M-channel Linear Phase Perfect Reconstruction Filter Bank with Integer Coefficients

Trac D. Tran

Department of Electrical and Computer Engineering
The Johns Hopkins University
Baltimore MD 21218 USA
ttran@ece.jhu.edu

ABSTRACT

This paper introduces a class of M-channel linear phase perfect reconstruction filter banks (LPPRFBs) with integer coefficients. The construction of these filter banks is based on a VLSI-friendly lattice structure which employs the minimum number of delay elements and robustly enforces both linear phase and perfect reconstruction properties. The lattice coefficients are parameterized as a series of lifting steps, providing fast, efficient, in-place computation of the subband coefficients. Image coding experiments show that these novel FBs are very competitive with current state-of-the-art transforms such as the 8×8 DCT and the wavelet transform with 9/7-tap biorthogonal floating-point wavelet.

Keywords: filter bank, transform, integer coefficients, multiplierless, lattice structure, lifting steps.

1. INTRODUCTION

There has been a tremendous growth in the field of filter banks (FBs) and multirate systems in the last fifteen years. These systems provide new and effective ways to represent signals for processing, understanding, and compression purposes. Filter banks find applications in virtually every signal processing field.[1–3] Obviously, of extreme importance is the ability to design a filter bank that can fully exploit (i) the statistical properties of a particular class of signals; (ii) the goals of the applications; and (iii) the computational resources available at the processing/communication system. One particular class of FBs that attracts a lot of recent interests is FBs with integer coefficients. First of all, integer-coefficient FBs can map integers to integers losslessly – a necessary condition for transform-based lossless coding. More importantly, floating-point implementations in hardware are slow, require too much space, and consume too much power. Many integer-coefficient FBs can also have multiplierless implementation with simple shift-and-add operations. This is critical in many wireless communication applications and mobile devices.

In this paper, we shall limit the discussions on discrete-time maximally decimated M-channel uniform filter banks as depicted in Fig. 1. At the analysis stage, the input signal $x[n]$ is passed through a bank of M analysis filters $H_i(z)$, each of which preserves a frequency band of uniform bandwidth $\frac{\pi}{M}$. These M filtered signals are then decimated by M to preserve the system's overall sampling rate. The resulting subband signals can be coded, processed, and/or transmitted independently or jointly. At the synthesis stage, the subbands are combined by a set of upsamplers and M synthesis filters $F_i(z)$ to form the reconstructed signal $\hat{x}[n]$. Assume that there is no information loss at the processing stage, filter banks that yield the output $\hat{x}[n]$ as a pure delayed version of the input $x[n]$, i.e. $\hat{x}[n] = x[n - n_0]$, are called perfect reconstruction (PR) filter banks. The PR property is highly desirable since it provides a lossless signal representation and it simplifies the error analysis significantly.

The M-channel filter bank in Fig. 1(a) can also be represented in terms of its polyphase matrices as shown in Fig. 1(b), where $\mathbf{E}(z)$ is the analysis bank's polyphase matrix and $\mathbf{R}(z)$ is the synthesis bank's polyphase matrix. Note that both $\mathbf{E}(z)$ and $\mathbf{R}(z)$ are $M \times M$ matrices whose elements are polynomials in z.[1] Now, if $\mathbf{E}(z)$ is invertible with minimum-phase determinant (stable inverse), one can obtain PR by simply choosing $\mathbf{R}(z) = \mathbf{E}^{-1}(z)$. In other words, any choice of $\mathbf{R}(z)$ and $\mathbf{E}(z)$ that satisfies

$$\mathbf{R}(z)\,\mathbf{E}(z) \;=\; \mathbf{E}(z)\,\mathbf{R}(z) \;=\; z^{-l}\,\mathbf{I}, \qquad l \geq 0 \tag{1}$$

yields perfect reconstruction, also commonly referred to as *biorthogonality* in the filter bank and wavelet community. One important PR subset is the *paraunitary* case where $\mathbf{E}(z)$ is constrained such that $\mathbf{R}(z) \;=\; z^{-K}\mathbf{E}^T(z^{-1})$.

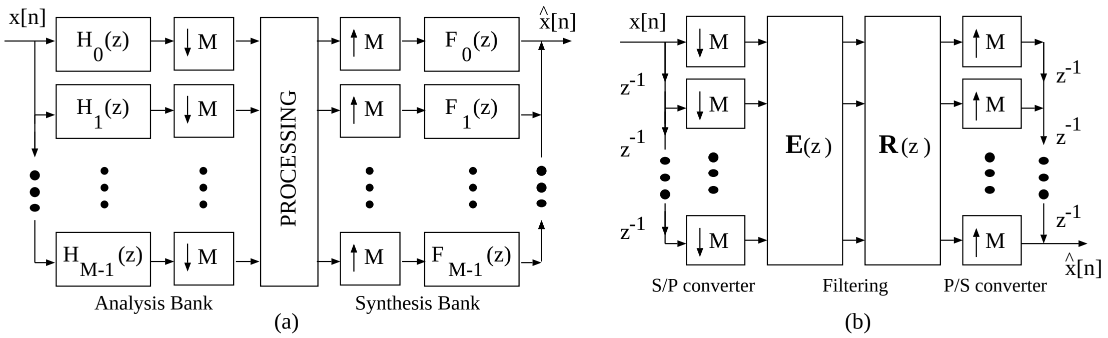

Figure 1. M-channel uniform-band maximally-decimated filter bank. (a) Conventional representation. (b) Polyphase representation.

Another property that attracts a lot of interests is FIR, or compact support, since FIR filters are usually cost-efficient and easy to implement. In fact, in image and video coding applications, the filters (especially the highpass one) should have relatively short length in order to limit ringing artifacts.[2,3] To obtain FIR filters in both banks, we need the determinants of both polyphase matrices to be monomials[1]:

$$|\mathbf{R}(z)| \;=\; z^{-m} \;\text{ and }\; |\mathbf{E}(z)| \;=\; z^{-n} \qquad m, n \text{ integers.} \tag{2}$$

In numerous applications, it is also crucial that all analysis and synthesis filters have linear phase (either symmetric or antisymmetric if the filters have real coefficients). Besides the elimination of the phase distortion, linear phase systems allow us to use simple symmetric extension methods to accurately handle the boundaries of finite-length signals. Furthermore, the linear phase property can be exploited, leading to faster and more efficient FB implementation. For the class of LPFBs whose filters all have length $L = KM, K \in \mathcal{Z}^+$, it has been well established that the polyphase matrices satisfy the following symmetric property[4,5]

$$\mathbf{E}(z) \;=\; z^{-(K-1)}\,\mathbf{D}\,\mathbf{E}(z^{-1})\,\mathbf{J} \quad \text{(analysis)} \tag{3}$$

$$\mathbf{R}(z) \;=\; z^{-(K-1)}\,\mathbf{J}\,\mathbf{R}(z^{-1})\,\mathbf{D} \quad \text{(synthesis)} \tag{4}$$

where $\mathbf{D}$ is the diagonal matrix with entries being $+1$ or -1 depending on the corresponding filter being symmetric or antisymmetric and $\mathbf{J}$ is the reversal matrix.

