Ricardo L. de Queiroz "Lapped Transforms for Image Compression"
The Transform and Data Compression Handbook
Ed. K. R. Rao et al.
Boca Raton, CRC Press LLC, 2001

Chapter 5

Lapped Transforms for Image Compression

Ricardo L. de Queiroz
Xerox Corporation

Trac D. Tran
The Johns Hopkins University

5.1 Introduction

This chapter covers the basic aspects of lapped transforms and their applications to image compression. It is a subject that has been extensively studied mainly because lapped transforms are closely related to filter banks, wavelets, and time-frequency transformations. Some of these topics are also covered in other chapters in this handbook. In any case it is certainly impractical to reference all the contributions in the field. Therefore, the presentation will be more focused rather than general. We refer the reader to excellent texts such as Malvar [26], Strang and Nguyen [55], Vaidyanathan [63], and Vetterli and Kovacevic [66] for a more detailed treatment of filter banks.

For the rest of this introductory section we will cover the basic notation, give a brief history of lapped transforms, and introduce block-based transforms. We will describe the principles of a block transform and its corresponding transform matrix along with its factorization. We will also introduce multi-input multi-output systems and relate them to block transforms. In Section 5.2, lapped transforms are introduced. Basic theory and concepts are presented for both orthogonal and nonorthogonal cases. In Section 5.3 lapped transforms are related to multi-input multi-output discrete systems with memory laying the theoretical basis for the understanding of the factorization of a lapped transform. Such a factorization is then presented in Section 5.4. Section 5.5 introduces hierarchical lapped transforms (which are constructed by connecting transforms hierarchically in a tree path), briefly introducing time-frequency diagrams and concepts such as the exchange of resolution between time and frequency. Another concept is also introduced in Section 5.5: variable length lapped transforms, which are also found through hierarchical connection of systems. Practical transforms are then presented. Transforms with symmetric bases including the popular lapped orthogonal transform, its bi-orthogonal, and generalized versions are described in Section 5.6, while fast transforms with variable-length are presented in Section 5.7. The transforms based on cosine modulation are presented in Section 5.8. In order to apply lapped transforms to images, one has to be able to transform signal segments of finite-length. Several methods for doing so are discussed in Section 5.9. Design issues for lapped transforms are discussed in Section 5.10, wherein the emphasis is given to compression applications. In Section 5.11, image compression systems are briefly introduced, including JPEG and other methods based on wavelet transforms. The performance analysis of lapped transforms in image compression is carried in Section 5.12 for different compression systems and several transforms. Finally, the conclusions are presented in Section 5.13.

5.1.1 Notation

Notation conventions used here: $\mathbf{I}_n$ is the $n \times n$ identity matrix; $\mathbf{0}_n$ is the $n \times n$ null matrix, while $\mathbf{0}_{n \times m}$ stands for the $n \times m$ null matrix. $\mathbf{J}_n$ is the $n \times n$ counter-identity, or exchange, or reversing matrix, illustrated by the following example:

$$\mathbf{J}_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} .$$

$\mathbf{J}$ reverses the ordering of elements of a vector. $[\ ]^T$ means transposition. $[\ ]^H$ means transposition combined with complex conjugation, where this combination is usually called the Hermitian conjugation of the vector or matrix. Unidimensional concatenation of matrices and vectors is indicated by a comma. In general, capital bold face letters are reserved for matrices, so that $\mathbf{a}$ represents a (column) vector while $\mathbf{A}$ represents a matrix.

5.1.2 Brief History

In the early 1980s, transform coding was maturing, and the discrete cosine transform (DCT) [45] was the preferred transformation method. At that time, DCT-based image compression was state-of-the-art, but researchers were uncomfortable with the blocking artifacts which are common (and annoying) artifacts found in images compressed at low bit rates using block transforms. To resolve the problem, the idea of a lapped transform (LT, for short) was developed in the early 1980s at M.I.T. The idea was to extend the basis function beyond the block boundaries, creating an overlap, in order to eliminate the blocking effect. This idea was not new at that time. However, the new ingredient was to preserve the number of transform coefficients and orthogonality, just as in the nonoverlapped case. Cassereau [5] introduced the lapped orthogonal transform (LOT). It was Malvar [18, 19, 20] who gave the LOT an elegant design strategy and a fast algorithm, thus making the LOT practical and a serious contender to replace the DCT for image compression.

It was also Malvar [22] who pointed out the equivalence between an LT and a multirate filter bank, which is now a very popular signal processing tool [63]. Based on cosine-modulated filter banks [33], modulated lapped transforms were designed [21, 48]. Modulated transforms were later generalized for an arbitrary overlap, creating the class of extended lapped transforms (ELT) [24]–[27]. Recently a new class of LTs with symmetric bases were developed yielding the class of generalized LOTs (GenLOT) [35, 37, 40]. The GenLOTs were made to have basis functions of arbitrary length (not a multiple of the block size) [57], extended to the nonorthogonal case [61], and even made to have filters of different lengths [60]. As mentioned before, filter banks and LTs are the same, although studied independently in the past. Because of this duality, it would be impractical to mention all related work in the field. Nevertheless, Vaidyanathan's book [63] is considered an excellent text on filter banks, while Malvar [26] is a good reference to bridge the gap between lapped transforms and filter banks. We usually refer to LTs as uniform critically sampled FIR filter banks with fast implementation algorithms based on special factorizations of the basis functions, with particular design attention for signal (mainly image) coding.

5.1.3 Block Transforms

We assume a one-dimensional input sequence $\{x(n)\}$ which is transformed into several coefficients sequences $\{y_i(n)\}$, where $y_i(n)$ would belong to the i-th subband. In traditional block-transform processing, the signal is divided into blocks of M samples, and each block is processed independently [6, 12, 26, 32, 43, 45, 46]. Let the samples in the m-th block be denoted as

$$\mathbf{x}_m^T = \left[x_0(m), x_1(m), \dots, x_{M-1}(m)\right] \ , \tag{5.1}$$

with $x_k(m) = x(mM + k)$, and let the corresponding transform vector be

$$\mathbf{y}_m^T = \left[y_0(m), y_1(m), \dots, y_{M-1}(m)\right] \ . \tag{5.2}$$

For a real unitary transform $\mathbf{A}$, $\mathbf{A}^T = \mathbf{A}^{-1}$. The forward and inverse transforms for the m-th block are, respectively,

$$\mathbf{y}_m = \mathbf{A}\mathbf{x}_m \ , \tag{5.3}$$

and

$$\mathbf{x}_m = \mathbf{A}^T\mathbf{y}_m \ . \tag{5.4}$$

The rows of $\mathbf{A}$, denoted $\mathbf{a}_n^T$ $(0 \le n \le M-1)$, are called the basis vectors because they form an orthogonal basis for the M-tuples over the real field [46]. The transform coefficients $[y_0(m), y_1(m), \dots, y_{M-1}(m)]$ represent the corresponding weights of vector $\mathbf{x}_m$ with respect to the above basis.

If the input signal is represented by vector $\mathbf{x}$ while the subbands are grouped into blocks in vector $\mathbf{y}$, we can represent the transform $\mathbf{H}$ which operates over the entire signal as a block diagonal matrix:

$$\mathbf{H} = diag\{\dots, \mathbf{A}, \mathbf{A}, \mathbf{A}, \dots\} , \tag{5.5}$$

where, of course, **H** is an orthogonal matrix if **A** is also. In summary, a signal is transformed by block segmentation followed by block transformation, which amounts to transforming the signal with a sparse matrix. Also, it is well known that the signal energy is preserved under a unitary transformation [12, 45], assuming stationary signals, i.e.,

$$M\sigma_x^2 = \sum_{i=0}^{M-1} \sigma_i^2 , \tag{5.6}$$

where σ_i^2 is the variance of $y_i(m)$ and σ_x^2 is the variance of the input samples.

