Lower Complexity Lifting Structures for Hierarchical Lapped Transforms Highly Compatible with JPEG XR Standard

Taizo Suzuki, *Member, IEEE* and Taichi Yoshida, *Member, IEEE*

***Abstract*—This paper presents lifting structures for hierarchical lapped transforms (HLTs) that are highly compatible with the JPEG XR standard and that are lower in complexity in terms of the number of operations and lifting steps than the existing HLT in JPEG XR. Two structures ($\mathbf{T}_{RR}$ and $\mathbf{T}_{HH}$), called Householder-lifting structures, are obtained by using a lifting factorization followed by a Householder factorization of a non-separable 2-D transform of rotation matrices. The third structure ($\mathbf{T}_{HR}$) is simply derived from a combination of a Hadamard transform ($\mathbf{T}_{HH}$) and two rotation matrices. The floating-point lifting coefficients are approximated as dyadic values as in the existing structures of JPEG XR, because doing so is very low cost, thanks to the structures having only adders and shifters without multipliers. Although the new $\mathbf{T}_{HH}$ does not outperform the existing structure, the new $\mathbf{T}_{RR}$ has one fewer adder, one fewer shifter, and four fewer lifting steps than the existing one. Moreover, the new $\mathbf{T}_{HR}$ not only has one fewer adder, three fewer shifters, and two fewer lifting steps; it also can reuse $\mathbf{T}_{HH}$; i.e., it can be used to make a more stylish codec. We show that these lower complexity HLTs are comparable in performance to the existing HLT at lossy-to-lossless image coding and at the same time highly compatible with JPEG XR.**

***Index Terms*—Hierarchical lapped transform, Householder factorization, JPEG XR standard, lifting factorization, lossy-to-lossless image coding.**

I. Introduction

JPEG (Joint Photographic Experts Group) [1] was the first image compression (coding) standard in telecommunications technology. JPEG helps to alleviate the burden on servers and free up communication bandwidth. However, the discrete cosine transform (DCT) [2] that it uses causes blocking artifacts in low bitrate compression because it ignores the continuity between adjacent blocks. In addition, the DCT, which maps integer input signals to real output signals, cannot be used to create a lossless mode, and particular data such as medical and satellite images should be saved without any loss of data. Moreover, the development of the multimedia applications and the spread of broadband have led to the requirement that image data be conveyed with various levels of quality.

JPEG 2000 [3] was developed to be the first lossy-to-lossless image coding standard that has scalability from lossless to lossy data. JPEG 2000 respectively employs 5/3 and 9/7-tap discrete wavelet transforms (DWTs) for lossless and lossy modes [4]. The DWTs are constructed using lifting structures that map integer input signals to integer output signals. JPEG 2000 is not only capable of lossy-to-lossless image coding; it also outperforms JPEG in terms of compression ratio and causes no blocking artifacts even in low bitrate compression. However, because of its high complexity, JPEG 2000 was not able to induce a generational change from JPEG to it.

JPEG XR (eXtended Range) [5] is a newer lossy-to-lossless image coding standard. JPEG XR has half the complexity of JPEG 2000 while preserving image quality. One of the factors contributing to its low complexity is its use of a four-channel hierarchical lapped transform (HLT) [6]. The HLT is implemented as cascading non-separable 2-D transforms of rotation matrices. The cascaded structures are factorized into lifting structures for achieving lossy-to-lossless image coding. Additionally, the floating-point lifting coefficients are approximated as dyadic values, as this is very low cost, thanks to the structures having only adders and shifters without multipliers. On the other hand, although many lifting-based lapped transforms have been proposed [7–11], they are incompatible with JPEG XR.

This paper presents lifting structures for HLTs that are highly compatible with the JPEG XR standard and that are lower in complexity in terms of the number of operations and lifting steps than the existing HLT in JPEG XR.[1] Two structures ($\mathbf{T}_{RR}$ and $\mathbf{T}_{HH}$), called Householder-lifting structures, are obtained by using a lifting factorization followed by a Householder factorization [13] of a non-separable 2-D transform of rotation matrices. The third structure ($\mathbf{T}_{HR}$) is simply derived from a combination of a Hadamard transform ($\mathbf{T}_{HH}$) and two rotation matrices. The floating-point lifting coefficients are approximated into dyadic values as in the existing structure of JPEG XR, because doing so is very low cost, thanks to the structures having only adders and shifters without multipliers. Although the new $\mathbf{T}_{HH}$ does not outperform the existing structure, the new $\mathbf{T}_{RR}$ has one fewer adder, one fewer shifter, and four fewer lifting steps than

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

T. Yoshida is with the Department of Electrical, Electronics and Information Engineering, Nagaoka University of Technology, Nagaoka, Niigata, 940-2137 Japan (email: yoshida@iwalab.org).

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

[1]Part of this paper was presented at the 22nd IEEE International Conference on Image Processing (ICIP 2015) [12].

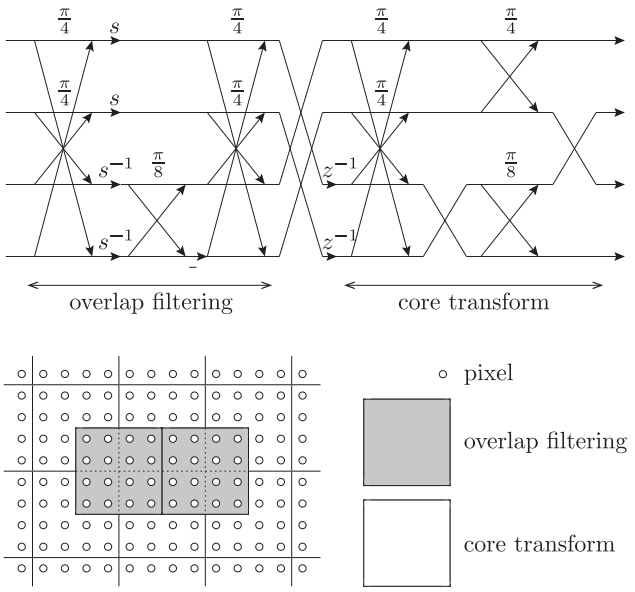

Fig. 1. Four-channel HLT in JPEG XR: (top) lattice structure, (bottom) regions of support for the basic core transform and overlap filtering operators.

the existing one. Moreover, the new $\mathbf{T}_{HR}$ not only has one fewer adder, three fewer shifters, and two fewer lifting steps; it also can reuse $\mathbf{T}_{HH}$; i.e., it can be used to make a more stylish codec. We show that the proposed HLTs are comparable to the existing HLT at lossy-to-lossless image coding while preserving high compatibility with the JPEG XR.

