Two-Dimensional Non-Separable Block-Lifting Structure and Its Application to M-Channel Perfect Reconstruction Filter Banks for Lossy-to-Lossless Image Coding

Taizo Suzuki, *Member, IEEE* and Hiroyuki Kudo, *Member, IEEE*

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

***Index Terms*—Lossy-to-lossless image coding, perfect reconstruction filter bank (PRFB), two-dimensional non-separable block-lifting structure (2D-NSBL)**

I. Introduction

THE amount of video being sent over communications networks has been steadily increasing as a result of developments in multimedia devices and communication tools. Filter banks (FBs) [1] have been widely researched as a way to efficiently compress such signals. The polyphase matrices of M-channel ($M = 2^n$, $n \in \mathbb{N}$) FBs shown in Fig. 1 are presented as

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

where $H_i(z)$, $F_i(z)$, 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). When $\mathbf{R}(z) = \mathbf{E}^T(z^{-1})$, the FBs are called paraunitary FBs (PUFBs) which are special classes of PRFBs. On the other hand, PRFBs that are not PUFBs are commonly called biorthogonal FBs (BOFBs). In particular, the JPEG series [2–4] and H.26x series [5], [6] of global standards use various classes of PRFBs, including the discrete cosine/sine transform (DCT and DST) [7], discrete wavelet transform (DWT) [8], and hierarchical lapped transform (HLT) [9]. However, there is a growing need for better FBs in order to alleviate the burden on servers and free up communication bandwidth.

T. Suzuki and H. Kudo are with the Faculty of Engineering, Information and Systems, University of Tsukuba, Tsukuba, Ibaraki, 305-8573 Japan (e-mail: {taizo, kudo}@cs.tsukuba.ac.jp).

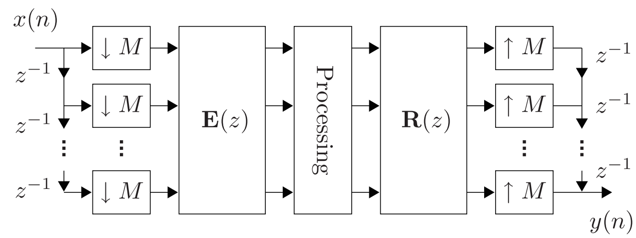

Fig. 1. Polyphase structure of M-channel FB.

TABLE I
Classification of lifting structures.

	1D-SL [10–12]	1D-SBL [13]	2D-NSL [14], [15]	2D-NSBL Prop.
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 [10–12] with which to achieve integer-to-integer transforms, and this structure has been applied to many FBs [16–23]. Although JPEG XR [4] has scalability ranging from lossless to lossy as a result of using a lifting-based HLT, its coding performance is not sufficient especially for images with high-frequency components (texture).

The one-dimensional separable block-lifting structure (1D-SBL) of BOFB was proposed in [13] for the purpose of designing lifting-based FBs with higher coding performance. Usually, the design parameters and structure of lifting-based FBs are constrained when factorizing the original FB into lifting structures, whereas the 1D-SBL-based BOFBs presented in [13] do not constrain them except in the initial block.[1] The 1D-SBL is better at lossy-to-lossless image coding because it uses fewer rounding operations in comparison with the

[1] The determinant of the initial block is constrained to be an integer value.

standard 1D separable lifting structure (1D-SL). Furthermore, the two-dimensional non-separable lifting structure (2D-NSL) for DWTs proposed in [14], [15] performs even better at coding because it uses fewer rounding operations than the 1D-SL.

Here, we propose a 2D non-separable block-lifting structure (2D-NSBL) that is easily formulated from the 1D-SBL and 2D-NSL methods. The 2D-NSBL can be regarded as an extension of the 2D-NSL because a 2D-NSBL with $M = 2$ is completely equivalent to a 2D-NSL. We apply the 2D-NSBL to M-channel PRFBs and show that the PRFBs perform better at lossy-to-lossless coding than the conventional 1D-SBL-based PRFBs do because their rounding errors are reduced by merging many rounding operations.

The remaining part of this paper is organized as follows. We review and define the block-lifting structure, 1D-SBL-based PRFBs, and 2D-NSL-based DWTs in Section II. Section III presents the derivation of the 2D-NSBL and its application to M-channel PRFBs. Design examples, lossy-to-lossless image coding simulations, and comparisons with the conventional PRFBs are shown in Section IV. Section V concludes this paper.

Notations: A classification of lifting structures is shown in Table. I. $\mathbf{I}_m$, $\mathbf{0}$, $\det(\cdot)$, and $\mathrm{round}(\cdot)$ denote an $m \times m$ identity matrix, a null matrix, determinant of a matrix, and a rounding operation, 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 ($w = x$ or y), and horizontally and vertically, respectively. For example,

$$\mathbf{T}^x\mathbf{x} = \mathbf{x}\mathbf{T}^T, \ \mathbf{T}^y\mathbf{x} = \mathbf{T}\mathbf{x}, \text{ and } \mathbf{T}^{2d}\mathbf{x} = \mathbf{T}\mathbf{x}\mathbf{T}^T,$$

where $\mathbf{T}$ and $\mathbf{x}$ are a transform matrix and a 2D input signal, respectively.

II. Review and Definitions

A. Block-Lifting Structure

A lifting structure [10–12] is a transform that map integers to integers by implementing rounding operation in each lifting step; i.e., it is a means of lossy-to-lossless image coding. The elementary matrices are identity matrices with one single nonzero off-diagonal element. However, an FB with too many lifting steps cannot perform good coding because it generates a rounding error in each lifting step.