5.1.4 Factorization of Discrete Transforms

For our purposes, discrete transforms of interest are linear and governed by a square matrix with real entries. Square matrices can be factorized into a product of sparse matrices of the same size. Notably, orthogonal matrices can be factorized into a product of plane (Givens) rotations [10]. Let **A** be an $M \times M$ real orthogonal matrix, and let $\mathbf{\Theta}(i, j, \theta_n)$ be a matrix with entries Θ_{kl}, which is like the identity matrix $\mathbf{I}_M$ except for four entries:

$$\Theta_{ii} = \cos(\theta_n) \quad \Theta_{jj} = \cos(\theta_n) \quad \Theta_{ij} = \sin(\theta_n) \quad \Theta_{ji} = -\sin(\theta_n) \ . \tag{5.7}$$

$\mathbf{\Theta}(i, j, \theta_n)$ corresponds to a rotation by the angle θ_n about an axis normal to the i-th and the j-th axes. Then, **A** can be factorized as

$$\mathbf{A} = \mathbf{S} \prod_{i=0}^{M-2} \prod_{j=i+1}^{M-1} \mathbf{\Theta}(i, j, \theta_n) \tag{5.8}$$

where n is increased by one for every matrix, and **S** is a diagonal matrix with entries ± 1 to correct for any sign error [10]. This correction is not necessary in most cases and is not required if we can apply variations of the rotation matrix defined in Eq. (5.7) as

$$\Theta_{ii} = \cos(\theta_n) \quad \Theta_{jj} = -\cos(\theta_n) \quad \Theta_{ij} = \sin(\theta_n) \quad \Theta_{ji} = \sin(\theta_n) \ . \tag{5.9}$$

All combinations of pairs of axes shall be used for a complete factorization. Fig. 5.1(a) shows an example of the factorization of a 4×4 orthogonal matrix into plane rotations [the sequence of rotations is slightly different than the one in Eq. (5.8) but it is equally complete]. If the matrix is not orthogonal, we can always decompose the matrix using singular value decomposition (SVD) [10]. **A** is decomposed through SVD as

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{V} \tag{5.10}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and $\mathbf{\Lambda}$ is a diagonal matrix containing the singular values of $\mathbf{A}$. While $\mathbf{\Lambda}$ is already a sparse matrix, we can further decompose the orthogonal matrices using Eq. (5.8):

$$\mathbf{A} = \mathbf{S}\left(\prod_{i=0}^{M-2}\prod_{j=i+1}^{M-1}\mathbf{\Theta}\left(i, j, \theta_n^U\right)\right)\mathbf{\Lambda}\left(\prod_{i=0}^{M-2}\prod_{j=i+1}^{M-1}\mathbf{\Theta}\left(i, j, \theta_n^V\right)\right) \tag{5.11}$$

where $\{\theta_n^U$ and $\theta_n^V\}$ compose the set of angles for $\mathbf{U}$ and $\mathbf{V}$, respectively. Fig. 5.1(c) illustrates the factorization for a 4×4 nonorthogonal matrix, where α_i are the singular values.

The reader will later see that the factorization above is an invaluable tool for the design of block and lapped transforms. In the orthogonal case, all of the degrees of freedom are contained in the rotation angles. In an $M \times M$ orthogonal matrix, there are $M(M-1)/2$ angles, and by spanning all the angles' space (0 to 2π for each one) one spans the space of all $M \times M$ orthogonal matrices. The idea is to span the space of all possible orthogonal matrices through varying arbitrarily and freely the rotation angles in an unconstrained optimization. In the general case, there are M^2 degrees of freedom, and we can either utilize the matrix entries directly or employ the SVD decomposition. However, we are mainly concerned with invertible matrices. Hence, using the SVD-based method, one can stay in the invertible matrix space by freely spanning the angles. The only mild constraint here is to assure that all singular values in the diagonal matrix are nonzero. The authors commonly use the unconstrained nonlinear optimization based on the simplex search provided by MATLAB$^{\text{TM}}$ to search for the optimal rotation angles and singular values.

5.1.5 Discrete MIMO Linear Systems

Let a multi-input multi-output (MIMO) [63] discrete linear FIR system have M input and M output sequences with respective Z-transforms $X_i(z)$ and $Y_i(z)$, for $0 \leq i \leq M-1$. Then, $X_i(z)$ and $Y_i(z)$ are related by

$$\begin{bmatrix} Y_0(z) \\ Y_1(z) \\ \vdots \\ Y_{M-1}(z) \end{bmatrix} = \begin{bmatrix} E_{0,0}(z) & E_{0,1}(z) & \cdots & E_{0,M-1}(z) \\ E_{1,0}(z) & E_{1,1}(z) & \cdots & E_{1,M-1}(z) \\ \vdots & \vdots & \ddots & \vdots \\ E_{M-1,0}(z) & E_{M-1,1}(z) & \cdots & E_{M-1,M-1}(z) \end{bmatrix} \begin{bmatrix} X_0(z) \\ X_1(z) \\ \vdots \\ X_{M-1}(z) \end{bmatrix} \tag{5.12}$$

where $E_{ij}(z)$ are entries of the given MIMO system $\mathbf{E}(z)$. $\mathbf{E}(z)$ is called the transfer matrix of the system, and we have chosen it to be square for simplicity. It is a regular matrix whose entries are polynomials. Of relevance to us is the case wherein the

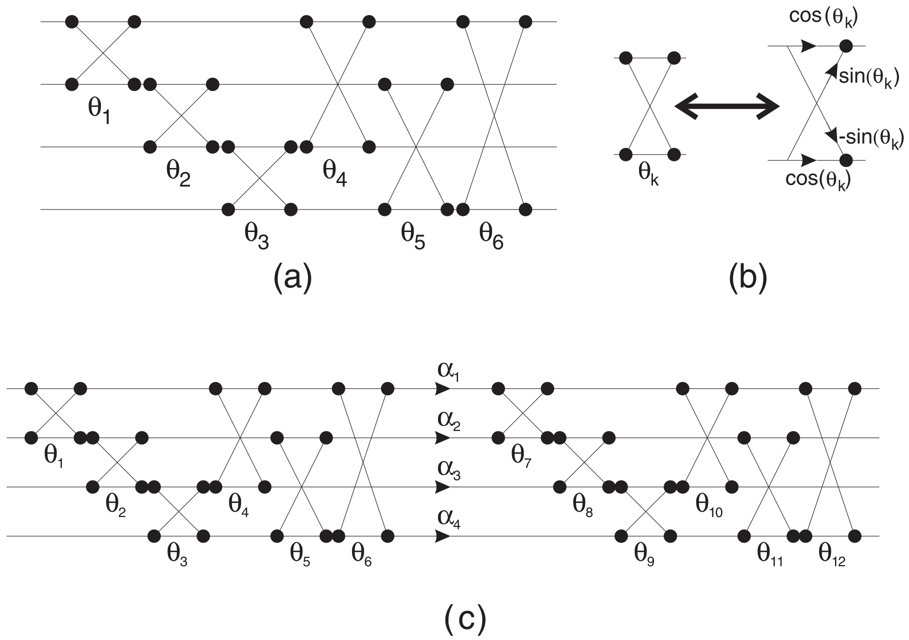


FIGURE 5.1

Factorization of a 4x4 matrix. (a) Orthogonal factorization into Givens rotations. (b) Details of the rotation element. (c) Factorization of a nonorthogonal matrix through SVD with the respective factorization of SVD's orthogonal factors into rotations.

entries belong to the field of real-coefficient polynomials of z^{-1}; i.e., the entries represent real-coefficient FIR filters. The degree of $\mathbf{E}(z)$ (or the McMillan degree, N_z) is the minimum number of delays necessary to implement the system. The order of $\mathbf{E}(z)$ is the maximum degree among all $E_{ij}(z)$. In both cases we assume that the filters are causal and FIR.

A special subset of great interest is comprised of the transfer matrices that are normalized paraunitary. In the paraunitary case, $\mathbf{E}(z)$ becomes a unitary matrix when evaluated on the unit circle:

$$\mathbf{E}^H\left(e^{j\omega}\right)\mathbf{E}\left(e^{j\omega}\right) = \mathbf{E}\left(e^{j\omega}\right)\mathbf{E}^H\left(e^{j\omega}\right) = \mathbf{I}_M \ . \tag{5.13}$$

Furthermore:

$$\mathbf{E}^{-1}(z) = \mathbf{E}^T\left(z^{-1}\right) \ . \tag{5.14}$$

For causal inverses of paraunitary systems,

$$\mathbf{E}'(z) = z^{-n}\mathbf{E}^T\left(z^{-1}\right) \tag{5.15}$$

is often used, where n is the order of $\mathbf{E}(z)$, since $\mathbf{E}'(z)\mathbf{E}(z) = z^{-n}\mathbf{I}_M$.

