$
$$\mathfrak{B}_U^{2d}(z_{2d}) = \mathfrak{B}_{U1}^{2d}(z_{2d})\mathfrak{B}_{U0}^{2d}(z_{2d}) \quad (10)$$

where

$$\mathfrak{B}_{L0}^{2d}(z_{2d}) = \begin{bmatrix} \mathbf{I}_{3M/2} & \mathbf{0} \\ [\mathfrak{L}^{2d}(z_{2d}) \quad \mathfrak{L}^y(z_y) \quad \mathfrak{L}^x(z_x)] & \mathbf{I} \end{bmatrix}$$
$$\mathfrak{B}_{L1}^{2d}(z_{2d}) = \begin{bmatrix} \mathbf{I} & [\mathfrak{L}^x(z_x)^T \quad \mathfrak{L}^y(z_y)^T \quad \mathbf{0}] \\ \mathbf{0} & \mathbf{I}_{3M/2} \end{bmatrix}^T$$
$$\mathfrak{B}_{U0}^{2d}(z_{2d}) = \begin{bmatrix} \mathbf{I}_{3M/2} & \mathbf{0} \\ [\mathbf{0} \quad \mathfrak{U}^y(z_y)^T \quad \mathfrak{U}^x(z_x)^T] & \mathbf{I} \end{bmatrix}^T$$
$$\mathfrak{B}_{U1}^{2d}(z_{2d}) = \begin{bmatrix} \mathbf{I} & [\mathfrak{U}^x(z_x) \quad \mathfrak{U}^y(z_y) \quad -\mathfrak{U}^{2d}(z_{2d})] \\ \mathbf{0} & \mathbf{I}_{3M/2} \end{bmatrix}.$$

Consequently, $\mathfrak{W}^{2d}(z_{2d})$ is expressed as

$$\mathfrak{W}^{2d}(z_{2d}) = \mathfrak{B}_{U1}^{2d}(z_{2d}) \underbrace{\mathfrak{B}_{U0}^{2d}(z_{2d})\mathfrak{B}_{L1}^{2d}(z_{2d})}_{\text{can be merged}} \mathfrak{B}_{L0}^{2d}(z_{2d})$$

from Eqs. (8)-(10). The resulting equation is completely the same as Eq. (6). □

3.2. Application to BOFBs

Here, we will apply the NSBL in Eq. (6) to the conventional SBL-BOFBs in Eq. (2). Let $\mathbf{E}^{2d}(z_{2d})$ be a 2D separable polyphase matrix based on a 1D separable polyphase matrix $\mathbf{E}(z)$ in Eq. (2). Since the 2D implementation of the separable block transform allows us to change the order in which the blocks are operated on, the polyphase matrix $\mathbf{E}^{2d}(z_{2d})$ can be expressed as

$$\mathbf{E}^{2d}(z_{2d}) = \mathbf{W}_K^{2d}(z_{2d}) \prod_{k=K-1}^{1} \{\mathbf{\Lambda}^{2d}(z_{2d})\mathbf{W}_k^{2d}(z_{2d})\}\mathbf{G}_0^{2d}$$

where

$$\mathbf{W}_k^{2d}(z_{2d}) = \begin{bmatrix} \mathbf{I} & [\hat{\mathbf{U}}_k^x(z_x) \quad \hat{\mathbf{U}}_k^y(z_y) \quad -\hat{\mathbf{U}}_k^{2d}(z_{2d})] \\ \mathbf{0} & \mathbf{I}_{3M/2} \end{bmatrix}$$
$$\times \begin{bmatrix} \mathbf{I} & [\hat{\mathbf{L}}_k^x(z_x)^T \quad \hat{\mathbf{L}}_k^y(z_y)^T] & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_M & \mathbf{0} \\ \mathbf{0} & [\hat{\mathbf{U}}_k^y(z_y)^T \quad \hat{\mathbf{U}}_k^x(z_x)^T] & \mathbf{I} \end{bmatrix}^T$$
$$\times \begin{bmatrix} \mathbf{I}_{3M/2} & \mathbf{0} \\ [\hat{\mathbf{L}}_k^{2d}(z_{2d}) \quad \hat{\mathbf{L}}_k^y(z_y) \quad \hat{\mathbf{L}}_k^x(z_x)] & \mathbf{I} \end{bmatrix}$$

