LA-UR-06-8583

*Approved for public release;
distribution is unlimited.*

Title:

Symmetry-preserving lattice vector quantization for reversible half-sample symmetric FIR filter banks

Author(s):

Christopher M. Brislawn
Brendt Wohlberg

Submitted to:

published in the Proceedings of the 2006
Asilomar Conference on Signals, Systems, and Computers
Asilomar, CA, 10/29-11/1, 2006
IEEE, pp. 878-882

Symmetry-Preserving Lattice Vector Quantization for Reversible Half-Sample Symmetric FIR Filter Banks

Christopher M. Brislawn
Computer & Computational Sciences Division
Los Alamos National Laboratory
Los Alamos, NM 87545–1663
E-mail: brislawn@lanl.gov

Brendt Wohlberg
Theoretical Division
Los Alamos National Laboratory
Los Alamos, NM 87545–1663
E-mail: brendt@lanl.gov

***Abstract*— Linear phase FIR filter banks form an integral part of the ISO/IEC JPEG 2000 image coding standard. One feature they enable is lossless subband coding based on reversible filter bank implementations. While this meshes well with symmetric boundary-handling techniques for whole-sample symmetric (odd-length) linear phase filters, there are obstructions with half-sample symmetric (even-length) filters, a fact that influenced the JPEG 2000 standard. We show how these obstructions can be overcome for a class of half-sample symmetric filter banks by employing lattice vector quantization to ensure symmetry-preserving rounding in reversible implementations.**

I. Introduction

We present some subtle problems in the lossless implementation of perfect reconstruction multirate filter banks [1], [2]. One approach for constructing invertible integer to integer transforms is to use rounded updates in a *lifting factorization* [3], [4]. This method is known as *reversible implementation* of a filter bank [5], [6]. Rounding operations applied to the output of each filter ensure that each lifting update uses integer addition and can therefore be inverted losslessly using integer subtraction. (Filter banks implemented *without* rounding are referred to as *irreversible*). A major application of reversible filter banks to date has been image coding, including the ISO/IEC JPEG 2000 image coding standard [7], [8].

A common method of handling boundary conditions for linear phase filter banks with finite-length input vectors is to extend the vectors by symmetric reflection, an approach known as *symmetric (pre-)extension* [9], [10]. When a filter bank and boundary scheme allow a vector of length N_0 to be transformed and reconstructed perfectly from just N_0 subband coefficients we say the scheme is *nonexpansive*. One advantage of symmetric pre-extension over circular convolution is that it is nonexpansive for *all* N_0, whereas circular convolution is only nonexpansive when N_0 is even.

Instead of pre-extending input vectors, one can also define extension operations *within* each lifting step (called *lifting step extensions* [11], [12], [13], [14], [15], [16]) in a way that yields invertible, nonexpansive transforms. Moreover, one can sometimes choose lifting step extensions so that the resulting transform is equivalent to symmetric pre-extension. This is the case for type 1 odd-length linear phase (*whole-sample symmetric,* or WS) filter banks, which factor using type 2

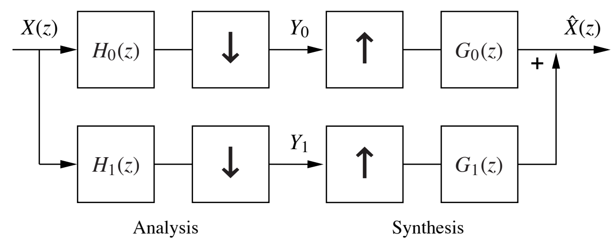

Fig. 1. Direct form representation of a two-channel filter bank.

even-length linear phase (*half-sample symmetric,* or HS) lifting filters. Thus, e.g., the specification of lifting step extension for arbitrary filter banks in JPEG 2000 Part 2 [8, Annex H], based on recommendations in [17], is compatible with the specification of symmetric pre-extension for WS filter banks in JPEG 2000 Part 1 [7, Annex F] and Part 2 [8, Annex G].

