Symmetric Extension for Lifted Filter Banks and Obstructions to Reversible Implementation

Brendt Wohlberg [1]

T-7 Mathematical Modeling and Analysis, MS-B284, Los Alamos National Laboratory, Los Alamos, NM 87545, USA [2]

Christopher M. Brislawn

CCS-3 Modeling, Algorithms, and Informatics, MS-B265, Los Alamos National Laboratory, Los Alamos, NM 87545, USA [2]

Abstract

Symmetric pre-extension is a standard approach to boundary handling for finite-length input vectors with linear phase filter banks. It works with both conventional linear implementations and so-called *reversible*, or integer-to-integer, implementations of odd-length linear phase (*whole-sample symmetric*) filter banks. In comparison, significant difficulties arise when using symmetric pre-extension on reversible filter banks with *even*-length (*half-sample symmetric*) linear phase filters. An alternative approach is presented using *lifting step extension*, in which boundary extensions are performed in each step of a lifting factorization, that avoids some of these difficulties while preserving reversibility and retaining the nonexpansive property of symmetric pre-extension. Another alternative that is capable of preserving both reversibility and subband symmetry for half-sample symmetric filter banks is developed based on ideas from the theory of lattice vector quantization. The practical ramifications of this work are illustrated by describing its influence on the specification of filter bank algorithms in Part 2 of the ISO/IEC JPEG 2000 image coding standard.

Key words: filter bank, symmetric extension, lifting, reversible, JPEG 2000

---

[1] Corresponding author. E-mail address: brendt@lanl.gov; Tel: (1) 505 667 6886; Fax: (1) 505 665 5757.

[2] This work was carried out under the auspices of the National Nuclear Security Administration of the U.S. Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396 and was supported in part by the DOE Accelerated Strategic Computing Initiative, the DOE Applied Mathematical Sciences Program, contract KC-07-01-01, and the NNSA's Laboratory Directed Research and Development Program.

1 Introduction

While a number of structures have been proposed for invertible integer to integer transforms [1,2], a particularly popular approach is based on the insertion of rounding operations within each step of a *lifting decomposition* [3,4] of a perfect reconstruction filter bank [5,6]. An advantage of the lifting approach is that it allows a single filter bank factorization to be used for either lossless or lossy subband coding, depending on whether rounding operations are applied to the lifting updates. A major application of reversible filter banks to date has been image coding, including the ISO/IEC JPEG 2000 image coding standard [7,8], but other applications have included audio coding [9], video coding [10], and image watermarking [11].

Symmetric pre-extension [12,13] is a common method of handling boundary conditions for linear phase filter banks with finite-length input vectors. One disadvantage of this approach is that it increases the length of the input vector. This additional boundary data is used by the filter bank for computing output samples near the boundaries; when the filter bank is implemented in cascade form, the boundary data must be carried forward from one cascade step to the next. With lifting it is possible to avoid carrying boundary extensions forward between cascade steps, thereby minimizing the working memory required. Instead of pre-extending input vectors, one can define *lifting step extension* operations *within* each lifting step in a way that yields invertible, nonexpansive transforms. Moreover, it is sometimes possible to choose lifting step extensions so that the resulting transform is mathematically equivalent to symmetric pre-extension. This is the case for linear phase FIR filter banks with odd-length impulse responses (the *whole-sample symmetric,* or WS, filter banks), which factor using *even*-length linear phase (*half-sample symmetric,* or HS) lifting filters. By comparison, the analogous issues for HS filter banks are considerably more complicated [14].

For irreversible linear phase filter banks, an important advantage of symmetric pre-extension is that it is performed as a pre-processing step applied to the input vector before it enters the filter bank. This means that symmetric pre-extension schemes yield nonexpansive, invertible transforms *independent* of how the filter bank is implemented. Thus, a signal decomposed using symmetric pre-extension can be reconstructed by *any* form of synthesis bank that understands symmetric extension, regardless of how the analysis was performed (e.g., direct-form, time-varying convolution [15], lifting, or some other polyphase factorization). In particular, the equivalence of symmetric pre-extension and lifting step extension for WS filter banks means that *all* representations of irreversible WS filter banks using symmetric pre-extension are equivalent to one particular lifting representation with appropriate lifting step extensions. Thus, e.g., the specification of lifting step extension for arbitrary filter banks in JPEG 2000 Part 2 [8, Annex H], based on recommendations made in [16], is backwards-compatible with the specification of symmetric pre-extension for WS filter banks in JPEG 2000 Part 1 [7, Annex F].

In comparison, reversible implementations require the encoder and the decoder to agree on a specific rounded lifting factorization to obtain lossless reconstruction. It is known that reversible lifting implementations are reversible for *any* choice of (agreed-upon) rounding operations, and they are similarly reversible for *any* choice of lifting step extensions. Moreover, they are also nonexpansive for all input lengths using any choice of lifting step extensions. While symmetric pre-extension might therefore appear to be of little benefit for reversible filter banks, there are three primary reasons for studying symmetric pre-extension and the existence of equivalent lifting step extensions for reversible filter banks.

One reason is the coding gain advantage that symmetric extension has over other nonexpansive boundary-handling techniques. While other extension methods are possible using lifting step extensions, symmetric pre-extension is well-studied and provides a natural baseline for comparison against proposed alternatives. A second reason to consider symmetric extension in reversible implementations is to maintain compatibility with irreversible implementations using lifting step extensions that are equivalent to symmetric pre-extension. It simplifies matters if both reversible and irreversible implementations can use the same lifting step extensions. A third reason is to ensure that reversible lifting schemes defined in terms of symmetric pre-extension (e.g., [7, Annex F]) can be inverted losslessly by implementations using lifting step extensions. While everything works out nicely for arbitrary reversible WS filter banks, this is not the case for arbitrary reversible HS filter banks, as we shall show.

1.1 Comparison to Prior Work

Lifting step extension schemes that are equivalent to symmetric pre-extension for some WS filter banks were known to the JPEG 2000 standards committee. For instance, they were presented in a committee report by the present authors [17] and had been exploited even earlier to some degree in research software such as the committee's Verification Model [18] and JasPer software [19]. The lifting step approach to symmetric extension for irreversible WS filter banks was described in Taubman and Marcellin [15, Section 6.5.3], emphasizing factorizations that use only first-order lifting filters; they do not cover HS filter banks. Adams and Ward [20,21] also proved that constant lifting step extension is equivalent to symmetric pre-extension for WS filter banks that factor using first-order lifting filters, including reversible implementations.

Very little has been written about lifting step extension for HS filter banks. Brislawn and Wohlberg [17] described how symmetric pre-extension for arbitrary irreversible HS filter banks can always be implemented in a mathematically equivalent fashion using a combination of symmetric pre-extension in the base HS filter bank and lifting step extensions in the WA lifting steps. For reversible HS filter banks lifted from the Haar filter bank, we showed that symmetric pre-extension always yields nonexpansive reversible transforms by generating symmetric subbands if the Haar lifting steps are rounded using the floor function and the WA lifting steps are rounded using an odd function such as symmetric truncation. This enables equivalent lifting step extension schemes for such filter banks, and we also explained why the well-known 2-tap/6-tap and 2-tap/10-tap HS filter banks are reversible even though their highpass WA lifting steps are not commonly rounded using odd functions.

Adams and Ward strengthened these results by showing that it is the integer invariance of the floor function that makes the reversible Haar filter bank produce symmetric lowpass subbands in symmetric pre-extension schemes. (A function is *integer invariant*, or *integer-bias invariant* [21], if it commutes with integer shifts: $f(x + n) = f(x) + n$ for all integers n and real numbers x.) For reversible HS analysis banks lifted from the Haar, they showed that the synthesis bank can always be renormalized to yield a new analysis bank lifted from the same factorization of the Haar. The renormalized synthesis bank thus provides another nonexpansive, reversible symmetric pre-extension transform. They also showed that certain odd rounding rules are invertible when applied to integer subbands *following* real-valued lifting updates, which yields an alternative construction of reversible symmetric pre-extension transforms for HS filter banks. Unfortunately, this novel idea is not compatible with the more widespread approach (e.g., [7,8]) based on rounding the lifting updates rather than the subbands. They did not analyze the case of HS filter banks lifted from higher-order HS base filter banks, however, nor did they demonstrate how symmetric extension can fail altogether for certain reversible lifting factorizations.

1.2 Overview of the Paper

The primary goal if this paper is to present the obstacles to symmetric extension for reversible HS filter banks. Section 2 reviews lifting and reversibility for perfect reconstruction filter banks, while Section 3 reviews the symmetric pre-extension method for irreversible and reversible WS filter banks. The equivalent lifting step extension scheme has been specified for general WS filter banks in JPEG 2000 Part 2 [8, Annex G]. Section 3 also introduces the polyphase approach used in the analysis of lifting step extension schemes, in preparation for the more difficult case of HS filter banks. Symmetric pre-extension for HS filter banks is studied in Section 4. In contrast to the WS case, symmetric pre-extension for irreversible HS filter banks is *never* equivalent to a scheme using only lifting step extensions, although this difficulty can be finessed by combining symmetric pre-extension in the HS base with lifting step extensions in subsequent WA lifting steps. Equivalence also holds for all *reversible* HS filter banks lifted via WA lifting steps from the Haar base with floor-function rounding. For some reversible HS filter banks, however, we show that symmetric pre-extension breaks down altogether: there are *no* scalar rounding rules that generate the subband symmetries needed for a nonexpansive reversible symmetric pre-extension transform. These results, which were originally presented in [17,22], are responsible for the lifting step extensions for arbitrary filter banks specified in JPEG 2000 Part 2 [8, Annex H]. Finally, Section 5 introduces vector rounding rules for reversible HS filter banks.