The rest of this paper is organized as follows: Sec. II reviews the existing HLT in JPEG XR and Householder-lifting factorization of an orthogonal matrix. Sec. III presents the lower complexity lifting structures. Sec. IV compares these HLTs with the existing HLT in terms of the number of operations and lifting steps, execution time, and in lossy-to-lossless image coding with a JPEG XR codec. It also discusses the precision of its compatibility with the existing HLT. Sec. V concludes this paper.

Notation: $\mathbf{I}_{[N]}$, $\mathbf{J}_{[N]}$, a superscript T, diag$(\cdots)$, $\otimes$, $\overline{x}$, and $\mathbf{P}_{[4]}$ respectively denote an $N\times N$ ($N\in\mathbb{N}$) identity matrix, an $N\times N$ reversal matrix, the transpose of a matrix, a diagonal matrix, the Kronecker product, $\overline{x}=-x$, and the following 4×4 permutation matrix:

$$\mathbf{P}_{[4]}=\begin{bmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{bmatrix}. \tag{1}$$

II. Review and Definitions

A. Hierarchical Lapped Transform for JPEG XR Standard

Tran *et al.* produced a time-domain lapped transform (TDLT) in which each overlap filtering operator is centered between the boundaries of four core transform operators [14].

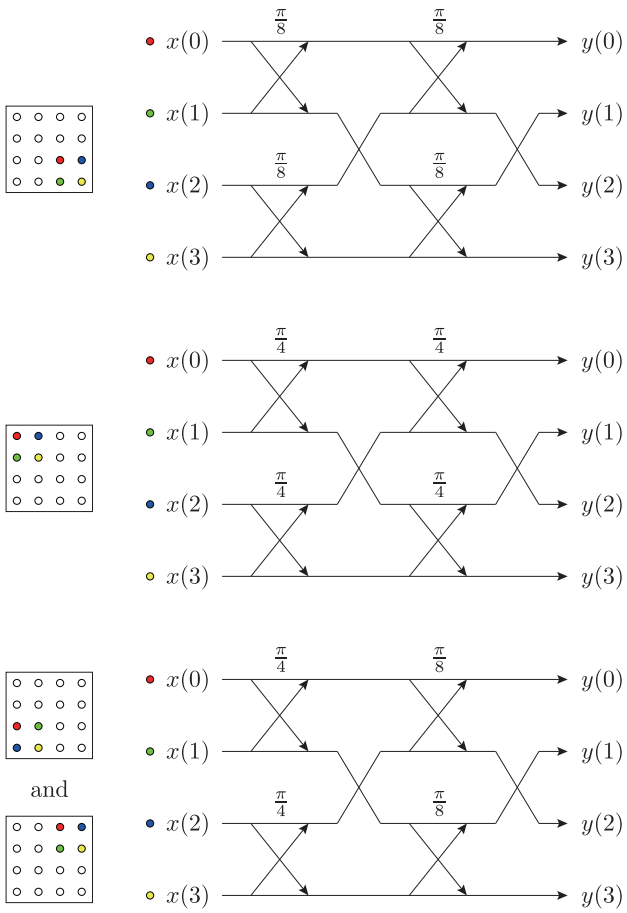

Fig. 2. 2-D implementation using separable rotation matrices on the last step of the TDLT in JPEG XR: (top-bottom) $\mathbf{R}_{\frac{\pi}{8}}\otimes\mathbf{R}_{\frac{\pi}{8}}$, $\mathbf{R}_{\frac{\pi}{4}}\otimes\mathbf{R}_{\frac{\pi}{4}}$, and $\mathbf{R}_{\frac{\pi}{8}}\otimes\mathbf{R}_{\frac{\pi}{4}}$.

JPEG XR employs a particular four-channel TDLT, as follows (see Fig. 1) [6]:

$$\mathbf{E}(z)=\underbrace{\mathbf{P}_{[4]}\begin{bmatrix}\mathbf{R}_{\frac{\pi}{4}}&\mathbf{0}\\\mathbf{0}&\mathbf{R}_{\frac{\pi}{8}}\mathbf{J}_{[2]}\end{bmatrix}\mathbf{W}_{[4]}\begin{bmatrix}\mathbf{I}_{[2]}&\mathbf{0}\\\mathbf{0}&z^{-1}\mathbf{I}_{[2]}\end{bmatrix}}_{\text{core transform}}\cdot\underbrace{\begin{bmatrix}\mathbf{0}&\mathbf{I}_{[2]}\\\mathbf{I}_{[2]}&\mathbf{0}\end{bmatrix}\mathbf{W}_{[4]}\begin{bmatrix}s\mathbf{I}_{[2]}&\mathbf{0}\\\mathbf{0}&s^{-1}\mathbf{R}'_{\frac{\pi}{8}}\end{bmatrix}\mathbf{W}_{[4]}}_{\text{overlap filtering}}, \tag{2}$$

where s is a scaling factor, $s=0.8272$, z^{-1} is a delay element, $\mathbf{R}_\theta$ and $\mathbf{R}'_\theta$ are rotation matrices with an arbitrary rotation angle θ:

$$\mathbf{R}_\theta=\begin{bmatrix}\cos\theta&\sin\theta\\\sin\theta&-\cos\theta\end{bmatrix}\triangleq\begin{bmatrix}\mathfrak{c}_\theta&\mathfrak{s}_\theta\\\mathfrak{s}_\theta&-\mathfrak{c}_\theta\end{bmatrix} \tag{3}$$

$$\mathbf{R}'_\theta=\begin{bmatrix}1&0\\0&-1\end{bmatrix}\mathbf{R}_\theta=\begin{bmatrix}\mathfrak{c}_\theta&\mathfrak{s}_\theta\\-\mathfrak{s}_\theta&\mathfrak{c}_\theta\end{bmatrix}, \tag{4}$$

and $\mathbf{W}_{[4]}$ has two $\mathbf{R}_{\frac{\pi}{4}}$s,

$$\mathbf{W}_{[4]}=\frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{I}_{[2]}&\mathbf{J}_{[2]}\\\mathbf{J}_{[2]}&-\mathbf{I}_{[2]}\end{bmatrix}. \tag{5}$$

It is clear that the TDLT in JPEG XR is easily constructed from only $\mathbf{R}_{\frac{\pi}{4}}$s, $\mathbf{R}_{\frac{\pi}{8}}$s, scaling factors, delay elements, permutations, and sign inversions.

