

PAPER

***M*-Channel Paraunitary Filter Banks Based on Direct Lifting Structure of Building Block and Its Inverse Transform for Lossless-to-Lossy Image Coding**

Taizo SUZUKI^{†a)}, *Student Member* and Masaaki IKEHARA[†], *Member*

SUMMARY This paper presents a paraunitary filter bank (PUFB) based on a direct lifting structure of a building block and its inverse transform for lossless-to-lossy image coding. Although the conventional lifting-based filter banks (LFBs), which are constructed by lifting structures with integer coefficients and rounding operations, suffer from degradation of coding performance due to much rounding error generated by cascading lifting structures, our proposals can be applied to any PUFB without losing many ones because building blocks can be applied to every lifting block as it is. It is constructed with very simple structures and many rounding operations are eliminated. Additionally, the number of rounding operations is reduced more by using two-dimensional block transform (2DBT) of separated transform to each building block. As result, even though the proposed PUFBs require a little side information block (SIB), they show better coding performance in lossless-to-lossy image coding than the conventional ones.

key words: paraunitary filter bank (PUFB), direct lifting, lossless-to-lossy image coding, side information block (SIB)

1. Introduction

With the rapid development of hardware such as PCs and mobile phones etc. and the continual expansion of broadband, lossless-to-lossy image coding, which is the unification of lossy and lossless image coding, is demanded to obtain higher quality and compression ratio. Because JPEG [1], which is the international standard in image coding, separately uses discrete cosine transform (DCT) [2] and differential pulse code modulation (DPCM) for lossy and lossless image coding, respectively, it can not achieve lossless-to-lossy image coding. Besides it, JPEG2000 [3], which is the next generation standard based on discrete wavelet transform (DWT) [4], has achieved it by lifting structures [5] with integer coefficients and rounding operations.

On the other hand, DCT and DWT in the international standards and the next generation one are the special forms of linear-phase filter banks (LPFBs) [6]. LPFBs correspond to a subclass of nonlinear-phase cases. Therefore, nonlinear-phase filter banks (NLPFBs) can be expected to have the potential to improve the performance or the quality in many applications. Paraunitary filter banks (PUFBs) [7] are the representative in NLPFBs and show good coding perfor-

mance in lossy image coding. Then, we have presented lifting-based PUFBs (LBPUBFs) for lossless-to-lossy image coding and solved a problem of error generated by many rounding operations in [8]. Moreover, many lifting-based filter banks (LFBs) have been researched [9]–[13]. But, any LFB suffer from degradation of coding performance due to much rounding error generated by cascading many lifting structures.

This paper presents a PUFB based on a direct lifting structure of a building block and its inverse transform for lossless-to-lossy image coding. The direct lifting structure is a family of block lifting structures [14] which achieves reduction of the number of rounding operations. Although our previous work [15] has reduced many rounding operations, a hard restriction must be imposed. Our proposals can be applied to any PUFB without losing many advantages since many lifting structures do not need to be cascaded because building blocks can be applied to every lifting block as it is. It is constructed with very simple structures and many rounding operations are eliminated. Additionally, the number of rounding operations is reduced more by using two-dimensional block transform (2DBT) of separated transforms on each building block. As result, even though the proposed PUFBs require a little side information block (SIB), they show better coding performance in lossless-to-lossy image coding than the conventional ones.

Notations: \mathbf{I} , \mathbf{M}^T , \mathbf{M}^\dagger , $\mathbf{M}^{[M]}$ and $\text{diag}\{\cdot\}$ are an identity matrix, a transpose of matrix, a conjugate transpose of matrix, $M \times M$ matrix and a block diagonal matrix, respectively. For simplicity, we omit matrix sizes when they are obvious.

2. Preliminaries

2.1 Lifting Structure

The lifting structure [5] is researched widely for various applications. Since the structure achieves integer-to-integer transform, lossless-to-lossy image coding can be implemented. In Fig. 1(a), which shows basic lifting structures, the analysis input signals x_i and x_j , the analysis output and synthesis input signals y_i and y_j , the synthesis output signals z_i and z_j and lifting coefficients T_0 and T_1 are represented by

$$\begin{aligned} y_j &= x_j + \text{round}[T_0 x_i] \\ y_i &= x_i + \text{round}[T_1 y_j] \end{aligned}$$

Manuscript received October 28, 2009.

Manuscript revised March 25, 2010.

[†]The authors are with the Department of Electronics and Electrical Engineering, Keio University, Yokohama-shi, 223-8522 Japan.

a) E-mail: suzuki@tkhm.elec.keio.ac.jp

DOI: 10.1587/transfun.E93.A.1457

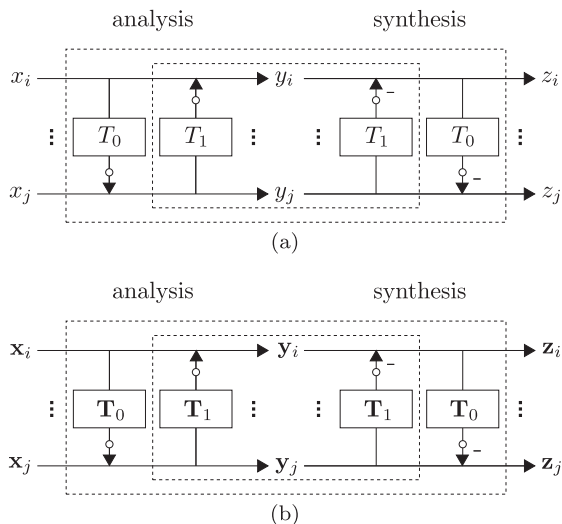


Fig. 1 Lifting structures: (a) basic lifting structures, (b) block lifting structures.