Now, the design of an FIR LPPRFB can be reduced to choosing appropriate polynomial matrices $\mathbf{E}(z)$ and $\mathbf{R}(z)$ such that Eqs. (1)-(4) are satisfied. This paper introduces a family of FBs whose polyphase matrices not only satisfy the symmetric, FIR, and invertible properties, but can also be decomposed into a series of dyadic lifting steps, leading to fast-computable, VLSI-friendly transforms for low-cost signal processing/coding applications.

2. GENERAL LATTICE STRUCTURE

2.1. Polyphase Matrix Factorization

The lattice structure is one of the most effective and elegant tools in M-channel FB design and implementation. This approach bases on various factorizations of the polyphase matrices $\mathbf{E}(z)$ and $\mathbf{R}(z)$.[4–7] From another viewpoint, in the lattice approach, the polyphase matrices $\mathbf{E}(z)$ and $\mathbf{R}(z)$ are constructed from a cascade of modular low-ordered structures that propagate certain desirable properties (such as LP and PR). The lattice structure offers fast implementation with a minimal number of delay elements, and it can retain desirable properties regardless of lattice coefficient quantization.

The most general lattice for M-channel FIR linear phase perfect reconstruction filter bank satisfying Eqs. (1)-(4) is presented in.[7,8] The polyphase matrix $\mathbf{E}(z)$ can be factorized as

$$\mathbf{E}(z) \;=\; \mathbf{G}_{K-1}(z)\,\mathbf{G}_{K-2}(z)\,\cdots\,\mathbf{G}_1(z)\;\mathbf{E}_0, \tag{5}$$

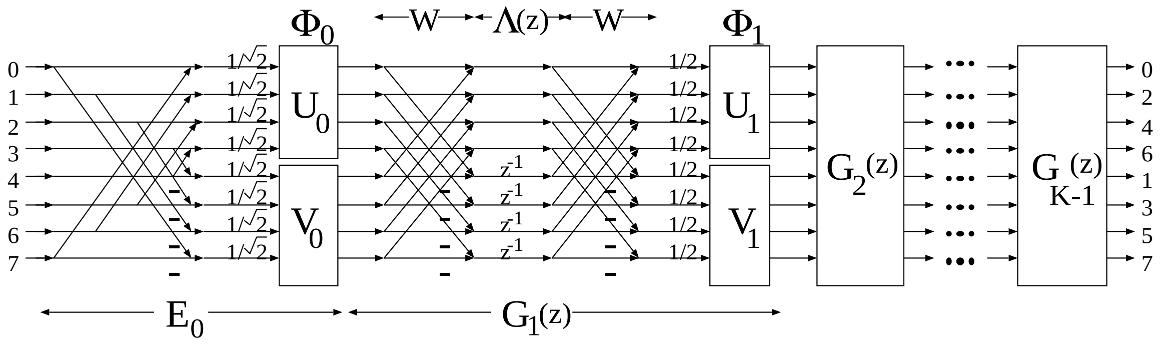

Figure 2. General lattice structure for M-channel LPPRFB with filter length $L = KM$ (drawn for $M = 8$).

where

$$\mathbf{G}_i(z) = \frac{1}{2}\begin{bmatrix} \mathbf{U}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_i \end{bmatrix}\begin{bmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{I} & -\mathbf{I} \end{bmatrix}\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & z^{-1}\mathbf{I} \end{bmatrix}\begin{bmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{I} & -\mathbf{I} \end{bmatrix} \triangleq \frac{1}{2}\,\mathbf{\Phi}_i\,\mathbf{W}\,\mathbf{\Lambda}(z)\,\mathbf{W}, \tag{6}$$

and

$$\mathbf{E}_0 = \frac{1}{\sqrt{2}}\begin{bmatrix} \mathbf{U}_0 & \mathbf{U}_0\,\mathbf{J}_{\frac{M}{2}} \\ \mathbf{V}_0\,\mathbf{J}_{\frac{M}{2}} & -\mathbf{V}_0 \end{bmatrix}. \tag{7}$$

This lattice results in all filters having length $L = KM$. K is often called the overlapping factor because of the FB connection to lapped transforms.[9,6] Each cascading structure $\mathbf{G}_i(z)$ increases the filter length by M. All $\mathbf{U}_i$ and $\mathbf{V}_i$, $i = 0, 1, \ldots, K-1$, are arbitrary $\frac{M}{2} \times \frac{M}{2}$ invertible matrices. The complete lattice of the analysis bank is depicted in Figure 2. The synthesis bank can be obtained by reversing the signal flow.

The *completeness* and *minimality* of the aforementioned structure are established in the following two theorems. For their proofs, the reader is referred to.[7]

Theorem I. The analysis polyphase matrix $\mathbf{E}(z)$ of any even-channel FIR LPPRFB with all analysis and synthesis filters of length $L = KM$ can always be factored as $\mathbf{E}(z) = \prod_{i=K-1}^{1} \mathbf{G}_i(z)\,\mathbf{E}_0$ where $\mathbf{G}_i(z)$ is as in Eq. (6) and $\mathbf{E}_0$ is as in Eq. (7). The corresponding synthesis polyphase matrix is $\mathbf{R}(z) = \mathbf{E}_0^{-1} \prod_{i=1}^{K-1} z^{-1}\,\mathbf{G}_i^{-1}(z)$.

Theorem II. The factorization in Eq. (5) is minimal, i.e. the resulting lattice structure employs the fewest number of delays in its implementation.

Theorem I and II present a powerful result in FB design. The set of $\frac{M}{2} \times \frac{M}{2}$ invertible matrices completely spans the space of all possible solutions. Therefore, the design problem is broken down to making the optimal choice of the invertible matrices $\mathbf{U}_i$ and $\mathbf{V}_i$, which contain all of the FB's degrees of freedom.

2.2. Parameterization of Invertible Matrices

The purpose of matrix parameterization is to completely capture the set of all matrices with a certain desirable property using the fewest number of independent parameters. From a design perspective, the lattice structure is a powerful FB design tool since the lattice coefficients can be varied independently and arbitrarily without affecting the most desirable FB characteristics. Unconstrained optimization can be applied to obtain secondary features such as high coding gain and low stopband attenuation.