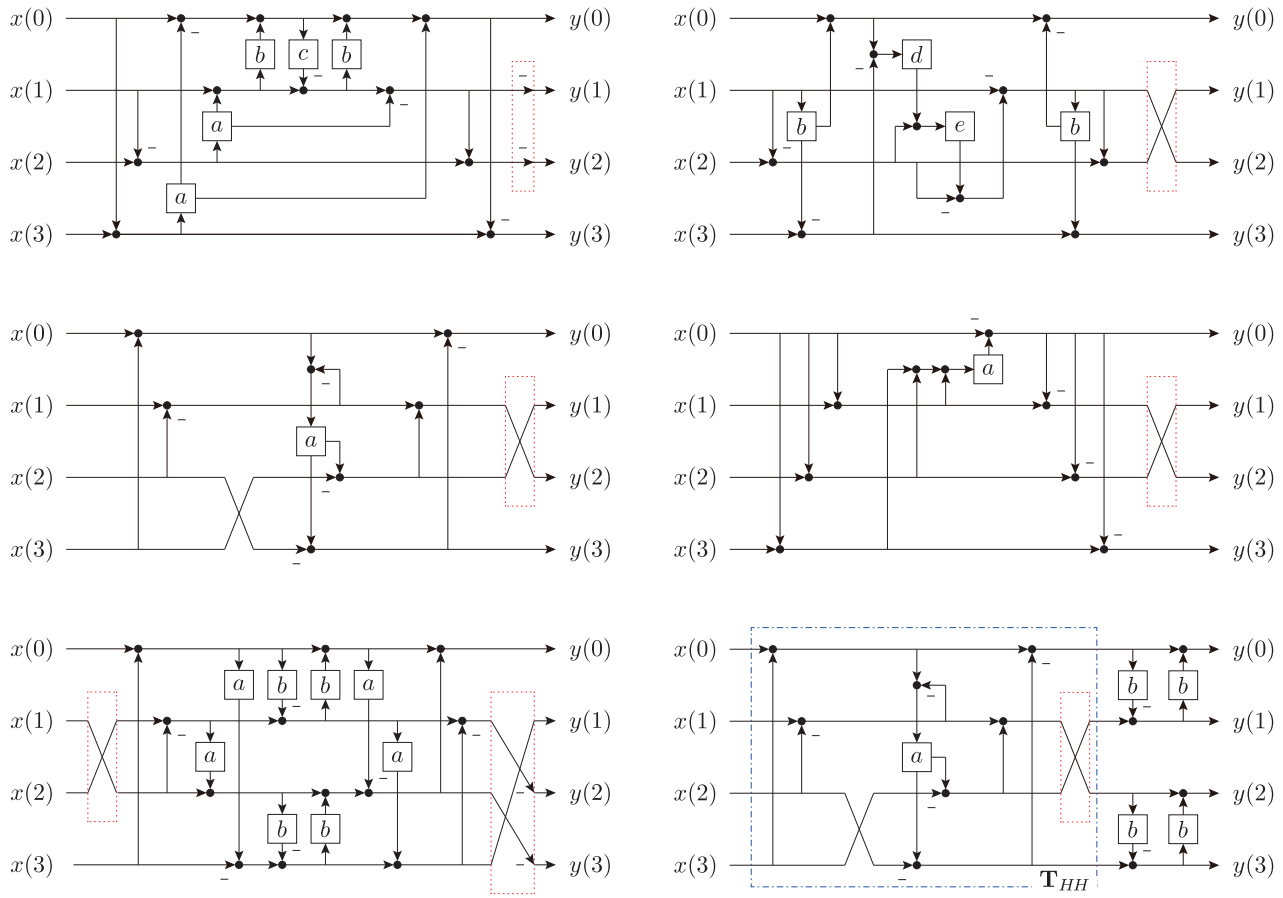

Fig. 3. Multiplierless lifting structures of non-separable 2-D transforms $\mathbf{T}_{RR}$, $\mathbf{T}_{HH}$, and $\mathbf{T}_{HR}$ (black circles, a, b, c, d, e, and areas framed by red dotted lines mean adders, $a = 1/2$, $b = 3/8 = (1+2)/2^3$, $c = 3/4 = 1 - 1/2^2$, $d = 3 = 1 + 2$, $e = 1/8 = 1/2^3$, and areas that can be omitted by arranging them appropriately in the real implementation): (top-to-bottom) $\mathbf{T}_{RR}$, $\mathbf{T}_{HH}$, and $\mathbf{T}_{HR}$; (left-to-right) the existing structures in JPEG XR and the proposed structures.

The TDLT in JPEG XR is implemented as cascading non-separable 2-D transforms of rotation matrices. When a 2×2 input block signal $\mathbf{X}$ is two-dimensionally implemented by using separable rotation matrices $\mathbf{R}_{\theta_0}$ and $\mathbf{R}_{\theta_1}$, it is expressed as

$$\mathbf{Y} = \mathbf{R}_{\theta_1} \mathbf{X} \mathbf{R}_{\theta_0}^T, \tag{6}$$

where $\mathbf{Y}$ is the output block signal of $\mathbf{X}$. By letting $\mathbf{x}$ be the 4×1 input vector signal $\mathbf{x} = [x(0)\ x(1)\ x(2)\ x(3)]^T$ obtained by rearranging the 2×2 input block signal $\mathbf{X}$, the operation (6) can be reformulated as

$$\mathbf{y} = \mathbf{R}_{\theta_0} \otimes \mathbf{R}_{\theta_1} \mathbf{x}, \tag{7}$$

where $\mathbf{y} = [y(0)\ y(1)\ y(2)\ y(3)]^T$ is the output vector signal of $\mathbf{x}$. Fig. 2 shows the last step of the TDLT in JPEG XR as an example of the 2-D implementation using separable rotation matrices. Let $\mathbf{T}_{\theta_0,\theta_1}$ be a non-separable 2-D transform,

$$\begin{aligned} \mathbf{T}_{\theta_0,\theta_1} &= \mathbf{R}_{\theta_0} \otimes \mathbf{R}_{\theta_1} \\ &= \mathbf{W}_{[4]} \mathrm{diag}\,(1,1,-1,-1) \\ &\quad \cdot \begin{bmatrix} \mathfrak{c}_{\theta_0-\theta_1} & 0 & \mathfrak{s}_{\theta_0-\theta_1} & 0 \\ 0 & \mathfrak{c}_{\theta_0+\theta_1} & 0 & \mathfrak{s}_{\theta_0+\theta_1} \\ \mathfrak{s}_{\theta_0-\theta_1} & 0 & -\mathfrak{c}_{\theta_0-\theta_1} & 0 \\ 0 & \mathfrak{s}_{\theta_0+\theta_1} & 0 & -\mathfrak{c}_{\theta_0+\theta_1} \end{bmatrix} \\ &\quad \cdot \mathbf{W}_{[4]} \mathrm{diag}\,(1,-1,-1,1)\,. \end{aligned} \tag{8}$$