2 Perfect Reconstruction Filter Banks

A two-channel perfect reconstruction filter bank is a system of the form in Figure 1 for which $\hat{X}(z) = Az^{-D}X(z)$ for some integer D and $A \neq 0$. In the remainder, we follow [4] in assuming that $D = 0$ and $A = 1$ (cf. [14, Section II.B]).

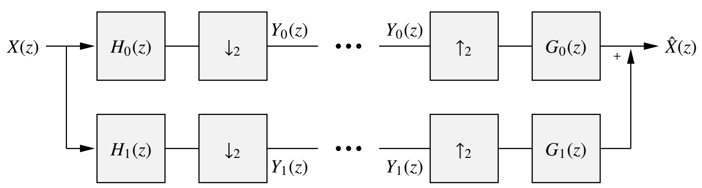

Fig. 1. Direct form representation of a two-channel filter bank.

2.1 Polyphase Decomposition

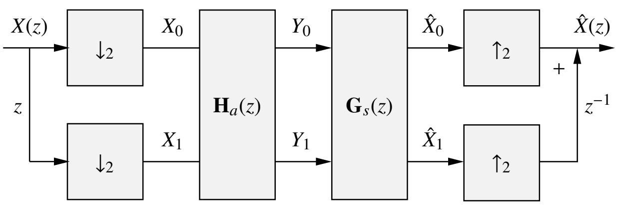

Fig. 2. Polyphase-with-advance representation.

The *polyphase-with-advance* representation [14] in Figure 2 is one of several forms of filter bank polyphase representation [5,6]. In particular, it is the form used in [4] and in the JPEG 2000 standard. It defines the even and odd polyphase components, $X_0(z)$ and $X_1(z)$, of the input so that $X(z) = X_0(z^2) + z^{-1}X_1(z^2)$. We use bold fonts for the polyphase vector form: $\boldsymbol{X}(z) = [X_0(z)\; X_1(z)]^T$. The polyphase decompositions of the analysis and synthesis filters [14] give the components of the analysis and synthesis polyphase matrices, $\mathbf{H}_a(z)$ and $\mathbf{G}_s(z)$. Under the normalization cited above, the perfect reconstruction condition becomes $\mathbf{G}_s(z)\mathbf{H}_a(z) = \mathbf{I}$, and $\hat{X} = X$.

It has been shown [4] that any FIR perfect reconstruction polyphase matrix factors into a diagonal gain scaling matrix, $\mathrm{diag}(1/K, K)$, $K \neq 0$, and alternating upper- and lower-triangular lifting matrices,

$$\mathbf{H}_a(z) = \mathrm{diag}(1/K, K)\,\mathbf{S}_{N_{LS}-1}(z)\cdots\mathbf{S}_1(z)\,\mathbf{S}_0(z)\,, \tag{1}$$

where

$$\mathbf{S}_k(z) = \begin{bmatrix} 1 & S_k(z) \\ 0 & 1 \end{bmatrix} \quad \text{or} \quad \mathbf{S}_k(z) = \begin{bmatrix} 1 & 0 \\ S_k(z) & 1 \end{bmatrix}$$

with FIR lifting filters $S_k(z)$. A block diagram of this structure is presented in Figure 3.

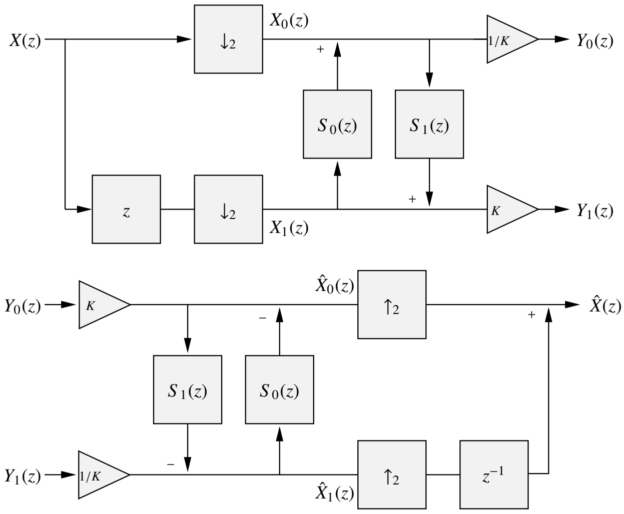

Fig. 3. Lifting implementation of a filter bank.

Some examples of lifting factorizations are presented in Table 1. The 5-tap/3-tap filter bank is the reversible filter bank in Part 1 of the JPEG 2000 standard, and the 6-tap/2-tap filter bank is the 2-tap/6-tap filter bank from [23] with the analysis and synthesis banks switched.

Table 1
Lifting factorizations for Haar, 2-tap/6-tap, 5-tap/3-tap, and 6-tap/2-tap filter banks.

Name	Haar	2-tap/6-tap	5-tap/3-tap	6-tap/2-tap
Num. steps	2	3	2	3
S_0 updates	Odd ch.	Odd ch.	Odd ch.	Even ch.
$S_0(z)$	$-z^0$	$-z^0$	$-\frac{1}{2}z^1 - \frac{1}{2}z^0$	z^0
$S_1(z)$	$\frac{1}{2}z^0$	$\frac{1}{2}z^0$	$\frac{1}{4}z^0 + \frac{1}{4}z^{-1}$	$-\frac{1}{2}z^0$
$S_2(z)$		$-\frac{1}{4}z^1 + \frac{1}{4}z^{-1}$		$-\frac{1}{4}z^1 + \frac{1}{4}z^{-1}$

Invertible integer-to-integer transforms [24] can be constructed by inserting rounding operations in each step of a lifting factorization as shown in Figure 4. (Note that the rounding function actually rounds the *time*-domain filter output samples rather than the complex transform-domain representation of the filter output.) These *reversible* implementations are nonlinear approximations of the corresponding linear filter banks. A reversible implementation provides bit-perfect reconstruction for *any* rounding operation, $R(\cdot)$, if the same rounded update is subtracted in each synthesis lifting step.

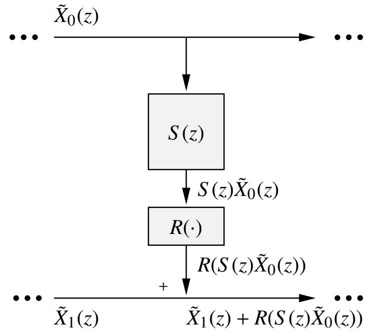

Fig. 4. A single lifting step in a reversible integer-to-integer transform.

3 Symmetric Pre-Extension

A boundary-handling strategy that preserves perfect reconstruction is required for finite-length input vectors in applications like image coding [6, Ch. 8]. One approach is the *extended filter bank* in Figure 5. In *symmetric pre-extension* [12,13] with linear phase filters, the operator E extends the input into a symmetric signal via reflection at each end, followed by periodic continuation. Under appropriate conditions, discussed below, the periodic subbands resulting from filtering and downsampling are also symmetric. This symmetry is exploited by projections P_0 and P_1 that retain only half of a symmetric period in each subband. (Although the extended input, $\tilde{x} \equiv Ex$, is in principle infinite in duration, in practice one only generates enough extended samples to compute half of a symmetric period in each subband; i.e., the nonredundant outputs that would be saved by P_0 and P_1 if $\tilde{x}$, and hence the subbands, were infinite in duration.) When Y_0 and Y_1 have a total of just N_0 samples between them, the extended filter bank is called *nonexpansive*. The full symmetric-periodic subbands are restored in the synthesis bank by symmetric extensions E_0 and E_1. The projection, P, ensures that the output, $\hat{x}$, has the same number of samples as the input.

Discrete-time signals may be symmetric about a sample, referred to as whole-sample (WS) symmetry, or about a midpoint between samples, referred to as half-

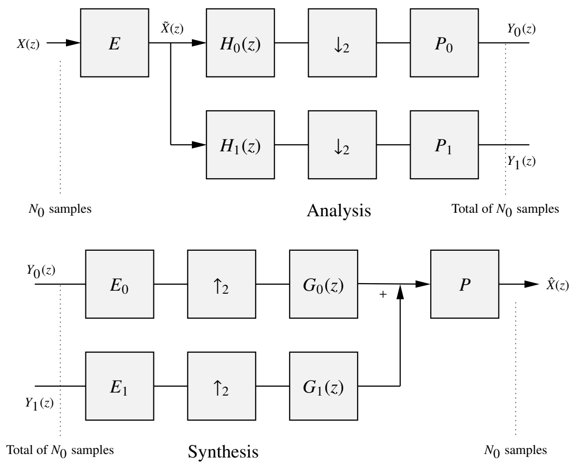

Fig. 5. Filter bank with symmetric pre-extension for nonexpansive transformation of finite-length input signals.

sample (HS) symmetry. WS symmetry of x about $i_0 \in \mathbb{Z}$ can be expressed in terms of the transform domain polyphase vector [14] as

$$\boldsymbol{X}(z^{-1}) = z^{i_0}\boldsymbol{\Lambda}(z)\boldsymbol{X}(z), \tag{2}$$

and HS symmetry about half-integer $i_0 \in \mathbb{Z}/2$ is expressed as

$$\boldsymbol{X}(z^{-1}) = z^{(2i_0-1)/2}\mathbf{J}\boldsymbol{X}(z), \quad \text{where} \tag{3}$$

$$\boldsymbol{\Lambda}(z) = \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \quad \text{and} \quad \mathbf{J} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{4}$$

The $E_s^{(1,1)}$ and $E_s^{(2,2)}$ extensions (notation follows [12]) are constructed by reflection with WS and HS symmetry respectively. Denote the extension constructed with WS symmetry at the left and HS symmetry at the right as $E_s^{(1,2)}$, and the reverse as $E_s^{(2,1)}$. Antisymmetric extension is obtained by negating the reflected samples, giving whole-sample antisymmetry (WA) and half-sample antisymmetry (HA). (A signal may only be WA about a zero sample.) Antisymmetric extensions use a subscript 'a'; e.g. $E_a^{(2,2)}$ for an extension which is HA at both ends.

