Reversible Symmetric Non-Expansive Convolution:  
An Effective Image Boundary Processing for  
M-Channel Lifting-based Linear-Phase Filter Banks

Taizo Suzuki, Member, IEEE and Masaaki Ikehara, Senior  
Member, IEEE

Abstract—We present an effective image boundary processing for  
M-channel (\(M \in \mathbb{N}, M \geq 2\)) lifting-based linear-phase filter banks  
(L-LPFBs) that are applied to unifed lossy and lossless image  
compression (coding), i.e., lossy-to-lossless image coding. The reversible  
symmetric extension (RevSE) we propose is achieved by manipulating  
building blocks on the image boundary and reawakening the symmetry  
of “each building block” that has been lost due to rounding error  
on each lifting step. Moreover, complexity is reduced by extending  
non-expansive convolution, called reversible symmetric non-expansive  
convolution (RevSNEC), because the number of input signals does not  
even temporarily increase. Our method not only achieves reversible  
boundary processing, but also is comparable to irreversible SE (IrrSE)  
in lossy image coding and outperformed periodic extension (PE) in lossy-  
to-lossless image coding.

Index Terms—Lifting-based linear-phase filter bank (L-LPFB), lossy-  
to-lossless image coding, reversible symmetric extension (RevSE),  
reversible symmetric non-expansive convolution (RevSNEC).

I. INTRODUCTION

Filter banks (FBs) [1] have been contributing to signal processing  
and communication tools for many years. They have often been  
employed as transforms in image compression (coding) because they  
have extensive frequency selectivity and high coding gain. FBs with  
linear-phase (LP) properties, i.e., LPFBs [2–8], including lapped  
transforms (LTs) [9–14] are particularly one of the most useful  
transformations for image coding. LPFBs can be easily designed and  
they simply overcome problems with image boundary distortion via  
symmetric extension (SE) [2] that maintain continuity at the image  
boundary. The output signals can be reconstructed without image  
boundary distortion by using symmetry even if the extended signals  
are not transmitted to the synthesis bank. Symmetry means that when  
an input signal vector for a building block is the reflected vector of  
another input signal vector, their output signal vectors also have the  
same relationship. Smith and Eddins [2] achieved such SE by using  
the symmetry of “whole FB.”

Lifting structures with rounding operations have also been presented  
[15–17]. A transform only composed of lifting structures and integer  
multipliers achieves an integer-to-integer transform that maps  
integer input signals to integer output signals. Thus, many lifting-  
based FBs (L-FBs) [18–23] including lifting-based LTs [24–26] have  
been proposed, and the structures have led FBs to achieve lossy-  
to-lossless image coding, which is unified lossy and lossless image  
coding, such as JPEG 2000 [27] and JPEG XR [28]. Discrete wavelet  
transforms (DWTs) [29] in JPEG 2000 have demonstrated excellent  
coding, whereas L-FBs have greater potential for lossy-to-lossless  
image coding due to their higher degrees of design freedom than DWTs.  
Several reversible smooth extensions have in fact been applied  
to 2-channel L-FBs including DWTs [21], [22], [29]. However,  
no smooth boundary extensions can directly be applied to generalized  
M-channel (\(M \in \mathbb{N}, M \geq 2\)) L-FBs. \(4 \times 8\) hierarchical lifted  
transform (HLT) [30] in JPEG XR is well-known as one of the most  
popular L-FBs. Although it uses a constrained case of the image  
boundary solution we propose later, \(M \times 2M\) L-LPFB case, it causes  
(a bit) boundary error because SE cannot be precisely achieved  
by ignoring the scaling coefficients. Periodic extension (PE), which  
causes image boundary distortion, is often reluctantly used for lossy-  
to-lossless image coding based on \(M\)-channel L-FBs even if it has  
LP properties because rounding error on each lifting step corrupts  
symmetry [26].

