PSEUDO REVERSIBLE SYMMETRIC EXTENSION FOR LIFTING-BASED NONLINEAR-PHASE PARAUNITARY FILTER BANKS

Taizo Suzuki¹, Naoki Tanaka², and Hiroyuki Kudo³

1: Faculty of Engineering, Information and Systems, University of Tsukuba, Japan
2: Department of Computer Science, University of Tsukuba, Japan
Email: taizo@cs.tsukuba.ac.jp, tanaka@imagelab.cs.tsukuba.ac.jp, and kudo@cs.tsukuba.ac.jp

ABSTRACT
This study presents a pseudo reversible symmetric extension (P-RevSE) that solves the signal boundary problem of lifting-based nonlinear-phase paraunitary filter banks (L-NLPPUFBs), which have high compression rates thanks to their not having a constraint on the linear-phase property unlike the existing transforms used in image coding standards. The conventional L-NLPPUFBs with a periodic extension (PE) yield annoying artifacts at the signal boundaries. However, the P-RevSE is implemented smoothly at the signal boundaries by using a nonexpansive convolution of a symmetric extension (SE) and determinant control for the lifting factorization. Although the determinant control causes a pseudo SE, not a true SE, the resulting L-NLPPUF with P-RevSE outperforms not only the L-NLPPUF with PE but also the current transform used in JPEG XR.

Index Terms—Lifting structure, lossy-to-lossless image coding, nonlinear-phase paraunitary filter bank, signal boundary problem, symmetric extension.

1. INTRODUCTION
Image compression (coding) standards help to alleviate the burden on servers and free up communication bandwidth. JPEG is the most common image coding standard, but it uses the discrete cosine transform (DCT), which causes blocking artifacts in low bitrate compression. JPEG XR (eXtended Range) [1] is a more effective image coding standard that uses a lapped transform (LT) [2], and it solves the blocking problem. Although LPFBs have to extend the signals at the signal boundaries because of the overlapping processing, the output signals should not be larger than the input signals. A periodic extension (PE), which is one of the simplest boundary processing, causes annoying artifacts due to discontinuities at the signal boundaries. LPFBs solve the problem by using a symmetric extension (SE) [4], which extends the boundary signals smoothly and does not require transmission of extra signals for reconstruction.

Nonlinear-phase filter banks (NLPPFBs) [5, 6] have high compression rates thanks to their not having a constraint on the linear-phase property unlike LPFBs. NLPPFBs also have the signal boundary problem, and some solutions for it have been presented [7–10]. On the other hand, lifting-based FBs (L-FBs) for lossy-to-lossless transmission of extra signals for reconstruction. Nonlinear-phase filter banks (NLPPFBs) [5, 6] have high compression rates thanks to their not having a constraint on the linear-phase property unlike LPFBs. NLPPFBs also have the signal boundary problem, and some solutions for it have been presented [7–10].

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

2. NONLINEAR-PHASE PARAUNITARY FILTER BANKS
Let \( M \times M K \) NLPPFB \((M, K \in \mathbb{N}, M \text{ is even}, K \geq 2)\) be an NLPPFB whose channel and filter length are \( M \) and \( M K \), respectively. The polyphase matrix \( E(z) \) is expressed as [13]

\[
E(z) = \left( \prod_{k=K-1}^{1} G_k \Lambda_k(z) \right) G_0,
\]

where \( G_k \) \((k = 0, 1, \ldots, K-1)\) is an \( M \times M \) arbitrary nonsingular matrix and \( \Lambda_k(z) \) is a delay matrix with a delay element \( z \) as:

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

Although \( \gamma_k \) is an arbitrary integer \( 1 \leq \gamma_k \leq M - 1 \), we set \( \gamma_k = M/2 \) for simplicity. Thus, \( \Lambda_k(z) \) is denoted as \( \Lambda(z) \). Since an NLPPFB is not constrained by the linear-phase property, it achieves a better frequency decomposition compared with the linear-phase case. When \( G_k \) is an arbitrary orthonormal matrix, \( G_k = G_k^* \) [5], an NLPPFB is called an NLPUPFB.