and

$$z_i = y_i - \text{round}[T_1 y_j] = x_i$$

$$z_j = y_j - \text{round}[T_0 z_i] = x_j$$

in analysis and synthesis part, respectively, where $\text{round}[\cdot]$ is rounding of (\cdot) .

In this case, the lifting matrices and their inverse matrices are as follows:

$$\begin{bmatrix} 1 & T_0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & T_0 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & -T_0 \\ 0 & 1 \end{bmatrix},$$

$$\begin{bmatrix} 1 & 0 \\ T_1 & 1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 & 0 \\ T_1 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ -T_1 & 1 \end{bmatrix}.$$

2.2 Block Lifting Structure

Conversely, the block lifting structure [14] is known as an efficient one. The structure achieves higher compression ratio because it can merge many rounding operations. It is clear that the number of rounding operations is reduced from N^2 to N when the size of the lifting coefficient matrix \mathbf{T} is $N \times N$. In Fig. 1(b), which shows block lifting structures, the analysis input signals vector \mathbf{x}_i and \mathbf{x}_j , the analysis output and synthesis input signals vector \mathbf{y}_i and \mathbf{y}_j , the synthesis output signals vector \mathbf{z}_i and \mathbf{z}_j and lifting coefficient matrices \mathbf{T}_0 and \mathbf{T}_1 are represented by

$$\mathbf{y}_j = \mathbf{x}_j + \text{round}[\mathbf{T}_0 \mathbf{x}_i]$$

$$\mathbf{y}_i = \mathbf{x}_i + \text{round}[\mathbf{T}_1 \mathbf{y}_j]$$

and

$$\mathbf{z}_i = \mathbf{y}_i - \text{round}[\mathbf{T}_1 \mathbf{y}_j] = \mathbf{x}_i$$

$$\mathbf{z}_j = \mathbf{y}_j - \text{round}[\mathbf{T}_0 \mathbf{y}_i] = \mathbf{x}_j$$

in analysis and synthesis part, respectively.

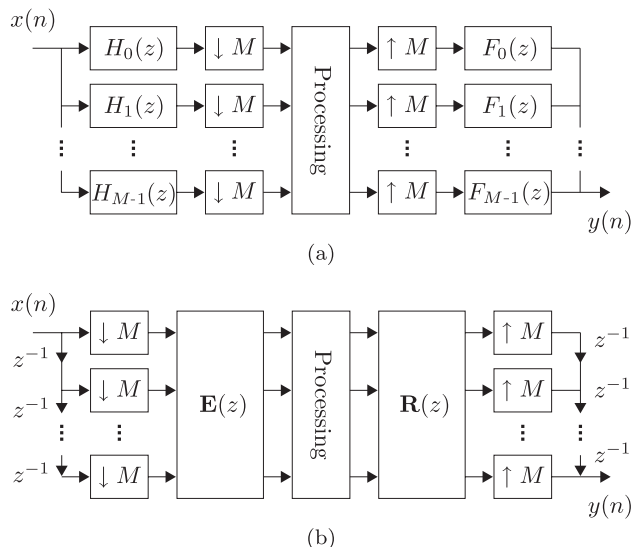


Fig. 2 M -channel maximally decimated filter bank (FB): (a) basic structure, (b) polyphase structure.

In this case, the block lifting matrices and their inverse matrices are as follows:

$$\begin{bmatrix} \mathbf{I} & \mathbf{T}_0 \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \begin{bmatrix} \mathbf{I} & \mathbf{T}_0 \\ \mathbf{0} & \mathbf{I} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{I} & -\mathbf{T}_0 \\ \mathbf{0} & \mathbf{I} \end{bmatrix},$$

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{T}_1 & \mathbf{I} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{T}_1 & \mathbf{I} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{T}_1 & \mathbf{I} \end{bmatrix}.$$

2.3 Paraunitary Filter Banks (PUFBs)

Filter banks (FBs) [6] have found many applications such as in speech, audio, image and video compression, statistical signal processing, discrete multitone modulation and channel equalization. Figure 2(a) shows an M -channel maximally decimated FB, where $H_k(z)$ and $F_k(z)$ are the k -th ($k = 0, 1, \dots, M-1$) analysis and synthesis filter, respectively. Also Fig. 2(b) shows a polyphase structure of FB. The analysis and synthesis filters are represented by using the polyphase matrices $\mathbf{E}(z)$ and $\mathbf{R}(z)$ as follows:

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

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

where

$$\mathbf{e}(z) = \begin{bmatrix} 1 & z^{-1} & z^{-2} & \cdots & z^{-(M-1)} \end{bmatrix}.$$

If $\mathbf{E}^\dagger(z^{-1})\mathbf{E}(z) = \mathbf{I}$ and $\mathbf{R}(z) = \mathbf{E}^\dagger(z^{-1})$, the FBs are called paraunitary filter banks (PUFBs).