2.2.1. Parameterization of Orthogonal Matrices via Rotation Angles

The reader can easily verify that imposing orthogonality on the free matrices $\mathbf{U}_i, \mathbf{V}_i$ in Eq. (5) results in paraunitary FBs. Every $N \times N$ orthogonal matrix can be completely characterized by $\frac{N(N-1)}{2}$ Givens rotations[10] as shown in Fig. 3(a) (drawn for the 4×4 case). The incorporation of the rotation angles into the lattice in Fig. 2 leads to the Generalized Lapped Orthogonal Transform (GenLOT).[6]

(a)

(b)

(c)

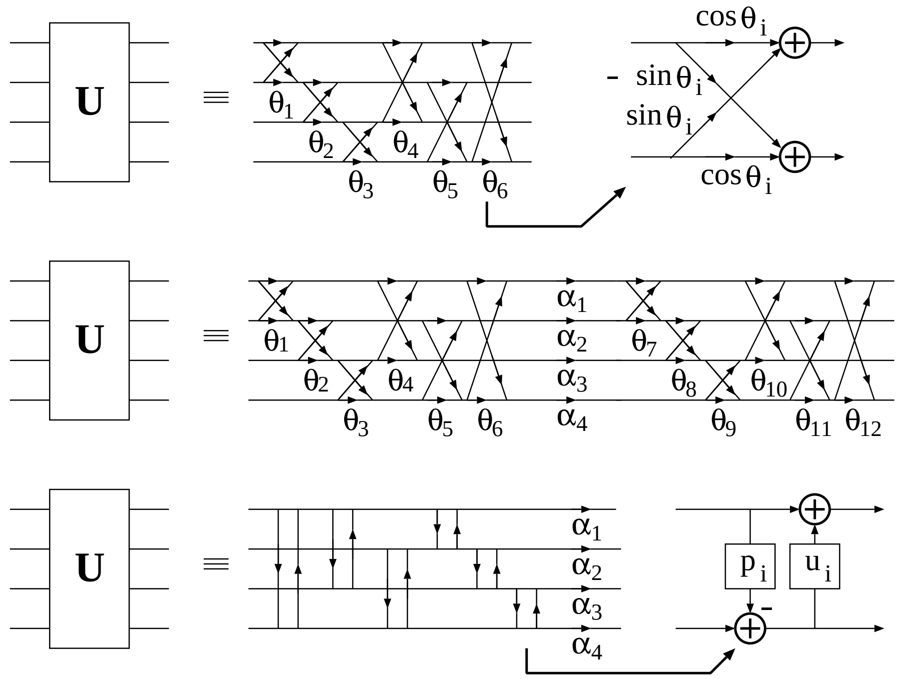

Figure 3. Parameterization of matrices. (a) Orthogonal matrix as product of rotation angles θ_i. (b) Invertible matrix as product of rotation angles θ_i and diagonal scaling factors α_i. (c) Invertible matrix as product of lifting steps p_i, u_i, and diagonal scaling factors α_i.

2.2.2. Parameterization of Invertible Matrices via the SVD

The difficulty in the more general PR case is obvious: how do we completely characterize a nonsingular square matrix $\mathbf{U}$ of size N? One naive solution is to choose $\mathbf{U}$'s elements as the lattice coefficients. However, there are many problems with this solution. First of all, it is difficult to guarantee exact reconstruction when the matrix elements are quantized. Secondly, this "parameterization" method does not provide a fast and efficient FB implementation. Furthermore, in order to obtain a high-performance FB, we have to synthesize the lattice by an optimization process to find the set of locally optimal lattice coefficients. This process typically involves thousands of iterative steps, so we have to face the costly matrix inversion problem. Finally, how can we prevent the optimization process from encountering singular or near-singular matrices?

One reasonable parameterization method of invertible matrices is proposed in[8,7] where the authors use the singular value decomposition (SVD) as illustrated in Fig. 3(b). Recall that every invertible matrix has an SVD representation: $\mathbf{U} = \mathbf{U}_0 \mathbf{\Gamma} \mathbf{U}_1$, where $\mathbf{U}_0$ and $\mathbf{U}_1$ are orthogonal matrices, and $\mathbf{\Gamma}$ is a diagonal matrix with positive elements.[10] Thus, every invertible matrix $\mathbf{U}$ of size N can be completely characterized by $N(N-1)$ rotation angles θ_i (from $\mathbf{U}_0$, $\mathbf{U}_1$) and $\frac{N}{2}$ diagonal multipliers α_i (from $\mathbf{\Gamma}$). Invertibility is guaranteed structurally under a mild condition – as long as none of the diagonal lattice coefficients α_i representing $\mathbf{\Gamma}$ is quantized to zero. Moreover, inverting $\mathbf{U}$ is now very fast, and singularity can be prevented by a simple cost function in the optimization process where a penalty is assigned whenever a diagonal coefficient (or its inverse) ventures too close to zero.

2.2.3. Parameterization of Invertible Matrices via Lifting Steps

However, one disadvantage of the aforementioned SVD-based parameterization is the possible redundancy in computational complexity. Since each rotation angle takes 4 multiplications and 2 additions (some manipulations yield 3 multiplications and 3 additions), the actual cost of performing a matrix multiplication in the SVD structure actually surpasses that of direct multiplication. In this paper, we propose to characterize these invertible matrices using the LU decomposition.

Proposition I. Every $N \times N$ invertible matrix can be completely characterized by $N(N-1)$ shears, N diagonal scaling factors, and possibly a permutation matrix.

It is not too difficult to see how one can systematically factor any given invertible matrix using permutation, diagonal scaling, and shearing. First, the invertible matrix of interest is permuted so that we have all non-zero elements on the diagonal. Next, the Gauss-Jordan elimination process is applied to each non-diagonal element, turning it to zero. This is a shear operation, also called a lifting step or a ladder structure in the context of this paper. Lastly, all of the diagonal elements can then be scaled to unity. The final ladder-based parameterization is illustrated in Fig. 3(c) (drawn for $N = 4$). Under this parameterization, the computational complexity is, at most, equal to that of direct multiplication. In most cases, the computational complexity can be reduced significantly by setting the diagonal scaling factors to unity and/or choosing the lifting coefficients to be dyadic as described in the next section.

The reader should note that this factorization, although complete, is not unique. Furthermore, if we restrict the determinants of the matrices to ± 1, the N scaling factors α_i can be converted into lifting steps as well.[11] However, these scaling factors are typically folded into the quantization stepsizes of the encoder. That is why we choose to place them at the end of the factorization.

2.2.4. Dyadic Lifting Steps and the Construction of Integer-Coefficient FBs

The lifting step and its versatility in constructing fast transforms that can map integers to integers are demonstrated in Fig. 4. First of all, inverting a lifting step is simple: just inverting its polarity when the diagonal elements of the matrix are unity with the same polarity (both +1 or -1). In the case that the diagonal elements have opposite polarity, the inverse of the lifting step is simply itself.