We proposed the block-lifting structure in [13], which is a special class of standard lifting structure (Fig. 2). It is good for lossy-to-lossless image coding because it reduces the rounding error by merging many rounding operations. In Fig. 2, 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:

$$\left.\begin{aligned} \mathbf{y}_j &= \mathbf{x}_j + \mathrm{round}(\mathfrak{L}(z)\mathbf{x}_i) \\ \mathbf{y}_i &= \mathbf{x}_i + \mathrm{round}(\mathfrak{U}(z)\mathbf{y}_j) \\ \mathbf{z}_i &= \mathbf{y}_i - \mathrm{round}(\mathfrak{U}(z)\mathbf{y}_j) = \mathbf{x}_i \\ \mathbf{z}_j &= \mathbf{y}_j - \mathrm{round}(\mathfrak{L}(z)\mathbf{x}_i) = \mathbf{x}_j \end{aligned}\right\} \quad \text{(Case A)}$$

$$\left.\begin{aligned} \mathbf{y}_i &= \mathbf{x}_i + \mathrm{round}(\mathfrak{U}(z)\mathbf{x}_j) \\ \mathbf{y}_j &= \mathbf{x}_j + \mathrm{round}(\mathfrak{L}(z)\mathbf{y}_i) \\ \mathbf{z}_j &= \mathbf{y}_j - \mathrm{round}(\mathfrak{L}(z)\mathbf{y}_i) = \mathbf{x}_j \\ \mathbf{z}_i &= \mathbf{y}_i - \mathrm{round}(\mathfrak{U}(z)\mathbf{x}_j) = \mathbf{x}_i \end{aligned}\right\} \quad \text{(Case B)}.$$

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

$$\mathfrak{W}(z) = \begin{cases}\mathfrak{B}_U(z)\mathfrak{B}_L(z) & \text{(Case A)}\\ \mathfrak{B}_L(z)\mathfrak{B}_U(z) & \text{(Case B)}\end{cases}$$

$$\mathfrak{W}^{-1}(z) = \begin{cases}\mathfrak{B}_L^{-1}(z)\mathfrak{B}_U^{-1}(z) & \text{(Case A)}\\ \mathfrak{B}_U^{-1}(z)\mathfrak{B}_L^{-1}(z) & \text{(Case B)}\end{cases}$$

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

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

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 “1D-SBL” to distinguish it from the “2D-NSBL” proposed in this paper. When $M = 2$, they will also be called “1D-SL” and “2D-NSL”.

B. 1D-SBL-based PRFBs

The polyphase matrix $\mathbf{E}(z)$ of an M-channel PRFB of filter length MK ($K \in \mathbb{N}, K \geq 2$) is expressed as [24]

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

where the initial block $\mathbf{G}_0$ is an $M \times M$ nonsingular matrix and the building block $\mathbf{E}_k(z)$ is expressed by

$$\mathbf{E}_k(z) = \mathbf{I} - \boldsymbol{\mathcal{U}}_k\boldsymbol{\mathcal{V}}_k^T + z^{-1}\boldsymbol{\mathcal{U}}_k\boldsymbol{\mathcal{V}}_k^T.$$

The $M \times \gamma_k$ parameter matrices $\boldsymbol{\mathcal{U}}_k$ and $\boldsymbol{\mathcal{V}}_k$ satisfy

$$\boldsymbol{\mathcal{V}}_k^T\boldsymbol{\mathcal{U}}_k = \begin{bmatrix}1 & \times & \cdots & \times\\ 0 & 1 & \ddots & \vdots\\ \vdots & \ddots & \ddots & \times\\ 0 & \cdots & 0 & 1\end{bmatrix}_{\gamma_k \times \gamma_k} \triangleq \boldsymbol{\mathcal{W}}_k,$$

where $\times$ indicates possibly nonzero elements and γ_k is a McMillan degree ($\gamma_k \in \mathbb{N}$, $1 \leq \gamma_k \leq M-1$). $\boldsymbol{\mathcal{W}}_k = \mathbf{I}$ when the filter lengths in analysis and synthesis banks are equal. A synthesis polyphase matrix $\mathbf{R}(z)$ is defined as one that has the PR property $\mathbf{R}(z)\mathbf{E}(z) = \mathbf{I}$. Moreover, $\boldsymbol{\mathcal{U}}_k$ and $\boldsymbol{\mathcal{V}}_k$ are defined as $\boldsymbol{\mathcal{U}}_k = [\mathbf{u}_{k0}^T, \mathbf{u}_{k1}^T]^T$ and $\boldsymbol{\mathcal{V}}_k = [\mathbf{v}_{k0}^T, \mathbf{v}_{k1}^T]^T$,

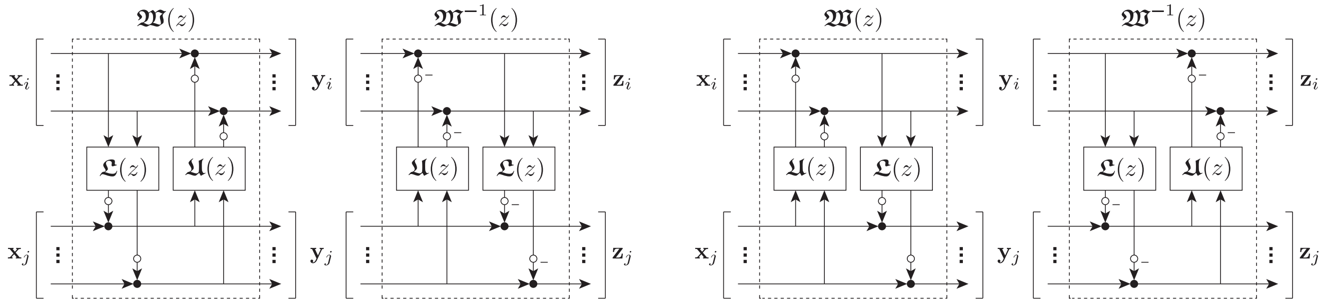

Fig. 2. Block-lifting structures. Black and white circles mean adders and rounding operations, respectively: (left) Case A and (right) Case B.