In this paper, we consider PUFBs where the number of channels is M (even) and all filter lengths are MK ($K \in \mathbb{N}$). They are efficiently designed and implemented by the lattice structure as [7]

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

where

$$\Lambda_k(z) = \begin{bmatrix} \mathbf{I}^{[M-\gamma_k]} & \mathbf{0} \\ \mathbf{0} & z^{-1}\mathbf{I}^{[\gamma_k]} \end{bmatrix}$$

and \mathbf{G}_k is an $M \times M$ arbitrary orthogonal matrix. Although γ_k is arbitrary integer $0 \leq \gamma_k \leq M-1$ and \mathbf{G}_k can be constructed by complex values, we set $\gamma_k = M/2$ and \mathbf{G}_k is constructed by real values for simplicity. Thus $\Lambda_k(z)$ is denoted as $\Lambda(z)$.

3. PUFB Based on a Direct Lifting Structure of a Building Block and Its Inverse Transform

In this section, an efficient lifting structure of PUFB for lossless-to-lossy image coding is presented. The system is implemented by consideration of the block parallel system [15] of each building block and its inverse transform in PUFB. It means that their original matrices are preserved and directly used as each lifting block even after the system was factorized into lifting structures.

3.1 Direct Lifting Structure

We consider to process two individual signals by each building block \mathbf{G} and its inverse transform $\mathbf{G}^{-1} = \mathbf{G}^T$ as shown in the left part of Fig. 3. The output signals are transformed by

$$\begin{bmatrix} \mathbf{y}_i \\ \mathbf{t}_i \end{bmatrix} = \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^T \end{bmatrix} \begin{bmatrix} \mathbf{x}_i \\ \mathbf{s}_i \end{bmatrix}. \quad (1)$$

First, let us define \mathbf{W}_k as

$$\mathbf{W}_k = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^T \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{G}^T \\ \mathbf{G} & \mathbf{0} \end{bmatrix}. \quad (2)$$

And, the block lifting matrix is multiplied from the right sides of (2) as

$$\mathbf{W}_k \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{G} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} -\mathbf{I} & \mathbf{G}^T \\ \mathbf{G} & \mathbf{0} \end{bmatrix}. \quad (3)$$

Next, the other block lifting matrix is multiplied from the right sides of (3) as

$$\begin{bmatrix} -\mathbf{I} & \mathbf{G}^T \\ \mathbf{G} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{G}^T \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} -\mathbf{I} & \mathbf{0} \\ \mathbf{G} & \mathbf{I} \end{bmatrix} \\ = \begin{bmatrix} -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{G} & \mathbf{I} \end{bmatrix}. \quad (4)$$

\mathbf{W}_k can be factorized into the lifting structures using (2)–(4) as

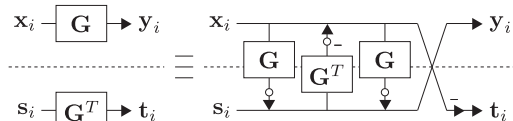


Fig. 3 A direct lifting structure of each building block and its inverse transform in PUFB (white circles: rounding operations).

$$\mathbf{W}_k = \begin{bmatrix} -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{G} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{G}^T \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{G} & \mathbf{I} \end{bmatrix}.$$

Consequently, $\text{diag}\{\mathbf{G}, \mathbf{G}^T\}$ in (1) can be factorized into the complete block lifting structures such as

$$\begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}^T \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{G} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{G}^T \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{G} & \mathbf{I} \end{bmatrix}. \quad (5)$$

Thus the block parallel system of each building block and its inverse transform can be efficiently implemented by block lifting structure as shown in the right part of Fig. 3. Any building block in PUFBs can be directly applied to lossless-to-lossy image coding even if they are designed for only lossy image coding. However the systems in (5) have a problem that the transformed signals by inverse transforms of building block are not suitable for image compression. It is a solution where a condition $\mathbf{G}^T = \mathbf{G}$ is imposed as in [15], but it may make filter performance degrade.

3.2 Realization of Efficient PUFB Based on a Direct Lifting Structure by Side Information Block (SIB)

We can easily come up with a simple realization for lossless-to-lossy image coding which is applied after an $N \times N$ image \mathbf{X} was segmented to $M \times M$ block \mathbf{x}_k ($0 \leq k \leq n-1$, $n = (N/M)^2$). The segmented block \mathbf{x}_k is sequentially transformed by building block in PUFB. In parallel, $M \times M$ block \mathbf{s}_k have to be transformed by inverse transform of building block in (5). However, the transformed signals by it are not suitable for image compression as described in the previous subsection. Then we prepare $M \times M$ side information block (SIB) \mathbf{s}_0 as null matrix and \mathbf{s}_k is iteratively transformed from \mathbf{s}_0 by

$$\mathbf{s}_k = \mathbf{G}^T \mathbf{s}_{k-1} \quad \text{for } k = 1, 2, \dots, n-1$$