$\mathbf{T}_{\theta_0,\theta_1}$ is factorized into multiplierless lifting structures with dyadic lifting coefficients. For example, $\mathbf{T}_{\frac{\pi}{8},\frac{\pi}{8}} \triangleq \mathbf{T}_{RR}$ and $\mathbf{T}_{\frac{\pi}{4},\frac{\pi}{8}} \triangleq \mathbf{T}_{HR}$ are shown at the top left and bottom left of Fig. 3, respectively. $\mathbf{T}_{\frac{\pi}{4},\frac{\pi}{4}} \triangleq \mathbf{T}_{HH}$ is a special case, i.e., a four-channel Hadamard transform [15] (see the middle left of Fig. 3)[2]:

$$\mathbf{T}_{HH} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}. \tag{9}$$

[2]Note that although there are some differences between the manuscript [6] and the released JPEG XR codec [16], we will describe the structures based on [16].

Moreover, we will not describe multiplierless lifting structures in detail except for $\mathbf{T}_{RR}$, $\mathbf{T}_{HH}$, and $\mathbf{T}_{HR}$ in JPEG XR because we directly employ them.

The TDLT in JPEG XR is commonly called a hierarchical lapped transform (HLT) because it is implemented hierarchically as follows:

1) overlap filtering (optional first stage)
2) core transform (first stage)
3) overlap filtering for the DC components obtained by the first stage (optional second stage)
4) core transform for the DC components obtained by the first stage (second stage).

B. Householder-Lifting Factorization of an Orthogonal Matrix

An $M \times M$ Householder matrix $\mathbf{H}_{[M]}[\mathbf{u}_k]$ is expressed as follows [17]:

$$\mathbf{H}_{[M]}[\mathbf{u}_k] = \mathbf{I}_{[M]} - 2\mathbf{u}_k\mathbf{u}_k^T, \tag{10}$$

where

$$\mathbf{u}_k = \begin{bmatrix} u_{k,0} & u_{k,1} & \cdots & u_{k,M-1} \end{bmatrix}^T \tag{11}$$

and $u_{k,l}$ ($l = 0, 1, \cdots, M-1$) is an arbitrary value that satisfies $||\mathbf{u}_k|| = 1$. Also, any Householder matrix is identical to its inverse because it is a symmetric orthogonal matrix, i.e.,

$$(\mathbf{H}_{[M]}[\mathbf{u}_k])^{-1} = (\mathbf{H}_{[M]}[\mathbf{u}_k])^T = \mathbf{H}_{[M]}[\mathbf{u}_k]. \tag{12}$$

Any $M \times M$ orthogonal matrix $\mathbf{G}$ can always be factorized into $(M-1)$ cascading Householder matrices as follows:

$$\mathbf{G} = \mathbf{H}_{[M]}[\mathbf{u}_0]\,\mathbf{H}_{[M]}[\mathbf{u}_1]\cdots\mathbf{H}_{[M]}[\mathbf{u}_{M-2}], \tag{13}$$

where

$$\begin{bmatrix} \mathbf{u}_0 & \mathbf{u}_1 & \cdots & \mathbf{u}_{M-2} \end{bmatrix} = \begin{bmatrix} u_{0,0} & 0 & \cdots & 0 \\ u_{0,1} & u_{1,1} & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ u_{0,M-2} & u_{1,M-2} & \cdots & u_{M-2,M-2} \\ u_{0,M-1} & u_{1,M-1} & \cdots & u_{M-2,M-1} \end{bmatrix}. \tag{14}$$

Furthermore, Chen and Amaratunga introduced a lifting factorization of an $M \times M$ Householder matrix [13]:

$$\begin{aligned} \mathbf{H}_{[M]}[\mathbf{u}_k] = &\begin{bmatrix} \mathbf{I}_{[r]} & \begin{bmatrix} \mathfrak{a}_{k,0} \\ \vdots \\ \mathfrak{a}_{k,M-1} \end{bmatrix} & \mathbf{0} \\ \mathbf{0} & & \mathbf{I}_{[M-r-1]} \end{bmatrix} \\ &\cdot \begin{bmatrix} \mathbf{I}_{[r]} & \mathbf{0} \\ \begin{bmatrix} \mathfrak{b}_{k,0} & \cdots & \mathfrak{b}_{k,M-1} \end{bmatrix} & \\ \mathbf{0} & \mathbf{I}_{[M-r-1]} \end{bmatrix} \\ &\cdot \begin{bmatrix} \mathbf{I}_{[r]} & \begin{bmatrix} \overline{\mathfrak{a}_{k,0}} \\ \vdots \\ \overline{\mathfrak{a}_{k,M-1}} \end{bmatrix} & \mathbf{0} \\ \mathbf{0} & & \mathbf{I}_{[M-r-1]} \end{bmatrix}, \end{aligned} \tag{15}$$

where

$$\mathfrak{a}_{k,l} = \begin{cases} 1 & (l = r) \\ \frac{u_{k,l}}{u_{k,r}} & (\text{otherwise}) \end{cases} \tag{16}$$

$$\mathfrak{b}_{k,l} = \begin{cases} -1 & (l = r) \\ -2u_{k,l}u_{k,r} & (\text{otherwise}) \end{cases} \tag{17}$$

and r ($r = 0, 1, \cdots, M-1$) for $u_{k,r} \neq 0$ is selected. It has $3(M-1)$ adders, multipliers, and lifting steps. Consequently, an $M \times M$ orthogonal matrix can be factorized into $3\sum_{m=0}^{M-2}(M-m-1)$ adders, multipliers, and lifting steps after being factorized into $(M-1)$ cascading Householder matrices. Depending on the structure, the operations can be simplified and the lifting steps can be implemented in parallel; i.e., the complexity can be reduced even more.