We solve the image boundary problem in lossy-to-lossless image  
coding based on \(M\)-channel lifting-based LPFBs (L-LPFBs) by  
focusing on the symmetry of “each building block” unlike that of  
the “whole FB” [2]. Although the proposed reversible SE (RevSE) is not  
completely equivalent to SE, which is called irreversible SE (IrrSE)  
to distinguish it from RevSE, it can obtain very similar smoothness  
to IrrSE at the image boundary even if rounding operations are  
used. Moreover, complexity is reduced by extending non-expansive  
convolution [13], called reversible symmetric non-expansive  
convolution (RevSNEC), because the number of input signals does not  
even temporarily increase. The proposed RevSNEC can be applied  
to \(M \times MK\) (\(K \in \mathbb{N}, K \neq 0, K\) and \(K\) must only be odd in odd  
channel case) L-LPFBs which are extensions of \(M \times 2M\) L-LPFBs  
as HLS in JPEG XR. The RevSNEC not only achieves reversible  
boundary processing, but also is comparable to IrrSE in lossy image  
coding and outperformed PE in lossy-to-lossless image coding.

The remaining part of this paper is organized as follows: Sec. II  
reviews the lattice structures of LPFBs and explains how we derive  
their lifting structures. Sec. III presents IrrSE using symmetry  
of “each building block” newly in both even and odd channel cases.  
This structures are extended to RevSE and RevSNEC by simple  
matrix manipulations. Filter design examples, lossy-to-lossless image  
coding simulations, and comparisons to the conventional methods are  
presented in Sec. IV. Sec. V concludes the paper.

Boldface letters represent vectors or matrices. \(I_M, J_N, 0, T,  
diag(\cdot), \) and \(det(\cdot)\) respectively denote an \(N \times N\) (\(N \in \mathbb{N}, N \neq 0)\)  
identity matrix, \(N \times N\) reversal identity matrix, null matrix, transpose  
of a matrix, block diagonal matrix, and determinant of a matrix.

II. LATTICE AND LIFTING STRUCTURES OF LPFBs

A. Linear-Phase Filter Banks (LPFBs)

When \(m \in \mathbb{N}, m \neq 0, M = 2m (M\ is\ even)\ and\ M = 2m + 1\  
(M is odd), the type-II analysis polyphase matrix \(E(z)\ in\ M \times MK\  
LPFBs are presented as in Tran [6]:

\[
E(z) = E_0 G_1(z) \cdots G_{K-2}(z) G_{K-1}(z)
\]  

where

\[
E_0 = \Phi_0 W, \quad G_k(z) = \Lambda(z) W \Phi_k W,
\]

\[
\Phi_k = \begin{cases}
\{ \text{diag} \{ U_{e,k}, V_{e,k} \} & (M\ is\ even) \\
\{ \text{diag} \{ U_{o,k}, V_{o,k} \} & (M\ is\ odd, k\ is\ even) \\
\{ \text{diag} \{ U_{o,k}, U_{o,k} \} & (M\ is\ odd, k\ is\ odd),
\end{cases}
\]

\[
\Lambda(z) = \begin{cases}
\{ I_m, z^{-1} I_m \} & (M\ is\ even) \\
\{ I_{m+1}, z^{-1} I_{m+1} \} & (M\ is\ odd),
\end{cases}
\]

\[
W = \begin{cases}
\frac{1}{\sqrt{2}} \begin{bmatrix} I_m & J_m \\
J_m & -I_m \end{bmatrix} & (M\ is\ even) \\
\frac{1}{\sqrt{2}} \begin{bmatrix} I_m & 0 \\
0 & \sqrt{2} \end{bmatrix} & (M\ is\ odd),
\end{cases}
\]

\(U_{e,k}, V_{e,k}, \) and \(V_{o,k}\ are m \times m\ arbitrary\ nonsingular\ matrices,\  
U_{o,k} is an (m + 1) \times (m + 1) or m \times m\ arbitrary\ nonsingular\ matrix.
LPFBs are paraunitary LPFBs (PULPFBs) if all matrices factorizing respectively mean matrices factorizing reconstruction (PR) property matrix is even or odd, and \( w_k \) is a nonzero scalar. Also, either \( U_{e,k} \) or \( V_{e,k} \) and \( U_{o,k} \) and \( V_{o,k} \) are usually replaced by \( I_m \) when \( k \geq 1 \) to eliminate redundancy. If \( E(z) \) is invertible, synthesis polyphase matrix \( R(z) \) can be chosen as the inverse of \( E(z) \), i.e., the perfect reconstruction (PR) property \( R(z)E(z) = I_M \) is satisfied. The LPFBs are paraunitary LPFBs (PULPFBs) if all \( U_{e,k}, V_{e,k}, U_{o,k}, \) and \( V_{o,k} \) are orthogonal matrices and \( w_k \) is 1, and the others are biorthogonal LPFBs (BOLPFBs). In PULPFBs, \( m \) must be \( m \geq 2 \).