where $\mathbf{u}_{k0}$ and $\mathbf{v}_{k0}$ are $(M-\gamma_k)\times\gamma_k$ matrices and $\mathbf{u}_{k1}$ and $\mathbf{v}_{k1}$ are $\gamma_k\times\gamma_k$ square matrices. When $\boldsymbol{\mathcal{V}}_k=\boldsymbol{\mathcal{U}}_k$, the PRFB has the paraunitary property, i.e., PUFB. In this paper, we fix $\boldsymbol{\mathcal{W}}_k=\mathbf{I}$ and $\gamma_k=M/2$.

We factorized the PRFBs into the 1D-SBL in [13]. The 1D-SBL-based PRFBs represent $\mathbf{E}_k(z)$ in Eq. (1) as

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

where $\mathbf{L}_k=\mathbf{v}_{k1}^{-T}\mathbf{v}_{k0}^T$, $\mathbf{U}_k=\mathbf{u}_{k0}\mathbf{v}_{k1}^T$, and

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

In addition, $\det(\mathbf{G}_0)$ is constrained to be $\det(\mathbf{G}_0)=\pm n$ ($n\in\mathbb{N}$) for the purpose of making a lifting factorization. If paraunitariness is not required, $\mathbf{L}_k$ and $\mathbf{U}_k$ can be arbitrary $M/2\times M/2$ matrices. To improve coding performance, Eq. (1) can be rewritten as (Fig. 3)

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

where

$$\mathbf{W}_k(z)=\begin{bmatrix}\mathbf{I} & \widehat{\mathbf{U}}_k(z)\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \widehat{\mathbf{L}}_k(z) & \mathbf{I}\end{bmatrix}$$

$$\widehat{\mathbf{U}}_k(z)=\begin{cases}\mathbf{0} & (k=K)\\ (z^{-1}-1)\mathbf{U}_k & (\text{otherwise})\end{cases}$$

$$\widehat{\mathbf{L}}_k(z)=\begin{cases}\mathbf{L}_1 & (k=1)\\ -\mathbf{L}_{K-1} & (k=K)\\ \mathbf{L}_k-\mathbf{L}_{k-1} & (\text{otherwise})\end{cases}.$$

In comparison with the 1D-SBL-based PRFBs in Eq. (1), the 1D-SBL-based PRFBs are more effective at lossy-to-lossless image coding because they reduce the rounding error by merging more rounding operations.

C. 2D-NSL-based DWTs

1D-SL-based 9/7-tap and 5/3-tap DWTs (9/7-DWT and 5/3-DWT) [8] are used in the JPEG 2000 [3] lossy and lossless modes, respectively. Let $\mathbf{e}(z)$ be a polyphase matrix of 1D-SL-based DWTs, expressed as

$$\mathbf{e}(z)=\begin{bmatrix}s & 0\\ 0 & s^{-1}\end{bmatrix}\prod_{k=N-1}^{0}\mathbf{w}_k(z),$$

where

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

For example, 5/3-DWT has $N=1$, $s=1$, $u_0(z)=(1+z)/4$ and $l_0(z)=-(1+z^{-1})/2$. If an image is transformed by the 2D-NSL-based DWT polyphase matrix $\mathbf{e}_k^{2d}(z_{2d})$, one can write

$$\begin{aligned}&\begin{bmatrix}Y_{LL}^T & Y_{HL}^T & Y_{LH}^T & Y_{HH}^T\end{bmatrix}^T\\ &\qquad=\mathbf{e}_k^{2d}(z_{2d})\begin{bmatrix}X_{LL}^T & X_{HL}^T & X_{LH}^T & X_{HH}^T\end{bmatrix}^T,\end{aligned}$$

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

$$\mathbf{e}_k^{2d}(z_{2d})=\begin{bmatrix}s^2 & \mathbf{0} & 0\\ \mathbf{0} & \mathbf{I}_2 & \mathbf{0}\\ 0 & \mathbf{0} & s^{-2}\end{bmatrix}\prod_{k=N-1}^{0}\mathbf{w}_k^{2d}(z_{2d}).$$

$\mathbf{w}_k^{2d}(z_{2d})$ in $\mathbf{e}^{2d}(z_{2d})$ is represented as [14], [15]

$$\begin{aligned}\mathbf{w}_k^{2d}(z_{2d})=&\begin{bmatrix}1 & \begin{bmatrix}u_k^x(z_x) & u_k^y(z_y) & -u_k^{2d}(z_{2d})\end{bmatrix}\\ 0 & \mathbf{I}_3\end{bmatrix}\\ &\cdot\begin{bmatrix}1 & \mathbf{0} & 0\\ \begin{bmatrix}l_k^x(z_x)\\ l_k^y(z_y)\end{bmatrix} & \mathbf{I}_2 & \begin{bmatrix}u_k^y(z_y)\\ u_k^x(z_x)\end{bmatrix}\\ 0 & \mathbf{0} & 1\end{bmatrix}\\ &\cdot\begin{bmatrix}\mathbf{I}_3 & 0\\ \begin{bmatrix}l_k^{2d}(z_{2d}) & l_k^y(z_y) & l_k^x(z_x)\end{bmatrix} & 1\end{bmatrix}.\end{aligned} \tag{3}$$

The 2D-NSL is also more effective at lossy-to-lossless image coding than the 1D-SL is because it uses fewer rounding operations.