3. SYMMETRIC EXTENSION FOR NLPUPFB
Tanaka et al. proposed a nonexpansive convolution for NLPUPFBs [9]. The top images of Fig. 1 shows the upper boundary
processing of the analysis banks of NLPPUFBs when $K = 2$ and 3. They mean that $(K - 1)/M/2$ extra signals have to be extended at each boundary. To obtain smooth signals and for simplicity, the extra signals are commonly extended by using $J_x[(K - 1)/M/2]$; i.e., the extension is an SE.

Here, we have a nonexpansive problem because the signals transmitted to the synthesis bank should only be $y_n$ ($n = 0, 1, 2, \ldots$). This section presents a reconstruction method at the synthesis bank without any extra signal; i.e., $x_n$ is reconstructed from only $a_n$, which can be reconstructed from $y_n$. Let $G_k$ be

$$G_k = \begin{bmatrix} A_k & B_k \\ C_k & D_k \end{bmatrix},$$

where each submatrix is $M/2 \times M/2$. Hereafter, we consider only the upper boundary processing. The signals at the lower boundary processing can be reconstructed in the same way as in the upper boundary processing.

3.1. Case of $K = 2$

From the top left of Fig. 1, we obtain

$$a_0 = \begin{bmatrix} A_1 & B_1 \end{bmatrix} \begin{bmatrix} Jx_0 \\ x_0 \end{bmatrix} = (A_1J + B_1)x_0. \quad (4)$$

The problem is to reconstruct $x_0$ from $a_0$, which is transmitted to the synthesis bank. $x_0$ is represented as

$$x_0 = U^{-1}a_0, \quad (5)$$

where the boundary matrix $U$ has to be nonsingular, i.e., $\det(U) \neq 0$. The nonexpansive convolution of the analysis banks in case of $K = 2$ is shown at the bottom left of Fig. 1.

3.2. Case of $K = 3$

From the top right of Fig. 1, similar to the case of $K = 2$, we obtain

$$b_{-1} = \begin{bmatrix} A_2 & B_2 \end{bmatrix} \begin{bmatrix} Jx_1 \\ x_1 \end{bmatrix} = [B_2J A_2J]x_1. \quad (6)$$

Also, $x_0$ and $x_1$ in Eq. (6) can be recalculated from $b_0$ and $b_1$ as

$$\begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} A_2^T & C_2^T \\ B_2^T & D_2^T \end{bmatrix} \begin{bmatrix} b_1 \\ b_0 \end{bmatrix}, \quad (7)$$

where $G_2$ is an orthonormal matrix. We cannot easily calculate $b_0$ from the limited signals $a_n$ unlike $b_1$. Substituting Eq. (7) into Eq. (6) yields

$$b_{-1} = [B_2J A_2J] \begin{bmatrix} A_2^T & C_2^T \\ B_2^T & D_2^T \end{bmatrix} \begin{bmatrix} b_1 \\ b_0 \end{bmatrix}. \quad (8)$$

Also, we obtain

$$a_0 = [A_1 B_1] \begin{bmatrix} b_{-1} \\ b_1 \end{bmatrix}, \quad (9)$$

Consequently, $b_0$ can be calculated from Eqs. (8) and (9) as

$$b_0 = \begin{pmatrix} (A_1 \left( B_2JC_2^T + A_2JD_2^T \right) + B_1)^{-1} \\ \vdots \end{pmatrix} \begin{bmatrix} a_0 - A_1 \left( B_2JA_2^T + A_2JB_2^T \right) b_1 \\ \vdots \end{bmatrix}, \quad (10)$$

where the boundary matrix $V$ has to be nonsingular, i.e., $\det(V) \neq 0$. The nonexpansive convolution of the analysis banks in case of $K = 3$ is shown at the bottom right of Fig. 1.

For any $K$, the signals can be reconstructed as the solution of a simultaneous matrix equation with $(K - 1)$ unknowns.

4. PSEUDO SYMMETRIC EXTENSION FOR L-NLPPUFBS

This section presents a P-RevSE for L-NLPPUFBS that uses the nonexpansive convolution described in Sec. 3 and determinant control for the lifting factorization. From the bottom of Fig. 1, if the structures are expressed as lifting structures, they achieve reversible transforms for lossless image coding. We can consider that the processing with $W$ is already a lifting structure. Consequently, the residual matrices $G_k$, $U$, and $V$ should be factorized into lifting structures. A minimum condition to realize a lifting factorization of a
Table 1. Coding gain $C_{cg}$ of the resulting $4 \times 12$ L-NLPPUFBs.