Second, if a *floor* (or *round*, or *ceiling*) operator is placed in each lifting step, the FB can now map integers to integers with perfect reconstruction. Third, if all lifting steps and diagonal scaling factors are constrained to be rational, the resulting FB now has integer coefficients (up to finite scaling factor). Finally, if the lifting step is chosen to be dyadic (i.e., a rational that can be written in the form of $\frac{k}{2^m}$; $k, m \in \mathcal{Z}$), the nonlinear operation can be incorporated into the division using binary bit shift. Division by $\frac{1}{2^m}$ followed by a truncation is equivalent to a right binary shift by m places. The numerator k can be easily implemented using bit shift and add operations as well. With all scaling factors set to unity, multiplierless FBs can be easily constructed using this method. The ± 1 butterflies in Eq. (5) can be retained since their outputs are integers.

3. DESIGN

3.1. FB Optimization

The FB structure presented in the previous already have high practical value: FIR, linear phase, and perfect reconstruction. However, for the FBs to achieve high coding performance, several other properties are also needed. Practical FBs can be obtained using unconstrained nonlinear optimization where the FB coding gain is maximized and the DC leakage is minimized. In the particular field of image compression, these two criteria are well-known desired properties in yielding the best reconstructed image quality.[3] The cost function in the optimization process is a weighted linear combination of these measures.

3.1.1. Coding Gain

The coding gain of a FB is defined as the reduction in transform coding mean-square error over pulse-code modulation (PCM) which simply quantizes the samples of the signal with the desired number of bits per sample. Define σ_x^2 as the variance of the input signal $x[n]$, $\sigma_{x_i}^2$ as the variance of the i-th subband, and $||f_i||^2$ as the $\mathcal{L}^2$ norm of the i-th synthesis filter. With several assumptions including scalar quantization and a sufficient large bit rate, the generalized coding gain can be formulated as[12–14]:

$$C_{\text{coding gain}} = 10 \log_{10} \frac{\sigma_x^2}{\left(\prod_{i=0}^{M-1} \sigma_{x_i}^2 ||f_i||^2 \right)^{\frac{1}{M}}}. \tag{8}$$

The signal $x[n]$ is the commonly-used AR(1) process with intersample autocorrelation coefficient $\rho = 0.95$. The coding gain can be thought of as an approximate measure of the FB's energy compaction capability. Among the

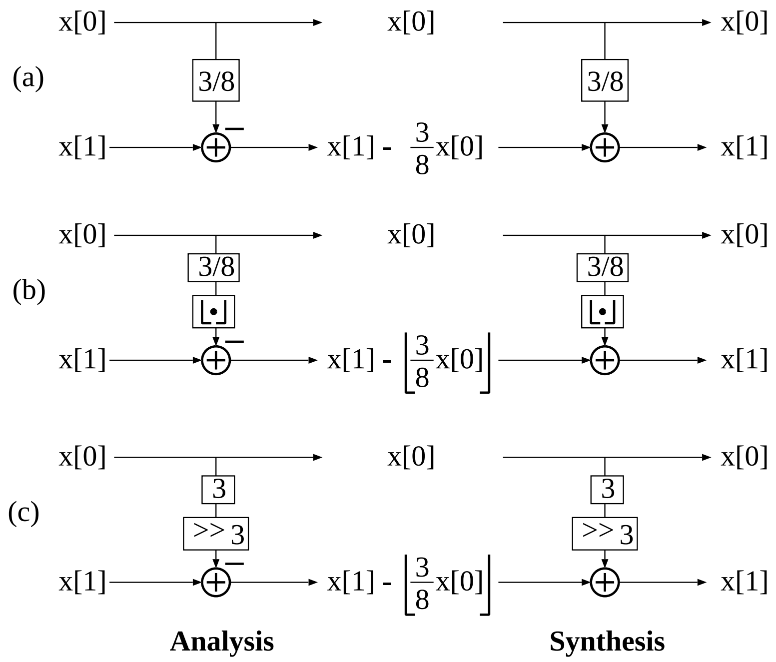

Figure 4. Example of a lifting step and its implementations. (a) Original lifting step. (b) An approximation that can map integers to integers. (c) The equivalent approximation using only shift-and-add operations.

listed criteria, higher coding gain correlates most consistently with higher objective performance (measured in MSE or PSNR). FBs with higher coding gain compact more signal energy into a fewer number of coefficients, and this leads to more efficient entropy coding.

3.1.2. Low DC Leakage

The DC leakage cost function measures the amount of DC energy that leaks out to the bandpass and highpass subbands. The main idea is to concentrate all signal energy at DC into the DC coefficients. This proves to be advantageous in both signal decorrelation and in the prevention of discontinuities in the reconstructed signals. Low DC leakage can prevent the annoying checkerboard artifact that usually occurs when high frequency bands are severely quantized.[3] The DC cost function can be defined as

$$C_{\mathrm{DC\ analysis}} = \sum_{i=1}^{M-1} \sum_{n=0}^{L-1} h_i[n] \quad \text{and} \quad C_{\mathrm{DC\ synthesis}} = \sum_{i=1}^{M-1} \sum_{n=0}^{L-1} f_i[n]. \tag{9}$$

The readers should note that all antisymmetric filters have a zero at DC (zero frequency). Therefore, the above formula only needs to apply to symmetric filters to reduce the complexity of the optimization process. It is also interesting that the zero leakage condition is equivalent to having one vanishing moment – a necessary condition in the construction of wavelets. In fact, all of the design examples presented in the next sections are M-band wavelets with 1 degree of regularity in both analysis and synthesis bank.

3.2. Design Example 1: 4-channel 8-tap integer-coefficient LPPRFB

In this 4-channel 8-tap design example, we demonstrate the feasibility of designing nontrivial integer-coefficient LPPRFBs using the lifting-based structure presented in previous sections. Floating-point lifting coefficients are first found via the nonlinear unconstrained optimization *simplex* routine in Matlab. Integer-coefficient solution is then obtained by quantized the lifting coefficients to dyadic rationals. The final four matrices used to construct the analysis bank are

$$\mathbf{U}_0 = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{V}_0 = \begin{bmatrix} 2 & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & \frac{3}{8} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}, \tag{10}$$

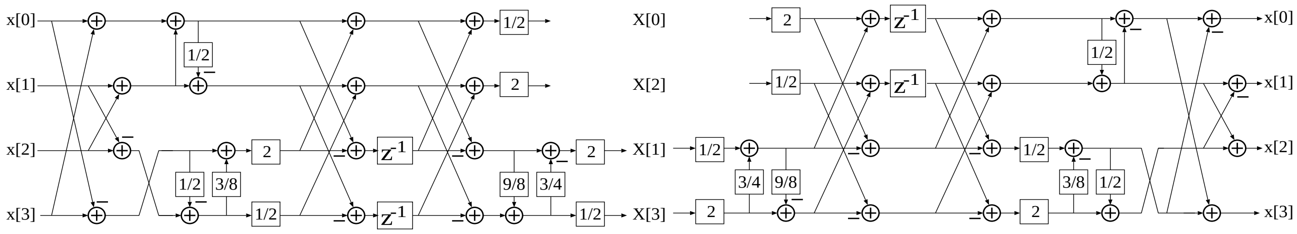

Figure 5. Fast implementation of the 4-channel 8-tap integer FB in Design Example 1. Left: analysis bank. Right: synthesis bank.