III. 2D-NSBL-BASED PRFBS

A. Derivation of 2D-NSBL

We introduce 2D-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. 2 as follows:

$$\begin{aligned}&\begin{bmatrix}\mathbf{Y}_{LL}^T & \mathbf{Y}_{HL}^T & \mathbf{Y}_{LH}^T & \mathbf{Y}_{HH}^T\end{bmatrix}^T\\ &\qquad=\mathfrak{W}^{2d}(z_{2d})\begin{bmatrix}\mathbf{X}_{LL}^T & \mathbf{X}_{HL}^T & \mathbf{X}_{LH}^T & \mathbf{X}_{HH}^T\end{bmatrix}^T,\end{aligned} \tag{4}$$

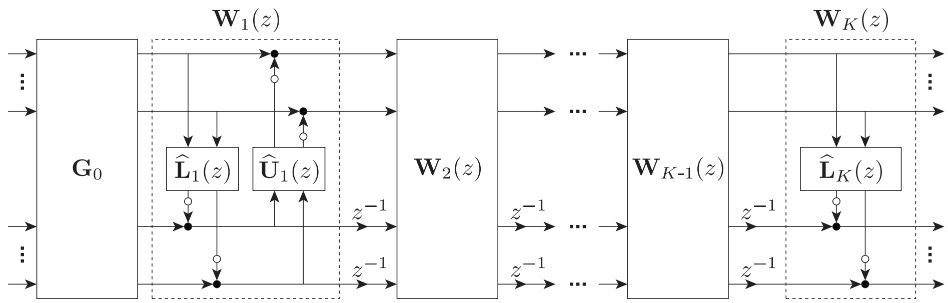

Fig. 3. 1D-SBL-based PRFB. Black and white circles mean adders and rounding operations, respectively.

where

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

$$\mathfrak{W}^w(z_w) = \begin{cases}\mathfrak{B}_U^w(z_w)\mathfrak{B}_L^w(z_w) & \text{(Case A)}\\ \mathfrak{B}_L^w(z_w)\mathfrak{B}_U^w(z_w) & \text{(Case B)}\end{cases}$$

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

$\mathbf{X}_{LL}$, $\mathbf{X}_{HL}$, $\mathbf{X}_{LH}$, 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 (Fig. 4). $\mathfrak{W}^{2d}(z_{2d})$ in Eq. (4) can be factorized into three 2D-NSBL matrices, as follows (Fig. 5):

$$\mathfrak{W}^{2d}(z_{2d}) = \mathfrak{W}_2^{2d}(z_{2d})\mathfrak{W}_1^{2d}(z_{2d})\mathfrak{W}_0^{2d}(z_{2d}), \tag{5}$$

where

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

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

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

It is clear that the 2D-NSBL is an extension of the 2D-NSL in [14], [15] because the 2D-NSBL with $M = 2$ in Eq. (5) (Case A) is completely equivalent to the 2D-NSL in Eq. (3). Furthermore, a 2D-NSBL with the McMillan degree $\gamma_k \neq M/2$ can be easily obtained.

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 [25]

$$\mathbf{y} = \mathfrak{T}\mathbf{x}\mathfrak{T}^T = \mathbf{T}_{n-1}\cdots\mathbf{T}_0\mathbf{x}\mathbf{T}_0^T\cdots\mathbf{T}_{n-1}^T. \tag{6}$$

This Eq. (6) means that the 2D implementation of $\mathbf{T}_k$ is performed after that of $\mathbf{T}_{k-1}$ ($1 \leq k \leq n-1$), i.e., the two block-lifting matrices $\mathfrak{B}_L^w(z_w)$ and $\mathfrak{B}_U^w(z_w)$ in Eq. (4) can be operated separately. The resulting representation of $\mathfrak{W}^{2d}(z_{2d})$ is

$$\mathfrak{W}^{2d}(z_{2d}) = \begin{cases}\mathfrak{B}_U^{2d}(z_{2d})\mathfrak{B}_L^{2d}(z_{2d}) & \text{(Case A)}\\ \mathfrak{B}_L^{2d}(z_{2d})\mathfrak{B}_U^{2d}(z_{2d}) & \text{(Case B)},\end{cases} \tag{7}$$

where

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

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

$$\mathfrak{B}_L^{2d}(z_{2d}) = \begin{cases}\mathfrak{B}_{L1}^{2d}(z_{2d})\mathfrak{B}_{L0}^{2d}(z_{2d}) & \text{(Case A)}\\ \mathfrak{B}_{L2}^{2d}(z_{2d})\mathfrak{B}_{L1}^{2d}(z_{2d}) & \text{(Case B)}\end{cases} \tag{8}$$

$$\mathfrak{B}_U^{2d}(z_{2d}) = \begin{cases}\mathfrak{B}_{U2}^{2d}(z_{2d})\mathfrak{B}_{U1}^{2d}(z_{2d}) & \text{(Case A)}\\ \mathfrak{B}_{U1}^{2d}(z_{2d})\mathfrak{B}_{U0}^{2d}(z_{2d}) & \text{(Case B)},\end{cases} \tag{9}$$