Nontrivial linear phase FIR filter banks consist of two types: odd-length symmetric filters (WS filter banks), or even-length symmetric lowpass and antisymmetric highpass filters (HS filter banks) [25], [6, pg. 111]. In preparation for tackling the main subject of this paper, symmetric extension for reversible HS filter banks, we now review the results for the considerably more straightforward WS case.

The standard way of constructing invertible, nonexpansive symmetric pre-extension transforms for WS filter banks is to extend the input vector using the $E_s^{(1,1)}$ extension so that the extended input, $\tilde{x} \equiv E_s^{(1,1)}x$, is itself whole-sample symmetric. Regardless of whether the length, N_0, of the input vector is even or odd, the period of the extended signal is even $(2N_0 - 2)$ and both the lowpass and highpass subbands, $\tilde{y}_0$ and $\tilde{y}_1$, are symmetric. An important advancement in our understanding of these symmetries was made in [15], where the authors showed that if a WS filter bank acts on a WS input signal and generates *interleaved* output (i.e., alternating lowpass and highpass outputs), then the interleaved output will also be WS. The subband symmetries are thus implicit in the symmetry of the interleaved output. This was proven in [15, Section 6.5.2] using a novel form of *time-varying convolution*; the proof was subsequently simplified in [14, Theorem 4] using polyphase techniques. The result may be stated more precisely as follows.

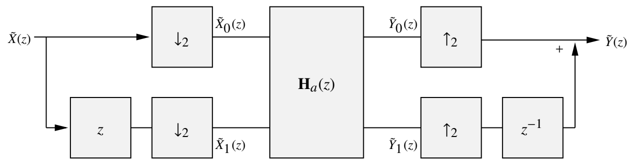

Fig. 6. Polyphase analysis filtering with interleaved output.

Let $\{H_0, H_1\}$ be a WS filter bank that satisfies the *delay-minimized* convention [14, Section III-B-1] adopted by the FBI WSQ Specification [26,27] and the JPEG 2000 standard: the lowpass impulse response, h_0, is symmetric about $\gamma_0 = 0$ and the highpass impulse response, h_1, is symmetric about $\gamma_1 = -1$. This is characterized in the polyphase domain by [14, equation (28)]:

$$\mathbf{H}_a(z^{-1}) = \mathbf{\Lambda}(z)\mathbf{H}_a(z)\mathbf{\Lambda}(z^{-1}) \,, \tag{5}$$

where $\mathbf{\Lambda}(z)$ is defined in (4). By *interleaved output* we mean the signal defined by Figure 6. The output, $\tilde{y}$, is formed by interleaving the subbands, $\tilde{y}_0$ and $\tilde{y}_1$, generated by the analysis filter bank. This interleaved subband structure is identical to the interleaved structure generated by time-varying convolution in [15]. With this notation, Theorem 4 from [14] can be stated as

Theorem 1 *Suppose that the input, $\tilde{x}(n)$, to Figure 6 is whole-sample symmetric about $i_0 \in \mathbb{Z}$, which is equivalent to saying that $\tilde{\mathbf{X}}(z)$ satisfies (2). If $\mathbf{H}_a(z)$ satisfies (5) then $\tilde{\mathbf{Y}}(z)$ also satisfies (2), so $\tilde{y}(n)$ is also symmetric about i_0.*

The symmetry properties of the subbands $\tilde{y}_0$ and $\tilde{y}_1$ are therefore identical to the symmetries of the extended input polyphase components, $\tilde{x}_0$ and $\tilde{x}_1$. These symmetries depend on the channel and on the parity of i_0. Explicit formulas for these symmetries when $i_0 = 0$ were derived in [12, Section 3.2 and Table 5] from the direct-form representation in Figure 5 and were rederived in [21, Section VI] via lifting factorization arguments. Alternatively, the symmetries can be deduced from (2).

For WS filter banks satisfying (5), an important characterization of lifting factorizations (1), is given in [14, Theorem 9] and [15, Section 6.4.4]:

Theorem 2 *Every WS filter bank satisfying (5) factors completely into lifting matrices (1) whose lifting filters, $S_k(z)$, are* half-*sample symmetric. It follows that the lifting matrices, $\mathbf{S}_k(z)$, also satisfy (5).*

This result can be used to construct an alternative extension procedure for WS filter banks that is mathematically equivalent to symmetric pre-extension.

3.2 Lifting Step Extension

Let $\mathbf{H}_a(z)$ be an FIR perfect reconstruction filter bank with a lifting factorization as shown in Figure 3. Let $x(n)$, $n = 0, \ldots, N_0 - 1$, be a finite input signal of length N_0. Demultiplex x into its (finite-length) polyphase components, x_0 and x_1, *without* performing any sort of extension. In order to input x into the first lifting step in Figure 3, it is only necessary to extend the finite-length component x_1 prior to applying the first lifting filter, $S_0(z)$, and it is only necessary to compute enough filtered samples to update the finitely many samples in component x_0. Let E_0 denote an extension operator to be applied to channel 1 for such a purpose. Using transform-domain notation, the intermediate result, $X_0^{(1)}(z)$, of the first lifting update can be written as

$$X_0^{(1)}(z) = S_0(z)\,(E_0 X_1)(z)\ .$$

This process is illustrated in Figure 7 and can be performed on subsequent lifting steps as well. Thus, the result of the next lifting update in Figure 7 is

$$X_1^{(2)}(z) = S_1(z)\left(E_1 X_0^{(1)}\right)(z)\ ,$$

for some extension operator E_1 (not necessarily the same as E_0). Note that the channel providing the input to each lifting filter remains unchanged: $X_1^{(1)} = X_1$ and $X_0^{(2)} = X_0^{(1)}$. The important observation is that all of the intermediate update results are the same length as the initial polyphase components, that is,

$$\text{length}\left(X_0^{(k)}\right) = \text{length}\left(X_0\right) \quad \text{and} \quad \text{length}\left(X_1^{(k)}\right) = \text{length}\left(X_1\right)\ .$$

This means that we are not carrying forward additional "boundary samples" generated by some form of pre-extension from one lifting step to the next. Moreover, the scheme is invertible for *any* choice of extension operators, E_k, as long as the same extensions are used in the synthesis bank, as shown in Figure 7. The scheme also provides bit-perfect reconstruction with reversible implementations. This approach has gone by a variety of names: *interleaved extension* [17], *iterated extension* [28],

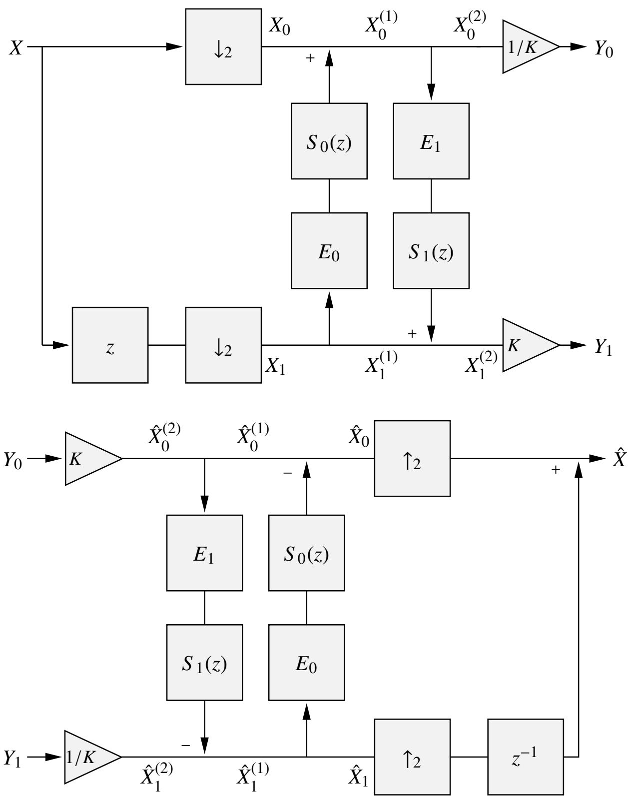

Fig. 7. Lifted filter bank with lifting step extensions.

per-displace-step extension [20], and *per-lifting-step extension* [21]. We think the term *lifting step extension* is both short and descriptive and will use it henceforth.