As result, the proposed realization for lossless-to-lossy image coding is presented by

$$\begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_{n-1} \\ \mathbf{s}_{n-1} \end{bmatrix} = \begin{bmatrix} \mathbf{G} & & & & \\ & \mathbf{G} & & & \\ & & \ddots & & \\ & & & \mathbf{G} & \\ \mathbf{0} & & & & (\mathbf{G}^T)^n \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{n-1} \\ \mathbf{s}_0 \end{bmatrix} \quad (6)$$

and \mathbf{s}_{n-1} is also encoded with all of \mathbf{y}_k s. The left part of Fig. 4 shows the system in (6). Note that $\mathbf{s}_{n-1} \neq \mathbf{0}$ due to rounding error in each of lifting structures. Also it is clear that the combination of \mathbf{G} and \mathbf{G}^T can be factorized into lifting structures by (5) as shown in dashed line area in Fig. 4.

Finally the proposed lossless-to-lossy image coding is summarized as follows:

- 1) Lossless bit stream with SIB \mathbf{s}_{n-1} can be obtained by using (6) and a progressive encoder with PSNR scalability like SPIHT [16] and EZW-IP [17].
- 2) In lossless mode, the image is reconstructed from all

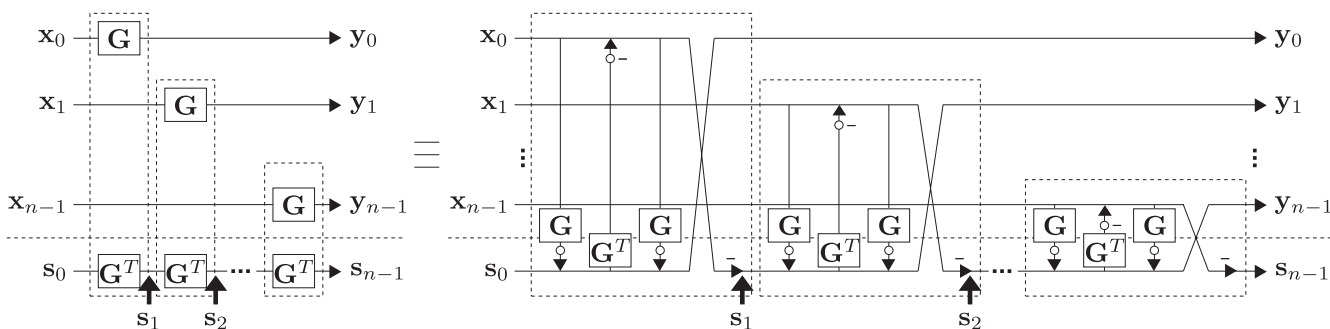


Fig. 4 Realization of a direct lifting structure of a building block and its inverse transform (white circles: rounding operations).

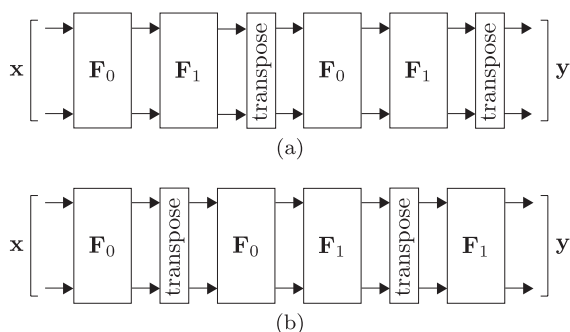


Fig. 5 Two-dimensional block transform (2DBT) when $n = 2$: (a) standard transform, (b) separated transform to each building block.

lossless bit stream and s_{n-1} . The each block s_k is successively inverse transformed by $s_k = \mathbf{G}s_{k+1}$ without any loss.

- 3) In lossy mode, the image is reconstructed by interrupting the lossless bit stream without using s_{n-1} .

3.3 More Efficient Improvement by Two-Dimensional Block Transform (2DBT)

When we apply a block transform matrix \mathbf{F} to a two-dimensional (2D) input signal \mathbf{x} in column- and row-wise separately, the 2D output signal \mathbf{y} is expressed by

$$\mathbf{y} = (\mathbf{F}(\mathbf{F}\mathbf{x})^T)^T = \mathbf{F}\mathbf{x}\mathbf{F}^T. \quad (7)$$

We call it a 2D separable block transform. If the transform \mathbf{F} is factorized as $\mathbf{F} = \mathbf{F}_{n-1} \cdots \mathbf{F}_1 \mathbf{F}_0$ ($n \in \mathbb{N}$), (7) is represented by

$$\begin{aligned} \mathbf{y} &= \mathbf{F}_{n-1} \cdots \mathbf{F}_1 \mathbf{F}_0 \mathbf{x} (\mathbf{F}_{n-1} \cdots \mathbf{F}_1 \mathbf{F}_0)^T \\ &= \mathbf{F}_{n-1} \cdots \mathbf{F}_1 \mathbf{F}_0 \mathbf{x} \mathbf{F}_0^T \mathbf{F}_1^T \cdots \mathbf{F}_{n-1}^T. \end{aligned}$$