The analogous issues for HS filter banks are considerably more complicated [16]. HS filter banks are lifted using whole-sample *anti*symmetric (WA) lifting filters to lift from a "base" HS filter bank, $\mathbf{B}(z)$, with equal-length filters. The equal-length base filter bank in turn factors using non-WA lifting steps. For instance, the Haar filter bank, normalized as in [4], [8], factors using zeroth-order lifting steps:

$$\mathbf{B}_{\text{haar}}(z) = \begin{bmatrix} 1/2 & 1/2 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1/2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}. \qquad (1)$$

Further complications arise when considering reversible implementations. While everything works out nicely for arbitrary reversible WS filter banks, this is not the case for arbitrary reversible HS filter banks.

A. Perfect Reconstruction Filter Banks

The direct-form representation of a two-channel multirate filter bank is shown in Figure 1. This is a *perfect reconstruction* filter bank if $\hat{X}(z) = Az^{-D}X(z)$ for some $A \neq 0$ and integer D. We follow [4] in assuming that $A = 1$ and $D = 0$.

The *polyphase-with-advance* representation [4], [18] is shown in Figure 2. This representation forms the basis for the lifting factorizations used in the JPEG 2000 standard. In this representation, the perfect reconstruction condition becomes $\det \mathbf{H}_a(z) = 1$, with $\mathbf{G}_s(z) = \mathbf{H}_a^{-1}(z)$.

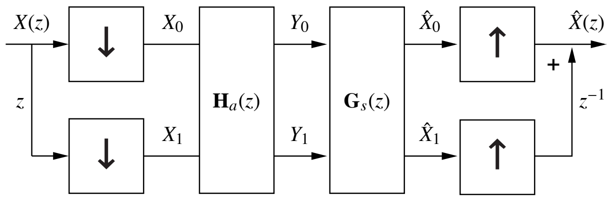

Fig. 2. Polyphase-with-advance representation of a filter bank.

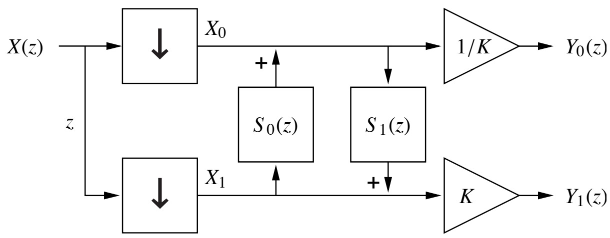

Fig. 3. Lifting decomposition of an analysis filter bank with two lifting steps.

B. Lifting Factorization

Any FIR polyphase matrix of determinant 1 has a *lifting factorization* [4] into a diagonal gain matrix and alternating upper and lower triangular matrices. This has the form

$$\mathbf{H}_a(z) = \mathrm{diag}(1/K, K)\, \mathbf{S}_{N_{LS}-1}(z) \cdots \mathbf{S}_1(z)\, \mathbf{S}_0(z)\ , \qquad (2)$$

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

where the *lifting filters*, $S_k(z)$, are Laurent polynomials. The block diagram in Figure 3 illustrates a factorization of an analysis bank with two lifting steps, corresponding to

$$\mathbf{H}_a(z) = \begin{bmatrix} 1/K & 0 \\ 0 & K \end{bmatrix} \begin{bmatrix} 1 & 0 \\ S_1(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & S_0(z) \\ 0 & 1 \end{bmatrix}.$$

C. Reversible Lifting Implementations

Reversible filter bank implementations [5], [6] are formed by inserting rounding operations, R, in each step of a lifting factorization as shown in Figure 4. For numerical robustness and implementation efficiency, it is customary to use lifting filters whose coefficients are dyadic rationals.