III. Lower Complexity Lifting Structures for HLTs

A. Householder-Lifting Factorization of a Non-Separable 2-D Transform $\mathbf{T}_{\theta,\theta}$

When the non-separable 2-D transform $\mathbf{T}_{\theta_0,\theta_1}$ in Eq. (8) is such that $\theta_0 = \theta_1 = \theta$, it is a particular symmetric orthogonal matrix $\mathbf{T}_{\theta,\theta}$:

$$\mathbf{T}_{\theta,\theta} = \mathbf{R}_\theta \otimes \mathbf{R}_\theta = \begin{bmatrix} \mathfrak{c}_\theta^2 & \mathfrak{c}_\theta\mathfrak{s}_\theta & \mathfrak{c}_\theta\mathfrak{s}_\theta & \mathfrak{s}_\theta^2 \\ \mathfrak{c}_\theta\mathfrak{s}_\theta & -\mathfrak{c}_\theta^2 & \mathfrak{s}_\theta^2 & -\mathfrak{c}_\theta\mathfrak{s}_\theta \\ \mathfrak{c}_\theta\mathfrak{s}_\theta & \mathfrak{s}_\theta^2 & -\mathfrak{c}_\theta^2 & -\mathfrak{c}_\theta\mathfrak{s}_\theta \\ \mathfrak{s}_\theta^2 & -\mathfrak{c}_\theta\mathfrak{s}_\theta & -\mathfrak{c}_\theta\mathfrak{s}_\theta & \mathfrak{c}_\theta^2 \end{bmatrix}. \tag{18}$$

Since any $M \times M$ orthogonal matrix can be factorized into $(M-1)$ cascading Householder matrices as described in Sec. II-B, the symmetric orthogonal matrix $\mathbf{T}_{\theta,\theta}$ in Eq. (18) can also be factorized into them. Fortunately, the symmetric orthogonal matrix $\mathbf{T}_{\theta,\theta}$ is easily composed of only a 4×4 permutation matrix $\mathbf{P}_{[4]}$ and a 4×4 Householder matrix $\mathbf{H}_{[4]}[\mathbf{u}_\theta]$:

$$\mathbf{T}_{\theta,\theta} = \mathbf{P}_{[4]}\mathbf{H}_{[4]}[\mathbf{u}_\theta], \tag{19}$$

where

$$\mathbf{u}_\theta = \frac{1}{\sqrt{2}}\begin{bmatrix} \pm\mathfrak{s}_\theta & \mp\mathfrak{c}_\theta & \mp\mathfrak{c}_\theta & \mp\mathfrak{s}_\theta \end{bmatrix}^T. \tag{20}$$

In accordance with Eqs. (15)-(17), the Householder matrix $\mathbf{H}_{[4]}[\mathbf{u}_\theta]$ in Eq. (19) can be factorized into four types of three

lifting matrices as follows:

$$\mathbf{H}_{[4]}\left[\mathbf{u}_\theta\right] = \begin{cases} \begin{bmatrix} 1 & 0 & 0 & 0 \\ \overline{\alpha_\theta} & 1 & 0 & 0 \\ \overline{\alpha_\theta} & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & \beta_\theta & \beta_\theta & \gamma_\theta \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ \alpha_\theta & 1 & 0 & 0 \\ \alpha_\theta & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} & (r=0) \\ \begin{bmatrix} 1 & \overline{\delta_\theta} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & \delta_\theta & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ \beta_\theta & -1 & \overline{\epsilon_\theta} & \overline{\beta_\theta} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \delta_\theta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & \overline{\delta_\theta} & 0 & 1 \end{bmatrix} & (r=1) \\ \begin{bmatrix} 1 & 0 & \overline{\delta_\theta} & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \delta_\theta & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \beta_\theta & \overline{\epsilon_\theta} & -1 & \overline{\beta_\theta} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \delta_\theta & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \overline{\delta_\theta} & 1 \end{bmatrix} & (r=2) \\ \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & \alpha_\theta \\ 0 & 0 & 1 & \alpha_\theta \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \gamma_\theta & \overline{\beta_\theta} & \overline{\beta_\theta} & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & \overline{\alpha_\theta} \\ 0 & 0 & 1 & \overline{\alpha_\theta} \\ 0 & 0 & 0 & 1 \end{bmatrix} & (r=3) \end{cases} \tag{21}$$

where

$$\{\alpha_\theta,\ \beta_\theta,\ \gamma_\theta,\ \delta_\theta,\ \epsilon_\theta\} = \left\{ \frac{\mathfrak{c}_\theta}{\mathfrak{s}_\theta},\ \mathfrak{c}_\theta\mathfrak{s}_\theta,\ \mathfrak{s}_\theta^2,\ \frac{\mathfrak{s}_\theta}{\mathfrak{c}_\theta},\ \mathfrak{c}_\theta^2 \right\}. \tag{22}$$

The Householder-lifting structure has nine adders, four multipliers, and three lifting steps.

B. Householder-Lifting Structures of $\mathbf{T}_{RR}$ *and* $\mathbf{T}_{HH}$

To obtain a multiplierless structure, any lifting coefficient with a floating-point value must be approximated as a dyadic value $n/2^{bit}$ ($n, bit \in \mathbb{N}$). The resulting multiplierless structure will yield fast implementations at the expense of resolution performance of the transform. The lifting coefficients of the non-separable 2-D transform $\mathbf{T}_{RR}$ proposed in Sec. III-A are approximated using 3-bit ($bit = 3$) values, as is done with the existing structures in [6]. The following multiplierless Householder-lifting structure in $\mathbf{T}_{RR} = \mathbf{P}_{[4]}\mathbf{H}_{[4]}\left[\mathbf{u}_{\frac{\pi}{8}}\right]$, in which $r = 1$ was experimentally determined to give the best coding performance, can be derived using approximations of the coefficients from Eq. (21).

$$\mathbf{H}_{[4]}[\mathbf{u}_{\frac{\pi}{8}}] = \begin{bmatrix} 1 & -\frac{3}{8} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & \frac{3}{8} & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{3}{8} & -1 & -\frac{7}{8} & -\frac{3}{8} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{3}{8} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -\frac{3}{8} & 0 & 1 \end{bmatrix} \tag{23}$$

where

$$\left\{\beta_{\frac{\pi}{8}},\ \delta_{\frac{\pi}{8}},\ \epsilon_{\frac{\pi}{8}}\right\} = \left\{ \mathfrak{c}_{\frac{\pi}{8}}\mathfrak{s}_{\frac{\pi}{8}},\ \frac{\mathfrak{s}_{\frac{\pi}{8}}}{\mathfrak{c}_{\frac{\pi}{8}}},\ \mathfrak{c}_{\frac{\pi}{8}}^2 \right\} \approx \left\{ \frac{3}{8},\ \frac{3}{8},\ \frac{7}{8} \right\}. \tag{24}$$

This structure is shown at the top right of Fig. 3. It controls the dynamic range in the process.