This equation means that 2D block transform (2DBT) by \mathbf{F}_{k+1} is applied after 2DBT by \mathbf{F}_k ($0 \leq k \leq n-2$, $k \in \mathbb{N}$). Therefore standard transform and separated transform to each building block when $n = 2$ are shown in Fig. 5(a) and (b), respectively. In this paper, we regard \mathbf{F} is $\mathbf{E}(z)$. Thus when each building block \mathbf{G} in PUFB is applied to an $M \times M$ input signal \mathbf{x} in column- and row-wise separately, it is expressed by

Table 1 The numbers of rounding operations.

filter	Conv. PUFBs [8]	Prop. PUFBs
4×8	23	12
4×12	31	18
8×16	62	24
8×24	82	36

$$(\mathbf{G}(\mathbf{G}\mathbf{x})^T)^T = \mathbf{G}\mathbf{x}\mathbf{G}^T \triangleq \mathbf{G}_{2D}\mathbf{x}.$$

Consequently, (5) and (6) are represented by

$$\begin{bmatrix} \mathbf{G}_{2D} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_{2D}^T \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{G}_{2D} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{G}_{2D}^T \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{G}_{2D} & \mathbf{I} \end{bmatrix}$$

and

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \\ s_{n-1} \end{bmatrix} = \begin{bmatrix} \mathbf{G}_{2D} & & & & \\ & \mathbf{G}_{2D} & & & \\ & & \ddots & & \\ & & & \mathbf{G}_{2D} & \\ \mathbf{0} & & & & (\mathbf{G}_{2D}^T)^n \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \\ s_0 \end{bmatrix}.$$

Obviously, delay matrix $\Lambda(z)$ in PUFB is also denoted as $\Lambda_{2D}(z)\mathbf{x} \triangleq \Lambda(z)\mathbf{x}\Lambda(z)^T$. These lifting matrices by 2DBT of separated transform to each building block are more efficient improvement for lossless-to-lossy image coding because the number of rounding operations in one system is reduced from $3M$ to $3M/2$ and the rounding error is not generated so much. When filter sizes are 4×8 , 4×12 , 8×16 and 8×24 , the number of rounding operations is shown in Table 1.

4. Results

In this paper, the proposed PUFBs are validated by lossless bit rate [bpp] in lossless image coding, and PSNR [dB] in lossy image coding. For lossless-to-lossy image coding, we used 4×8 , 4×12 , 8×16 and 8×24 PUFBs which have same filter properties as LBPUB [8], and applied the proposed direct lifting factorization to each building block in them. Frequency responses of the designed PUFBs are shown in Fig. 6. As targets for comparison, LBPUBs, which use same filters as our PUFBs and different lifting factorization, and 5/3- and 9/7-tap DWTs [4] were chosen.