D. Symmetric Extensions of Finite-Length Signals

Filter bank boundary-handling for a finite-length input vector, $x(n)$, $n = 0, \ldots, N_0 - 1$, can be accomplished by extending x symmetrically at its left and right ends. The $E_s^{(1,1)}$ and $E_s^{(2,2)}$ extensions, constructed by reflection with whole- and half-sample symmetry, respectively, are shown in Figures 5(b) and 6(a) for N_0 even. The extended signal, $\tilde{x}$, is continued periodically as an infinite-duration, periodic-symmetric signal. The symmetries of the polyphase components, $\tilde{x}_0$ and $\tilde{x}_1$, for the WS extension, $\tilde{x} \equiv E_s^{(1,1)}x$, of period $2N_0 - 2$ are shown in Figures 5(c–d). Both polyphase components have linear phase, with period $N_0 - 1$. Open dots indicate even-indexed samples and filled dots indicate odd-indexed samples. The figures show two complete periods of $\tilde{x}$ and two periods of $\tilde{x}_0$ and $\tilde{x}_1$.

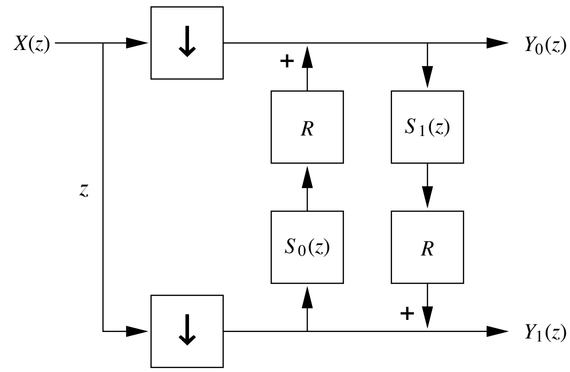

Fig. 4. Reversible lifting implementation of an analysis filter bank.

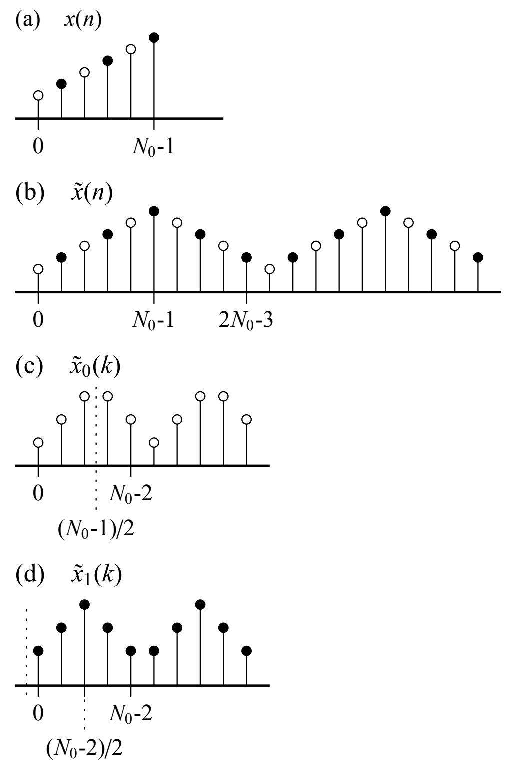

Fig. 5. Polyphase component symmetries for WS extension (even number of input samples). (a) Even-length input. (b) Periodic WS extension, $\tilde{x} = E_s^{(1,1)}x$. (c) Even channel, $\tilde{x}_0(k)$. (d) Odd channel, $\tilde{x}_1(k)$.

II. Linear Phase Filter Banks

A. Subband Symmetries for Linear Phase Filter Banks

1) Whole-sample symmetric filter banks: Let $\{H_0, H_1\}$ be a WS analysis filter bank that satisfies the *delay-minimized* convention [18, Section III-B-1]: h_0 is symmetric about 0 and h_1 is symmetric about -1. Let the input signal be $\tilde{x}(n) = E_s^{(1,1)}x$. It follows from [18, Theorem 4] that the output subbands, $\tilde{y}_0$ and $\tilde{y}_1$, have the same symmetries as the polyphase input components (cf. Figure 5), specifically:

$\tilde{y}_0$ is symmetric about 0 and $(N_0 - 1)/2$, and
$\tilde{y}_1$ is symmetric about $-1/2$ and $(N_0 - 2)/2$.