As described in Sec. II-A, a non-separable 2-D transform $\mathbf{T}_{HH}$ is equivalent to a four-channel Hadamard transform. The following multiplierless Householder-lifting structure in $\mathbf{T}_{HH} = \mathbf{P}_{[4]}\mathbf{H}_{[4]}\left[\mathbf{u}_{\frac{\pi}{4}}\right]$, in which $r = 0$ was experimentally determined to give the coding performance, is easily derived without any approximation from Eq. (21).

$$\mathbf{H}_{[4]}[\mathbf{u}_{\frac{\pi}{4}}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \tag{25}$$

where

$$\left\{\alpha_{\frac{\pi}{4}},\ \beta_{\frac{\pi}{4}},\ \gamma_{\frac{\pi}{4}}\right\} = \left\{ \frac{\mathfrak{c}_{\frac{\pi}{4}}}{\mathfrak{s}_{\frac{\pi}{4}}},\ \mathfrak{c}_{\frac{\pi}{4}}\mathfrak{s}_{\frac{\pi}{4}},\ \mathfrak{s}_{\frac{\pi}{4}}^2 \right\} = \left\{ 1,\ \frac{1}{2},\ \frac{1}{2} \right\}. \tag{26}$$

The structure is shown at the middle right of Fig. 3. However, the existing $\mathbf{T}_{HH}$ has two fewer adders than the new one, as described in Sec. IV-A. Thus, we used the existing $\mathbf{T}_{HH}$ in the experiments.

C. Lifting Structure of $\mathbf{T}_{HR}$

Although a non-separable 2-D transform $\mathbf{T}_{HR}$ is a symmetric orthogonal matrix, it is not equivalent to the particular symmetric orthogonal matrix in Eq. (18) because of $\theta_0 \neq \theta_1$. However, the non-separable 2-D transforms $\mathbf{T}_{HR}$ and $\mathbf{T}_{HH}$ are related as follows:

$$\mathbf{T}_{HR} = \begin{cases} \begin{bmatrix} \mathbf{R}'_{\frac{\pi}{8}} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}'_{\frac{\pi}{8}} \end{bmatrix} \mathbf{T}_{HH} & \text{(type A)} \\ \mathbf{T}_{HH} \begin{bmatrix} \mathbf{R}'_{\frac{\pi}{8}} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}'_{\frac{\pi}{8}} \end{bmatrix}^T & \text{(type B)} \end{cases}. \tag{27}$$

Type A of Eq. (27) was experimentally selected because it performed better than type B. $\mathbf{R}'_{\frac{\pi}{8}}$ is approximated as two multiplierless lifting structures, i.e.,

$$\mathbf{R}'_{\frac{\pi}{8}} \approx \begin{bmatrix} 1 & \frac{3}{8} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{3}{8} & 1 \end{bmatrix} \tag{28}$$

in accordance with the existing $\mathbf{T}_{HR}$ in [6]. Since this structure can reuse $\mathbf{T}_{HH}$ clearly, it can be used to make a more stylish codec. Consequently, the $\mathbf{T}_{HR}$ based on the existing $\mathbf{T}_{HH}$ can be expressed as shown at the bottom right of Fig. 3.

TABLE I
NUMBER OF OPERATIONS AND LIFTING STEPS (ADD., SHIFT., ROUND., STEP., AND PARA. RESPECTIVELY MEAN ADDER, SHIFTER, ROUNDING OPERATION (RIGHT SHIFTER), LIFTING STEP, AND LIFTING STEP IN PARALLEL PROCESSING).

		Add.	Shift. (Round.)	Step. (Para.)
# in $\mathbf{T}_{RR}$	Exist.	14	7 (5)	11 (7)
	Prop.	**13**	**6** (**3**)	**7** (**3**)
# in $\mathbf{T}_{HH}$	Exist.	**7**	**1** (**1**)	**6** (**3**)
	Prop.	9	**1** (**1**)	7 (**3**)
# in $\mathbf{T}_{HR}$	Exist.	16	12 (8)	12 (6)
	Prop.	**15**	**9** (**5**)	**10** (**5**)

TABLE II
AVERAGE EXECUTION TIMES IN TRANSFORMING THE 8-BIT 1280×1600 GRAYSCALE IMAGE *Cafe* $1,000$ TIMES ([SEC]).

Exist.	Prop. (Type I)	Prop. (Type II)
0.532	0.527	**0.520**

IV. EXPERIMENTAL RESULTS

A. Design and Complexity of JPEG XR Codecs

Table I shows the number of operations and lifting steps. Here, the new $\mathbf{T}_{RR}$ has one fewer adder, one fewer shifter, and four fewer lifting steps, the new $\mathbf{T}_{HH}$ has two more adders, the same number of shifters, and one more lifting step, and the new $\mathbf{T}_{HR}$ has one fewer adder, three fewer shifters, and two fewer lifting steps than the existing structures in [16]. Clearly, the resulting HLTs with the new $\mathbf{T}_{RR}$ and $\mathbf{T}_{HR}$ have fewer operations and lifting steps compared with the existing HLT. Moreover, since the new $\mathbf{T}_{RR}$ and $\mathbf{T}_{HR}$ respectively have two and three fewer rounding operations (which degrade coding performance by introducing rounding errors), we can expect some improvements in performance. Additionally, these new structures can be run in parallel for the hardware implementation. "Para." in Table I means the number of lifting steps in parallel processing.

We designed two new JPEG XR codecs by replacing the transform and its inverse in the JPEG XR codec [16] with the proposed ones, as follows:

1) The type I codec incorporated the new $\mathbf{T}_{RR}$ in the HLT.
2) The type II codec incorporated the new $\mathbf{T}_{RR}$ and $\mathbf{T}_{HR}$ in the HLT.

Note that the $\mathbf{T}_{RR}$, $\mathbf{T}_{HH}$, and $\mathbf{T}_{HR}$ algorithms used in JPEG XR codec have trivial differences from those shown in Fig. 3; i.e., some permutations and sign inversions have been added or omitted. Thus, in accordance with these differences, we put some permutations and sign inversions in the new structures so that they would be highly comparable with the existing ones.

To evaluate the complexity, we measured the average execution times in transforming the 8-bit 1280×1600 grayscale image *Cafe* in [18] $1,000$ times by using MATLAB on a machine with an Intel Core i7-4770 CPU 3.40 GHz and RAM 32.0 GB (Table II). The transform in the type I codec provided a 1.01 % speed up. The transform in the type II codec not only provided a 2.27 % speed up but also achieved a stylish codec because of the reuse of the existing $\mathbf{T}_{HH}$ for the new $\mathbf{T}_{HR}$ as described in Sec. III-C. As described above, these new structures have the potential for a faster implementation through parallel processing in hardware.