<table>
<thead>
<tr>
<th>Boundary</th>
<th>Not</th>
<th>Upper</th>
<th>Lower</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{cg}$</td>
<td>8.3168</td>
<td>8.2852</td>
<td>8.3173</td>
</tr>
</tbody>
</table>

Table 2. Lossless image coding results (LBR [bpp]).

<table>
<thead>
<tr>
<th>Test Images</th>
<th>LT [2]</th>
<th>L-NLPPUFBs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barbara</td>
<td>4.801</td>
<td>4.775</td>
</tr>
<tr>
<td>Boat</td>
<td>5.124</td>
<td>5.093</td>
</tr>
<tr>
<td>Elaine</td>
<td>5.166</td>
<td>5.106</td>
</tr>
<tr>
<td>Lena</td>
<td>4.587</td>
<td>4.587</td>
</tr>
<tr>
<td>Pepper</td>
<td>4.954</td>
<td>4.907</td>
</tr>
<tr>
<td>Room</td>
<td>4.344</td>
<td>4.427</td>
</tr>
</tbody>
</table>

matrix is that the determinant is $\pm 1$ [12]. Since $G_k$ is an orthonormal matrix, i.e., $\det(G_k) = \pm 1$, the matrix can be factorized into lifting structures. On the other hand, note that the determinants of $\mathbf{U}$ and $\mathbf{V}$ might not satisfy this condition, i.e., $\det(\mathbf{U}) \neq \pm 1$ and $\det(\mathbf{V}) \neq \pm 1$. Thus, we control the matrices in

$$
\mathbf{\tilde{U}} = \frac{\mathbf{U}}{\sqrt[4]{\det(\mathbf{U})}} \quad \text{and} \quad \mathbf{\tilde{V}} = \frac{\mathbf{V}}{\sqrt[4]{\det(\mathbf{V})}},
$$

where $\det(\mathbf{U}) \neq 0$ and $\det(\mathbf{V}) \neq 0$, and these conditions are completely equivalent to those as described in Sec. 3. Since these matrices $\mathbf{U}$ and $\mathbf{V}$ satisfy the condition for a lifting factorization, $\det(\mathbf{U}) = \pm 1$ and $\det(\mathbf{V}) = \pm 1$, $\mathbf{U}$ and $\mathbf{V}$ can be factorized into lifting structures. This means that the top half of $G_1$ at the signal boundaries is scaled by $\sqrt[4]{\det(\mathbf{U})}$ and $\sqrt[4]{\det(\mathbf{V})}$.

$$
G_1 = \frac{1}{2} I \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} G_1 = \begin{bmatrix} \frac{1}{2} A_1 \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} B_1 \\ C_1 & D_1 \end{bmatrix},
$$

where $s = \sqrt[4]{\det(\mathbf{U})}$ ($K = 2$) or $\sqrt[4]{\det(\mathbf{V})}$ ($K = 3$); i.e., the NLPPUFBs losing the original properties are used at the signal boundaries. Therefore, if $\sqrt[4]{\det(\mathbf{U})}$ and $\sqrt[4]{\det(\mathbf{V})}$ are quite different from 1, the boundary processing cannot be implemented smoothly. Accordingly, we have to take this into consideration when designing the NLPPUFB.

5. EXPERIMENTAL RESULTS

5.1. Design of NLPPUFBs

We designed $4 \times 12$ ($M = 4$ and $K = 3$) NLPPUFB with the cost function $\phi$, which is a weighted linear combination of the coding gain $C_k$ [14], the symmetric property of filters $C_{sym}$ [9], and the determinant control of matrices $C_{det}$, as follows:

$$
\phi = -w_0 C_{cg} + w_1 C_{sym} + w_2 C_{det},
$$

where

$$
C_{det} = [(\det(\mathbf{\nabla}_a)) - 1]^2 + (\det(\mathbf{\nabla}_l)) - 1)^2.
$$