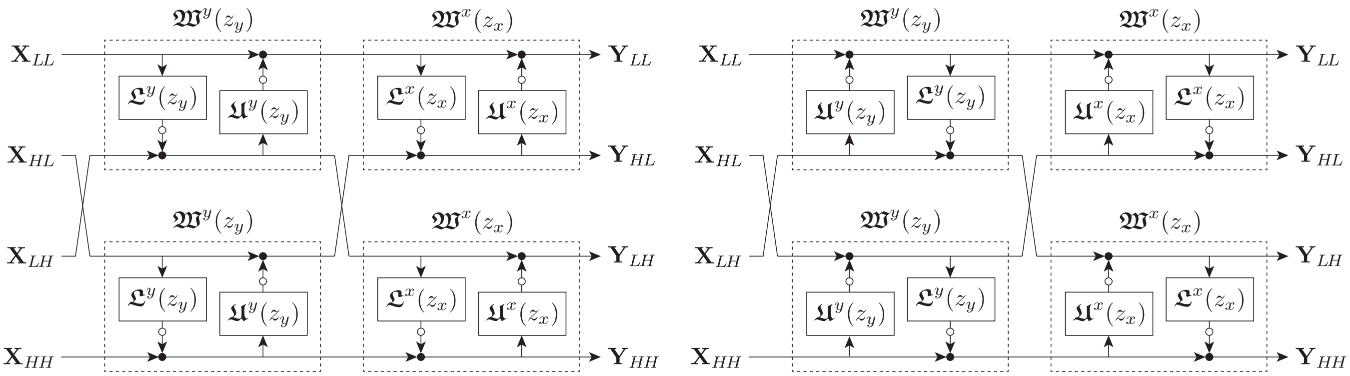

Fig. 4. 2D implementation of 1D-SBL. Black and white circles mean adders and rounding operations, respectively: (left) Case A and (right) Case B.

where

$$\mathfrak{B}_{L0}^{2d}(z_{2d}) = \begin{bmatrix} \mathbf{I}_{3M/2} & \mathbf{0} \\ \begin{bmatrix}\mathfrak{L}^{2d}(z_{2d}) & \mathfrak{L}^y(z_y) & \mathfrak{L}^x(z_x)\end{bmatrix} & \mathbf{I} \end{bmatrix}$$

$$\mathfrak{B}_{L1}^{2d}(z_{2d}) = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \begin{bmatrix}\mathfrak{L}^x(z_x) \\ \mathfrak{L}^y(z_y)\end{bmatrix} & \mathbf{I}_M & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}$$

$$\mathfrak{B}_{L2}^{2d}(z_{2d}) = \begin{bmatrix} \mathbf{I}_{3M/2} & \mathbf{0} \\ \begin{bmatrix}-\mathfrak{L}^{2d}(z_{2d}) & \mathfrak{L}^y(z_y) & \mathfrak{L}^x(z_x)\end{bmatrix} & \mathbf{I} \end{bmatrix}$$

$$\mathfrak{B}_{U0}^{2d}(z_{2d}) = \begin{bmatrix} \mathbf{I} & \begin{bmatrix}\mathfrak{U}^x(z_x) & \mathfrak{U}^y(z_y) & \mathfrak{U}^{2d}(z_{2d})\end{bmatrix} \\ \mathbf{0} & \mathbf{I}_{3M/2} \end{bmatrix}$$

$$\mathfrak{B}_{U1}^{2d}(z_{2d}) = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_M & \begin{bmatrix}\mathfrak{U}^y(z_y) \\ \mathfrak{U}^x(z_x)\end{bmatrix} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}$$

$$\mathfrak{B}_{U2}^{2d}(z_{2d}) = \begin{bmatrix} \mathbf{I} & \begin{bmatrix}\mathfrak{U}^x(z_x) & \mathfrak{U}^y(z_y) & -\mathfrak{U}^{2d}(z_{2d})\end{bmatrix} \\ \mathbf{0} & \mathbf{I}_{3M/2} \end{bmatrix}.$$

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

$$\mathfrak{W}^{2d}(z_{2d}) = \begin{cases} \mathfrak{B}_{U2}^{2d}(z_{2d})\mathfrak{B}_{UL}^{2d}(z_{2d})\mathfrak{B}_{L0}^{2d}(z_{2d}) & \text{(Case A)} \\ \mathfrak{B}_{L2}^{2d}(z_{2d})\mathfrak{B}_{LU}^{2d}(z_{2d})\mathfrak{B}_{U0}^{2d}(z_{2d}) & \text{(Case B)}, \end{cases}$$

where

$$\mathfrak{B}_{UL}^{2d}(z_{2d}) = \mathfrak{B}_{LU}^{2d}(z_{2d}) = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \begin{bmatrix}\mathfrak{L}^x(z_x) \\ \mathfrak{L}^y(z_y)\end{bmatrix} & \mathbf{I}_M & \begin{bmatrix}\mathfrak{U}^y(z_y) \\ \mathfrak{U}^x(z_x)\end{bmatrix} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}$$

from Eqs. (7)-(9). The resulting equation is completely the same as Eq. (5). □

B. Application to PRFBs

Here, we will apply the 2D-NSBL in Eq. (5) to the conventional 1D-SBL-based PRFBs 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

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

where

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

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

$$\mathbf{G}_0^{2d} = \begin{bmatrix} \mathbf{G}_0^x & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_0^x \end{bmatrix} \mathbf{P} \begin{bmatrix} \mathbf{G}_0^y & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_0^y \end{bmatrix} \mathbf{P}.$$

Applying Case A of the proposed 2D-NSBL in Eq. (5) to $\mathbf{W}_k^{2d}(z_{2d})$ yields

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

For $\mathbf{G}_0^{2d}$, a 1D-SL factorization is used. As is done in [13], we use the single-row elementary reversible matrix (SERM) presented in [16] for each initial block $\mathbf{G}_0^w$, where any other 1D-SL factorization can be applied to $\mathbf{G}_0^w$.