For paraunitary systems, the determinant of $\mathbf{E}(z)$ is of the form az^{-N_z} for a real constant a [63], where we recall that N_z is the McMillan degree of the system. For FIR causal entries, they are also said to be lossless systems [63]. In fact, a familiar orthogonal matrix is one where all $E_{ij}(z)$ are constant for all z.

We also have interest in invertible, although nonparaunitary, transfer matrices. In this case, it is required that the matrix be invertible on the unit circle, i.e., for all $z = e^{j\omega}$ and real ω. Nonparaunitary systems are also called bi-orthogonal or perfect reconstruction (PR) [63].

5.1.6 Block Transform as a MIMO System

The sequences $\{x_i(m)\}$ in Eq. (5.1) are called the polyphase components of the input signal $\{x(n)\}$. On the other hand, the sequences $\{y_i(m)\}$ in Eq. (5.2) are the subbands resulting from the transform process. In an alternative view of the transformation process, the signal samples are "blocked" or parallelized into polyphase components through a sequence of delays and decimators as shown in Fig. 5.2. Each block is transformed by system $\mathbf{A}$ into M subband samples (transformed samples). Inverse transform (for orthogonal transforms) is accomplished by system $\mathbf{A}^T$ whose outputs are polyphase components of the reconstructed signal, which are then serialized by a sequence of upsamplers and delays. In this system, blocks are processed independently. Therefore, the transform can be viewed as a MIMO system of order 0, i.e., $\mathbf{E}(z) = \mathbf{A}$, and if $\mathbf{A}$ is unitary, so is $\mathbf{E}(z)$ which is obviously also paraunitary. The system matrix relating the polyphase components to the subbands is referred to as the polyphase transfer matrix (PTM).

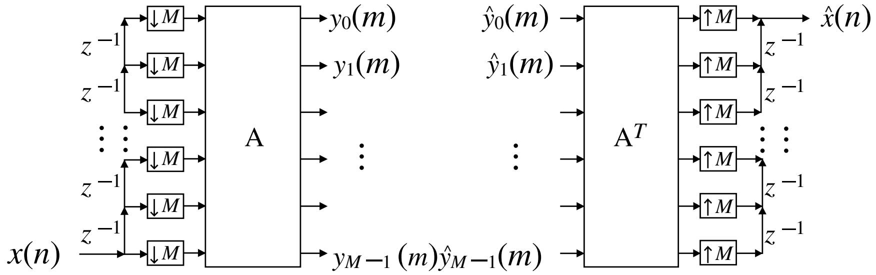


FIGURE 5.2
The signal samples are parallelized into polyphase components through a sequence of delays and decimators (↓ M means subsampling by a factor of M). The signal is "blocked," and each block is transformed by system A into M subband samples (transformed samples). Inverse transform (for orthogonal transforms) is accomplished by system $\mathbf{A}^T$ whose outputs are polyphase components of the reconstructed signal, which are then serialized by a sequence of upsamplers (↑ M means upsampling by a factor of M, padding the signal with $M - 1$ zeros) and delays.

5.2 Lapped Transforms

The motivation for a transform with overlap, as mentioned in the introduction, is to try to improve the performance of block (nonoverlapped) transforms for image and signal compression. Compression commonly implies signal losses due to quantization [12]. As the bases of block transforms do not overlap, there may be discontinuities along the boundary regions of the blocks. Different approximations of those boundary regions on each side of the border may cause an artificial "edge" between blocks, the so-called *blocking* effect. Fig. 5.3 shows an example signal that is to be projected into bases, by segmenting the signal into blocks and projecting each segment into the desired bases. Alternatively, one can view the process as projecting the whole signal into several translated bases (one translation per block). Fig. 5.3 shows, on the left, translated versions of the first basis of the DCT in order to account for all the different blocks. The same figure, on the right, shows the same diagram for the first basis of a typical short LT. Note that the bases overlap spatially. The idea is that overlap would help decrease, if not eliminate, the *blocking* effect.

There are M basis functions for either the DCT or the LT, although Fig. 5.3 shows only one of them. An example of the bases for $M = 8$ is shown in Fig. 5.4 which plots the bases for the DCT and for the LOT, a particular LT discussed later. The reader may note that not only are the LOT bases longer but they are also smoother than the DCT counterpart. Fig. 5.5(a) is an example of an image compressed using the standard JPEG baseline coder [32], and the blocking artifacts at the boundaries of 8×8 pixels blocks are readily seen. By replacing the DCT with the LOT and keeping the same compression ratio, we obtain the image shown in Fig. 5.5(b), where blocking is largely reduced. This brief introduction to the motivation behind the development of LTs illustrates only the overall problem. We have not described the details on how to apply LTs. The following section develops the LT framework.

5.2.1 Orthogonal Lapped Transforms

A lapped transform [26] can be generally defined as any transform whose basis vectors have length L, such that $L > M$, extending across traditional block boundaries. Thus, the transform matrix is no longer square, and most of the equations valid for block transforms do not apply to an LT. We will concentrate our efforts on *orthogonal* LTs [26] and consider $L = NM$, where N is the overlap factor. Note that N, M, and hence L are all integers. As in the case of block transforms, we define the transform matrix as containing the orthonormal basis vectors as its rows. A lapped transform matrix $\mathbf{P}$ of dimensions $M \times L$ can be divided into square $M \times M$ submatrices $\mathbf{P}_i$ $(i = 0, 1, \ldots, N-1)$ as follows:

$$\mathbf{P} = \left[\mathbf{P}_0 \; \mathbf{P}_1 \; \cdots \; \mathbf{P}_{N-1}\right] . \tag{5.16}$$

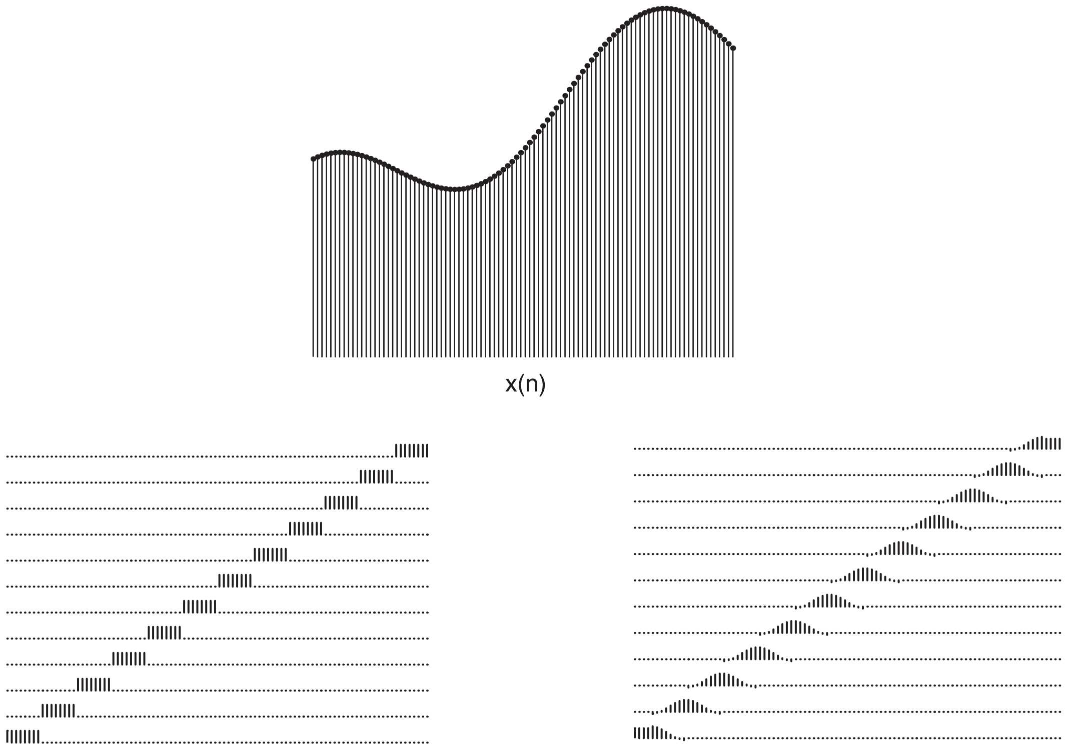


FIGURE 5.3
The example discrete signal $x(n)$ is to be projected onto a number of bases. Left: spatially displaced versions of the first DCT basis. Right: spatially displaced versions of the first basis of a typical short LT.