$\mathbf{\nabla}_a$ and $\mathbf{\nabla}_l$ mean $\mathbf{\nabla}$ at the upper and lower boundary processing of the NLPPUFBs. $w_k$ was experimentally-determined as $\{w_0, w_1, w_2\} = \{1, 0.01, 0.15\}$. Regularity, which is one of the most important image coding properties to prevent the DC leakage, was considered structurally [12]. As described in the previous section, smooth boundary processing cannot be implemented if $\sqrt[4]{\det(\mathbf{\nabla})}$ is quite different from 1. However, the resulting $4 \times 12$ NLPPUFB achieved $\sqrt[4]{\det(\mathbf{\nabla}_a)} = 0.9339 \approx 1$ and $\sqrt[4]{\det(\mathbf{\nabla}_l)} = 0.9635 \approx 1$ thanks to the cost functions $C_{sym}$ and $C_{det}$. Table 1 and Fig. 2 respectively show the coding gain $C_{cg}$ and the frequency responses of the resulting $4 \times 12$ NLPPUFBs at non-signal boundaries, upper signal boundaries, and lower signal boundaries, respectively: (top) analysis banks, (bottom) synthesis banks.

5.2. Lossy-to-Lossless Image Coding

We evaluated the $4 \times 8$ LT (JPEG XR) with RevSE$^1$ and the resulting $4 \times 12$ L-NLPPUFBs with PE and P-RevSE through lossy-to-lossless image coding. A single-row elementary reversible matrix (SERM) [15] was used for the lifting factorization of $G_k$ and $\mathbf{\nabla}$. As described in Sec. 4, the processing with $\mathbf{\nabla}$ is considered to be a lifting structure without any lifting factorization. The resulting $4 \times 12$ L-NLPPUFBs were implemented with a rounding operation at each lifting step and compared in terms of the lossless bitrate (LBR) [bpp] in lossless image coding and peak signal-to-noise ratio (PSNR) [dB].

$^1$Note that the RevSE in JPEG XR is not equivalent to [11] because it is customized for JPEG XR.
Table 3. Lossy image coding results (PSNR [dB]).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.25</td>
<td>26.569</td>
<td>27.436</td>
<td>27.578</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Barbara</td>
<td>0.50</td>
<td>30.334</td>
<td>31.097</td>
<td>31.234</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>34.952</td>
<td>35.601</td>
<td>35.728</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>27.261</td>
<td>28.129</td>
<td>28.219</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>30.727</td>
<td>31.296</td>
<td>31.371</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>34.213</td>
<td>34.758</td>
<td>34.805</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>30.825</td>
<td>31.178</td>
<td>31.381</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>32.502</td>
<td>32.876</td>
<td>33.016</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>34.179</td>
<td>34.746</td>
<td>34.894</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>31.603</td>
<td>31.965</td>
<td>32.251</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>35.024</td>
<td>35.300</td>
<td>35.524</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>38.247</td>
<td>38.572</td>
<td>38.708</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>31.026</td>
<td>31.078</td>
<td>31.586</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>33.892</td>
<td>34.207</td>
<td>34.328</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>35.428</td>
<td>36.200</td>
<td>36.273</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>27.818</td>
<td>28.684</td>
<td>29.048</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>32.715</td>
<td>32.838</td>
<td>33.116</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>38.288</td>
<td>37.927</td>
<td>38.407</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To evaluate transform performance fairly, we employed two-level decompositions on all transforms. The image set included six 512 × 512 eight-bit standard grayscale images in [16]. A quadtree-based embedded image coder EZW-IP [17] was used to encode the transformed images.

Table 2, Table 3, and Fig. 3 show the lossless and lossy image coding results. The results show the advantage of the L-NLPPUFB with P-RevSE over the L-NLPPUFB with PE. Especially, in Fig. 3, the L-NLPPUFB with PE produces annoying artifacts, whereas no artifacts are apparent in the image from the L-NLPPUFB with P-RevSE. In addition, the L-NLPPUFB with P-RevSE outperformed the 4 × 8 LT with RevSE.

6. CONCLUSION

This study presented a P-RevSE for L-NLPPUFBs. The conventional L-NLPPUFBs without any smooth boundary processing have the annoying artifacts at signal boundaries. The L-NLPPUFB with P-RevSE solved this problem by using a nonexpansive convolution of an SE and determinant control for the lifting factorization. Although the determinant control caused a pseudo SE, not a true SE, the L-NLPPUFB with P-RevSE outperformed not only the L-NLPPUFB with PE but also the current transform used in JPEG XR.

7. REFERENCES