How does one define a lifting step extension scheme that is equivalent to symmetric pre-extension when $\mathbf{H}_a(z)$ is a WS filter bank satisfying (5)? Form the WS extension of the finite length input signal, $\tilde{x} \equiv E_s^{(1,1)} x$, and let

$$\mathbf{H}_a(z) = \mathrm{diag}(1/K, K)\, \mathbf{S}_{N_{LS}-1}(z) \cdots \mathbf{S}_1(z)\, \mathbf{S}_0(z)$$

be a lifting factorization of $\mathbf{H}_a(z)$ into HS lifting steps whose lifting matrices, $\mathbf{S}_k(z)$, satisfy (5), as guaranteed by Theorem 2. Using transform-domain polyphase vector notation, input $\tilde{\boldsymbol{X}}(z)$ to the lifting factorization of $\mathbf{H}_a(z)$ and denote the intermediate vectors as

$$\tilde{\boldsymbol{X}}^{(k+1)}(z) = \mathbf{S}_k(z)\, \tilde{\boldsymbol{X}}^{(k)}(z) \quad , \quad \tilde{\boldsymbol{X}}^{(0)}(z) \equiv \tilde{\boldsymbol{X}}(z)\ .$$

Since each lifting matrix satisfies (5), Theorem 1 implies that each intermediate vector, $\tilde{\boldsymbol{X}}^{(k)}(z)$, will satisfy (2) for $i_0 = 0$ and $i_0 = N_0 - 1$. As pointed out in Section 3.1, this implies that the individual subbands, $\tilde{x}_0^{(k)}$ and $\tilde{x}_1^{(k)}$, possess the same symmetry properties as the initial extended input polyphase components, $\tilde{x}_0$ and $\tilde{x}_1$. It is therefore possible to truncate the intermediate subbands to minimal, nonredundant sets, $x_0^{(k)}$ and $x_1^{(k)}$, with N_0 samples combined and carry this minimal amount of data forward to the next lifting step. The symmetric intermediate data can then be *recreated* by applying an appropriate lifting step extension to the subband being input to the next lifting filter. This results in a lifting step extension scheme that is mathematically equivalent to WS symmetric pre-extension. Since rounding operations do not affect the symmetries of the intermediate subbands, this same lifting step extension scheme is (bit-wise) equivalent to symmetric pre-extension for reversible lifting implementations of WS filter banks.

The general equivalence of symmetric pre-extension and lifting step extension for WS filter banks was noted in [17] and [15, Section 6.5.3], where it was pointed out that the equivalent lifting step extensions, E_k, in the case of first-order HS lifting filters consist of constant (zeroth-order) replication of the subband end-samples. The general equivalence was also noted by Adams and Ward [21, Section IX], who presented a detailed analysis of the case of WS filter banks with first-order HS lifting filters. In Section 4 we will apply this approach to the question of symmetric pre-extension versus lifting step extension for HS filter banks.

4 HS Filter Banks and Obstructions to Reversibility

4.1 Subband Symmetries

Symmetric pre-extension for HS filter banks uses the (2,2)-symmetric extension, $\tilde{x} = E_s^{(2,2)} x$. We begin by reviewing the subband symmetries in the HS case. An HS analysis bank satisfying the concentric delay-minimized convention [14, Section III-C.1] consists of a lowpass filter, $H_0(z)$, symmetric about $\gamma_0 = -1/2$, and a highpass filter, $H_1(z)$, antisymmetric about $\gamma_1 = -1/2$, which can be expressed

$$\mathbf{H}_a(z^{-1}) = \mathbf{L}\,\mathbf{H}_a(z)\,\mathbf{J}, \quad \text{where} \quad \mathbf{L} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{6}$$

The subband symmetries generated by such a filter bank follow from [14, Theorem 7]:

Theorem 3 *If the input, $\tilde{x}(n)$, to a concentric delay-minimized HS filter bank is symmetric about an odd multiple of 1/2 (i.e., $i_0 = (2k+1)/2$), then the lowpass subband, $\tilde{y}_0(n)$, is symmetric about* $(2i_0 - 1)/4$ *while the highpass subband, $\tilde{y}_1(n)$, is antisymmetric about* $(2i_0 - 1)/4$.

Symmetric pre-extension for HS filter banks uses input vectors that are extended symmetrically about $-1/2$ and $N_0 - 1/2$. By Theorem 3, this results in the subband symmetries shown in Figure 8 for N_0 even, and the symmetries shown in Figure 9 for N_0 odd. It is these symmetries that enable nonexpansive symmetric pre-extension transforms for HS filter banks. With reversible implementations, however, we shall see that the rounding operators in the lifting steps create obstacles to obtaining symmetric subbands.

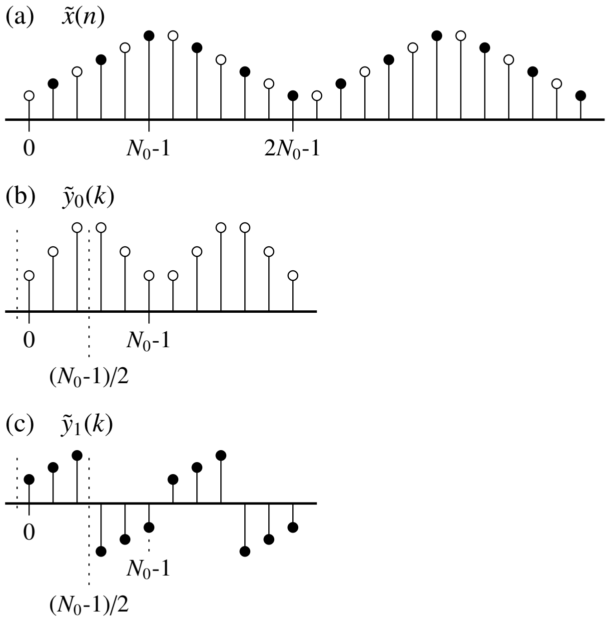

Fig. 8. Subband symmetries generated by an HS filter bank from an even-length input. (a) Periodic HS extension, $\tilde{x} = E_s^{(2,2)}x$. (b) Lowpass subband, $\tilde{y}_0(k)$. (c) Highpass subband, $\tilde{y}_1(k)$.

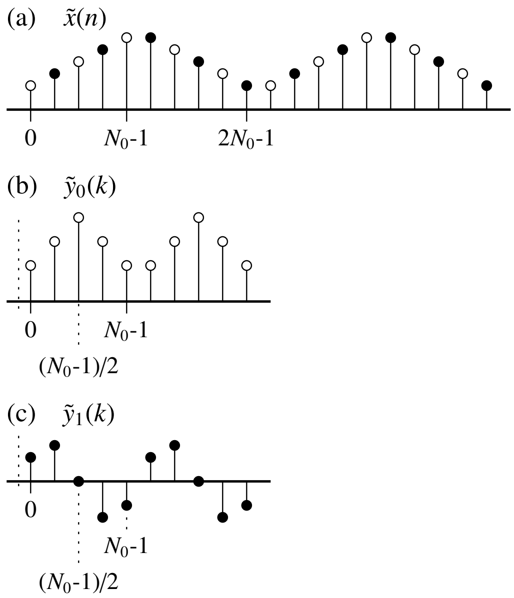

Fig. 9. Subband symmetries generated by an HS filter bank from an odd-length input. (a) Periodic HS extension, $\tilde{x} = E_s^{(2,2)}x$. (b) Lowpass subband, $\tilde{y}_0(k)$. (c) Highpass subband, $\tilde{y}_1(k)$.

4.2 Lifting Factorizations

Factoring HS filter banks into linear phase lifting steps is not as straightforward as factoring WS filter banks into half-sample symmetric lifting steps per Theorem 2.

The linear phase filters that lift an HS filter bank to a higher-order HS filter bank are the whole-sample *anti*symmetric (WA) filters of the form

$$H(z) = \sum_{n=1}^{k} h(n)\,(z^{-n} - z^{n}) \ .$$

Unfortunately, it is never possible to factor an HS filter bank completely into WA lifting steps [14, Theorem 13]. For instance, the Haar filter bank factors into constant (zeroth-order) lifting steps. In general, when factoring an HS filter bank into WA lifting steps one always reaches a point at which one is left with a lower-order HS factor containing filters of *equal lengths* (called an *equal-length HS base*), which cannot be factored further using WA lifting steps. The general situation is given by [14, Theorem 14]:

Theorem 4 *Every HS filter bank satisfying (6) can be lifted using WA lifting steps from an equal-length HS base filter bank,* $\mathbf{B}(z)$*, which also satisfies (6):*

$$\mathbf{H}_a(z) = \mathbf{S}_{N_{LS}-1}(z) \cdots \mathbf{B}(z) \ . \tag{7}$$

A good example is the 2-tap/10-tap HS filter bank [8, Annex H.4.1.1.3], which is lifted from the Haar filter bank via a fourth-order WA highpass lifting step, $\mathbf{S}(z)$:

$$\mathbf{H}_a(z) = \mathbf{S}(z)\,\mathbf{B}_{\text{haar}}(z) \ . \tag{8}$$

Theorem 4 is the key to understanding symmetric pre-extension with reversible HS filter banks since, as we shall show, the only unavoidable problems that arise can be traced back to the lifting factorization of the base HS filter bank, $\mathbf{B}(z)$.