Table 1. Design example 1: 4-channel LPPRFB with 8-tap integer-coefficient filters. Top: coefficients of analysis filters h_i. Bottom: coefficients of synthesis filters f_i.

h_0	-5/32	1/16	7/16	21/32	21/32	7/16	1/16	-5/32
h_1	-61/256	9/128	95/128	141/256	-141/256	-95/128	-9/128	61/256
h_2	-1/4	0	1	-3/4	-3/4	1	0	-1/4
h_3	23/256	5/128	-61/128	249/256	-249/256	61/128	-5/128	-23/256
f_0	0	1/8	3/8	1/2	1/2	3/8	1/8	0
f_1	-5/256	23/512	249/512	61/256	-61/256	-249/512	-23/512	5/256
f_2	-1/32	-5/64	21/64	-7/32	-7/32	21/64	-5/64	-1/32
f_3	9/256	61/512	-141/512	95/256	-95/256	141/512	-61/512	-9/256

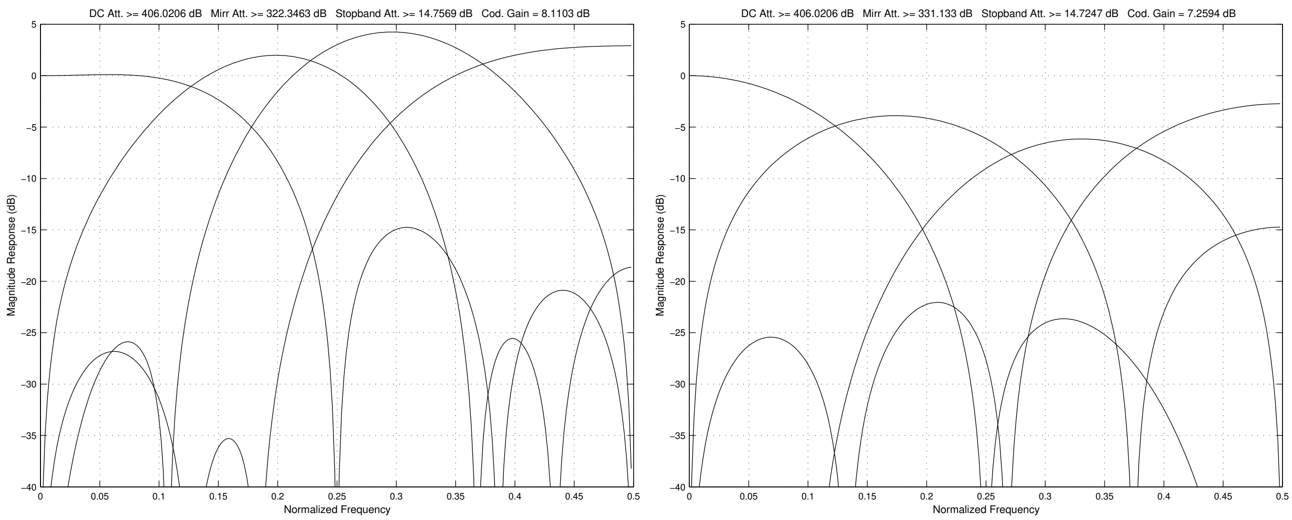

Figure 6. Frequency responses of the 4-channel 8-tap integer LPPRFB in Design Example 1. Left: analysis bank. Right: synthesis bank.

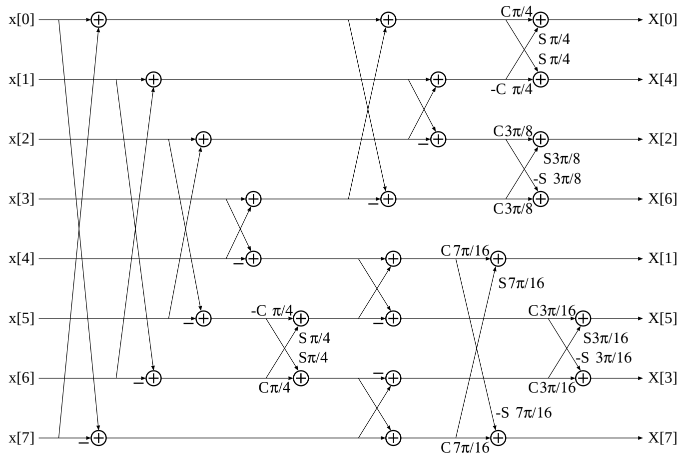

Figure 7. Fast implementation of the 8×8 DCT.

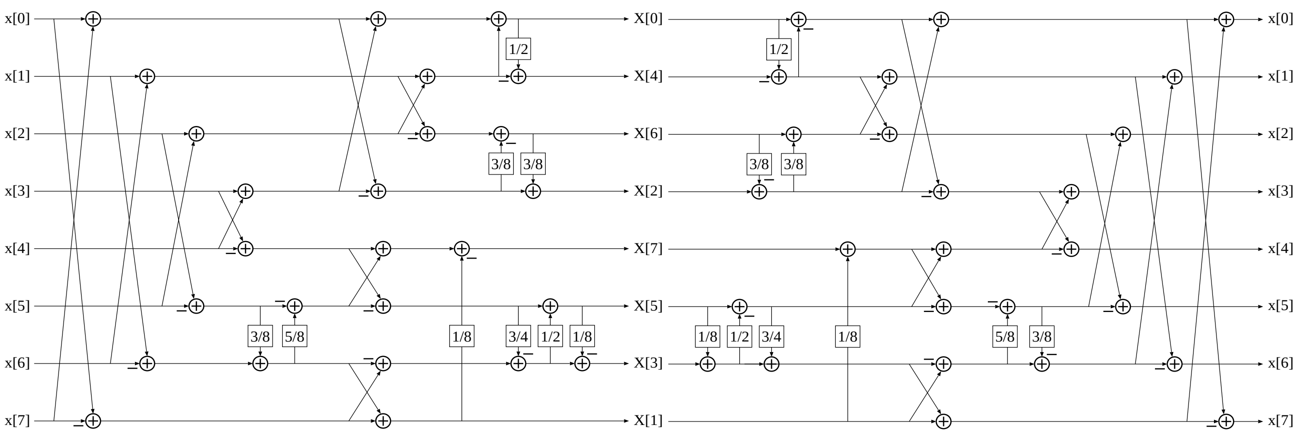

Figure 8. Fast implementation of the 8-channel 8-tap binDCT in Design Example 2. Left: analysis bank. Right: synthesis bank.

and

$$\mathbf{U}_1 = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 2 \end{bmatrix}, \quad \mathbf{V}_1 = \begin{bmatrix} 2 & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & -\frac{3}{4} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{9}{8} & 1 \end{bmatrix}. \tag{11}$$

The corresponding fast implementation is depicted in Fig. 5. All of the filter coefficients are indeed dyadic, and they are tabulated in Table 1. Note that the scaling factors 2 and $\frac{1}{2}$ are present to balance the gain of the subbands only; they should be absorbed into the quantizer or moved to the other bank in practice. The frequency responses of the filters are shown in Fig. 6.