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

Test Images	Exist.	Prop. (Type I)	Prop. (Type II)
Bike	4.453	4.451	**4.449**
Building	3.069	**3.062**	**3.062**
Cafe	5.614	5.614	**5.604**
Car	3.556	3.550	**3.546**
Falls	3.617	3.612	**3.610**
Flower	4.394	4.391	**4.379**
Girl	3.345	**3.343**	3.346
House	3.683	**3.679**	**3.679**
Sakura	3.435	3.430	**3.427**
Woman	4.071	4.069	**4.067**
Big_building	3.820	**3.815**	**3.815**
Big_tree	4.652	4.650	**4.649**
Bridge	4.607	4.604	**4.594**
Deer	5.397	5.396	**5.387**
Spider_web	2.470	**2.461**	2.463

TABLE IV
COMPATIBILITY IN LOSSLESS MODE (AVEPSNR [DB] / AVESSIM).

Encoder	Decoder		
	Exist.	Prop. (Type I)	Prop. (Type II)
Exist.	Lossless	47.321 / 1.000	43.902 / 0.999
Prop. (Type I)	47.344 / 1.000	Lossless	—
Prop. (Type II)	44.089 / 0.999	—	Lossless

B. Application to Lossy-to-Lossless Image Coding

We compared the designed codecs with the existing one in terms of lossless bitrate (LBR) [bpp], peak signal-to-noise ratio (PSNR) [dB], and structural similarity (SSIM) [20] in lossy-to-lossless image coding:

$$\text{LBR [bpp]} = \frac{\text{Total number of bits [bit]}}{\text{Total number of pixels [pixel]}} \tag{29}$$

$$\text{PSNR [dB]} = 10 \log_{10} \left(\frac{255^2}{\text{MSE}} \right) \tag{30}$$

$$\text{SSIM} = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2\mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{31}$$

where MSE is the mean squared error, σ_x and σ_y are the variances, σ_{xy} is the covariance, and μ_x and μ_y are the average values of the original and reconstructed images. The 8-bit color images with 1280×1600 and larger sizes in [18], [19] were selected for the experiments (see Fig. 4).

Table III and Fig. 5 show the results of lossless and lossy image coding. The new codecs were comparable to the existing codec despite having structures with fewer operations and lifting steps. In particular, the new codecs slightly outperformed the existing codec in lossless mode. The rounding operations were fewer than in the existing structure, and the proposed approximations of the lifting coefficients lost less of the original transfer function than the existing approximations did.

C. Precision of Compatibility with JPEG XR Standard

The new structures are not completely compatible with JPEG XR because of rounding errors and differences in the

Fig. 4. Minified 8-bit color images in [18], [19]: (top) *Bike*, *Building*, *Cafe*, *Car*, *Falls*, *Flower*, *Girl*, *House*, *Sakura*, and *Woman* (all 1280×1600), (bottom) *Big_building* (7216×5412), *Big_tree* (6088×4550), *Bridge* (2749×4049), *Deer* (4043×2641), and *Spider_web* (4256×2848).

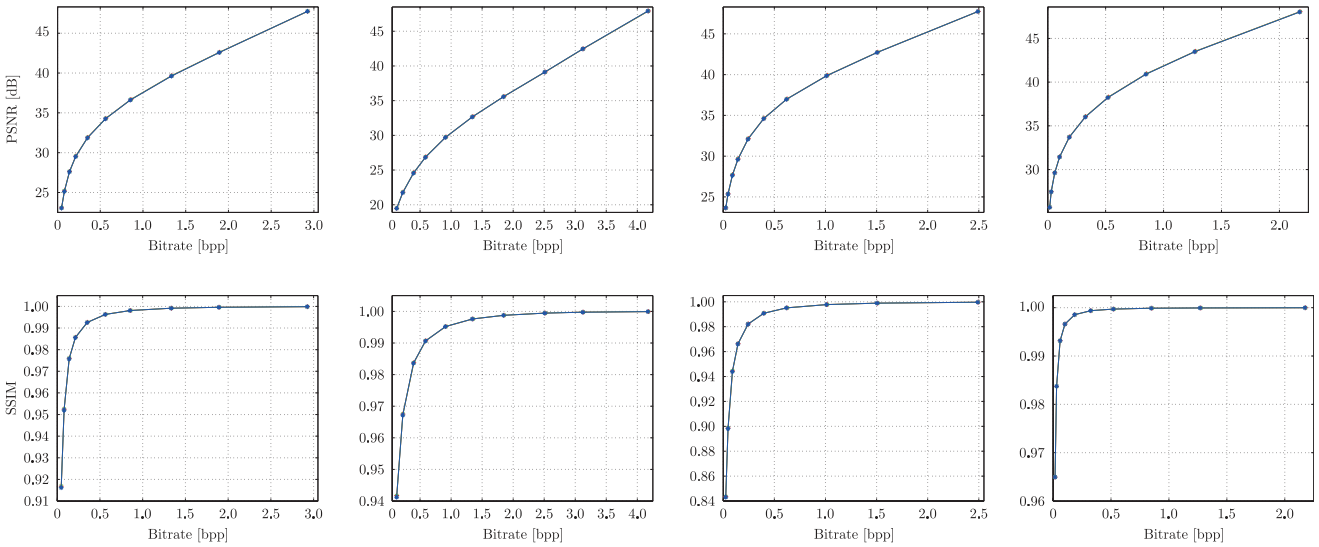

Fig. 5. Rate-distortion (R-D) curves for lossy compression and reconstruction using each of the tested codecs (red ($+$), green ($\times$), and blue ($*$) lines mean the conventional, proposed type I, and type II codecs, respectively): (top) PSNR, (bottom) SSIM, (left-to-right) *Bike*, *Cafe*, *Woman*, and *Big_building*.

approximate lifting coefficients. First, we investigated the precision of compatibility in lossless image coding using the existing encoder and decoding each of the decoders. 8-bit color images in [18], [19] were transformed and reconstructed by applying the same or different codecs. The average PSNR (AvePSNR) [dB] and average SSIM (AveSSIM) were used as objective indicators. Next, we investigated the precision of compatibility in lossy image coding using the existing encoder and decoding each of the decoders by using PSNR and SSIM (Eqs. (30) and (31)).

Table IV shows the lossless image coding results. As a matter of course, the values are indicated as "Lossless" when the same codecs were used. Values not required for the precision investigation are indicated as "—". The table shows that the type I codec outperformed the type II codec. We can thus consider that the type I codec has higher compatibility with the existing codec. However, since the all PSNRs were more than 40 [dB] and all SSIMs were more than 0.999, we can also say that both of the these codecs have high compatibility with the JPEG XR. Figs. 6 and 7 show the lossy image coding results of the existing encoder and each of the decoders. We can see that the type II decoder degraded PSNRs of high bitrate compression; e.g., the PSNRs of *Cafe* lossy compressed with 4.179 bpp were 47.919 [dB], 47.662 [dB], and 40.982 [dB], respectively. However, the SSIMs of the decoders were almost the same, and we could not find any perceptual differences, as shown in Fig. 7. Consequently, although the combination of the existing encoder and the existing decoder naturally achieved the best PSNRs, the new lifting structures have high compatibility with JPEG XR.