4.3 Lifting Step Extension

As seen in Figure 8(a) and Figure 9(a), the polyphase components of $\tilde{x} = E_s^{(2,2)}x$ are not symmetric, but form *mirror images* of one another. This is expressed algebraically by equation (3). One consequence is that there is *no way* to extend x_1 in the first lifting step of Figure 7 to obtain the extended polyphase component defined by (2,2)-symmetric pre-extension, $\tilde{x}_1$ as we did for (1,1)-symmetric pre-extension. As a result of the mirror-image property for the initial polyphase components of $\tilde{x} = E_s^{(2,2)}x$, a lifting implementation of an HS filter bank must be initialized with (2,2)-symmetric pre-extension in the base HS filter bank, $\mathbf{B}(z)$, in order to obtain symmetric intermediate subbands. Since subsequent WA lifting steps, $\mathbf{S}_k(z)$, will lift $\mathbf{B}(z)$ to higher-order intermediate HS filter banks, the symmetries described by Theorem 3 also hold for the output of all subsequent WA lifting steps. This means that, following $\mathbf{B}(z)$, we can stop carrying forward the boundary extensions for the intermediate subbands and instead proceed via lifting step extension, carrying forward a combined total of just N_0 samples for channels 0 and 1 and reconstructing the symmetries given in Theorem 3 using appropriate extension operators, E_i.

This scheme, which was described in [17], is particularly simple if the base filter bank is the Haar filter bank, $\mathbf{B}_{\text{haar}}(z)$, since the lifting steps that factor $\mathbf{B}_{\text{haar}}(z)$ are zeroth-order. If N_0 is even, the Haar filter bank does not require any extensions at all. If N_0 is odd then the highpass channel needs to be extended by just one sample, which is the right end-point of the input vector. In both cases, subsequent WA lifting steps can be implemented with symmetric or antisymmetric lifting step extensions to achieve a result equivalent to $(2, 2)$-symmetric pre-extension.

4.4 Preservation of Subband Symmetry with Reversible Lifting Steps

In Figure 4, $S(z)$ represents a WA lifting step following an HS base filter bank that has generated a symmetric even channel, $\tilde{x}_0$, and an antisymmetric odd channel, $\tilde{x}_1$. Since $S(z)\tilde{X}_0(z)$ is antisymmetric in the absence of rounding, the updated odd channel, $\tilde{X}_1(z) + R(S(z)\tilde{X}_0(z))$, will remain antisymmetric if $R(\cdot)$ preserves the antisymmetry of $S(z)\tilde{X}_0(z)$. This forces the rounding rule to be an odd function:

$$R(-x) = -R(x) \; .$$

While this constraint is not satisfied by the floor function, it is satisfied by fractional-part truncation, which returns the integer part of its argument, possibly with a symmetric bias offset:

$$R_\beta(x) = \begin{cases} \lfloor x + \beta \rfloor & \text{if} \quad x \geq 0 \\ \lceil x - \beta \rceil & \text{if} \quad x < 0 \end{cases} \qquad\qquad 0 \leq \beta < 1. \tag{9}$$

In contrast, if $S(z)$ is a WA filter in a *lowpass* lifting step following the HS base, then the update, $S(z)\tilde{X}_1(z)$, will be symmetric about 0. Since any rounding rule will preserve symmetry, there are no constraints on the choice of rounding rule for reversible lowpass updates.

4.4.1 Rounding in the equal-length HS base filter bank

We now raise a more serious difficulty. As shown above, any WS filter bank factors completely into HS lifting steps, and update symmetry is preserved by rounding within each step. HS filter banks, in contrast, generate a more complicated sequence of lifting steps [14, Sec. VI-B]. For instance, the *ELASF* family of Adams and Ward [21] consists of all reversible filter banks lifted via WA steps from the reversible Haar base filter bank. They show that *ELASF* filter banks always yield nonexpansive, reversible symmetric pre-extension transforms and that, moreover, the *ELASF* family is closed under filter bank inversion if the inverse filter bank is renormalized suitably. As we will now show, however, symmetric pre-extension may break down (i.e., fail to produce symmetric subbands) for other reversible HS filter banks due to rounding in the equal-length HS base. In fact, this happens for something as simple as the unrenormalized inverse of an *ELASF* filter bank.

Consider the Haar analysis bank, $\mathbf{H}_a(z) = \mathbf{B}_{\text{haar}}(z)$, and the corresponding inverse, $\mathbf{G}_s(z) = \mathbf{H}_a^{-1}(z)$. Since $\mathbf{G}_s\mathbf{H}_a = I$, we also have $\mathbf{H}_a^T\mathbf{G}_s^T = I$. The matrix $\mathbf{G}_s^T$ will be referred to as the *dual Haar* analysis filter bank. (Note that $\mathbf{H}_a$ and $\mathbf{G}_s^T$ are equivalent up to gain normalization factors.) This new analysis bank demonstrates the rounding problems, shown in Figure 10, that may arise in an equal-length HS base filter bank. Given a length-4 input vector, (a, b, c, d), the $E_s^{(2,2)}$ extension generates the extended input $\tilde{x} = [a, b, c, d, d, c, b, a]$, where the notation $[w, y, z]$ denotes a single period of the infinite periodic signal $\ldots, w, y, z, w, y, z, \ldots$. The polyphase components of $\tilde{x}$ are $\tilde{x}_0 = [a, c, d, b]$ and $\tilde{x}_1 = [b, d, c, a]$, and the lifting filters $S_0(z) = 1$ and $S_1(z) = -1/2$ generate lowpass and highpass channels

$$\tilde{x}_0^{(2)} = [a + b, c + d, c + d, a + b] \tag{10}$$

$$\begin{aligned}\tilde{x}_1^{(2)} = [&b + R(-(a + b)/2), d + R(-(c + d)/2),\\ &c + R(-(c + d)/2), a + R(-(a + b)/2)] \quad .\end{aligned} \tag{11}$$

Imposing the constraint that the highpass channel be antisymmetric, we obtain

$$b + R\left(-\frac{a+b}{2}\right) = -\left(a + R\left(-\frac{a+b}{2}\right)\right) \;, \quad \forall a, b \in \mathbb{Z}.$$

The substitution $n = -(a + b)$ implies

$$\frac{n}{2} = R\left(\frac{n}{2}\right) \;, \quad \forall n \in \mathbb{Z},$$

which is not satisfied by any valid rounding procedure. While a renormalization of this filter bank results in a filter bank that is compatible with symmetric pre-extension [20,21], equivalence of the filter bank and its renormalization holds only in the sense of irreversible filter banks — the rounding operations of the reversible filter bank structure destroy this equivalence, so that the renormalized reversible filter bank is a *different* transform, and not merely an alternative implementation of the same transform.

The reader may have noticed that the broken antisymmetry in Figure 10 is irrelevant to reversibility, since erroneous extension of the highpass subband has no effect on the synthesis lifting steps, both of which consist of zeroth-order filters. The broken antisymmetry is relevant, however, if the dual Haar filter bank is lifted to a higher-order filter bank. We demonstrate this with an example input (abandoning the general algebraic expressions used previously) to the 6-tap/2-tap filter bank in Table 1, using the odd rounding function (9) with $\beta = 0$ to show that odd rounding is

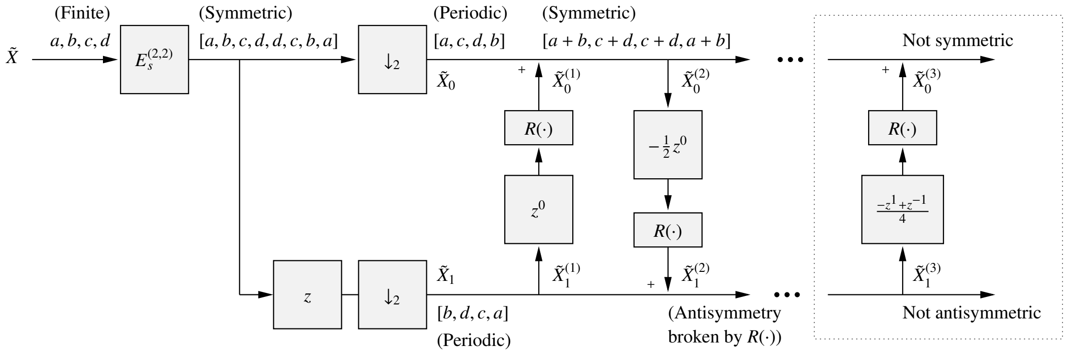

Fig. 10. The dual Haar filter bank with a third lifting step that lifts it to the 6-tap/2-tap filter bank from Table 1.

necessary, but not sufficient, for symmetric pre-extension. Listing the intermediate subbands for the 4-sample input vector (0,3,-3,0):

$$
\begin{array}{llll}
\tilde{x}_0 = & [0,\ -3,\ 0,\ 3] & \tilde{x}_1 = & [3,\ 0,\ -3,\ 0] \\
\tilde{x}_0^{(1)} = & [3,\ -3,\ -3,\ 3] & \tilde{x}_1^{(1)} = & [3,\ 0,\ -3,\ 0] \\
\tilde{x}_0^{(2)} = & [3,\ -3,\ -3,\ 3] & \tilde{x}_1^{(2)} = & [2,\ 1,\ -2,\ -1] \\
\tilde{x}_0^{(3)} = & [3,\ -2,\ -3,\ 2] & \tilde{x}_1^{(3)} = & [2,\ 1,\ -2,\ -1]
\end{array} \quad .
$$