3.3. Design Example 2: 8-channel 8-tap binDCT

This 8-channel 8-tap design example is designed to approximate the 8×8 floating-point DCT.[15] Hence, it is named the binDCT[16] and it follows the DCT's factorization in Fig. 7 very closely. The resulting multiplierless lattice structure is depicted in Fig. 8. In this design example, we took a divide-and-conquer optimization approach. First, the 4×4 DCT is first approximated (this is $\mathbf{U}_0$ in Fig. 2), then the result is used to find $\mathbf{V}_0$.

For the $\mathbf{U}_0$ matrix which yields the symmetric basis functions, the following lifting choices yield a transform with high coding gain:

Table 2. Design example 2: 8-channel LPPRFB with 8-tap integer-coefficient filters. Top: coefficients of analysis filters h_i. Bottom: coefficients of synthesis filters f_i.

h_0	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2
h_1	1/2	1/2	3/16	0	0	-3/16	-1/2	-1/2
h_2	55/128	3/16	-3/16	-55/128	-55/128	-3/16	3/16	55/128
h_3	15/32	-215/1024	-4037/8192	-53/128	53/128	4037/8192	215/1024	-15/32
h_4	1/4	-1/4	-1/4	1/4	1/4	-1/4	-1/4	1/4
h_5	1/4	-57/128	149/1024	5/16	-5/16	-149/1024	57/128	-1/4
h_6	-3/16	1/2	-1/2	3/16	3/16	-1/2	1/2	-3/16
h_7	-1/16	1/4	-13/32	1/2	-1/2	13/32	-1/4	1/16
f_0	1/4	1/4	1/4	1/4	1/4	1/4	1/4	1/4
f_1	1/2	13/32	1/4	1/16	-1/16	-1/4	-13/32	-1/2
f_2	1/2	3/16	-3/16	-1/2	-1/2	-3/16	3/16	1/2
f_3	5/16	-149/1024	-57/128	-1/4	1/4	57/128	149/1024	-5/16
f_4	1/2	-1/2	-1/2	1/2	1/2	-1/2	-1/2	1/2
f_5	53/128	-4037/8192	215/1024	15/32	-15/32	-215/1024	4037/8192	-53/128
f_6	-3/16	55/128	-55/128	3/16	3/16	-55/128	55/128	-3/16
f_7	0	3/16	-1/2	1/2	-1/2	1/2	-3/16	0

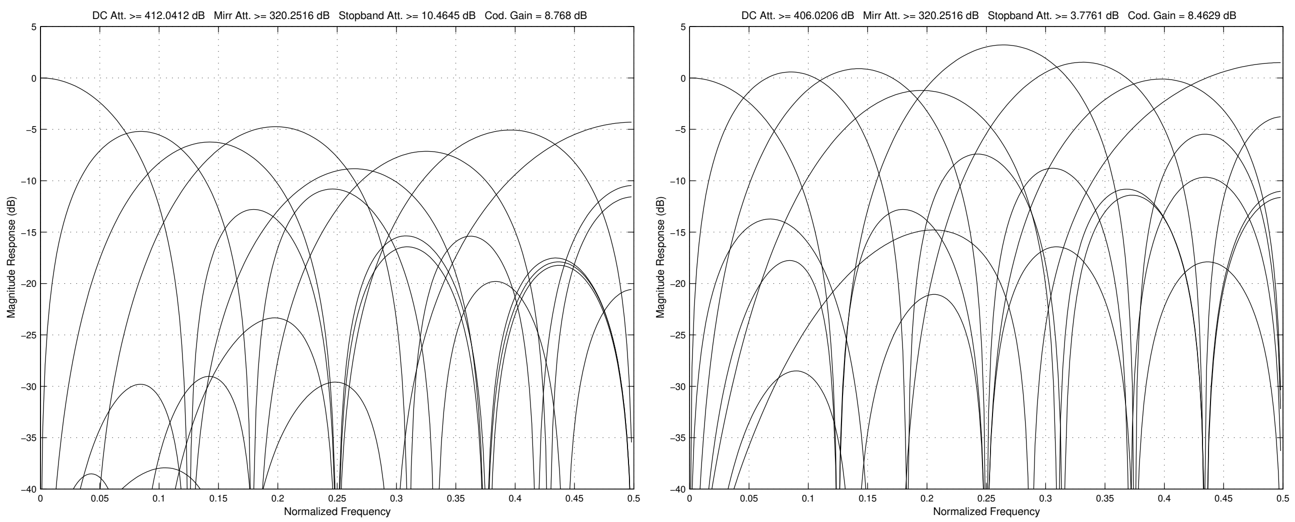

Figure 9. Frequency responses of the 8-channel 8-tap integer LPPRFB in Design Example 2. Left: analysis bank. Right: synthesis bank.

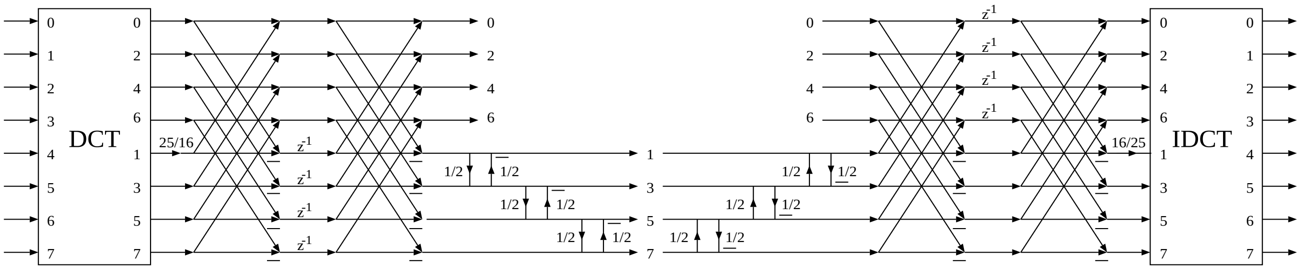

Figure 10. Fast implementation of the 8-channel 16-tap LiftLT. Left: analysis bank. Right: synthesis bank.

1. Rotation $\frac{\pi}{4} \approx \begin{bmatrix} 1 & 0 \\ \frac{1}{2} & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$.