The orthogonality property does not hold because **P** is no longer a square matrix, and it is replaced by the perfect reconstruction (PR) property [26], defined by

$$\sum_{i=0}^{N-1-l} \mathbf{P}_i \mathbf{P}_{i+l}^T = \sum_{i=0}^{N-1-l} \mathbf{P}_{i+l}^T \mathbf{P}_i = \delta(l)\mathbf{I}_M \ , \tag{5.17}$$

for $l = 0, 1, \ldots, N-1$, where $\delta(l)$ is the Kronecker delta; i.e., $\delta(0) = 1$ and $\delta(l) = 0$ for $l \neq 0$. As will be seen later, Eq. (5.17) states the PR conditions and orthogonality of the transform operating over the entire signal.

If we divide the signal into blocks, each of size M, we would have vectors $\mathbf{x}_m$ and $\mathbf{y}_m$ as in Eqs. (5.1) and (5.2). These blocks are not used by LTs in a straightforward manner. The actual vector that is transformed by the matrix **P** has to have L samples, and, at block number m, it is composed of the samples of $\mathbf{x}_m$ plus $L-M$ samples from the neighboring blocks. These samples are chosen by picking $(L-M)/2$ samples on each side of the block $\mathbf{x}_m$, as shown in Fig. 5.6, for $N = 2$. However, the number

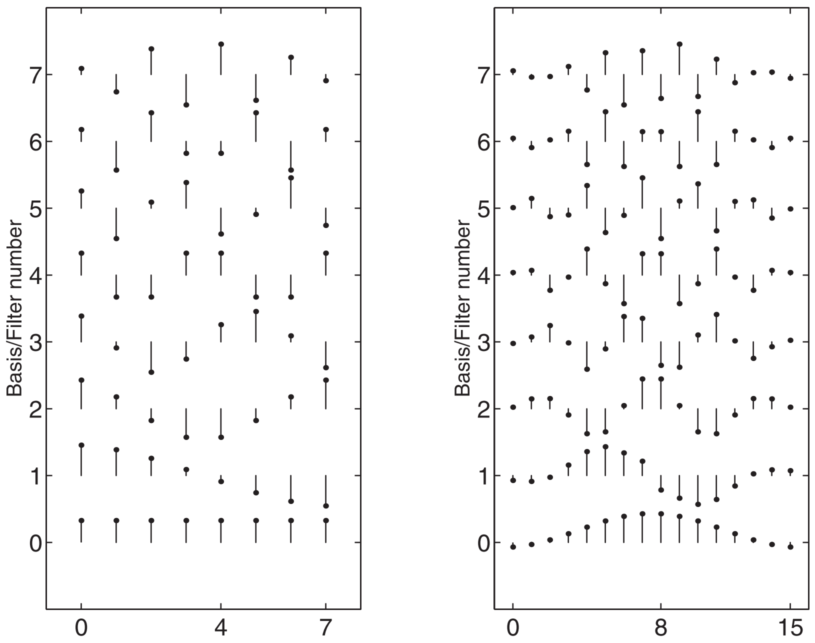


FIGURE 5.4
Bases for the 8-point DCT ($M = 8$) (left) and for the LOT (right) with $M = 8$. The LOT is a particular LT which will be explained later.

of transform coefficients at each step is M, and, in this respect, there is no change in the way we represent the transform-domain blocks $\mathbf{y}_m$.

The input vector of length L is denoted as $\mathbf{v}_m$, which is centered around the block $\mathbf{x}_m$, and is defined as

$$\mathbf{v}_m^T = \left[x\left(mM - (N-1)\frac{M}{2} \right) \cdots x\left(mM + (N+1)\frac{M}{2} - 1 \right) \right] . \tag{5.18}$$

Then, we have

$$\mathbf{y}_m = \mathbf{P}\mathbf{v}_m \ . \tag{5.19}$$

The inverse transform is not direct as in the case of block transforms; with the knowledge of $\mathbf{y}_m$ we know neither the samples in the support region of $\mathbf{v}_m$, nor those in the support region of $\mathbf{x}_m$. We can reconstruct a vector $\hat{\mathbf{v}}_m$ from $\mathbf{y}_m$, as

$$\hat{\mathbf{v}}_m = \mathbf{P}^T \mathbf{y}_m \ , \tag{5.20}$$

where $\hat{\mathbf{v}}_m \neq \mathbf{v}_m$. To reconstruct the original sequence, it is necessary to accumulate the results of the vectors $\hat{\mathbf{v}}_m$, in a sense that a particular sample $x(n)$ will be reconstructed from the sum of the contributions it receives from all $\hat{\mathbf{v}}_m$. This additional

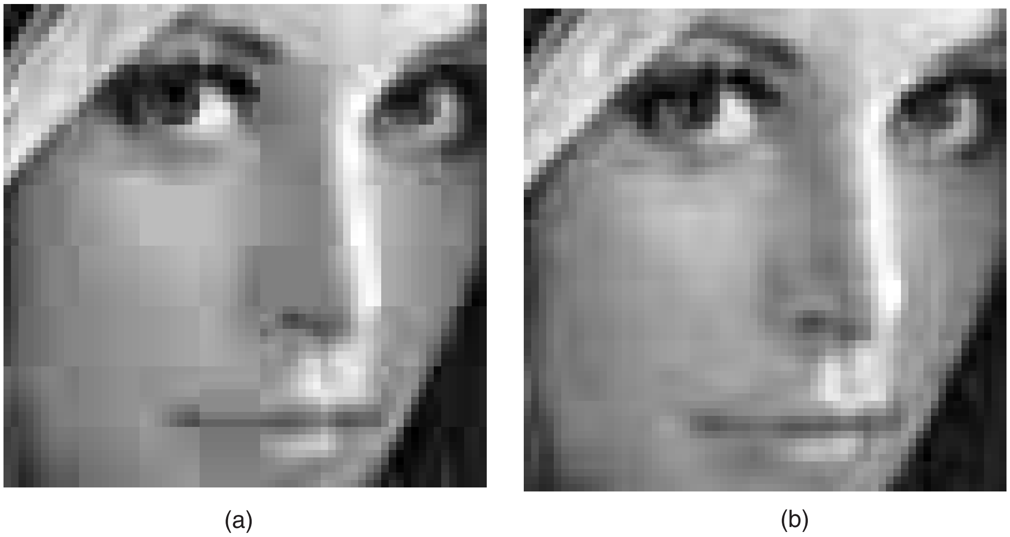


FIGURE 5.5
Zoom of image compressed using JPEG at 0.5 bits/per pixel. (a) DCT, (b) LOT. Reproduced by Special Permission of *Playboy* magazine. Copyright ©1972, 2000 by Playboy.

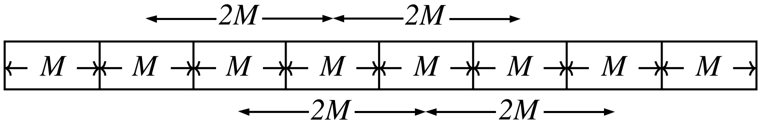


FIGURE 5.6
The signal samples are divided into blocks of M samples. The lapped transform uses neighboring blocks samples, as in this example for $N = 2$; i.e., $L = 2M$, yielding an overlap of $(L - M)/2 = M/2$ samples on either side of a block.

complication comes from the fact that $\mathbf{P}$ is not a square matrix [26]. However, the entire analysis-synthesis system (applied to the entire input vector) is still orthogonal, assuring the PR property using Eq. (5.20).

We can also describe the above process using a sliding rectangular window applied over the samples of $\{x(n)\}$. As an M-sample block, $\mathbf{y}_m$ is computed using $\mathbf{v}_m$, and $\mathbf{y}_{m+1}$ is computed from $\mathbf{v}_{m+1}$ which is obtained by shifting the window to the right by M samples, as shown in Fig. 5.7.

As the reader may have noticed, the region of support of all vectors $\mathbf{v}_m$ is greater than the region of support of the input vector. Hence, a special treatment has to be given to the transform at the borders. We will discuss this operation later and assume infinite-length signals until then. We can also assume that the signal length is very large and the borders of the signal are far enough from the region on which we are focusing our attention.