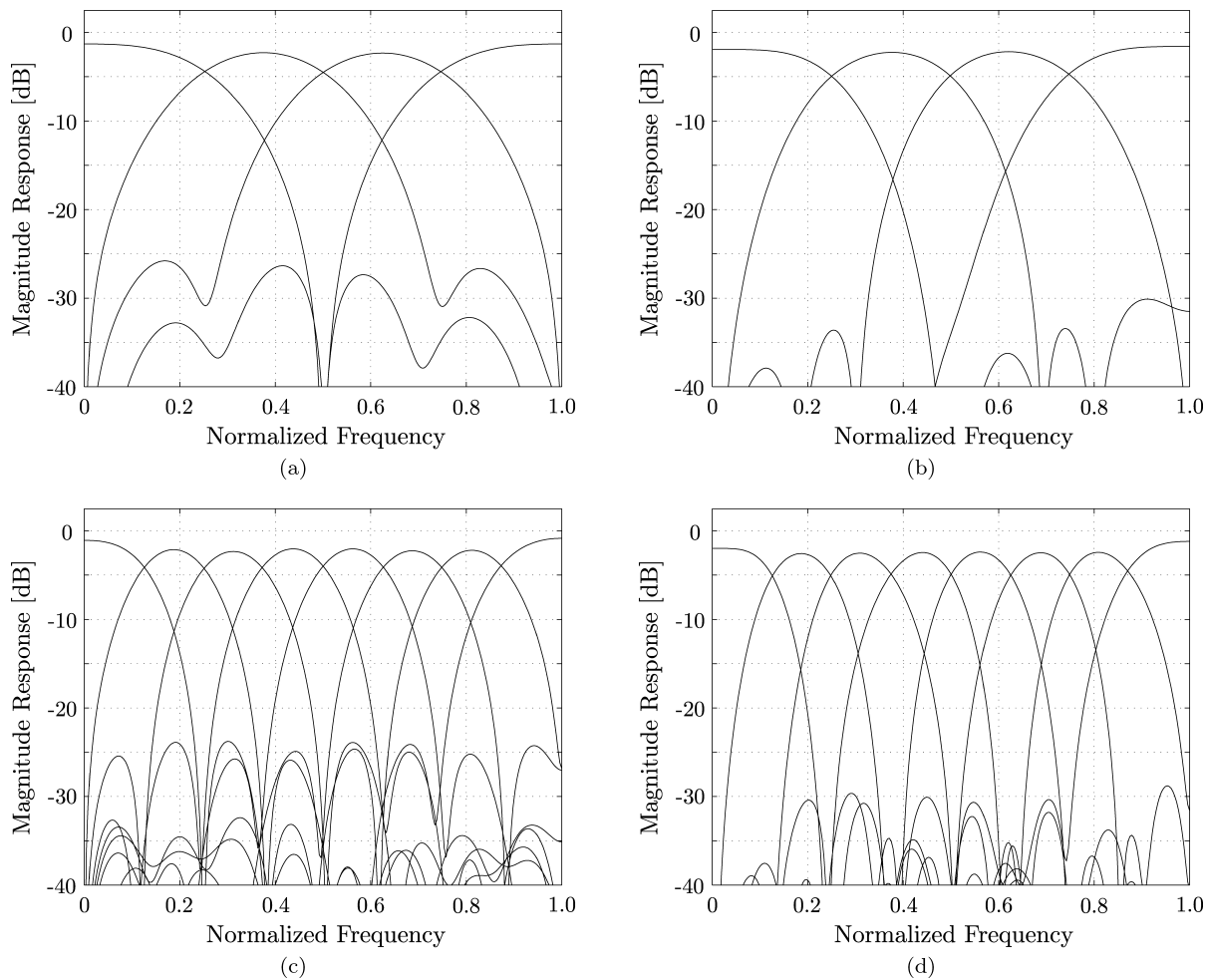


Fig. 6 Frequency responses of the proposed PUFBs and the conventional PUFBs: (a) 4×8 , (b) 4×12 , (c) 8×16 , (d) 8×24 .

Table 2 Comparison of lossless image coding.

Image	5/3-tap DWT [4]	Lossless bit rate [bpp]							
		Conv. PUFBs [8]				Prop. PUFBs			
512x512		4x8	4x12	8x16	8x24	4x8	4x12	8x16	8x24
Baboon	6.17	6.24	6.21	6.25	6.23	6.23	6.20	6.24	6.22
Barbara	4.87	4.88	4.79	4.88	4.82	4.83	4.74	4.77	4.70
Boat	5.10	5.14	5.10	5.16	5.13	5.11	5.07	5.10	5.07
Elaine	5.11	5.17	5.12	5.12	5.06	5.16	5.11	5.08	5.01
Finger1	5.84	5.82	5.74	5.70	5.68	5.81	5.73	5.67	5.64
Finger2	5.60	5.63	5.50	5.48	5.43	5.60	5.48	5.42	5.36
Goldhill	5.01	5.14	5.11	5.15	5.14	5.11	5.09	5.09	5.08
Grass	6.06	6.11	6.07	6.07	6.05	6.10	6.07	6.06	6.04
Lena	4.49	4.66	4.61	4.72	4.68	4.62	4.58	4.63	4.59
Pepper	4.85	4.95	4.90	5.00	4.95	4.92	4.88	4.95	4.89
Avg.	5.31	5.37	5.32	5.35	5.32	5.35	5.30	5.30	5.26

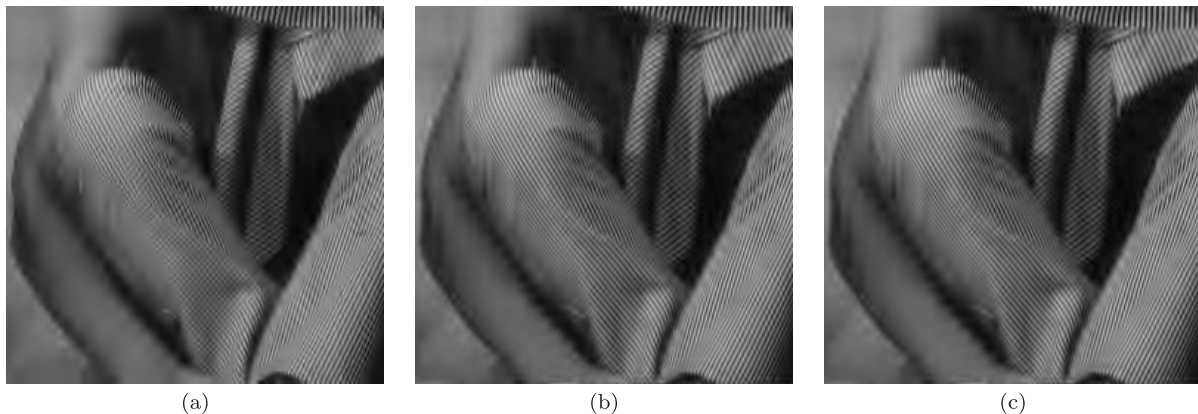
To evaluate transform performance equally, a very common wavelet-based coder EZW-IP [17] was adopted for all. Also, for image boundary, the periodic extension was used in our proposals and LBPUBs, and the symmetric extension was used in 5/3- and 9/7-tap DWTs. Moreover, we used ten 512×512 gray scale images such as “Barbara” and “Lena” etc.

4.1 Application to Lossless Image Coding

First, the designed PUFBs are applied to lossless image coding. SIB s_{n-1} is just encoded with no compression because it has only very few bit data size:

Table 3 Comparison of lossy image coding.