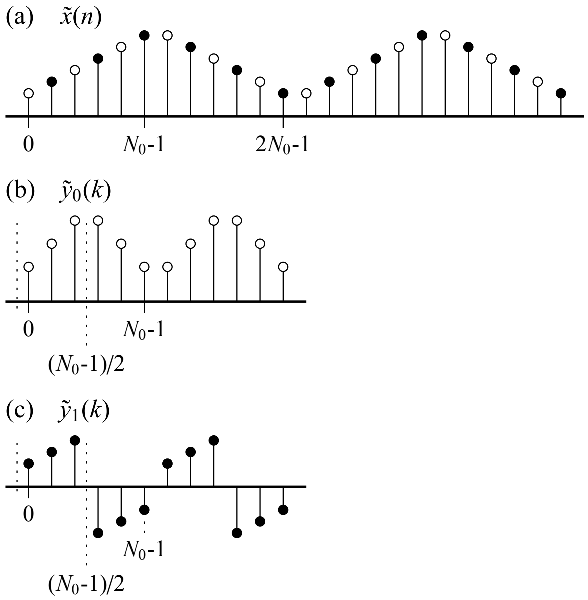

Fig. 6. Subband symmetries generated by an HS filter bank from an even-length input. (a) Periodic HS extension, $\tilde{x} = E_s^{(2,2)}x$. (b) Lowpass subband, $\tilde{y}_0(k)$. (c) Highpass subband, $\tilde{y}_1(k)$.

2) Half-sample symmetric filter banks: An HS analysis bank satisfying the concentric delay-minimized convention [18, Section III-C.1] has h_0 symmetric about $-1/2$ and h_1 antisymmetric about $-1/2$. If $\tilde{x}(n) = E_s^{(2,2)}x$, the subband symmetries follow from [18, Theorem 7]:

$\tilde{y}_0(n)$ is symmetric about $-1/2$ and $(N_0 - 1)/2$, and
$\tilde{y}_1(n)$ is antisymmetric about $-1/2$ and $(N_0 - 1)/2$.

B. Lifting Factorization of Linear Phase Filter Banks

1) Whole-sample symmetric filter banks: The lifting factorization of WS filter banks is described by [18, Theorem 9].

Theorem 1: A delay-minimized WS filter bank of determinant 1 factors completely into lifting matrices (2) whose lifting filters, $S_k(z)$, are *half*-sample symmetric. It follows that the lifting matrices, $\mathbf{S}_k(z)$, are themselves WS polyphase matrices.

2) Half-sample symmetric filter banks: Factoring HS filter banks into linear phase lifting steps is more complicated. The linear phase filters that lift an HS filter bank to a higher-order HS filter bank are whole-sample *anti*symmetric (WA, or type 3) filters. Unfortunately, HS filter banks never factor completely into WA lifting steps [18, Theorem 13]; cf. (1). 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*, which cannot be factored using WA lifting steps [18, Theorem 14].

Theorem 2: Every concentric delay-minimized HS filter bank can be lifted using WA lifting steps from an equal-length HS base filter bank, $\mathbf{B}(z)$:

$$\mathbf{H}_a(z) = \mathbf{S}_{N_{LS}-1}(z) \cdots \mathbf{B}(z) \ . \qquad (3)$$

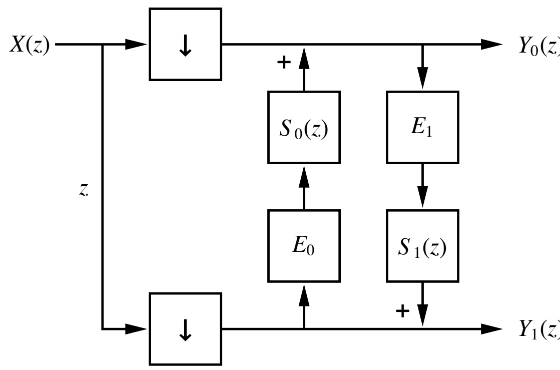

Fig. 8. Lifted filter bank with lifting step extensions.