IV. Experimental Results

A. Filter Design

By following the method presented in [13], 8×16 and 8×24 BOFBs with order-1 building blocks were designed by using the cost function Φ and `fminunc.m` in `Optimization ToolBox` of `MATLAB`. Φ was a weighted linear combination of the coding gain C_{CG} and the stopband attenuation values of analysis and synthesis filters C_{SAa} and C_{SAs}:

$$\Phi = (w_1 C_{SAa} + w_2 C_{SAs}) - w_3 C_{CG},$$

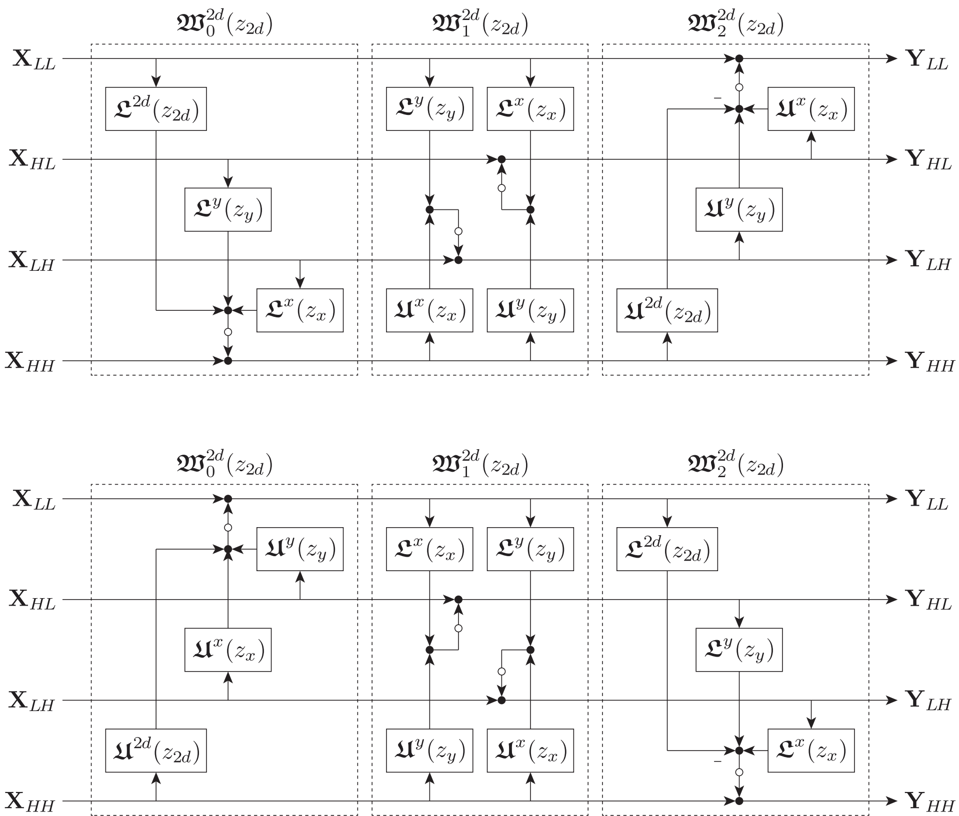

Fig. 5. 2D-NSBL. Black and white circles mean adders and rounding operations, respectively: (top) Case A and (bottom) Case B.

TABLE II
LOSSLESS IMAGE CODING RESULTS (LBR [BPP]).

	Conventional Methods					Proposed Method	
Test	Not Lifting	1D-NSL		1D-SBL [13]		2D-NSBL	
Images	8×8 DCT [26]	4×8 HLT [9]	5/3-DWT [14], [15]	8×16 BOFB	8×24 BOFB	8×16 BOFB	8×24 BOFB
Barbara	—	4.81	4.86	4.79	4.76	4.76	**4.75**
Boat	—	5.13	5.09	5.09	5.10	**5.08**	5.09
Finger	—	5.71	5.83	5.66	5.65	5.65	**5.64**
Grass	—	**6.05**	6.06	**6.05**	**6.05**	**6.05**	**6.05**
Lena	—	4.61	**4.48**	4.63	4.62	4.61	4.61
Pepper	—	4.96	**4.85**	4.93	4.92	4.92	4.92
Bridge	—	**4.63**	4.65	4.65	4.67	**4.63**	**4.63**
Deer	—	4.77	4.76	4.76	4.76	**4.74**	**4.74**
Arri	—	11.28	11.40	**11.27**	**11.27**	**11.27**	**11.27**
Face	—	10.33	10.37	**10.28**	**10.28**	**10.28**	**10.28**

where w_ks are weighted coefficients. C_{CG}, C_{SAa}, and C_{SAs} are

$$C_{CG} = 10\log_{10}\frac{\sigma_x^2}{\prod_{k=0}^{M-1}\sigma_{x_i}^2 \parallel f_i \parallel^2}$$