As described in Sec. III-A, IrrSE can simply be applied to the LPFBs to improve coding performance.

### B. Lifting-based Linear-Phase Filter Banks (L-LPFBs)

This subsection presents L-LPFBs based on the polyphase matrix in Eq. (1). When all lifting steps are implemented with rounding operations, L-LPFBs achieve integer-to-integer mapping, i.e., the lossless mode. The building block \( G_k(z) \) in Eq. (1) to achieve this is represented as

\[
G_k(z) = \Lambda(z)W\Phi_kW = \Lambda(z)W_L\Phi_kW_R = \Lambda(z)\Xi_k
\]

where

\[
W_L = \begin{bmatrix}
I_m & 0 & I_m & -\frac{1}{2}J_m \\
0 & I_m & 0 & I_m \\
0 & I_m & I_{m+1} & 0 \\
-\frac{1}{2}J_m & 0 & I_m & I_{m+1}
\end{bmatrix} \quad \text{(M is even)}
\]

and

\[
W_R = \begin{bmatrix}
I_m & \frac{1}{2}J_m & 0 & I_m \\
0 & 0 & I_m & -J_m \\
0 & I_{m+1} & 0 & I_{m+1}
\end{bmatrix} \quad \text{(M is odd)}.
\]

The top half of Fig. 1 outlines the building block of L-LPFB in even channel case. Let \( \gamma_k = \Xi_k \phi_k \) and \( \xi_k \) correspond to an \( M \times 1 \) input signal vector for a building block \( \Xi_k \), a reflected output signal vector of \( \Xi_k \), and \( \xi_k \) be an input signal vector for a building block \( \Xi_k \).

### III. IMAGE BOUNDARY PROCESSING FOR L-LPFBs

Let \( \text{symmetry} \) be that of “each building block” unlike that of the “whole FB” in Smith and Eddins [2]. PR is satisfied by using such \( \text{symmetry} \) without receiving redundant signals in the synthesis bank. First, IrrSE is investigated in both case of even and odd channels. Furthermore, RevSE for the lossless mode is presented with \( \text{Cases I and II. Case I} \) means building blocks that do not step over the image boundary, and \( \text{Case II} \) means those that just step over the image boundary. Moreover, the redundancy of RevSE is eliminated by extending non-expansive convolution [13], called RevSNEC.

#### A. Irreversible Symmetric Extension (IrrSE)

Fig. 2 shows IrrSE in even channel case. Let \( \xi_k, J_M\xi_k, \gamma_k, \) and \( \xi_k \) correspond to an \( M \times 1 \) input signal vector for a building block \( \Xi_k, \), a reflected output signal vector of \( \Xi_k, \), and \( \xi_k \) be an input signal vector for a building block \( \Xi_k, \). Symmetry means

\[
\gamma_k = J_M\xi_k.
\]

We demonstrate this \( \text{symmetry} \) is satisfied in both \( \text{Cases I and II.} \)

1) \( \text{Case I (Not Stepping Over Image Boundary):} \) \( \xi_k \) is an input signal vector for \( \Xi_k \) that does not step over the image boundary in this subsection. Let \( \xi_k \) be \( \xi_k = [A_k^T, B_k^T]^T \) \( (M \text{ is even}) \) or \([A_k^T, c_k, B_k^T]^T \) \( (M \text{ is odd}) \) where \( A_k \) and \( B_k \) are \( m \times 1 \) vectors and \( c_k \) is a scalar. Also, let \( U_{o,k} (k \text{ is even}) \) be redefined as

\[
U_{o,k} = [u_k, s_k, t_k]
\]