III. Symmetric Extension for WS Filter Banks

The *extended filter bank* in Figure 7 is a boundary-handling scheme that preserves perfect reconstruction. In *symmetric pre-extension* [9], [10], E extends the input vector, x, into a periodic-symmetric signal, $\tilde{x}$, as in Section I-D. For a WS filter bank with $\tilde{x} \equiv E_s^{(1,1)}x$, the subband symmetries described in Section II-A.1 are exploited by projections P_0 and P_1 that retain only half of a symmetric period in each subband. When Y_0 and Y_1 have a total of just N_0 samples between them, as they do in the WS case for both even and odd N_0, the extended filter bank is called *nonexpansive*. In the synthesis bank, the full periodic-symmetric subbands are restored by symmetric extension operators E_0 and E_1. The synthesis filters then reconstruct the extended input, $\tilde{x}$. The projection, P, gives the output vector $\hat{x}$ the same length as the input.

A. Lifting Step Extensions

One problem with symmetric pre-extension for cascade implementations is that the resulting transforms are "internally" expansive because they must carry boundary extension data forward from one cascade step to the next. We can eliminate this internal expansiveness if $\mathbf{H}_a(z)$ is WS. Consider a lifting factorization of $\mathbf{H}_a(z)$ into HS lifting steps. Each lifting matrix, $\mathbf{S}_k(z)$, is a WS filter bank in its own right, so all intermediate subbands (the outputs of the $\mathbf{S}_k(z)$) are symmetric. Thus, we can construct an "internally" nonexpansive implementation that's equivalent to WS pre-extension without carrying boundary data from one lifting step to the next. Instead, we carry just N_0 samples forward for both channels combined and then *create* the symmetrically extrapolated input for the next lifting step by extending the subband being filtered, as shown in Figure 8. We call this strategy *lifting step extension*.

Because symmetric pre-extension is based on an extension applied prior to the filter bank (see Figure 7), symmetric pre-extension transforms are insensitive to how the filter bank is implemented; e.g., via direct-form filtering or lifting. In contrast, *reversible* implementation is tied to one particular lifting factorization. Reversibility does not, however, depend on boundary-handling so long as the same process is used for both analysis and synthesis. For instance, one can apply a reversible lifting factorization of a WS filter bank to a

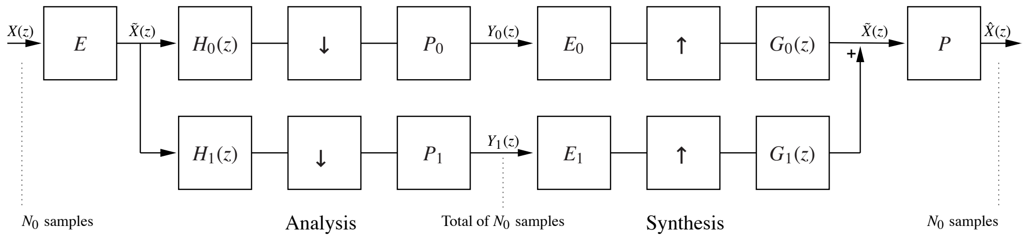

Fig. 7. Filter bank with symmetric pre-extension for nonexpansive transformation of finite input vectors.

symmetrically pre-extended input to obtain a reversible, nonexpansive transform for both even- and odd-length inputs. This is reversible because the finite-length integer subbands, y_0 and y_1, in Figure 7 are obtained by truncating symmetric integer subbands, which can be reconstructed losslessly by symmetric extensions in the synthesis bank. Similar reasoning shows that lifting step extension provides lossless reconstruction with reversible WS filter banks and that the resulting transform is equivalent to reversible symmetric pre-extension.