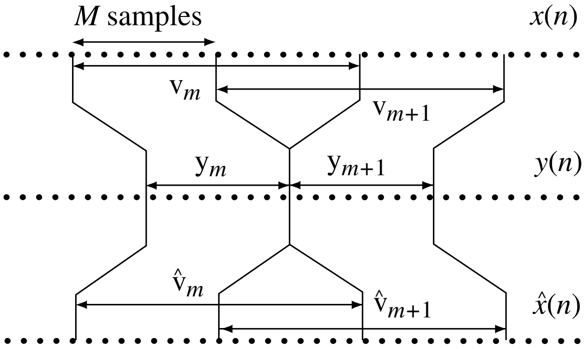


FIGURE 5.7
Illustration of a lapped transform with $N = 2$ applied to signal $x(n)$, yielding transform domain signal $y(n)$. The input L-tuple as vector $\mathbf{v}_m$ is obtained by a sliding window advancing M samples, generating $\mathbf{y}_m$. This sliding is also valid for the synthesis side.

If we denote by $\mathbf{x}$ the input vector and by $\mathbf{y}$ the transform-domain vector, we can be consistent with our notation of transform matrices by defining a matrix $\mathbf{H}$ such that $\mathbf{y} = \mathbf{Hx}$ and $\hat{\mathbf{x}} = \mathbf{H}^T\mathbf{y}$. In this case, we have

$$\mathbf{H} = \begin{bmatrix} \ddots & & & & \mathbf{0} \\ & \mathbf{P} & & & \\ & & \mathbf{P} & & \\ & & & \mathbf{P} & \\ \mathbf{0} & & & & \ddots \end{bmatrix} , \tag{5.21}$$

where the displacement of the matrices $\mathbf{P}$ obeys the following:

$$\mathbf{H} = \begin{bmatrix} \ddots & \ddots & & \ddots & & & \mathbf{0} \\ & \mathbf{P}_0 & \mathbf{P}_1 & \cdots & \mathbf{P}_{N-1} & & \\ & & \mathbf{P}_0 & \mathbf{P}_1 & \cdots & \mathbf{P}_{N-1} & \\ \mathbf{0} & & & \ddots & \ddots & & \ddots \end{bmatrix} . \tag{5.22}$$

$\mathbf{H}$ has as many block-rows as transform operations over each vector $\mathbf{v}_m$.

Let the rows of $\mathbf{P}$ be denoted by $1 \times L$ vectors $\mathbf{p}_i^T$ $(0 \le i \le M - 1)$, so that $\mathbf{P}^T = [\mathbf{p}_0, \ldots, \mathbf{p}_{M-1}]$. In an analogy to the block transform case, we have

$$y_i(m) = \mathbf{p}_i^T \mathbf{v}_m \ . \tag{5.23}$$

The vectors $\mathbf{p}_i$ are the basis vectors of the lapped transform. They form an orthogonal basis for an M-dimensional subspace (there are only M vectors) of the L-tuples over the real field. As a remark, assuming infinite length signals, from the orthogonality of the basis vectors and from the PR property in Eq. (5.17), the energy is preserved, such that Eq. (5.6) is valid.

In order to compute the variance of the subband signals of a block or lapped transform, assume that $\{x(n)\}$ is a zero-mean stationary process with a given autocorrelation function. Let its $L \times L$ autocorrelation matrix be $\mathbf{R}_{xx}$. Then, from Eq. (5.23)

$$E\left[y_i(m)\right] = \mathbf{p}_i^T E\left[\mathbf{v}_m\right] = \mathbf{p}_i^T \mathbf{0}_{L\times 1} = 0 \, , \tag{5.24}$$

so that

$$\sigma_i^2 = E\left[y_i^2(m)\right] = \mathbf{p}_i^T E\left[\mathbf{v}_m \mathbf{v}_m^T\right] \mathbf{p}_i = \mathbf{p}_i^T \mathbf{R}_{xx} \mathbf{p}_i \; ; \tag{5.25}$$

i.e., the output variance is easily computed from the input autocorrelation matrix for a given set of bases **P**.

Assuming that the entire input and output signals are represented by the vectors **x** and **y**, respectively, and that the signals have infinite length, we have, from Eq. (5.21),

$$\mathbf{y} = \mathbf{H}\mathbf{x} \tag{5.26}$$

and, if **H** is orthogonal,

$$\mathbf{x} = \mathbf{H}^T \mathbf{y} \, . \tag{5.27}$$

Note that **H** is orthogonal if and only if Eq. (5.17) is satisfied. Thus, the meaning for Eq. (5.17) becomes clear, as it forces the transform operating over the entire input-output signals to be orthogonal. Hence, the resulting LT is called orthogonal. For block transforms, as there is no overlap, it is sufficient to state the orthogonality of **A** because **H** will be a block-diagonal matrix.

These formulations for LTs are general, and if the transform satisfies the PR property described in Eq. (5.17), then the LTs are independent of the contents of the matrix **P**. The definition of **P** with a given N can accommodate any lapped transform whose length of the basis vectors lies between M and NM. For the case of block transforms, $N = 1$; i.e., there is no overlap.

Causal notation — If one is not concerned with particular localization of the transform with respect to the origin $x(0)$ of the signal $\{x(n)\}$, it is possible to change the notation to apply a causal representation. In this case, we can represent $\mathbf{v}_m$ as

$$\mathbf{v}_m^T = \left[\mathbf{x}_{m-N+1}^T, \ldots, \mathbf{x}_{m-1}^T, \mathbf{x}_m^T\right] \, , \tag{5.28}$$

which is identical to the previous representation, except for a shift in the origin to maintain causality. The block $\mathbf{y}_m$ is found in a similar fashion as

$$\mathbf{y}_m = \mathbf{P}\mathbf{v}_m = \sum_{i=0}^{N-1} \mathbf{P}_{N-1-i} \mathbf{x}_{m-i} \, . \tag{5.29}$$

Similarly, $\hat{\mathbf{v}}_m$ can be reconstructed as in Eq. (5.20) where the support region for the vector is the same, except that the relation between it and the blocks $\hat{\mathbf{x}}_m$ will be changed accordingly.

5.2.2 Nonorthogonal Lapped Transforms

So far, we have discussed orthogonal LTs where a segment of the input signal is projected onto the basis functions of **P**, yielding the coefficients (subband samples). The signal is reconstructed by the overlapped projection of the same bases weighted by the subband samples. In the nonorthogonal case, we define another LT matrix **Q** as

$$\mathbf{Q} = \left[\mathbf{Q}_0\, \mathbf{Q}_1\, \cdots\, \mathbf{Q}_{N-1}\right] , \tag{5.30}$$

in the same way as we did for **P** with the same size. The difference is that **Q** instead of **P** is used in the reconstruction process so that Eq. (5.20) is replaced by

$$\hat{\mathbf{v}}_m = \mathbf{Q}^T \mathbf{y}_m . \tag{5.31}$$

We also define another transform matrix as

$$\mathbf{H}' = \begin{bmatrix} \ddots & \ddots & & \ddots & & & \mathbf{0} \\ & \mathbf{Q}_0 & \mathbf{Q}_1 & \cdots & \mathbf{Q}_{N-1} & & \\ & & \mathbf{Q}_0 & \mathbf{Q}_1 & \cdots & \mathbf{Q}_{N-1} & \\ \mathbf{0} & & & \ddots & \ddots & & \ddots \end{bmatrix} . \tag{5.32}$$

The forward and inverse transformations are now

$$\mathbf{y} = \mathbf{H}_F \mathbf{x} \quad , \quad \mathbf{x} = \mathbf{H}_I \mathbf{y} . \tag{5.33}$$

In the orthonormal case, $\mathbf{H}_F = \mathbf{H}$ and $\mathbf{H}_I = \mathbf{H}^T$. In the general case, it is required that $\mathbf{H}_I = \mathbf{H}_F^{-1}$. With the choice of **Q** as the inverse LT, then $\mathbf{H}_I = \mathbf{H}'^T$, while $\mathbf{H}_F = \mathbf{H}$. Therefore the perfect reconstruction condition is

$$\mathbf{H}'^T \mathbf{H} = \mathbf{I}_\infty . \tag{5.34}$$

The reader can check that the above equation can also be expressed in terms of the LTs **P** and **Q** as

$$\sum_{k=0}^{N-1-m} \mathbf{Q}_k^T \mathbf{P}_{k+m} = \sum_{k=0}^{N-1-m} \mathbf{Q}_{k+m}^T \mathbf{P}_k = \delta(m) \mathbf{I}_M , \tag{5.35}$$

which establish the general necessary and sufficient conditions for the perfect reconstruction of the signal by using **P** as the forward LT and **Q** as the inverse LT. Unlike the orthogonal case in Eq. (5.17), here both sets are necessary conditions; there is a total of $2N - 1$ matrix equations.