where \( u_k, s_k, t_k, \) and \( u_k \) correspond to \( m \times m, m \times 1, 1 \times m \) matrices, and a scalar. The output signal vectors \( \gamma_k \) and \( \xi_k \) are expressed by

\[
\gamma_k = \Xi_k\phi_k
\]

\[
\begin{align*}
\frac{1}{2} & \begin{bmatrix}
U_{e,k}U_k + J_mV_{e,k}V_k \\
J_mU_{e,k}U_k - V_{e,k}V_k \\
U_{o,k}U_k + \sqrt{2c_k}s_k + J_mV_{o,k}V_k \\
J_m(U_{o,k}U_k + \sqrt{2c_k}s_k) - V_{o,k}V_k
\end{bmatrix} \\
\frac{1}{2} & \begin{bmatrix}
U_{o,k}U_k + J_mV_{o,k}V_k \\
J_mU_{o,k}U_k - V_{o,k}V_k
\end{bmatrix}
\end{align*}
\]

(M is odd, \( k \) is even)

(M is odd, \( k \) is odd)
and

$$Z_k = \Xi_k J_M \chi_k$$

$$Z_k = \begin{cases} 
U_{e,k} u_k - J_m V_{e,k} V_k & (M \text{ is even}) \\
J_m u_k + \sqrt{2} c_k b_k - J_m V_{o,k} V_k & (M \text{ is odd}, k \text{ is even}) \\
J_m (u_k + \sqrt{2} c_k b_k) + V_{o,k} V_k & (M \text{ is odd}, k \text{ is odd}),
\end{cases}$$

where $U_k = A_k + J_m B_k$ and $V_k = J_m A_k - B_k$, respectively.

Consequently, it is clear that symmetry is satisfied as $Y_k = J_M Z_k$ in Eq. (2). Precisely, this means that when an input signal vector is the reflected vector of another input signal vector, their output signal vectors also have the same relationship.

2) Case II (Just Stepping Over Image Boundary): If $\chi_k$ is an input signal vector for $\Xi_k$ that just steps over the image boundary in this subsection. Let $\chi_k = \chi_k = [(J_m B_k)^T, B_k^T]^T (M \text{ is even})$ or $[(J_m B_k)^T, c_k, B_k^T]^T (M \text{ is odd})$ where $B_k$ is an $m \times 1$ vector and $c_k$ is a scalar. Similar to Case I, it is clear that symmetry is satisfied as $Y_k = J_M Z_k$ in Eq. (2) where

$$Y_k = Z_k = \begin{cases} 
U_{e,k} J_m B_k & (M \text{ is even}) \\
J_m U_{e,k} J_m B_k & (M \text{ is odd}).
\end{cases}$$

B. Reversible Symmetric Extension (RevSE)

InrSE in Sec. III-A is only for the lossy mode. We solve the problem with symmetry lost due to rounding errors in the lossless mode in this subsection. It can be solved with a very simple matrix manipulation where $\Xi_k$ for extended signals is replaced with $J_M \Xi_k J_M$ except for the case where the process involves just stepping over the image boundary. Fig. 3 shows a realization of the symmetry of L-LPFBs with rounding operations in even channel case.

1) Case I (Not Stepping Over Image Boundary): When $\chi_k$ is an input signal vector for $\Xi_k$ that does not step over the image boundary and rounding operations are considered, this is expressed as $Y'_k = J_M \Xi_k \chi_k \neq \mathbf{Y}_k$ and $Z'_k = J_M \Xi_k J_M \chi_k \neq Z_k$. Obviously, symmetry is lost as $Z'_k \neq J_M Y'_k$ due to rounding error on each lifting step. Therefore, we cannot use this extension as it is. We need to refocus on $\Xi_k$ before it is factorized into lifting structures. According to Eq. (2), $\Xi_k$ can be represented by

$$\Xi_k = J_M \Xi_k J_M.$$ (3)

However, when $\Xi_k$ is used instead of $\Xi_k$, this relationship is not preserved completely as $\Xi_k \neq J_M \Xi_k J_M$, where each building block $\Xi_k$ for extended signals is replaced by $J_M \Xi_k J_M$. Although this transform at the boundary is different from a normal transform with $\Xi_k$, this difference is trivial. By replacing $\Xi_k$ for extended signals, the implementation in the case of reflected input signal vector $J_M \chi_k$ is expressed as $J_M \Xi_k J_M \chi_k = J_M \Xi_k \chi_k = J_M \chi_k'$, Fig. 1 shows the symmetry in building blocks $\Xi_k$ and $J_M \Xi_k J_M$ of $L$-LPFB in even channel case. As a result, it is clear that symmetry can be satisfied by a simple matrix manipulation for extended signals as Eq. (3) even if rounding operations are implemented.