IV. Symmetric Extension for HS Filter Banks

As shown in [9], symmetric pre-extension via Figure 7 also provides nonexpansive perfect reconstruction for all irreversible HS filter banks if the input extension is $\tilde{x} = E_s^{(2,2)}x$. Problems arise, however, if we attempt to construct lifting step extension schemes for HS filter banks that are equivalent to symmetric pre-extension. As seen in Figure 6(a), the polyphase components of $E_s^{(2,2)}x$ are not symmetric but instead form *mirror images* of one another. Thus, unlike the WS case, there is no way to extend x_1 in the first lifting step of Figure 8 to obtain the same extended polyphase component, $\tilde{x}_1$, as that defined by symmetric pre-extension. Instead, a lifting implementation of an HS filter bank must be initialized with symmetric pre-extension in the base HS filter bank, $\mathbf{B}(z)$, to obtain symmetric/antisymmetric intermediate subbands. Subsequent WA lifting steps, $\mathbf{S}_k(z)$, will preserve these symmetries so, after $\mathbf{B}(z)$, we can stop carrying forward boundary extensions for intermediate subbands and instead proceed via lifting step extension. This scheme, which was described in [12], is particularly simple for the Haar base filter bank, $\mathbf{B}_{\text{haar}}(z)$, since the lifting steps (1) are zeroth-order.

Things are more complicated for *reversible* HS filter banks. Consider a symmetrically pre-extended reversible HS base filter bank that generates a symmetric even channel, $\tilde{x}_0$, and an antisymmetric odd channel, $\tilde{x}_1$. For instance, the Haar filter bank with lifting factorization (1) generates symmetric/antisymmetric subbands using floor function rounding. Let $\mathbf{S}(z)$ be a subsequent highpass WA lifting update. Since $s * \tilde{x}_0$ is antisymmetric, the updated odd channel, $\tilde{x}_1 + R(s * \tilde{x}_0)$, will be antisymmetric if R preserves the antisymmetry of $s * \tilde{x}_0$. This forces R to be an odd function: $R(-x) = -R(x)$. This constraint is satisfied by, e.g., fractional-part (integer) truncation. In contrast, if $S(z)$ is a WA filter in a *lowpass* lifting update then $s*\tilde{x}_1$ will be symmetric. Since *any* rounding rule will preserve symmetry, there are no constraints on the rounding rule for reversible lowpass updates.

A. Rounding in the Equal-Length HS Base Filter Bank

In some cases, symmetric pre-extension may fail to produce symmetric/antisymmetric subbands due to rounding in the equal-length HS base. Consider the Haar analysis bank, $\mathbf{B}_{\text{haar}}(z)$, and its inverse, $\mathbf{G}_s(z) = \mathbf{B}_{\text{haar}}^{-1}(z)$. The matrix $\mathbf{B}'_{\text{haar}} \equiv \mathbf{G}_s^T$ will be referred to as the *dual Haar* filter bank. Let the dual Haar have the lifting factorization

$$\mathbf{B}'_{\text{haar}}(z) = \begin{bmatrix} 1 & 1 \\ -1/2 & 1/2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -1/2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}. \qquad (4)$$

In contrast to $\mathbf{B}_{\text{haar}}$, which generates symmetric/antisymmetric subbands from $\tilde{x} = E_s^{(2,2)}x$ using floor rounding, it is shown in [16] that there is *no* rounding function for which the lifting factorization (4) produces antisymmetric highpass subbands.

While broken antisymmetry doesn't break reversibility of $\mathbf{B}'_{\text{haar}}$, it is problematic if $\mathbf{B}'_{\text{haar}}$ is used as a base filter bank. In [16] we analyze the 6-tap/2-tap wavelet filter bank shown in Figure 9: symmetry is broken in both channels and symmetric extension in the synthesis filter bank incorrectly regenerates the truncated subbands, leading to incorrect signal synthesis.