5.3 LTs as MIMO Systems

As previously discussed in Sections 5.1.3 and 5.1.6, the input signal can be decomposed into M polyphase signals $\{x_i(m)\}$, each sequence having one M-th of the original rate. As there are M subbands $\{y_i(m)\}$ under the same circumstances, and only linear operations are used to transform the signal, there is a MIMO system $\mathbf{F}(z)$ that converts the M polyphase signals to the M subband signals. Those transfer matrices are also called PTM (Section 5.1.6). The same is true for the inverse transform (from subbands $\{\hat{y}_i(m)\}$ to polyphase $\{\hat{x}_i(m)\}$). Therefore, we can use the diagram shown in Fig. 5.8 to represent the forward and inverse transforms. Note that Fig. 5.8 is identical to Fig. 5.2 except that the transforms have memory; they depend not only on the present input vector, but on past input vectors also. One can view the system as a clocked one, in which at every clock a block is input, transformed, and output. The parallelization and serialization of blocks are performed by the chain of delays, upsamplers, and downsamplers shown in Fig. 5.8. If we express the forward and inverse PTM as matrix polynomials

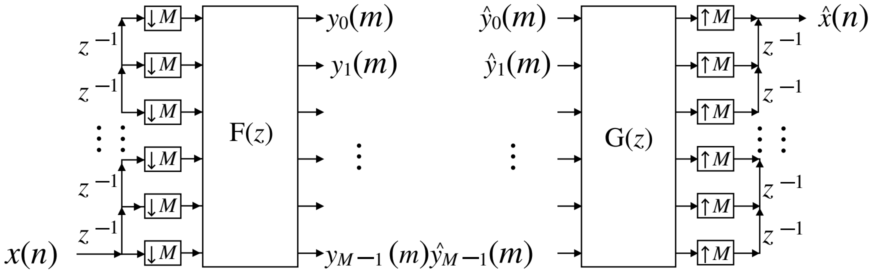


FIGURE 5.8

The filter bank represented as a MIMO system is applied to the polyphase components of the signal. The matrices $\mathbf{F}(z)$ and $\mathbf{G}(z)$ are called polyphase transfer matrices. For a PR system, both must be inverses of each other and for paraunitary filter banks they must be paraunitary matrices; i.e., $\mathbf{G}(z) = \mathbf{F}^{-1}(z) = \mathbf{F}^T(z^{-1})$. For a PR paraunitary causal system of order N, we must choose $\mathbf{G}(z) = z^{-(N-1)}\mathbf{F}^T(z^{-1})$.

$$\mathbf{F}(z) = \sum_{i=0}^{N-1} \mathbf{F}_i z^{-1}\,, \tag{5.36}$$

$$\mathbf{G}(z) = \sum_{i=0}^{N-1} \mathbf{G}_i z^{-1}\,, \tag{5.37}$$

then the forward and inverse transforms are given by

$$\mathbf{y}_m = \sum_{i=0}^{N-1} \mathbf{F}_i \mathbf{x}_{m-i}\,, \tag{5.38}$$

$$\hat{\mathbf{x}}_m = \sum_{i=0}^{N-1} \mathbf{G}_i \hat{\mathbf{y}}_{m-i}\,. \tag{5.39}$$

In the absence of any processing, $\hat{\mathbf{y}}_m = \mathbf{y}_m$ and $\mathbf{F}(z)$ and $\mathbf{G}(z)$ are connected together back-to-back, so PR is possible if they are inverses of each other. Since the inverse of a causal FIR MIMO system may be noncausal, we can delay the entries of the inverse matrix to make it causal. Since the MIMO system's PTM is assumed to have order N (N is the overlap factor of the equivalent LT), PR requires that

$$\mathbf{G}(z)\mathbf{F}(z) = z^{-N+1}\mathbf{I}_M \quad \rightarrow \quad \mathbf{G}(z) = z^{-N+1}\mathbf{F}^{-1}(z) \ . \tag{5.40}$$

In this case, $\hat{\mathbf{x}}_m = \mathbf{x}_{m-N+1}$; the signal is perfectly reconstructed after a system's delay. Because of the delay chains combined with the block delay (system's order), the reconstructed signal delay is $\hat{x}(n) = x(n - NM + 1) = x(n - L - 1)$.

By combining Eqs. (5.38), (5.39), and (5.40), we can restate the PR conditions as

$$\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \mathbf{G}_i \mathbf{F}_i z^{-i-j} = z^{-N+1}\mathbf{I}_M \ , \tag{5.41}$$

which, by equating the powers of z, can be rewritten as

$$\sum_{k=0}^{N-1-m} \mathbf{G}_k \mathbf{F}_{k+m} = \sum_{k=0}^{N-1-m} \mathbf{G}_{k+m}\mathbf{F}_k = \delta(m)\mathbf{I}_M \ . \tag{5.42}$$

The reader should note the striking similarity of the above equation with Eq. (5.35). In fact, the simple comparison of the transformation process in the space domain notation Eq. (5.33) against the MIMO system notation in Eqs. (5.38) and (5.39) would reveal the following relations

$$\mathbf{F}_k = \mathbf{P}_{N-1-k}, \qquad \mathbf{G}_k = \mathbf{Q}_k^T \tag{5.43}$$

for $0 \leq k < N$. In fact, the conditions imposed in Eqs. (5.34), (5.35), (5.40), and (5.42) are equivalent and each one implies the others. This is a powerful tool in the design of lapped transforms. As an LT, the matrix is nonsquare, but the entries are real. As a MIMO system, the matrix is square, but the entries are polynomials. One form may complement the other, facilitating tasks such as factorization, design, and implementation.

As mentioned earlier, paraunitary (lossless) systems belong to a class of MIMO systems of high interest. Let $\mathbf{E}(z)$ be a paraunitary PTM so that $\mathbf{E}^{-1}(z) = \mathbf{E}^T(z^{-1})$, and let

$$\mathbf{F}(z) = \mathbf{E}(z), \qquad \mathbf{G}(z) = z^{-(N-1)}\mathbf{E}^T\left(z^{-1}\right) \ . \tag{5.44}$$

As a result, the reader can verify that the equations above imply that $\mathbf{P}_i = \mathbf{Q}_i$, and that

$$\sum_{i=0}^{N-1-l} \mathbf{P}_i \mathbf{P}_{i+l}^T \quad = \quad \sum_{i=0}^{N-1-l} \mathbf{P}_i^T \mathbf{P}_{i+l} = \delta(l)\mathbf{I}_M \ , \tag{5.45}$$

$$\mathbf{H}\mathbf{H}^T \quad = \quad \mathbf{H}^T\mathbf{H} = \mathbf{I}_\infty \ . \tag{5.46}$$

In other words, if the system's PTM is paraunitary, then the corresponding LT ($\mathbf{H}$) is orthogonal and vice-versa.

5.4 Factorization of Lapped Transforms

There is an important result for paraunitary PTM which states that any paraunitary $\mathbf{E}(z)$ can be decomposed into a series of orthogonal matrices and delay stages [8, 64]. In this decomposition, there are N_z delay stages and $N_z + 1$ orthogonal matrices, where N_z is the McMillan degree of $\mathbf{E}(z)$ (the degree of the determinant of $\mathbf{E}(z)$). Then,

$$\mathbf{E}(z) = \mathbf{B}_0 \prod_{i=1}^{N_z} \left(\boldsymbol{\Upsilon}(z)\mathbf{B}_i\right) \tag{5.47}$$

where $\boldsymbol{\Upsilon}(z) = diag\{z^{-1}, 1, 1, \ldots, 1\}$, and $\mathbf{B}_i$ are orthogonal matrices. It is well known that an $M \times M$ orthogonal matrix can be expressed as a product of $M(M-1)/2$ plane rotations. However, in this case, only $\mathbf{B}_0$ is a general orthogonal matrix, while the matrices $\mathbf{B}_1$ through $\mathbf{B}_{N_z}$ have only $M - 1$ degrees of freedom [64].