2. Rotation $\frac{3\pi}{8} \approx \begin{bmatrix} 1 & 0 \\ \frac{3}{8} & 1 \end{bmatrix} \begin{bmatrix} 1 & -\frac{3}{8} \\ 0 & 1 \end{bmatrix}$.

The $\mathbf{V}_0$ matrix poses a greater challenge. Specifically, the replacement of the rotation angle $\frac{\pi}{4}$ in $\mathbf{V}_0$ has to be chosen wisely since it precedes, and hence greatly influences, the remaining lifting coefficients in the lattice. For example, the unnormalized Haar transform mentioned above does not yield very good result here. We found that the following choices yields good performance:

1. Rotation $\frac{\pi}{4} \approx \begin{bmatrix} -1 & \frac{5}{8} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{3}{8} & 1 \end{bmatrix}$.

2. Rotation $\frac{7\pi}{16} \approx \begin{bmatrix} 1 & -\frac{1}{8} \\ 0 & 1 \end{bmatrix}$.

3. Rotation $\frac{3\pi}{16} \approx \begin{bmatrix} 1 & 0 \\ -\frac{1}{8} & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{2} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{3}{4} & 1 \end{bmatrix}$.

The 8×8 binDCT's dyadic transform coefficients are listed in Table. 2. Its frequency responses are presented in Fig. 9.

3.4. Design Example 3: 8-channel 16-tap LiftLT

In this 8-channel 16-tap design example, we set the initial stage $\mathbf{E}_0$ to be the DCT itself because there are numerous fast DCT implementations in both software and hardware. Moreover, although the DCT is a floating-point transform, it has several integer implementations as well (where the perfect reconstruction property is sacrificed).[15] Following the excellent observation in,[14] we apply a scaling to the first DCT's antisymmetric basis to generate synthesis LT basis functions whose end values decay smoothly to exact zero – a crucial requirement in blocking artifacts elimination. However, instead of scaling the analysis by $\sqrt{2}$ and the synthesis by $\frac{1}{\sqrt{2}}$, we opt for $\frac{25}{16}$ and its inverse $\frac{16}{25}$ as shown in Fig. 10 since our interest is in integer-coefficient FBs. Another nice value that also works almost as well as $\frac{25}{16}$ is $\frac{5}{4}$.

After 2 series of ± 1 butterflies $\mathbf{W}$ and the delay chain $\mathbf{\Lambda}(z)$, the LT symmetric basis functions already have good attenuation, especially at DC ($\omega = 0$). Hence, we can comfortably set $\mathbf{U}_1 = \mathbf{I}_{\frac{M}{2}}$ as in.[9] Now, there is only the parameterization of $\mathbf{V}_1$ left. In the Type-I fast 8×16 LOT, the three rotation angles that yield near-optimal solution are $\{0.13\pi, 0.16\pi, 0.13\pi\}$. It turns out that each of these rotation angles can be loosely approximated by two lifting steps

$$\begin{bmatrix} 1 & -\frac{1}{2} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{2} \\ 0 & 1 \end{bmatrix} \tag{12}$$

without losing any performance. The final LiftLT's lattice structure is presented in Fig. 10. The frequency responses of the 8×16 LiftLT's basis functions are depicted in Fig. 11.

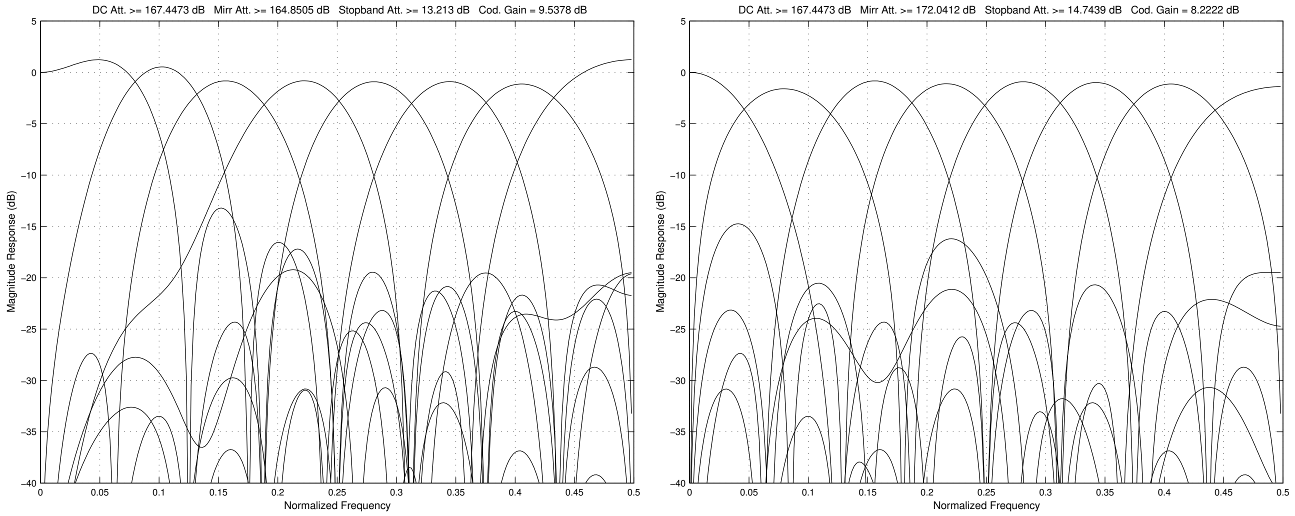

Figure 11. Frequency responses of the 8-channel 16-tap LPPRFB in Design Example 3. Left: analysis bank. Right: synthesis bank.

3.5. Computational Complexity and Performance

This section provide a comparison of the computational complexity and the theoretical performance in 1D between the new integer FBs and current state-of-the-art transforms. Table 3 tabulates the number of multiplications, additions, and/or shifting operations needed to process 8 input samples. Table 4 lists the coding gain of each transform given an $AR(1)$ signal with intersample autocorrelation coefficient $\rho = 0.95$. The evidence in Table 3-4 shows that high-performance, yet low-complexity, FBs can be constructed based on our proposed lattice structure.

The novel 4-channel 8-tap integer FB, 8-channel 8-tap binDCT, and 8-channel 16-tap LiftLT are all faster than the 9/7-tap wavelet transform, even in its lifting implementation. The LiftLT is slower than the DCT. However, the LiftLT should be sufficiently fast for many applications, especially in hardware, since most of the additional computation comes from the 2 butterflies and the 6 shift-and-add lifting steps. It is even faster than the type-I fast LOT.[9] Moreover, the LiftLT achieves a coding gain of 9.54 dB, significantly higher than that of the DCT (8.83 dB). In fact, the LiftLT's performance is already very close to that of the optimal GLBT[8,7] which attains 9.63 dB coding gain. The improvement in coding performance is tremendous as demonstrated by the image coding experiments in the next section.

Table 3. Comparison of transform complexity: number of operations needed per 8 transform coefficients.