V. Symmetry-Preserving Lattice Vector Quanitzation

Since scalar rounding can't generate the subband symmetries for symmetric pre-extension in the dual Haar, we consider alternative rounding strategies based on lattice vector quantization. Consider the case $N_0 = 4$. Extend an integer-valued input vector $x = [a, b, c, d]$ to a symmetric-periodic signal with period $\tilde{x} = [a, b, c, d, d, c, b, a]$. The polyphase components have periods $\tilde{x}_0 = [a, c, d, b]$ and $\tilde{x}_1 = [b, d, c, a]$. The result, $\tilde{x}_0^{(1)}$, of the first lifting step in Figure 10 is symmetric and integer-valued without rounding since $S_0(z) = 1$. Define a simple lattice vector quantizer $Q_1 : \mathbb{R}^{N_0} \to \mathbb{Z}^{N_0}$ for rounding the output of $S_1(z) = -1/2$:

$$Q_1([w, x, y, z]) \equiv [\lfloor w \rfloor, \lfloor x \rfloor, \lceil y \rceil, \lceil z \rceil] .$$

Apply Q_1 to the output of S_1 and update channel 1:

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

Now add the first and last elements in $\tilde{x}_1^{(2)}$:

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

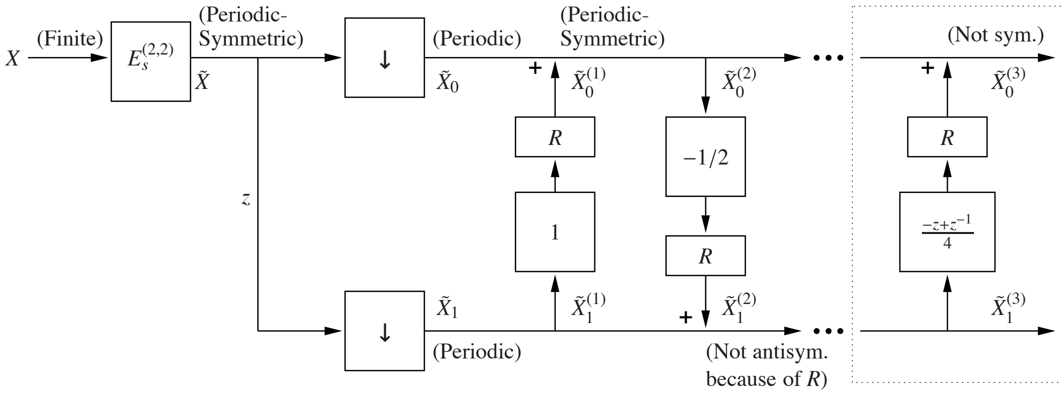

Fig. 9. The dual Haar base filter bank with a third lifting step that lifts it to the 6-tap/2-tap HS filter bank.

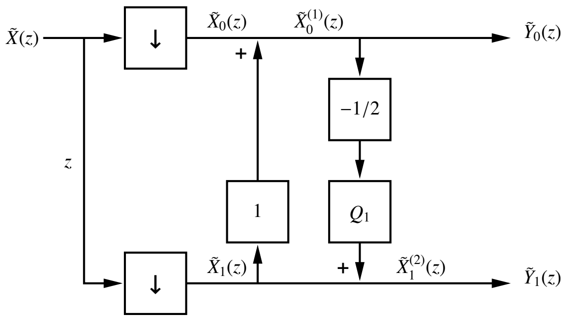

Fig. 10. The dual Haar filter bank rounded with a lattice vector quantizer.

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.

This allows reversible HS filter banks lifted from the dual Haar to be implemented using symmetric pre-extension or an equivalent lifting step extension scheme. A considerably more general theoretical treatment of symmetric extension for reversible HS filter banks using lattice vector quantization rounding functions is presented in the forthcoming paper [16].

Acknowledgments

Los Alamos National Laboratory is operated for the U.S. Department of Energy by Los Alamos National Security LLC under Contract No. DE-AC52-06NA25396. This work was supported in part by the DOE Accelerated Strategic Computing Initiative, the DOE Applied Mathematical Sciences Program, contract KC-07-01-01, and the DOE Laboratory-Directed Research and Development Program.