Note that the symmetry is broken in both channels, so that truncation and extension in the synthesis filter bank incorrectly gives $\tilde{y}_0 = [3, -2, -2, 3]$ and $\tilde{y}_1 = [2, 1, -1, -2]$. Computing the intermediate subbands in the synthesis filter bank,

$$
\begin{array}{llll}
\tilde{y}_0 = & [3,\ -2,\ -2,\ 3] & \tilde{y}_1 = & [2,\ 1,\ -1,\ -2] \\
\tilde{y}_0^{(3)} = & [3,\ -2,\ -2,\ 3] & \tilde{y}_1^{(3)} = & [2,\ 1,\ -1,\ -2] \\
\tilde{y}_0^{(2)} = & [3,\ -2,\ -2,\ 3] & \tilde{y}_1^{(2)} = & [3,\ 0,\ -2,\ -1] \\
\tilde{y}_0^{(1)} = & [0,\ -2,\ 0,\ 4] & \tilde{y}_1^{(1)} = & [3,\ 0,\ -2,\ -1]
\end{array} \quad .
$$

This gives a reconstructed output of $[0, 3, -2, 0]$, which is not equal to the input.

4.5 An Exceptional Class of HS Filter Banks

Due to an interesting coincidence, the well known 2-tap/6-tap [8, Sec. H.4.1.1.2] and 2-tap/10-tap [8, Sec. H.4.1.1.3] filter banks (as well as any others lifted from the Haar by a single WA highpass update) are reversible with the floor rounding rule. This is probably why the failure of symmetric pre-extension for certain reversible HS filter banks was not observed earlier. While the correct symmetry and antisymmetry are generated by the Haar base filter bank using floor-function rounding, the same rounding destroys antisymmetry in the highpass channel after the final WA lifting step, as described above. Thus, the (non-symmetric) highpass channel is incorrectly extended in the synthesis bank. Due to the structure of the synthesis lifting, though, this error has no effect on the synthesis. The error is irrelevant to the first synthesis step since it is an odd channel update, and the final two steps both consist of zeroth-order lifting filters, which therefore do not make use of erroneous extended values when updating the unextended parts of their channels (see Figure 11).

5 Range Space Preserving Rounding

Since no scalar rounding operation suffices for reversible rounding in the equal-length HS base filter bank (recall that an odd rounding rule is suitable for the subsequent lifting steps, once the desired symmetries have been established), we now examine a more general class of operations that, in principle enables the implementation of reversible non-expansive HS filter banks with symmetric pre-extension. This discussion is most conveniently presented by considering lifting within a linear algebra framework rather than in the usual filter bank form. Within this framework, input signals are represented as column vectors, and filters are represented by

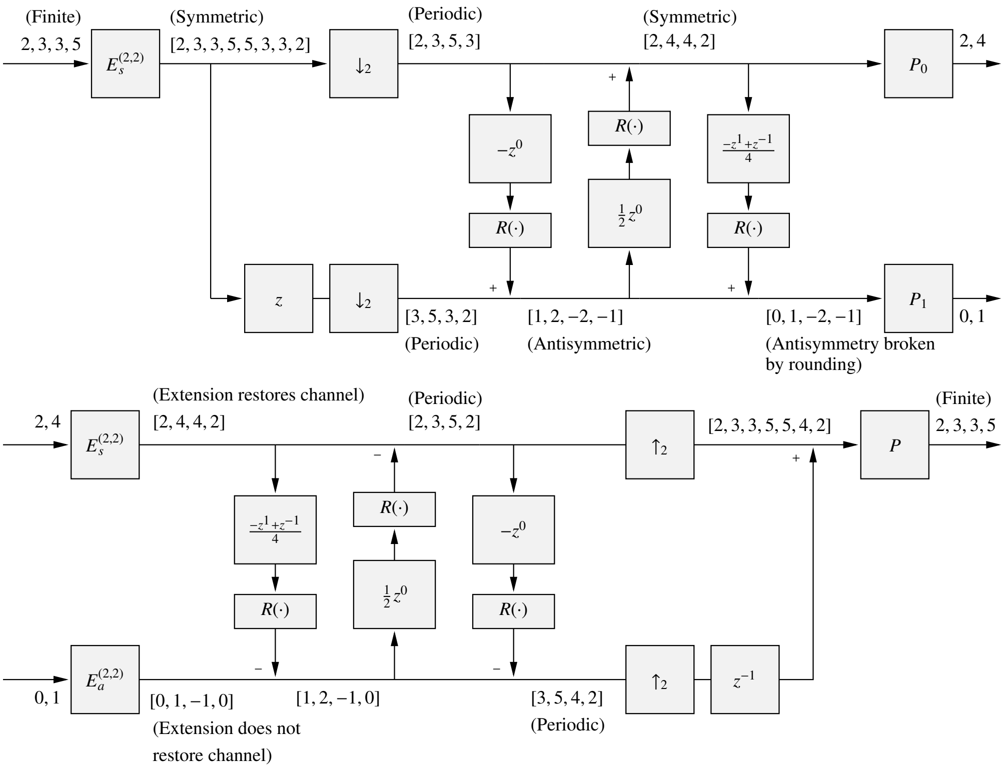

Fig. 11. Reversible 2-tap/6-tap filter bank. A single period of an in-principle infinite signal is denoted as in Figure 10. The rounding operation is $R(x) = \lfloor x \rfloor$. While synthesis does not correctly restore the full symmetric input, the original finite extent part of the input is correctly reconstructed because the first synthesis step is computed from the even channel, which is correct, and the subsequent steps are single taps at z^0 and therefore do not transfer information from the incorrect extension into the correct original part of the signal.

the equivalent linear operators. An infinite symmetrically extended signal is represented by a finite-length vector of a single period, since this retains all of the information present in the infinite length symmetric signal. Where filters are applied to such symmetrically extended signals, the corresponding linear operator representation may always be constructed so that they are equivalent when applied to the vector representing a single period of that signal.

We simplify the following discussion by considering input vector $\mathbf{x} \in \mathbb{R}^{N_0}$, with N_0 assumed to be even. In the WS case, the $E_s^{(1,1)}$ extension followed by the polyphase decomposition maps a length N_0 input into even and odd channels, each of which have a period of length $N_0 - 1$, and symmetry such that they each contain $N_0/2$ unique samples (for example, the input a, b, c, d, e, f is mapped to even and odd channels $[a, c, e, e, c]$ and $[b, d, f, d, b]$ respectively). In vector form, therefore, the $E_s^{(1,1)}$ extension followed by the polyphase decomposition maps input vector $\mathbf{x} \in \mathbb{R}^{N_0}$ into even and odd channels $\mathbf{x}_0 \in \mathbb{R}^{N_0-1}$ and $\mathbf{x}_1 \in \mathbb{R}^{N_0-1}$ respectively. In the HS case, the $E_s^{(2,2)}$ extension followed by the polyphase decomposition maps a length N_0 input into even and odd channels each of which have a period of length N_0, and are reflections of one another (for example, the input a, b, c, d, e, f is mapped to even and odd channels $[a, c, e, f, d, b]$ and $[b, d, f, e, c, a]$ respectively). In vector form, therefore, the $E_s^{(2,2)}$ extension followed by the polyphase decomposition maps input vector $\mathbf{x} \in \mathbb{R}^{N_0}$ into even and odd channels $\mathbf{x}_0 \in \mathbb{R}^{N_0}$ and $\mathbf{x}_1 \in \mathbb{R}^{N_0}$ respectively.

By concatenating $\mathbf{x}_0$ and $\mathbf{x}_1$ into a single column vector $(\mathbf{x}_0^T \;\; \mathbf{x}_1^T)^T$, we may represent the action of an entire lifting step (i.e., both the filter and the channel update) by the linear operator $S_k = I + U_k$, where

$$U_k = \begin{pmatrix} 0 & F_k \\ 0 & 0 \end{pmatrix} \quad \text{for an even update, and}$$

$$U_k = \begin{pmatrix} 0 & 0 \\ F_k & 0 \end{pmatrix} \quad \text{for an odd update,}$$

and F_k is the linear operator representing the lifting filter. We denote the operators taking $\mathbf{x}$ to $(\mathbf{x}_0^T \;\; \mathbf{x}_1^T)^T$ by $P_{\mathrm{WS}} : \mathbb{R}^{N_0} \mapsto \mathbb{R}^{2N_0-2}$ and $P_{\mathrm{HS}} : \mathbb{R}^{N_0} \mapsto \mathbb{R}^{2N_0}$ for the WS and HS cases respectively, allowing the action of the entire filter bank may be expressed as $S_{N_{LS}-1} \dots S_1 S_0 P_{\mathrm{WS}}$ or $S_{N_{LS}-1} \dots S_1 S_0 P_{\mathrm{HS}}$.