2) Case II (Just Stepping Over Image Boundary): When $\chi_k$ is an input signal vector for $\Xi_k$ that just steps over the image boundary and rounding operations are considered, it is clear that symmetry as

$$Y'_k = J_M Z'_k,$$ (4)

where

$$Y'_k = Z'_k = \begin{cases} 
U_{e,k} J_m B_k & (M \text{ is even}) \\
J_m U_{e,k} J_m B_k & (M \text{ is odd}).
\end{cases}$$

is structurally satisfied even if the rounding operations are implemented in this case similar to those in Sec. III-A2. $\Xi_k$ for extended signals can be replaced by $J_M \Xi_k J_M$ in this case because it is clear that $\Xi_k = J_M \Xi_k J_M$ unlike that in Sec. III-B1, where we did not replace it for the sake of simplicity.
C. Reversible Symmetric Non-Expansive Convolution (RevSNEC)

It is important for the input and output signals for $E_k$ to always achieve symmetry as explained in Sec. III-B. Therefore, only $E_k$ that just stepping over the image boundary should be considered, and only $m$ or $m+1$ output signals are used to the next block as follows:

1) When $M$ is even, either $U_{e,k}J_mB_k$ or $J_mU_{e,k}J_mB_k$ in the output signals in Eq. (4) are used.

2) When $M$ is odd, either $[(U_{a,k}J_mB_k)^T, (w_{a,k})^T]^T$ or $J_mU_{a,k}J_mB_k$ in the output signals in Eq. (4) are used.

Consequently, RevSE in the above subsection can be replaced by non-expansive convolution [13], called RevSNEC, as seen in Fig. 4. This RevSNEC is less complex because it does not need any temporarily extensions to the input and output signals at the image boundary. Also, since $U_{e,k}$ ($k \neq 0$) usually adopts $I_m$ as discussed in Sec. II-A, $J_mU_{e,k}$ and $U_{e,k}J_m$ are simply replaced by $J_m$.

IV. RESULTS

A. Filter Optimization

We designed $8 \times 16$ and $8 \times 24$ PULPFBs, which have $U_{e,k} = I_m$ ($k \neq 0$), $U_{e,0} = U_{e,0}^T$ and $V_{e,k} = V_{e,k}^T$, based on Sec. II-B and $8 \times 16$ and $8 \times 24$ BOLPFBs based on TDLTs in [14] and lifting-based DCT in [31]. We optimized the design parameters by using fminunc.m in the Optimization ToolBox of MATLAB and only coding gain $CCG$ as the cost function [1] for simple design. Moreover, since less DC leakage is one of the most important properties in FB theory for image compression, we parameterized the initial blocks of LPFBs for one degree of regularity [32].