References

[1] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ, USA: Prentice Hall, 1992.
[2] G. Strang and T. Nguyen, *Wavelets and Filter Banks*. Wellesley, MA, USA: Wellesley-Cambridge Press, 1996.
[3] W. Sweldens, “The lifting scheme: a custom-design construction of biorthogonal wavelets,” *Applied and Computational Harmonic Analysis*, vol. 3, no. 2, pp. 186–200, 1996.
[4] I. Daubechies and W. Sweldens, “Factoring wavelet transforms into lifting steps,” *Journal Of Fourier Analysis And Applications*, vol. 4, no. 3, pp. 247–269, 1998.
[5] A. Zandi, J. D. Allen, E. L. Schwartz, and M. Boliek, “Crew: Compression with reversible embedded wavelets.” *Proceedings of the Data Compression Conference*, pp. 212 – 221, Mar. 1995.
[6] A. R. Calderbank, I. Daubechies, W. Sweldens, and B. L. Yeo, “Wavelet transforms that map integers to integers,” *Applied and Computational Harmonic Analysis*, vol. 5, no. 3, pp. 332–369, July 1998.
[7] *JPEG 2000 Image Coding System, Part 1*, ser. ISO/IEC Int’l. Standard 15444-1, ITU-T Rec. T.800. Int’l. Org. for Standardization, Dec. 2000.
[8] *JPEG 2000 Image Coding System, Part 2 (Extensions)*, ser. ISO Int’l. Standard 15444-2, ITU-T Rec. T.801. Int’l. Org. for Standardization, Dec. 2001.
[9] C. M. Brislawn, “Classification of nonexpansive symmetric extension transforms for multirate filter banks,” *Applied and Computational Harmonic Analysis*, vol. 3, pp. 337–357, 1996.
[10] ——, “Symmetric Extension Transforms,” in *Wavelet Image and Video Compression*, P. N. Topiwala, Ed. Boston, MA: Kluwer, 1998, ch. 5, pp. 83–91.
[11] M. D. Adams and F. Kossentini, “JasPer: A software-based JPEG-2000 codec implementation,” in *Proc. Int’l. Conf. Image Process.* Vancouver, BC: IEEE Signal Process. Soc., Sept. 2000, pp. 53–56.
[12] C. M. Brislawn and B. E. Wohlberg, “Boundary extensions and reversible implementation for half-sample symmetric filter banks,” ISO/IEC Standards Committee JTC1/SC29/WG1, Singapore, Tech. Rep. WG1N2119, Mar. 2001.
[13] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards, and Practice*. Boston, MA, USA: Kluwer, 2002.
[14] M. D. Adams, “Reversible integer-to-integer wavelet transforms for image coding,” Ph.D. dissertation, Dept. of Electrical and Computer Engineering, Univ. of British Columbia, Vancouver, BC, Canada, Sept. 2002.
[15] M. D. Adams and R. K. Ward, “Symmetric-extension-compatible reversible integer-to-integer wavelet transforms,” *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2624–2636, 2003.
[16] B. E. Wohlberg and C. M. Brislawn, “Symmetric extension for lifted filter banks and obstructions to reversible implementation,” Los Alamos Nat’l. Lab, Tech. Rep. LAUR-05-906, July 2006, submitted for publication.
[17] C. M. Brislawn, S. M. Mniszewski, M. D. Pal, A. G. Percus, B. E. Wohlberg, T. Acharya, P.-S. Tsai, and M. Lepley, “Report on core experiment CodEff03: Even-length filter bank option,” ISO/IEC Standards Committee JTC1/SC29/WG1, Stockholm, Sweden, Tech. Rep. WG1N2209, July 2001.
[18] C. M. Brislawn and B. E. Wohlberg, “The polyphase-with-advance representation and linear phase lifting factorizations,” *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2022–2034, June 2006.