$$C_{SAa} = \sum_{k=0}^{M-1}\int_{\omega\in\Omega_i} W_i^a|H_i(e^{j\omega})|^2 d\omega$$

$$C_{SAs} = \sum_{k=0}^{M-1}\int_{\omega\in\Omega_i} W_i^s|F_i(e^{j\omega})|^2 d\omega,$$

where σ_x^2, $\sigma_{x_i}^2$, $\parallel f_i \parallel^2$, W_i^a, W_i^s, and Ω_i are the variance of the input signal, the variance of the i-th subbands, the norm of the i-th synthesis filter, weighting functions for the stopband attenuation of the analysis, synthesis bank, and the stopband region of $H_i(z)$ and $F_i(z)$, respectively. The input signal $x(n)$ is the AR(1) process with an intersample autocorrelation coefficient $\rho = 0.95$ in common use.

B. Lossy-to-Lossless Image Coding

The resulting BOFBs were implemented with a rounding operation at each lifting step and compared in terms of the lossless bitrate (LBR) [bpp] in lossless image coding:

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

TABLE III
LOSSY IMAGE CODING RESULTS (PSNR [DB]).

Test Images	Bitrate [bpp]	Conventional Methods: Not Lifting					Proposed Method: 2D-NSBL	
		Not Lifting	1D-NSL		1D-SBL [13]		2D-NSBL	
		8×8 DCT [26]	4×8 HLT [9]	9/7-DWT [14], [15]	8×16 BOFB	8×24 BOFB	8×16 BOFB	8×24 BOFB
Barbara	0.25	26.96	26.85	27.24	28.04	28.64	28.05	**28.65**
	0.50	30.40	30.43	30.46	31.63	32.18	31.67	**32.20**
	1.00	34.98	35.05	34.85	35.88	36.26	35.94	**36.38**
Boat	0.25	27.85	27.62	28.45	28.26	28.62	28.25	**28.63**
	0.50	30.87	30.87	31.38	31.35	31.61	31.36	**31.63**
	1.00	34.39	34.31	34.48	34.66	34.85	34.70	**34.91**
Finger	0.25	22.78	22.96	23.49	23.52	**23.86**	23.51	**23.86**
	0.50	25.42	25.56	25.98	26.43	26.93	26.43	**26.95**
	1.00	29.17	29.01	29.07	30.06	30.78	30.07	**30.81**
Grass	0.25	24.00	23.99	24.35	24.27	**24.50**	24.28	**24.50**
	0.50	25.94	25.86	26.09	26.30	26.61	26.30	**26.62**
	1.00	28.70	28.68	28.68	29.08	29.42	29.10	**29.44**
Lena	0.25	30.55	31.65	**32.50**	32.20	31.85	32.20	31.86
	0.50	34.43	35.03	35.49	35.41	35.65	35.44	**35.73**
	1.00	38.87	38.65	38.63	38.68	38.76	38.78	**38.95**
Pepper	0.25	29.84	31.20	**31.90**	31.45	31.08	31.45	31.10
	0.50	32.83	33.95	**34.45**	33.97	33.70	34.01	33.76
	1.00	35.75	35.59	36.08	35.92	36.01	35.97	**36.11**
Bridge	0.25	31.11	31.14	31.79	31.93	32.19	31.94	**32.22**
	0.50	33.76	34.06	34.25	34.57	34.54	34.60	**34.63**
	1.00	36.64	37.35	36.83	37.24	37.27	37.38	**37.42**
Deer	0.25	**34.14**	33.96	34.10	33.88	34.02	33.90	34.06
	0.50	35.10	34.83	34.88	34.99	35.06	35.03	**35.13**
	1.00	**37.45**	36.97	36.83	37.11	37.04	37.20	37.27
Arri	0.25	31.92	33.22	33.28	33.63	**34.14**	33.63	**34.14**
	0.50	36.82	36.75	37.30	38.26	**38.98**	38.26	**38.98**
	1.00	41.80	42.28	41.97	43.87	**43.93**	43.87	**43.93**
Face	0.25	45.01	45.49	45.96	46.27	**46.67**	46.27	**46.67**
	0.50	47.85	48.40	48.72	48.98	**49.31**	48.98	**49.31**
	1.00	50.87	51.47	51.68	51.89	**52.22**	51.89	**52.22**

and the peak signal-to-noise ratio (PSNR) [dB] in lossy image coding:

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

where MAX_p and MSE are the maximum possible pixel value of the image and the mean squared error, respectively. To evaluate transform performance fairly, we employed two-, three-, six-, and two-level decompositions, respectively, on the eight-channel DCT [26] (H.265/HEVC)[2], HLT [9] (JPEG XR), 1D-NSL-based DWTs without adaptive directionalities [14], [15], and eight-channel BOFBs. The 1D-SBL-based BOFBs had the same transfer function as the proposed FBs. The image set included six 512×512 eight-bit standard grayscale images in [27], two 2048×2048 eight-bit clipped grayscale images in [28], and two 2816×1600 16-bit grayscale images in [29]. A quadtree-based embedded image coder EZW-IP [30] was used to encode the transformed images. EZW-IP is more suited to block transforms than are the popular zerotree-based coders, e.g., EZW [31] and SPIHT [32]. 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 processings of the HLT and DWTs.