This result states that it is possible to implement an orthogonal lapped transform using a sequence of delays and orthogonal matrices. It also defines the total number of degrees of freedom in a lapped transform; if one changes arbitrarily any of the plane rotations composing the orthogonal transforms, one will span all possible orthogonal lapped transforms, for given values of M and L. It is also possible to prove [35] that the (McMillan) degree of $\mathbf{E}(z)$ is bounded by $N_z \leq (L - M)/2$ with equality for a general structure to implement all orthogonal LTs whose bases have length up to $L = NM$, i.e., $\mathbf{E}(z)$ of order $N - 1$.

In fact Eq. (5.47) may be used to implement all lapped transforms (orthogonal or not) whose degree is N_z. To accomplish that, all of the multiplicative factors that compose the PTM must be invertible. Let us consider a more particular factorization:

$$\mathbf{F}(z) = \prod_{i=0}^{(N-1)/(K-1)} \mathbf{B}_i(z) \tag{5.48}$$

where $\mathbf{B}_i(z) = \sum_{k=0}^{K-1} \mathbf{B}_{ik} z^{-k}$ is a stage of order $K - 1$. If $\mathbf{F}(z)$ is paraunitary, then all $\mathbf{B}_i(z)$ must be paraunitary, so that perfect reconstruction is guaranteed if

$$\mathbf{G}(z) = z^{-N+1}\mathbf{F}^T\left(z^{-1}\right) = \prod_{i=(N-1)/(K-1)}^{0} \left(\sum_{k=0}^{K-1} \mathbf{B}_{ik}^T z^{-(K-1-k)}\right) . \tag{5.49}$$

If PTM is not paraunitary, all factors have to be invertible in the unit circle for PR. More strongly put, there must be factors $\mathbf{C}_i(z)$ of order $K - 1$ such that

$$\mathbf{C}_i(z)\mathbf{B}_i(z) = z^{-K+1}\mathbf{I}_M . \tag{5.50}$$

Then the inverse PTM is simply given by

$$\mathbf{G}(z) = \prod_{i=(N-1)/(K-1)}^{0} \mathbf{C}_i(z) . \tag{5.51}$$

With this factorization, the design of $\mathbf{F}(z)$ is broken down to the design of $\mathbf{B}_i(z)$. Lower-order factors simplify the constraint analysis and facilitate the design of a useful transform, either paraunitary or invertible. It is even more desirable to factor the PTM as

$$\mathbf{F}(z) = \mathbf{B}_0 \prod_{i=0}^{N-1} \mathbf{\Lambda}(z)\mathbf{B}_i \tag{5.52}$$

where $\mathbf{B}_i$ are square matrices and $\mathbf{\Lambda}(z)$ is a paraunitary matrix containing only entries 1 and z^{-1}. In this case, if the PTM is paraunitary

$$\mathbf{G}(z) = \left(\prod_{i=N-1}^{0} \mathbf{B}_i^T \tilde{\mathbf{\Lambda}}(z) \right) \mathbf{B}_0^T \tag{5.53}$$

where $\tilde{\mathbf{\Lambda}}(z) = z^{-1}\mathbf{\Lambda}(1/z)$. If the PTM is not paraunitary, then

$$\mathbf{G}(z) = \left(\prod_{i=N-1}^{0} \mathbf{B}_i^{-1} \tilde{\mathbf{\Lambda}}(z) \right) \mathbf{B}_0^{-1} \; ; \tag{5.54}$$

in other words, the design can be simplified by applying only invertible real matrices $\mathbf{B}_i$. This factorization approach is the basis for most useful LTs. It allows efficient implementation and design. We will discuss some useful LTs later on. For example, for M even, the symmetric delay factorization (SDF) is quite useful. In that,

$$\mathbf{\Lambda}(z) = \begin{bmatrix} z^{-1}\mathbf{I}_{M/2} & 0 \\ 0 & \mathbf{I}_{M/2} \end{bmatrix}, \qquad \tilde{\mathbf{\Lambda}}(z) = \begin{bmatrix} \mathbf{I}_{M/2} & 0 \\ 0 & z^{-1}\mathbf{I}_{M/2} \end{bmatrix}. \tag{5.55}$$

The flow graph for implementing an LT which can be parameterized using SDF is shown in Fig. 5.9.

If we are given the SDF matrices instead of the basis coefficients, one can easily construct the LT matrix. For this, start with the last stage and recur the structure in Eq. (5.52) using Eq. (5.55). Let $\mathbf{P}^{(i)}$ be the partial reconstruction of $\mathbf{P}$ after including up to the i-th stage. Then,

$$\mathbf{P}^{(0)} = \mathbf{B}_{N-1} \tag{5.56}$$

$$\mathbf{P}^{(i)} = \mathbf{B}_{N-1-i} \begin{bmatrix} \mathbf{I}_{M/2} & \mathbf{0}_{M/2} & \mathbf{0}_{M/2} & \mathbf{0}_{M/2} \\ \mathbf{0}_{M/2} & \mathbf{0}_{M/2} & \mathbf{0}_{M/2} & \mathbf{I}_{M/2} \end{bmatrix} \begin{bmatrix} \mathbf{P}^{(i-1)} & \mathbf{0}_M \\ \mathbf{0}_M & \mathbf{P}^{(i-1)} \end{bmatrix} \tag{5.57}$$

$$\mathbf{P} = \mathbf{P}^{(N-1)} \, . \tag{5.58}$$

Similarly, one can find $\mathbf{Q}$ from the factors $\mathbf{B}_i^{-1}$.

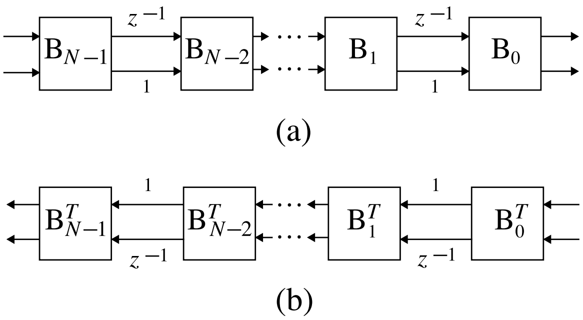


FIGURE 5.9
Flow graph for implementing an LT where $\mathbf{F}(z)$ can be factorized using symmetric delays and N stages. Signals $\{x(n)\}$ and $\{y(n)\}$ are segmented and processed using blocks of M samples, all branches carry $M/2$ samples, and blocks $\mathbf{B}_i$ are $M \times M$ orthogonal or invertible matrices. (a) Forward transform section, (b) inverse transform section.

5.5 Hierarchical Connection of LTs: An Introduction

So far we have focused on the construction of a single LT resulting in M subband signals. What happens if we cascade LTs by connecting them hierarchically in such a way that a subband signal is the actual input for another LT? Also, what are the consequences of submitting only part of the subband signals to further stages of LTs? We will try to introduce the answers to these questions.

This subject has been intensively studied and many publications are available. Our intent, however, is to provide only a basic introduction, while leaving more detailed analysis to the references. Again, the relation between filter banks and discrete wavelets [53, 63, 65] is well known. Under conditions that are easily satisfied [63], an infinite cascade of filter banks will generate a set of continuous orthogonal wavelet bases. In general, if only the lowpass subband is connected to another filter bank, for a finite number of stages, we call the resulting filter bank a discrete wavelet transform (DWT) [63, 65]. A free cascading of filter banks, however, is better known as discrete wavelet packet (DWP) [7, 68, 34, 53]. As LTs and filter banks are generally equivalent, the same relations apply to LTs and wavelets. The system resulting from the hierarchical association of several LTs is called a hierarchical lapped transform (HLT) [23].

5.5.1 Time-Frequency Diagram

The cascaded connection of LTs is better described with the aid of simplifying diagrams. The first is the time-frequency (TF) diagram. It is based on the TF plane, which is well known from the fields of spectral and time-frequency analysis [31, 3, 4]. The time-frequency representation of signals is a well-known method (for example the time-dependent discrete Fourier transform — DFT — and the construction of spectrograms; see for example [31, 3, 4] for details on TF signal representation). The TF representation is obtained by expressing the signal $\{x(n)\}$ with respect to bases which are functions of both frequency and time. For example, the size-r DFT of a sequence extracted from $\{x(n)\}$ (from $x(n)$ to $x(n+r-1)$) [31] can be

$$\alpha(k,n) = \sum_{i=0}^{r-1} x(i+n) \exp\left(-\frac{j2\pi ki}{r}\right) . \tag{5.59}$$

Using a sliding window $w(m)$ of length r which is nonzero only in the interval $n \leq m \leq n+r-1$ (which in this case is rectangular), we can rewrite the last equation as

$$\alpha(k,n) = \sum_{i=-\infty}^{\infty} x(i)w(i) \exp\left(-\frac{jk(i-n)2\pi}{r}\right) . \tag{5.60}$$

For more general bases we may write

$$\alpha(k,n) = \sum_{i=-\infty}^{\infty} x(i)\phi(n-i,k) \tag{5.61}$$

where $\phi(n,k)$ represents the bases for the space of the signal, n represents the index where the base is located in time, and k is the frequency index.