V. Conclusion

This paper presented lifting structures for HLTs that are highly compatible with the JPEG XR standard and that are lower in complexity in terms of the number of operations and lifting steps compared with JPEG XR. Although the new $\mathbf{T}_{HH}$ does not outperform the existing structure, the new $\mathbf{T}_{RR}$ and $\mathbf{T}_{HR}$ have fewer operations and lifting steps than the corresponding existing structures. Additionally, the new $\mathbf{T}_{HR}$ can reuse $\mathbf{T}_{HH}$; i.e., it can be used to make a more stylish codec. The resulting HLTs have fewer lifting steps and a faster implementation compared with the existing HLT. In spite of their simple structures, the new codecs were comparable in performance to the existing codec at lossy-to-lossless image coding. Furthermore, we confirmed that the new HLTs have high compatibility with JPEG XR. Not only

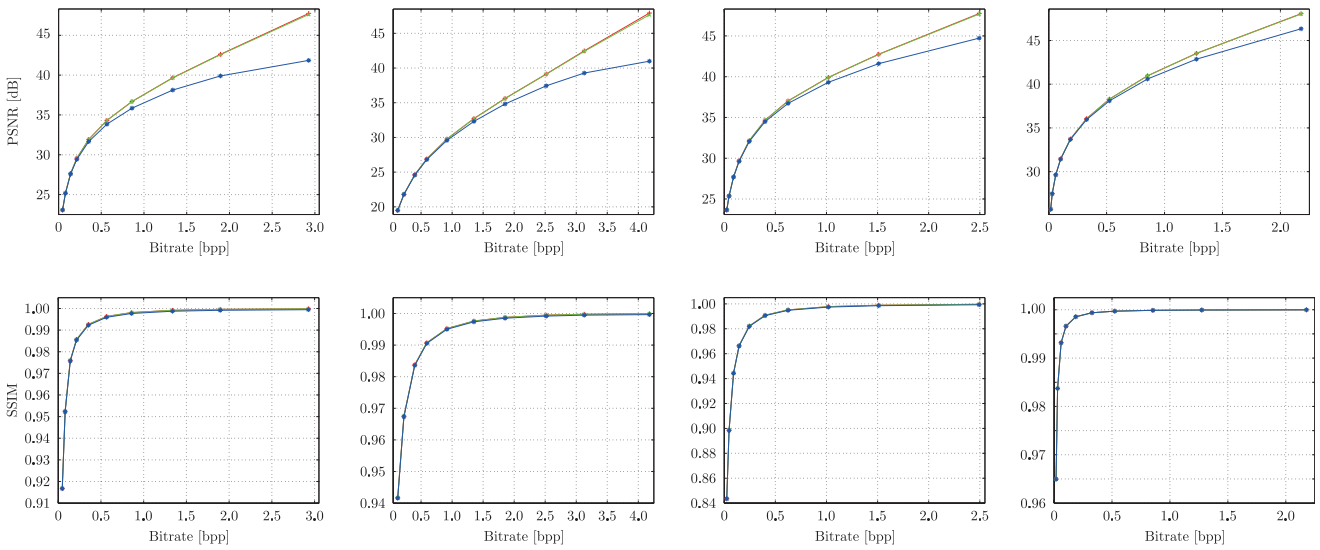

Fig. 6. Rate-distortion (R-D) curves for precision of compatibility when images were lossy compressed by using the existing encoder and reconstructed by each of the tested decoders (PSNR [dB]) (red (+), green (×), and blue (∗) lines mean the conventional, type I, and type II decoders, respectively): (top) PSNR, (bottom) SSIM, (left-to-right) *Bike*, *Cafe*, *Woman*, and *Big_building*.

Fig. 7. Particular areas of *Cafe* lossy compressed at 4.179 bpp by using the existing encoder and reconstructed by each of the tested decoders: (left-to-right) reconstructed with the existing decoder, type I decoder, and type II decoder, respectively.

that, these lifting structures will have many signal processing and communication applications because they are based on commonly used rotation matrices.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for providing many constructive suggestions that significantly improve the presentation of this paper.

REFERENCES

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

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

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

[5] 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.

[6] 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. 70730C–70730C–12.

[7] J. Xu, F. Wu, J. Liang, and W. Zhang, "Directional lapped transforms for image coding," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 85–97, Jan. 2010.

[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] 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.

[10] 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.
[11] T. Suzuki and H. Kudo, "Extended block-lifting-based lapped transforms," *IEEE Signal Process. Lett.*, vol. 22, no. 10, pp. 1657–1660, Oct. 2015.
[12] T. Suzuki, "Four-channel lifting-Householder-based Hadamard transform," in *Proc. of ICIP'15*, Québec City, Canada, Sep. 2015, pp. 1–5.
[13] 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.
[14] T. D. Tran, J. Liang, and C. Tu, "Lapped transform via time-domain pre- and post-filtering," *IEEE Trans. Signal Process.*, vol. 6, no. 6, pp. 1557–1571, June 2003.
[15] J. Manz, "A sequency-ordered fast Walsh transform," *IEEE Trans. Audio and Electroacoust.*, vol. 20, no. 3, pp. 204–205, Aug. 1972.
[16] "T.835 : Information technology - JPEG XR image coding system - Reference software," *ITU [Online]*, Available: https://www.itu.int/rec/T-REC-T.835-201201-I/.
[17] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Englewood Cliffs, NJ: Prentice Hall, 1992.
[18] "JPEG core experiment for the evaluation of JPEG XR image coding," *EPFL, Multimedia Signal Processing Group [Online]*, Available: http://mmspg.epfl.ch/iqa.
[19] "The New Test Images - Image Compression Benchmark," *Rawzor - Lossless Compression Software for Camera RAW Images [Online]*, Available: http://imagecompression.info/test_images/.
[20] Z. Wang, A. Conrad, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

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.

Taichi Yoshida (M'14) received B.Eng., M.Eng and Ph.D. degrees in Engineering from Keio University, Yokohama, Japan, in 2006, 2008, and 2013. In 2014, he joined Nagaoka University of Technology, where he is currently an Assistant Professor in the Department of Electrical, Electronics and Information Engineering, Faculty of Technology. His research interests are in the field of filter bank design, image coding, and image processing.