$$\mathbf{W}_K^{2d}(z_{2d}) = \begin{bmatrix} \mathbf{I} & [\hat{\mathbf{L}}_K^x(z_x)^T \quad \hat{\mathbf{L}}_K^y(z_y)^T \quad \mathbf{0}] \\ \mathbf{0} & \mathbf{I}_{3M/2} \end{bmatrix}^T$$
$$\times \begin{bmatrix} \mathbf{I}_{3M/2} & \mathbf{0} \\ [\hat{\mathbf{L}}_K^{2d}(z_{2d}) \quad \hat{\mathbf{L}}_K^y(z_y) \quad \hat{\mathbf{L}}_K^x(z_x)] & \mathbf{I} \end{bmatrix},$$

and $\mathbf{\Lambda}^{2d}(z_{2d}) = \mathrm{diag}\{\mathbf{I}, z_x^{-1}\mathbf{I}, z_y^{-1}\mathbf{I}, z_{2d}^{-1}\mathbf{I}\}$. As is done in [9], we use the single-row elementary reversible matrix (SERM) presented in [6] for each initial block $\mathbf{G}_0^w$, where any other SL factorization can be applied to $\mathbf{G}_0^w$.

Table 2. Lossless image coding results (LBR [bpp]): (A) 4×8 L-LT [4], (B) 5/3-NSL-DWT [10], (C-D) 8×16 and 8×24 SBL-BOFBs [9], and (E-F) 8×16 and 8×24 NSBL-BOFBs.

Images	(A)	(B)	(C)	(D)	(E)	(F)
Barbara	4.81	4.86	4.79	4.78	4.76	**4.75**
Boat	5.13	5.09	5.09	5.11	**5.08**	5.09
Elaine	5.17	5.11	5.06	5.05	5.05	**5.03**
Finger	5.71	5.83	5.66	5.65	5.65	**5.64**

Table 3. Lossy image coding results (PSNR [dB]): (A) 4×8 L-LT [4], (B) 9/7-NSL-DWT [10], (C-D) 8×16 and 8×24 SBL-BOFBs [9], and (E-F) 8×16 and 8×24 NSBL-BOFBs.

[bpp]	(A)	(B)	(C)	(D)	(E)	(F)
Barbara						
0.25	26.85	27.24	28.04	28.64	28.05	**28.65**
0.50	30.43	30.46	31.63	32.18	31.67	**32.20**
1.00	35.05	34.85	35.88	36.26	35.94	**36.38**
Boat						
0.25	27.62	28.45	28.26	28.62	28.25	**28.63**
0.50	30.87	31.38	31.35	31.61	31.36	**31.63**
1.00	34.31	34.48	34.66	34.85	34.70	**34.91**
Elaine						
0.25	30.81	31.50	31.34	31.56	31.36	**31.59**
0.50	32.47	32.93	33.07	32.98	33.10	**33.08**
1.00	34.22	34.61	35.06	35.24	35.11	**35.36**
Finger						
0.25	22.96	23.49	23.52	**23.86**	23.51	**23.86**
0.50	25.56	25.98	26.43	26.93	26.43	**26.95**
1.00	29.01	29.07	30.06	30.78	30.07	**30.81**

4. EXPERIMENTAL RESULTS

By following the method presented in [9], 8×16 and 8×24 BOFBs with order-1 building blocks were designed. They were compared in terms of the lossless bitrate (LBR) [bpp] in lossless image coding and the peak signal-to-noise ratio (PSNR) [dB] in lossy image coding. To evaluate transform performance fairly, we employed 3, 6, and 2-level decompositions, respectively, on the L-LT [4], NSL-DWTs without adaptive directionalities [10], and eight-channel BOFBs. The SBL-BOFBs had the same transfer function as the proposed