We now consider the range spaces of P_{WS} and $S_k \dots S_1 S_0 P_{\mathrm{WS}}$, which we denote by R_{WS} and $R_{\mathrm{WS},k}$ respectively. In this case $\mathbf{x}_0$ and $\mathbf{x}_1$ each have symmetry such that R_{WS} is an N_0 dimensional subspace of $\mathbb{R}^{2N_0-2}$. If we define R'_{WS} as the space of signals with the required symmetry properties for a non-expansive and invertible WS filter bank, we have that $R_{\mathrm{WS}} = R'_{\mathrm{WS}}$, so that the each lifting step S_k maps from $R_{\mathrm{WS},k-1} \subset R'_{\mathrm{WS}}$ to $R_{\mathrm{WS},k} \subset R'_{\mathrm{WS}}$, as illustrated in Figure 12.

Similarly, denote the range spaces of P_{HS} and $S_k \dots S_1 S_0 P_{\mathrm{HS}}$, by R_{HS} and $R_{\mathrm{HS},k}$ respectively. Since $\mathbf{x}_1$ is a reflection of $\mathbf{x}_0$, it is clear that R_{HS} is an N_0 dimensional

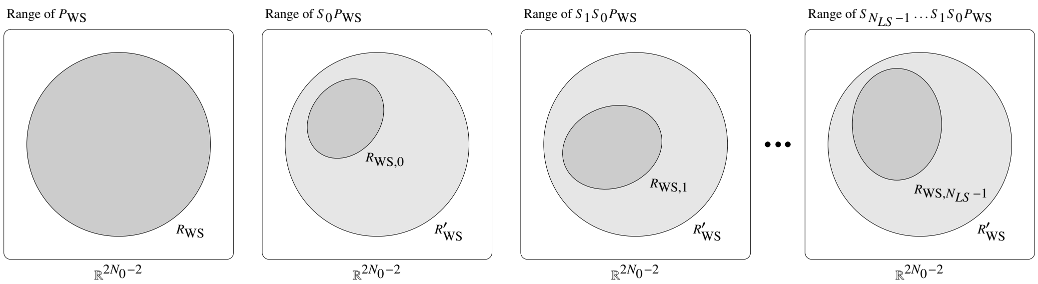

Fig. 12. Conceptual illustration of the evolution of range spaces in a WS filter bank.

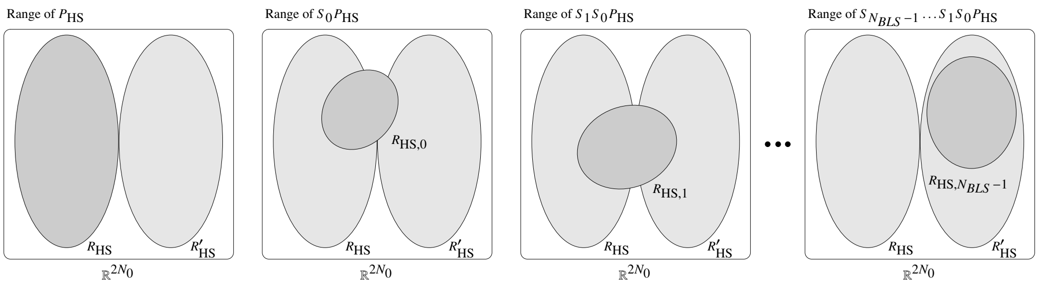

Fig. 13. Conceptual illustration of the evolution of range spaces in an HS filter bank.

subspace of $\mathbb{R}^{2N_0}$. The symmetry properties necessary for a non-expansive and invertible HS filter bank imply that $S_{N_{LS}-1} \ldots S_1 S_0 P_{\mathrm{HS}}\mathbf{x}$ should have a symmetric upper component and an antisymmetric lower component — we denote the corresponding space by R'_{HS}. In contrast to the WS case, $R_{\mathrm{HS}} \neq R'_{\mathrm{HS}}$, so that the required symmetry properties emerge progressively as each S_k maps from $R_{\mathrm{HS},k-1} \not\subset R'_{\mathrm{HS}}$ to $R_{\mathrm{HS},k} \not\subset R'_{\mathrm{HS}}$ (see Figure 13) until the the end of the HS base filter bank, at which point the remaining WA steps preserve this symmetry.

A reversible filter bank is constructed by replacing each step S_k with $S'_k = I + Q_k U_k$, where Q_k is a rounding operation for step k. In the WS case, range space R_{WS} has the required symmetry, which is preserved by each S_k. While the insertion of the rounding operations may modify each space $R_{\mathrm{WS},k}$, it does not perturb the fundamental symmetry property, so that $R_{\mathrm{WS},k} \subset R'_{\mathrm{WS}}$ for each step k. In contrast, in the HS case, the required symmetry properties emerge only after the HS base filter bank: in general, the modification of $R_{\mathrm{HS},k}$ by the insertion of the rounding operations will perturb subsequent $R_{\mathrm{HS},l} \quad l > k$ so that the essential symmetries are not preserved, and $R_{\mathrm{HS},N_{BLS}-1} \not\subset R'_{\mathrm{HS}}$ (where N_{BLS} denotes the number of lifting steps in the equal-length HS base filter bank).

A sufficient condition for the final symmetries to be preserved is that each rounding operation Q_k not remove its operand from the range space of $U_k S_{k-1} \ldots S_1 S_0 P_{\mathrm{HS}}$ — in general this requires a vector rounding operation such as

$$Q_k(\mathbf{u}) = \operatorname*{arg\,min}_{\mathbf{v} \in \mathbb{Z}^{2N_0}, \mathbf{v} \in \operatorname{ran}(U_k S_{k-1} \ldots S_1 S_0 P_{\mathrm{HS}})} \|\mathbf{u} - \mathbf{v}\| \; ,$$

which rounds input $\mathbf{u}$ to the intersection of that range space and the lattice of inte-

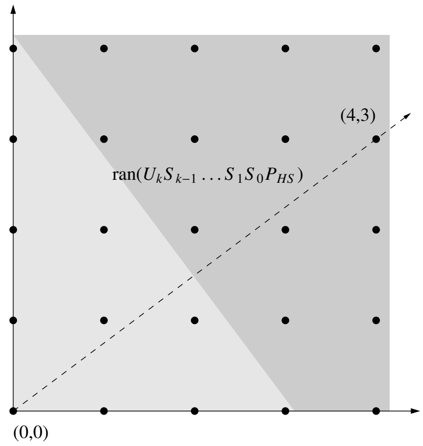

Fig. 14. Diagram illustrating the intersection of the lattice $\mathbb{Z}^2$ and the range space ran$(U_k S_{k-1} \ldots S_1 S_0 P_{\mathrm{HS}})$ of lifting step k. This intersection may be very sparse, resulting in large quantization errors — in this example, all vectors within the lightly shaded region would be quantized to the origin $(0,0)$, and all those within the darker region would be quantized to the lattice and range space intersection point $(4,3)$.

gers $\mathbb{Z}^{2N_0}$, and is a form of Lattice Vector Quantization [29]. The lattice resulting from this intersection may, depending on the filter bank, be less dense than the integer lattice, and therefore have a larger rounding error (see Figure 14) than scalar rounding. This is not necessarily the case, however, as is illustrated in the following example.

5.1 *Symmetry-Preserving Lattice Vector Quantization for the Dual Haar Filter Bank*

We now demonstrate how to apply the above theory to the rounding problem for the dual Haar analysis filter bank described in Section 4.4.1. Continue to assume $N_0 = 4$ as in Section 4.4.1; the initial range space, R_{HS}, is the 4-dimensional subspace of $\mathbb{R}^8$ consisting of vectors whose first 4 entries are the mirror image of the last 4,

$$R_{HS} = \{[a, c, d, b, b, d, c, a] \ : \ a, b, c, d \in \mathbb{R}\} .$$

In the absence of any rounding, the subbands generated by the dual Haar are given by (10) and (11). Thus, the range space after the second lifting step consists of vectors whose first 4 entries are symmetric and whose second 4 are antisymmetric:

$$R_{HS,1} = \{[w, x, x, w, y, z, -z, -y] \ : \ w, x, y, z \in \mathbb{R}\} .$$

Restrict attention to integer inputs, $a, b, c, d \in \mathbb{Z}$, for the sake of considering reversible implementations. (Note that the rounding operation in the first lifting step in Figure 10 is superfluous since the first lifting filter maps integer input to integer output.) As we saw, there is no scalar rounding rule, R, that will generate antisymmetry in the highpass subband, $\tilde{X}_1^{(2)}$, following the second lifting step in Figure 10. To remedy this, define the following simple (coordinate-wise) lattice vector quantizer:

$$Q_1([w, x, y, z]) = [\lfloor w \rfloor, \lfloor x \rfloor, \lceil y \rceil, \lceil z \rceil] \ . \tag{12}$$

Apply Q_1 to the output of the second lifting filter, S_1, in Figure 10 and update channel 1:

$$\begin{aligned}\tilde{x}_1^{(2)} &= \tilde{x}_1^{(1)} + Q_1(s_1 * \tilde{x}_0^{(1)}) \\ &= [\, b,\, d,\, c,\, a \,] \ + \ [\, \lfloor -(a+b)/2 \rfloor,\ \lfloor -(c+d)/2 \rfloor, \\ &\quad \lceil -(c+d)/2 \rceil,\ \lceil -(a+b)/2 \rceil \,] \ .\end{aligned}$$

To see that this subband is antisymmetric, add the first and last elements in a period:

$$\begin{aligned}\tilde{x}_1^{(2)}(0) + \tilde{x}_1^{(2)}(3) &= b + \lfloor -(a+b)/2 \rfloor + a + \lceil -(a+b)/2 \rceil \\ &= b + a - (a+b) \ = \ 0\end{aligned}$$

using the identity

$$\lfloor n/2 \rfloor + \lceil n/2 \rceil = n \ , \quad n \in \mathbb{Z} \ .$$

Similarly, $\tilde{x}_1^{(2)}(1) + \tilde{x}_1^{(2)}(2) = 0$, proving that $\tilde{x}_1^{(2)}$ has an antisymmetric period, which means that Q_1 preserves the range space $R_{HS,1}$. Moreover, the restriction of Q_1 to vectors of half-integers is a nearest-neighbor vector quantizer, so we expect the rate-distortion performance of this implementation to be comparable to that obtained using a scalar nearest-neighbor quantizer on the output of S_1, as in Figure 10.