Transform	*No. of Multiplications*	*No. of Additions*	*No. of Shifts*
9/7 Wavelet, 3-level, standard	63	98	0
9/7 Wavelet, 3-level, lifting	42	56	0
8×8 DCT	13	29	0
4×8 intFB	0	42	20
8×8 binDCT	0	31	14
8×16 LiftLT	14	51	6

4. IMAGE CODING EXPERIMENT

To be fair, the same SPIHT's quantizer and entropy coder[17] are utilized to encode the coefficients of every transform. The transforms in comparison are:

Table 4. Comparison of transform's coding gain with AR(1) image model, $\rho = 0.95$.

Transform	9/7 Wavelet, 1-level	8×8 DCT	4×8 intFB	8×8 binDCT	8×16 LiftLT
Coding Gain	5.92 dB	8.83 dB	8.11 dB	8.77 dB	9.54 dB

- DCT: 8-channel, 8-tap floating-point filters
- Wavelet: 9/7-tap biorthogonal.
- binDCT: 8-channel, 8-tap multiplierless transform
- LiftLT: 8-channel, 16-tap filters

In the M-channel cases, we use the modified zerotree structure in[18] where each block of transform coefficients are treated analogously to a full wavelet tree and three more levels of decomposition are employed to decorrelate the DC subband further. The objective coding results (PSNR in dB) for standard 512×512 Goldhill and Barbara test images are tabulated in Table 5.

The LiftLT outperforms all transforms on both test images at all bit rates. The visual quality of its reconstructed images is also much more superior as testified in Fig. 12: blocking is completely avoided whereas ringing is reasonably contained. Comparing to the wavelet transform, the LiftLT consistently surpasses the 9/7-tap wavelet. The PSNR improvement can sometimes reach as high as 1.5 dB.

The binDCT's objective coding performance is slightly below that of its DCT relative. However, subjectively, the binDCT yields more visually-pleasant reconstructed images.

Table 5. Objective coding result comparison (PSNR in dB).

	Goldhill				*Barbara*			
Comp. Ratio	9/7 Wavelet	8×8 DCT	8×8 binDCT	8×16 LiftLT	9/7 Wavelet	8×8 DCT	8×8 binDCT	8×16 LiftLT
1:8	36.55	36.25	35.82	36.56	36.41	36.31	35.71	37.57
1:16	33.13	32.76	32.45	33.22	31.40	31.11	30.67	32.82
1:32	30.56	30.07	29.87	30.63	27.58	27.28	26.90	28.93
1:64	28.48	27.93	27.79	28.54	24.86	24.58	24.24	25.93

5. CONCLUSIONS

We have presented in this paper the theory, design, and implementation of M-channel LPPRFB with integer coefficients. All new FBs are based on a fast, efficient, robust, and modular lattice structure. Several low-complexity design examples are presented. With only 1 more multiplication (which can also be implemented with shift-and-add operations), 22 more additions, and 4 more delay elements comparing to the DCT, the novel 8-channel 16-tap LiftLT offers a fast, low-cost, VLSI-friendly implementation while providing high-quality reconstructed images, both objectively and subjectively. The LiftLT even surpasses the 9/7-tap biorthogonal wavelet with irrational coefficients. The 8-channel 8-tap binDCT provides a fast multiplierless close approximation of the 8×8 DCT. Furthermore, the binDCT can map integers to integers with exact reconstruction. This property is pivotal in transform-based lossless coding. It also allows a unifying lossy/lossless coding framework.

REFERENCES

1. P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice Hall, New Jersey, 1993.
2. M. Vetterli and J. Kovačević, *Wavelets and Subband Coding*, Prentice Hall, New Jersey, 1995.
3. G. Strang and T. Q. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, 1996.

Figure 12. Enlarged 256 × 256 Barbara portions at 32:1 compression ratio. Top row, from left to right: original image; coded by 8 × 8 DCT, 27.28 dB; coded by 9/7-tap wavelet, 27.58 dB. Bottom row, from left to right: coded by 4 × 8 integer FB, 26.93 dB; coded by 8 × 8 binDCT, 26.90 dB; coded by 8 × 16 LiftLT, 28.93 dB.

4. A. K. Soman, P. P. Vaidyanathan, and T. Q. Nguyen, "Linear-phase paraunitary filter banks: theory, factorizations and applications," *IEEE Trans. on Signal Processing* **41**, pp. 3480–3496, Dec. 1993.
5. T. D. Tran and T. Q. Nguyen, "On m-channel linear-phase fir filter banks and application in image compression," *IEEE Trans. on Signal Processing* **45**, pp. 2175–2187, Sep. 1997.
6. R. L. de Queiroz, T. Q. Nguyen, and K. R. Rao, "The genlot: generalized linear-phase lapped orthogonal transform," *IEEE Trans. on Signal Processing* **40**, pp. 497–507, Mar. 1996.
7. T. D. Tran, R. de Queiroz, and T. Q. Nguyen, "Linear phase perfect reconstruction filter bank: lattice structure, design, and application in image coding," *IEEE Trans. on Signal Processing* , to appear, Jan. 2000.
8. T. D. Tran, R. de Queiroz, and T. Q. Nguyen, "The generalized lapped biorthogonal transform," *Proc. Int. Conf. on Acoust., Speech, and Signal Processing* **III**, pp. 1441–1444, Seattle, May 1998.
9. H. S. Malvar, *Signal Processing with Lapped Transforms*, Artech House, Boston, 1992.
10. F. E. Hohn, *Elementary Matrix Algebra*, MacMillan, New York, 1964.
11. I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Appl.* **4**, pp. 247–269, 1998.
12. N. S. Jayant and P. Noll, *Digital Coding of Waveforms*, Prentice-Hall, New Jersey, 1984.
13. J. Katto and Y. Yasuda, "Performance evaluation of subband coding and optimization of its filter coefficients," *SPIE Proc. Visual Comm. and Image Proc.* , pp. 95–106, Boston, Nov. 1991.
14. H. S. Malvar, "Biorthogonal and nonuniform lapped transforms for transform coding with reduced blocking and ringing artifacts," *IEEE Trans. on Signal Processing* **46**, pp. 1043–1053, Apr. 1998.

15. K. R. Rao and P. Yip, *Discrete Cosine Transform : Algorithms, Advantages, Applications*, Academic Press, San Diego, 1990.
16. T. D. Tran, "Fast multiplierless approximation of the dct," *Proc. 33rd Annual Conference on Information Sciences and Systems* , pp. 933–938, Baltimore, Mar. 1999.
17. A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. on Circuits and Systems for Video Technology* **6**, pp. 243–250, Jan. 1996.
18. T. D. Tran and T. Q. Nguyen, "A progressive transmission image coder using linear phase uniform filter banks as block transforms," *IEEE Trans. on Image Processing* , to appear, Nov. 1999.