B. Application to Lossy-to-Lossless Image Coding

The resulting LPFBs were applied to lossy-to-lossless image coding. Integer-to-integer transforms can be obtained by using a rounding operation at each lifting step. A wavelet-based coder (embedded zerotree wavelet based on intraband partitioning: EZW-IP) [33] was used in the simulation to fairly evaluate the performance of transforms. Also, RevSNEC, PE, and IrrSE were used for the image boundary processing in the designed LPFBs. We compared the lossy image coding results in Table I in the peak signal-to-noise ratio (PSNR): PSNR [dB] = $10 \log_{10}(\text{MAX}_p^2/\text{MSE})$, where MAX$_p$ and MSE are the maximum possible pixel value of the image and the mean squared error, respectively, at 0.25, 0.50, and 1.00 bit per pixel (bpp) for several test images: $512 \times 512$ 8-bit standard grayscale images, $512 \times 768$ 8-bit Kodak grayscale images, and $2816 \times 1600$ 16-bit clipped grayscale images in [34]. The bold numerals indicate the best PSNRs. 9/7-tap DWT (9/7-DWT) and $4 \times 8$ HLT are the transforms used in JPEG 2000 and JPEG XR lossy modes, respectively. Table I shows that most LPFBs with the RevSNEC achieves better lossy coding than the conventional methods. Fig. 5 illustrates the comparison of a particular area of the image Barbara. It is obvious that the proposed RevSNEC is better than PE and the boundary processing for HLT in JPEG XR in the image boundary at the right of the images in Fig. 5. Also, the LPFBs with the RevSNEC achieved almost same performance compared with the IrrSE which can achieve only lossy mode. Since the RevSNEC in $8 \times 16$ case is completely equivalent to the IrrSE in same case, Table II show the results of IrrSE only in $8 \times 24$ case.

Since the resulting LPFBs are integer-to-integer transforms, we can also obtain lossless reconstructed images at high bit rates. The performance of lossless coding at the lossless bit rate (LBR): LBR [bpp] = $(\text{Total number of bits [bit]})/(\text{Total number of pixels [pixel]})$ is summarized in Table II. The bold numerals mean the best LBRs. 5/3-tap DWT (5/3-DWT) and $4 \times 8$ HLT are the transforms used in JPEG 2000 and JPEG XR lossless modes, respectively. Although the LPFBs with the RevSNEC are often inferior as compared with DWT and HLT in Kodak images and 16-bit large images, they demonstrated their effectiveness in images with high frequency components.

V. CONCLUSION

This paper has presented reversible symmetric extension (RevSE) for $M$-channel lifting-based linear-phase filter banks (L-LPFBs) applied to lossy-to-lossless image coding. Since the proposed RevSE has similar smoothness to an irreversible symmetric extension (IrrSE) at the image boundary, it does not generate distortion at the image boundary even if rounding operations are used. Moreover, complexity is lessened by extending non-expansive convolution, called reversible...
symmetric non-expansive convolution (RevSNEC). As a result, it achieves better performance in lossy-to-lossless image coding than periodic extension (PE).

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and Dr. S. Kyochi 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.

REFERENCES


[22] , “Group lifting structures for multirate filter banks: II: Linear phase
### TABLE I
**Lossy image coding results (PSNR[dB]).**

<table>
<thead>
<tr>
<th>Test images</th>
<th>9/7-DWT</th>
<th>4 × 8 HLT</th>
<th>8 × 16 PULPB</th>
<th>8 × 16 BOLPB</th>
<th>8 × 24 PULPB</th>
<th>8 × 24 BOLPB</th>
<th>8 × 16 BOLPB</th>
<th>8 × 21 BOLPB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>bit rate: 0.25 [bpp]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Barbara</strong></td>
<td>34.78</td>
<td>34.03</td>
<td>34.14 (34.87)</td>
<td>34.14 (34.87)</td>
<td>34.15 (34.87)</td>
<td>34.15 (34.87)</td>
<td>34.16 (34.87)</td>
<td>34.17 (34.87)</td>
</tr>
<tr>
<td><strong>Elaine</strong></td>
<td>32.96</td>
<td>32.47</td>
<td>33.12 (32.42)</td>
<td>33.12 (32.42)</td>
<td>33.12 (32.42)</td>
<td>33.12 (32.42)</td>
<td>33.13 (32.42)</td>
<td>33.14 (32.42)</td>
</tr>
<tr>
<td><strong>Kodim20</strong></td>
<td>35.19</td>
<td>34.34</td>
<td>34.78 (34.69)</td>
<td>34.78 (34.69)</td>
<td>34.78 (34.69)</td>
<td>34.78 (34.69)</td>
<td>34.79 (34.69)</td>
<td>34.80 (34.69)</td>
</tr>
<tr>
<td><strong>Kodim21</strong></td>
<td>30.20</td>
<td>30.29</td>
<td>30.28 (30.15)</td>
<td>30.28 (30.15)</td>
<td>30.28 (30.15)</td>
<td>30.28 (30.15)</td>
<td>30.29 (30.15)</td>
<td>30.30 (30.15)</td>
</tr>
<tr>
<td><strong>Arri</strong></td>
<td>37.30</td>
<td>37.73</td>
<td>38.05 (37.93)</td>
<td>38.05 (37.93)</td>
<td>38.05 (37.93)</td>
<td>38.05 (37.93)</td>
<td>38.06 (37.93)</td>
<td>38.07 (37.93)</td>
</tr>
<tr>
<td><strong>Face</strong></td>
<td>40.20</td>
<td>43.48</td>
<td>40.94 (40.95)</td>
<td>40.94 (40.95)</td>
<td>40.94 (40.95)</td>
<td>40.94 (40.95)</td>
<td>40.95 (40.95)</td>
<td>40.96 (40.95)</td>
</tr>
<tr>
<td><strong>Lake Locked</strong></td>
<td>42.11</td>
<td>41.78</td>
<td>42.51 (42.41)</td>
<td>42.51 (42.41)</td>
<td>42.51 (42.41)</td>
<td>42.51 (42.41)</td>
<td>42.52 (42.41)</td>
<td>42.53 (42.41)</td>
</tr>
</tbody>
</table>