If the factorization of an equal-length HS base filter bank is more complicated than this, involving multiple lifting filters with rounded output, then the construction of range space-preserving lattice quantizers may be harder than it was in the above example, depending on how the intermediate range spaces are embedded in $\mathbb{R}^{N_0}$. This may also lead to increased granular quantization noise in the rounded lifting updates, which does not affect reversibility, but may have an impact on the coding gain and rate-distortion performance of the reversible filter bank.

6 Conclusions

WS and HS filter banks have very different lifting factorizations, which has a significant impact on their use with symmetric pre-extension schemes for finite-length inputs. All WS filter banks factor into a sequence of HS lifting steps, each of which preserves the symmetries of the polyphase components for whole-sample symmetric input, and any rounding operation may be used to form integer to integer filter bank implementations. HS filter banks factor into an equal-length HS base filter bank, which in turn can be factored using general lifting filters, followed by a sequence of WA lifting steps. The highpass channel generated by the HS base filter bank is antisymmetric, imposing the requirement that any symmetry-preserving rounding operation be an odd function. More seriously, there may be no scalar rounding operation that generates antisymmetry in the highpass channel for certain lifted HS base filter banks. This poses an obstruction to forming nonexpansive reversible symmetric pre-extension transforms for arbitrary reversible HS filter banks. We have shown that, in some cases, this obstruction may be overcome by using vectorized rounding rules, but it is not clear whether this is a universal solution, and rounding via lattice vector quantization may potentially degrade filter bank coding performance.

The discovery of the incompatibility of symmetric pre-extension with reversible HS filter banks has had a significant impact on Part 2 of the JPEG 2000 standard [8]. Prior to our discovery of these difficulties, drafts of Annexes G and H provided for user-specified WS and HS filter banks respectively, specified by signalling the lifting steps. Once reversible HS filter banks were found to require *lifting step* extension, there was no longer any point to constraining the filter bank symmetries for compatibility with symmetric pre-extension. Thus, in its final form, Annex H does not place any constraints on the user-specified lifting steps, and boundary handling is via lifting step extension.

References

[1] F. A. M. L. Bruekers, A. W. M. van den Enden, New networks for perfect inversion and perfect reconstruction, IEEE Journal on Selected Areas in Communications 10 (1) (1992) 129–137.

[2] A. Said, W. A. Pearlman, An image multiresolution representation for lossless and lossy image compression, IEEE Transactions on Image Processing 5 (1996) 1303–1310.

[3] W. Sweldens, The lifting scheme: a custom-design construction of biorthogonal wavelets, Applied and Computational Harmonic Analysis 3 (2) (1996) 186–200.

[4] I. Daubechies, W. Sweldens, Factoring wavelet transforms into lifting steps, Journal Of Fourier Analysis And Applications 4 (3) (1998) 247–269.

[5] P. P. Vaidyanathan, Multirate Systems and Filter Banks, Prentice Hall, Englewood Cliffs, NJ, USA, 1992.

[6] G. Strang, T. Nguyen, Wavelets and Filter Banks, Wellesley-Cambridge Press, Wellesley, MA, USA, 1996.

[7] JPEG 2000 Image Coding System, Part 1, ISO/IEC Int'l. Standard 15444-1, ITU-T Rec. T.800, Int'l. Org. for Standardization, 2000.

[8] JPEG 2000 Image Coding System, Part 2 (Extensions), ISO Int'l. Standard 15444-2, ITU-T Rec. T.801, Int'l. Org. for Standardization, 2001.

[9] C. D. Giurcaneanu, I. Tabus, J. Astola, Integer wavelet transform based lossless audio compression, in: Proceedings of the IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP'99), Vol. 1, Antalaya, Turkey, 1999, pp. 378–382.

[10] A. Secker, D. Taubman, Highly scalable video compression with scalable motion coding, IEEE Transactions on Image Processing 13 (8) (2004) 1029–1041.

[11] G. Xuan, J. Chen, J. Zhu, Y. Q. Shi, Z. Ni, W. Su, Lossless data hiding based on integer wavelet transform, in: Proceedings of IEEE 5th Workshop on Multimedia Signal Processing, St.Thomas, VI, USA, 2002, pp. 312–315.

[12] C. M. Brislawn, Classification of nonexpansive symmetric extension transforms for multirate filter banks, Applied and Computational Harmonic Analysis 3 (1996) 337–357.

[13] C. M. Brislawn, Symmetric Extension Transforms, in: P. N. Topiwala (Ed.), Wavelet Image and Video Compression, Kluwer, Boston, MA, 1998, Ch. 5, pp. 83–91.

[14] C. Brislawn, B. Wohlberg, The polyphase-with-advance representation and linear phase lifting factorizations, IEEE Transactions on Signal Processing 54 (6 part 1) (2006) 2022–2034.

[15] D. S. Taubman, M. W. Marcellin, JPEG2000: Image Compression Fundamentals, Standards, and Practice, Kluwer, Boston, MA, USA, 2002.

[16] C. M. Brislawn, S. M. Mniszewski, M. D. Pal, A. G. Percus, B. E. Wohlberg, T. Acharya, P.-S. Tsai, M. Lepley, Report on core experiment CodEff03: Even-length filter bank option, Tech. Rep. WG1N2209, ISO/IEC Standards Committee JTC1/SC29/WG1, Stockholm, Sweden (Jul. 2001).

[17] C. M. Brislawn, B. E. Wohlberg, Boundary extensions and reversible implementation for half-sample symmetric filter banks, Tech. Rep. WG1N2119, ISO/IEC Standards Committee JTC1/SC29/WG1, Singapore (Mar. 2001).

[18] JPEG 2000 Image Coding System, Part 5 (Reference Software), ISO/IEC Int'l. Standard 15444-5, Int'l. Org. for Standardization, 2003.

[19] M. D. Adams, F. Kossentini, JasPer: A software-based JPEG-2000 codec implementation, in: Proc. Int'l. Conf. Image Process., IEEE Signal Process. Soc., Vancouver, BC, 2000, pp. 53–56.

[20] M. D. Adams, Reversible integer-to-integer wavelet transforms for image coding, Ph.D. thesis, Dept. of Electrical and Computer Engineering, Univ. of British Columbia, Vancouver, BC, Canada (Sep. 2002).

[21] M. D. Adams, R. K. Ward, Symmetric-extension-compatible reversible integer-to-integer wavelet transforms, IEEE Transactions on Signal Processing 51 (10) (2003) 2624–2636.

[22] B. Wohlberg, C. Brislawn, Reversible integer-to-integer transforms and symmetric extension of even-length filter banks, in: Proceedings of Visual Communication and Image Processing (VCIP) 2003, Lugano, Switzerland, 2003, pp. 1709–1714.

[23] A. Zandi, J. D. Allen, E. L. Schwartz, M. Boliek, CREW: Compression with reversible embedded wavelets., Proceedings of the Data Compression Conference (1995) 212 – 221.

[24] A. R. Calderbank, I. Daubechies, W. Sweldens, B. L. Yeo, Wavelet transforms that map integers to integers, Applied and Computational Harmonic Analysis 5 (3) (1998) 332–369.

[25] T. Q. Nguyen, P. P. Vaidyanathan, Two-channel perfect-reconstruction FIR QMF structures which yield linear-phase analysis and synthesis filters, IEEE Trans. Acoust., Speech, Signal Process. 37 (5) (1989) 676–690.

[26] U.S. Federal Bureau of Investigation, WSQ Gray-Scale Fingerprint Image Compression Specification, IAFIS-IC-0110v2, Washington, DC, 1993.

[27] C. M. Brislawn, J. N. Bradley, R. J. Onyshczak, T. Hopper, The FBI compression standard for digitized fingerprint images, in: Appl. Digital Image Process., Vol. 2847 of Proc. SPIE, SPIE, Denver, CO, 1996, pp. 344–355.

[28] C. Brislawn, B. Wohlberg, A. Percus, Resolution scalability for arbitrary wavelet transforms in the JPEG-2000 standard, in: Visual Communications and Image Processing (VCIP) 2003, Lugano, Switzerland, 2003, pp. 774–784.

[29] A. Gersho, R. M. Gray, Vector Quantization and Signal Compression, Kluwer, Norwell, MA, USA, 1992.