Image 512×512	Bit rate [bpp]	PSNR [dB]									
		9/7-tap DWT [4]	Conv. PUFBs [8]				Prop. PUFBs				
			4×8	4×12	8×16	8×24	4×8	4×12	8×16	8×24	
Baboon	0.25	22.82	22.61	22.73	22.69	22.77	22.61	22.74	22.69	22.78	
	0.50	24.60	24.53	24.70	24.66	24.73	24.54	24.71	24.68	24.73	
	1.00	27.61	27.58	27.74	27.64	27.71	27.60	27.76	27.68	27.75	
Barbara	0.25	27.25	27.11	27.62	27.97	28.38	27.12	27.64	27.99	28.41	
	0.50	30.49	30.67	31.23	31.54	31.98	30.70	31.25	31.62	32.13	
	1.00	34.88	35.17	35.68	35.70	36.07	35.28	35.79	35.96	36.44	
Boat	0.25	28.46	28.11	28.33	28.32	28.41	28.11	28.33	28.35	28.44	
	0.50	31.39	31.18	31.36	31.25	31.38	31.20	31.38	31.32	31.47	
	1.00	34.51	34.58	34.76	34.56	34.65	34.68	34.85	34.74	34.88	
Elaine	0.25	31.51	30.97	31.29	31.08	31.34	30.99	31.32	31.12	31.40	
	0.50	32.94	32.26	32.96	32.60	33.29	32.37	33.00	32.88	33.44	
	1.00	34.61	34.51	34.80	34.86	35.24	34.61	34.89	35.12	35.54	
Finger1	0.25	23.51	22.87	23.20	23.66	23.79	22.87	23.19	23.66	23.79	
	0.50	25.99	25.70	26.12	26.52	26.59	25.70	26.10	26.53	26.71	
	1.00	29.07	29.37	29.80	30.11	30.21	29.40	29.80	30.20	30.46	
Finger2	0.25	24.18	22.52	23.33	23.82	24.55	22.53	23.33	23.82	24.58	
	0.50	27.53	25.83	26.61	27.03	27.62	25.84	26.61	27.07	27.88	
	1.00	30.90	29.76	30.61	31.65	32.40	29.80	30.64	31.89	32.66	
Goldhill	0.25	29.67	28.79	29.00	29.09	29.14	28.80	29.02	29.12	29.16	
	0.50	31.53	31.35	31.52	31.59	31.67	31.39	31.56	31.67	31.73	
	1.00	34.51	34.59	34.74	34.67	34.69	34.67	34.82	34.87	34.90	
Grass	0.25	24.36	24.18	24.35	24.48	24.57	24.19	24.35	24.49	24.59	
	0.50	26.09	26.14	26.30	26.41	26.46	26.15	26.30	26.43	26.52	
	1.00	28.68	28.89	29.08	29.15	29.22	28.93	29.10	29.22	29.32	
Lena	0.25	32.53	31.38	32.09	31.90	32.23	31.42	32.13	31.95	32.31	
	0.50	35.52	34.93	35.39	35.07	35.39	35.00	35.49	35.22	35.59	
	1.00	38.65	38.69	38.88	38.37	38.54	38.89	39.10	38.92	39.09	
Pepper	0.25	31.94	30.58	31.64	30.81	31.67	30.60	31.68	30.89	31.73	
	0.50	34.48	33.40	34.33	33.20	34.17	33.52	34.38	33.44	34.29	
	1.00	36.09	35.94	36.19	35.73	35.91	36.06	36.33	35.97	36.19	

**Fig. 7** Enlarged images of “Barbara” in lossy image coding (bit rate: 0.25 [bpp]): (a) 9/7-tap DWT [4], (b) conventional 8×24 PUFB [8], (c) proposed 8×24 PUFB.

$$\text{SIB bit rate [bpp]} = \frac{\lceil \log_2(|s_{n-1}|_{max}) + 1 \rceil M^2 [\text{bit}]}{\text{Total number of pixels [pixel]}}$$

where $\lceil \cdot \rceil$ is ceiling of (\cdot) . For example, the size is only 512 [bits] (0.01953125 [bpp]) in 512×512 image “Barbara” transformed by the proposed 8×24 PUFB. The comparison of

$$\text{Lossless bit rate [bpp]} = \frac{\text{Total number of bits [bit]}}{\text{Total number of pixels [pixel]}} + \text{SIB bit rate [bpp]}$$

in lossless image coding is shown in Table 2. The proposed PUFBs showed better coding performance than all of the conventional methods in especially 8×24 PUFB. Even though the proposed PUFBs and the conventional PUFBs have same transfer function, all proposed PUFBs with less rounding operations show better coding performance than the conventional PUFBs. We consider that this is mainly due to the reduction of rounding operations as shown in Table 1.

4.2 Application to Lossy Image Coding

If lossy compressed data is required, it can be achieved by interrupting the obtained lossless bit stream without SIB \mathbf{s}_{n-1} . In this paper, \mathbf{s}_{n-1} is replaced by $\mathbf{0}$ for simplicity. The comparison of

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

where MSE is the mean squared error in lossy image coding are shown in Table 3. Furthermore, Fig. 7 shows the enlarged images of “Barbara” which are lossy compressed images by the proposed PUFBs and the conventional methods when bit rate is 0.25 [bpp]. In Fig. 7, it is obvious that the high frequency region of the reconstructed image by our PUFB is well-approximated because M -channel PUFB has more excellent frequency resolution in high frequency than DWTs. Moreover, although JPEG2000 is implemented by two different filter in lossy and lossless mode, our proposals can show better coding performance with only one filter.