Tables II, III, and Fig. 6 show lossless and lossy image coding results. Although the conventional methods sometimes performed better on images with many low frequency components, overall, the 2D-NSBL-based BOFBs outperformed the conventional methods. These results are considered to be due to the merging (reducing) of many rounding operations in the 2D-NSBL-based BOFBs. Comparing Figs. 4 and 5, it is clear that the number of rounding operations of the 2D non-separable implementation is the almost half that of the 1D separable implementation. However, there were no differences between the 1D-SBL and 2D-NSBL of BOFBs in 16-bit images. In the future, we should solve the problem in high bit images.

[2]Since the DCT in H.265/HEVC is not composed of lifting structures, it is unsuitable for lossless image coding.

V. CONCLUSION

We devised a 2D-NSBL and applied it to M-channel PRFBs in lossy-to-lossless image coding. The 2D-NSBL is easily formulated from the 1D-SBL and 2D-NSL methods and can be regarded as an extension of the 2D-NSL because it is completely equivalent to a 2D-NSL when $M = 2$. A lossy-to-lossless image coding experiment confirmed the improvements that could be had with 2D-NSBL.

ACKNOWLEDGMENT

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

Fig. 6. Comparison of a particular area of an image *Barbara* (bitrate: 0.25 [bpp]): (left-right) 8×8 DCT, 4×8 HLT, 2D-NSL-based 9/7-DWT, 8×24 1D-SBL-based BOFB [13], and 8×24 2D-NSBL-based BOFB.

REFERENCES

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

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

[3] A. Skodras, C. Christopoulis, and T. Ebrahimi, "The JPEG2000 still image compression standard," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 36–58, Sep. 2001.

[4] 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.

[5] 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.

[6] 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.

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

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

[9] 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*, vol. 7073, San Diego, CA, Aug. 2008, pp. 1–12.

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

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

[12] ——, "The lifting scheme: A construction of second generation wavelets," *SIAM J. Math. Anal.*, vol. 29, no. 2, pp. 511–546, 1998.

[13] 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.

[14] 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.

[15] M. Iwahashi and H. Kiya, "Reversible 2D 9-7 DWT based on non-separable 2D lifting structure compatible with irreversible DWT," *IEICE Trans. Fundamentals.*, vol. E94-A, no. 10, pp. 1928–1936, Oct. 2011.

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

[17] 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.

[18] Y. J. Chen, S. Oraintara, and K. S. Amaratunga, "Dyadic-based factorizations for regular paraunitary filterbanks and M-band orthogonal wavelets with structural vanishing moments," *IEEE Trans. Signal Process.*, vol. 53, no. 1, pp. 193–207, Jan. 2005.

[19] Y. She, P. Hao, and Y. Paker, "Matrix factorizations for parallel integer transformation," *IEEE Trans. Signal Process.*, vol. 54, no. 12, pp. 4675–4684, Dec. 2006.

[20] C. M. Brislawn, "Group lifting structures for multirate filter banks I: Uniqueness of lifting factorizations," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2068–2077, Apr. 2010.

[21] ——, "Group lifting structures for multirate filter banks II: Linear phase filter banks," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2078–2087, Apr. 2010.

[22] 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.

[23] L. Jiao, L. Wang, J. Wu, J. Bai, S. Wang, and B. Hou, "Shape-adaptive reversible integer lapped transform for lossy-to-lossless ROI coding of remote sensing two-dimensional images," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 2, pp. 326–330, Mar. 2011.

[24] Y. J. Chen, S. Oraintara, and K. S. Amaratunga, "Theory and factorization for a class of structurally regular biorthogonal filter banks," *IEEE Trans. Signal Process.*, vol. 54, no. 2, pp. 691–700, Feb. 2006.

[25] 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.

[26] P. Kumar, S. Y. Park, B. K. Mohanty, K. S. Lim, and C. Yeo, "Efficient integer DCT architectures for HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 1, pp. 168–178, Jan. 2014.

[27] "The USC-SIPI Image Database," *Univ. Southern California, Signal and Image Processing Institute [Online]*, Available: http://sipi.usc.edu/database/.

[28] "The New Test Images - Image Compression Benchmark," *Rawzor - Lossless Compression Software for Camera RAW Images [Online]*, Available: http://www.imagecompression.info/test_images/.

[29] S. Andriani, H. Brendel, T. Seybold, and J. Goldstone, "Beyond the Kodak image set: A new reference set of color image sequences," in *Proc. of ICIP'13*, Melbourne, Australia, Sep. 2013, pp. 2289–2293.

[30] 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.

[31] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet

coefficients," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.
[32] 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.

Taizo Suzuki (M'11) 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.

Hiroyuki Kudo In 1985, Hiroyuki Kudo received the B.Sc. degree from Department of Electrical Communications, Tohoku University, Japan. In 1990, he received the Ph.D. degree from the Graduate School of Engineering, Tohoku University, Japan. In 1992, he joined the University of Tsukuba, Japan as an assistant professor. Currently, he is a professor at the Division of Information Engineering, Faculty of Engineering, Information and Systems, University of Tsukuba, Japan. His research areas include medical imaging, image processing, and inverse problems. In particular, he is actively working on tomographic image reconstruction for X-ray CT, PET, SPECT, and Electron Tomography. Since 2011, he is working as an Editor in Chief, Journal of Medical Imaging Technology (MIT). He obtained the best paper award several times from various domestic and international academic societies. In particular, his papers on interior tomography published in 2008 were selected as High Lights of the two journals, Physics in Medicine and Biology and Inverse Problems. He is a member of IEEE, Japanese Society of Medical Imaging Technology (JAMIT), and IEICE Japan.