### TABLE II
**Lossless image coding results (LBR [bpp]).**

<table>
<thead>
<tr>
<th>Test images</th>
<th>5/3-DWT</th>
<th>4 × 8 HLT</th>
<th>8 × 16 PULPB</th>
<th>8 × 16 BOLPB</th>
<th>8 × 16 BOLPB</th>
<th>8 × 16 BOLPB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>bit rate:1.00 [bpp]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Barbara</strong></td>
<td>34.87</td>
<td>35.05</td>
<td>35.95 (35.77)</td>
<td>35.95 (35.77)</td>
<td>35.95 (35.77)</td>
<td>35.95 (35.77)</td>
</tr>
<tr>
<td><strong>Elaine</strong></td>
<td>34.63</td>
<td>34.22</td>
<td>34.13 (34.07)</td>
<td>34.13 (34.07)</td>
<td>34.13 (34.07)</td>
<td>34.13 (34.07)</td>
</tr>
<tr>
<td><strong>Finger</strong></td>
<td>29.06</td>
<td>29.01</td>
<td>30.91 (30.05)</td>
<td>30.91 (30.05)</td>
<td>30.91 (30.05)</td>
<td>30.91 (30.05)</td>
</tr>
<tr>
<td><strong>Kodim19</strong></td>
<td>34.70</td>
<td>34.66</td>
<td>34.97 (34.89)</td>
<td>34.97 (34.89)</td>
<td>34.97 (34.89)</td>
<td>34.97 (34.89)</td>
</tr>
<tr>
<td><strong>Kodim20</strong></td>
<td>39.45</td>
<td>39.50</td>
<td>39.23 (39.29)</td>
<td>39.23 (39.29)</td>
<td>39.23 (39.29)</td>
<td>39.23 (39.29)</td>
</tr>
<tr>
<td><strong>Kodim21</strong></td>
<td>34.31</td>
<td>34.12</td>
<td>34.31 (34.15)</td>
<td>34.31 (34.15)</td>
<td>34.31 (34.15)</td>
<td>34.31 (34.15)</td>
</tr>
<tr>
<td><strong>Arri</strong></td>
<td>41.97</td>
<td>42.28</td>
<td>42.75 (42.67)</td>
<td>42.75 (42.67)</td>
<td>42.75 (42.67)</td>
<td>42.75 (42.67)</td>
</tr>
<tr>
<td><strong>Face</strong></td>
<td>51.68</td>
<td>51.47</td>
<td>52.05 (51.98)</td>
<td>52.05 (51.98)</td>
<td>52.05 (51.98)</td>
<td>52.05 (51.98)</td>
</tr>
<tr>
<td><strong>Lake Locked</strong></td>
<td>45.50</td>
<td>45.37</td>
<td>46.02 (45.95)</td>
<td>46.02 (45.95)</td>
<td>46.02 (45.95)</td>
<td>46.02 (45.95)</td>
</tr>
</tbody>
</table>