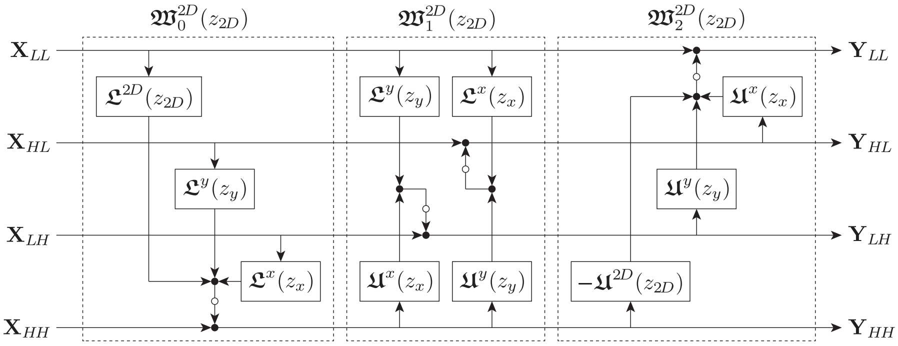

Fig. 3. NSBL (black and white circles mean adders and rounding operations, respectively).

FBs. The image set included several 512×512 grayscale test images, such as *Barbara*. A quadtree-based embedded image coder EZW-IP [12] was used to encode the transformed images. A periodic extension was used in the image boundary processing of the BOFBs, whereas the extensions used in JPEG XR and JPEG 2000 were used as the respective boundary processing of the L-LT and NSL-DWTs.

Tables 2 and 3 show lossless and lossy image coding results. Although NSL-DWTs sometimes performed better on images with many low frequency components, overall, the NSBL-BOFBs outperformed the conventional methods. These results are considered to be due to the merging (reducing) of many rounding operations in the NSBL-BOFBs. Comparing Figs. 2 and 3, it is clear that the number of rounding operations of the NSBL is the almost half that of the SBL.

5. CONCLUSION

We devised a NSBL and applied it to M-channel BOFBs in lossy-to-lossless image coding. The NSBL is easily formulated from the SBL and NSL methods and can be regarded as an extension of the NSL because it is completely equivalent to a NSL when $M = 2$. A lossy-to-lossless image coding experiment confirmed the improvements that could be had with NSBL.

REFERENCES

[1] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Englewood Cliffs, NJ: Prentice Hall, 1992.

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

[3] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Appl.*, vol. 4, no. 3, pp. 247–269, 1998.

[4] 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.

[5] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Appl. Comput. Harmon. Anal.*, vol. 3, no. 2, pp. 186–200, Apr. 1996.

[6] P. Hao and Q. Shi, "Matrix factorizations for reversible integer mapping," *IEEE Trans. Signal Process.*, vol. 49, no. 10, pp. 2314–2324, Oct. 2001.

[7] Y. J. Chen and K. S. Amaratunga, "M-channel lifting factorization of perfect reconstruction filter banks and reversible M-band wavelet transforms," *IEEE Trans. Circuits Syst. II*, vol. 50, no. 12, pp. 963–976, Dec. 2003.

[8] 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, Sep. 2010.

[9] 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.

[10] T. Yoshida, T. Suzuki, S. Kyochi, and M. Ikehara, "Two dimensional non-separable adaptive directional lifting structure of discrete wavelet transform," *IEICE Trans. Fundamentals.*, vol. E94-A, no. 10, pp. 1920–1927, Oct. 2011.

[11] T. Suzuki, S. Kyochi, Y. Tanaka, M. Ikehara, and H. Aso, "Multiplierless lifting based FFT via fast Hartley transform," in *Proc. of ICASSP'13*, Vancouver, Canada, May 2013, pp. 5603–5607.

[12] Z. Liu and L. J. Karam, "An efficient embedded zerotree wavelet image codec based on intraband partitioning," in *Proc. of ICIP'00*, Vancouver, British Columbia, Canada, Sep. 2000.