5. Conclusion

In this paper, we presented a paraunitary filter bank (PUFB) based on a direct lifting structure of a building block and its inverse transform for lossless-to-lossy image coding. Our proposals are obtained by the following processes:

- 1) The system $\text{diag}\{\mathbf{G}, \mathbf{G}, \dots, \mathbf{G}, \{\mathbf{G}^T\}^n\}$, where \mathbf{G} is a building block in PUFB, is prepared.
- 2) Side information block (SIB) \mathbf{s}_0 to be iteratively transformed by inverse transform \mathbf{G}^T is prepared.
- 3) Each combination of \mathbf{G} and \mathbf{G}^T in 1), which is $\text{diag}\{\mathbf{G}, \mathbf{G}^T\}$, is factorized into direct lifting structures.
- 4) Additionally, the number of rounding operations is reduced by using two-dimensional block transform (2DBT) of separated transforms on each building block.
- 5) In lossless mode, the image is reconstructed from all encoded lossless bit stream and SIB.
- 6) In lossy mode, the image is reconstructed by interrupting the lossless bit stream without SIB.

As result, our proposals show better coding performance than the conventional methods for lossless-to-lossy image coding. In future work, side information and computational cost should be deleted and reduced, respectively.

Acknowledgment

This work is supported in part by a Grant-in-Aid for the Global Center of Excellence for High-Level Global Cooperation for Leading-Edge Platform on Access Spaces from the Ministry of Education, Culture, Sport, Science, and Technology in Japan.

References

- [1] W. Pennebaker and J. Mitchell, JPEG, Still Image Data Compression Standard, Van Nostrand, NY, 1993.
- [2] K.R. Rao and P. Yip, Discrete Cosine Transform Algorithms, Academic Press, 1990.
- [3] A. Skodras, C. Christopoulos, and T. Ebrahimi, “The JPEG2000 still image compression standard,” IEEE Trans. Signal Process. Mag., vol.18, no.5, pp.36–58, Sept. 2001.
- [4] G. Strang and T. Nguyen, Wavelets and Filter Banks, Wellesley-Cambridge Press, 1996.
- [5] W. Sweldens, “The lifting scheme: A custom-design construction of biorthogonal wavelets,” Appl. Comput. Harmon. Anal., vol.3, no.2, pp.186–200, 1996.
- [6] P.P. Vaidyanathan, Multirate Systems and Filter Banks, Englewood Cliffs, Prentice Hall, NJ, 1992.
- [7] X. Gao, T.Q. Nguyen, and G. Strang, “On factorization of M -channel paraunitary filterbanks,” IEEE Trans. Signal Process., vol.49, no.7, pp.1433–1446, July 2001.
- [8] T. Suzuki, Y. Tanaka, and M. Ikehara, “Lifting-based paraunitary filterbanks for lossy/lossless image coding,” Proc. EUSIPCO’06, Florence, Italy, Sept. 2006.
- [9] K. Komatsu and K. Sezaki, “Design of lossless block transforms and filter banks for image coding,” IEICE Trans. Fundamentals, vol.E82-A, no.8, pp.1656–1664, Aug. 1999.
- [10] M. Okuda, S.K. Mitra, M. Ikehara, and S. Takahashi, “Filter banks with nonlinear lifting steps for lossless image compression,” IEICE Trans. Fundamentals, vol.E84-A, no.3, pp.797–801, March 2001.
- [11] T.D. Tran, “ M -channel linear phase perfect reconstruction filter bank with rational coefficients,” IEEE Trans. Circuits Syst. I, vol.49, no.7, pp.914–927, July 2002.
- [12] 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.
- [13] 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.
- [14] S. Iwamura, Y. Tanaka, and M. Ikehara, “An efficient lifting structure of biorthogonal filter banks for lossless image coding,” Proc. ICIP’07, San Antonio, TX, Sept. 2007.
- [15] T. Suzuki and M. Ikehara, “Lifting factorization based on block parallel system of M -channel perfect reconstruction filter banks,” Proc. EUSIPCO’09, Glasgow, Scotland, Aug. 2009.
- [16] 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, 1996.
- [17] Z. Liu and L.J. Karam, “An efficient embedded zerotree wavelet image codec based on intraband partitioning,” Proc. ICIP’00, Vancouver, British Columbia, Canada, Sept. 2000.



Taizo Suzuki received the B.E. and M.E. degrees in electrical engineering from Keio University, Yokohama, Japan, in 2004 and 2006, respectively. He joined Toppan Printing Co., Ltd., Tokyo, Japan, from 2006 to 2008. He is currently working toward a Ph.D. degree and belongs a Global Center of Excellence (G-COE) research assistant (RA) at Keio University. His research interests are in the field of filter bank design and its application for multimedia signal processing.



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