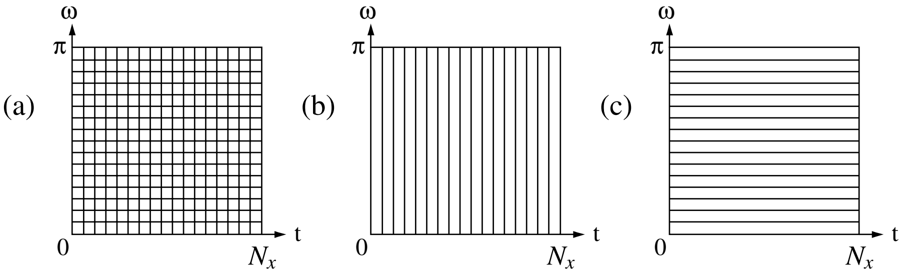


FIGURE 5.10
Examples of rectangular partitions of the time-frequency plane for a signal which has N_x samples. (a) Spectrogram with a N_x-length window, resulting in N_x^2 TF samples, (b) input signal, no processing, (c) a transform such as the DCT or DFT is applied to all N_x samples.

As the signal is assumed to have an infinite number of samples, consider a segment of N_x samples extracted from signal $\{x(n)\}$, which can be extended in any fashion in order to account for the overlap of the window of r samples outside the signal domain.

In such a segment we can construct a spectrogram with a resolution of r samples in the frequency axis and N_x samples in the time axis. Assuming a maximum frequency resolution, we can have a window with length up to $r = N_x$. The diagram for the spectrogram in this case is given in Fig. 5.10(a). We call such diagrams TF diagrams because they indicate only the number of samples used in the TF representation of the signal. Assuming an ideal partition of the TF plane (using filters with ideal frequency response and null transition regions), each TF coefficient represents a distinct region in a TF diagram. Note that in such representations, the signal is represented by N_x^2 TF coefficients. We are looking for a maximally decimated TF representation which is defined as a representation of the signal where the TF plane diagram would be partitioned into N_x regions; N_x TF coefficients will be generated. Also, we require that all N_x samples of $\{x(n)\}$ be able to be reconstructed from the N_x TF coefficients. If we use fewer than N_x samples in the TF plane, we clearly cannot reconstruct all possible combinations of samples in $\{x(n)\}$, from the TF coefficients, using solely linear relations.

Under these assumptions, Fig. 5.10(b) shows the TF diagram for the original signal (only resolution in the time axis) for $N_x = 16$. Also, for $N_x = 16$, Fig. 5.10(c) shows a TF diagram with maximum frequency resolution, which could be achieved by transforming the original N_x-sample sequence with an N_x-sample DCT or DFT.

5.5.2 Tree-Structured Hierarchical Lapped Transforms

The tree diagram is helpful in describing the hierarchical connection of filter banks. In this diagram we represent an M-band LT by nodes and branches of an M-ary tree. Fig. 5.11(a) shows an M-band LT, where all the M subband signals have sampling rates M times smaller than that of $\{x(n)\}$. Fig. 5.11(b) shows the equivalent notation for the LT in a tree diagram, i.e., a single-stage M-branch tree, which is called here a tree cell. Recalling Fig. 5.10, the equivalent TF diagram for an M-band LT is shown in Fig. 5.11(c), for a 16-sample signal and for $M = 4$. Note that the TF diagram of Fig. 5.11(c) resembles that of Fig. 5.10(a). This is because for each 4 samples in $\{x(n)\}$, there is a corresponding set of 4 transformed coefficients. So, the TF representation is maximally decimated. Compared to Fig. 5.10(b), Fig. 5.11(c) implies an exchange of resolution from time to frequency domain achieved by the LT.

The exchange of resolution in the TF diagram can be obtained from the LT. As we connect several LTs following the paths of a tree, each new set of branches (each new tree cell) connected to the tree will force the TF diagram to exchange from time to frequency resolution. We can achieve a more versatile TF representation by connecting cells in unbalanced ways. For example, Fig. 5.12 shows some examples of HLTs given by their tree diagrams and respective TF diagrams. Fig. 5.12(a) depicts the tree diagram for the 3-stages DWT. Note that only the lowpass subband is further processed. Also, as all stages are chosen to be 2-channel LTs, this HLT can be represented by a binary tree. Fig. 5.12(b) shows a more generic hierarchical connection of 2-channel LTs. First the signal is split into lowpass and highpass. Each output branch is further connected to another 2-channel LT. In the third stage, only the most

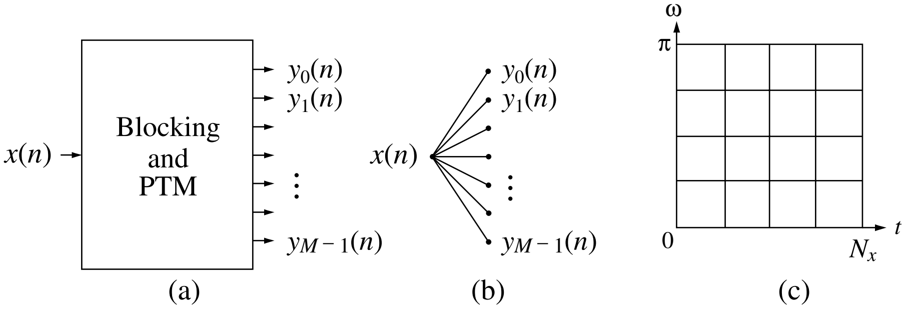


FIGURE 5.11

Representation of an M-channel LT as tree nodes and branches. (a) Forward section of an LT, including the blocking device. (b) Equivalent notation for (a) using an M-branch single-stage tree. (c) Equivalent TF diagram for (a) or (b) assuming $M = 4$ and $N_x = 16$.

lowpass subband signal is connected to another 2-channel LT. Fig. 5.12(c) shows a 2-stage HLT obtaining the same TF diagram as Fig. 5.12(b). Note that the succession of 2-channel LTs was substituted by a single stage 4-channel LT, i.e., the signal is split into four subbands and then one subband is connected to another LT. Fig. 5.12(d) shows the TF diagram corresponding to Fig. 5.12(a), while Fig. 5.12(e) shows the TF diagram corresponding to Figs. 5.12(b) and (c). The reader should note that, as the tree paths are unbalanced, we have irregular partitions of the TF plane. For example, in the DWT, low-frequency TF coefficients have poor time localization and good frequency resolution, while high-frequency ones have poor frequency resolution and better time localization.

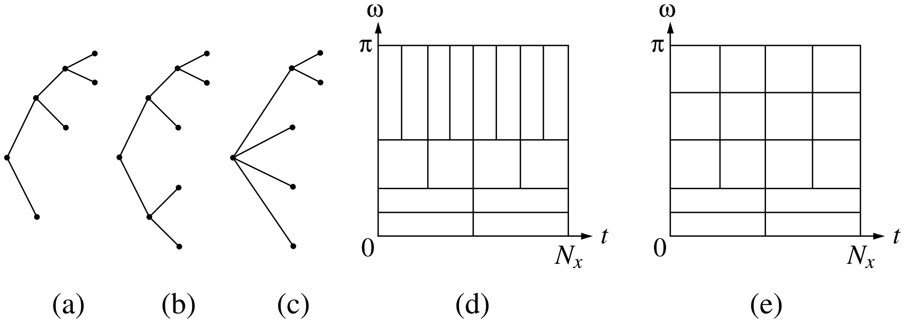


FIGURE 5.12

Tree and TF diagrams. (a) The 3-stage DWT binary-tree diagram, where only the lowpass subband is submitted to further LT stages. (b) A more generic 3-stage tree diagram. (c) A 2-stage tree-diagram resulting in the same TF diagram as (b). (d) TF diagram for (a). (e) TF diagram for (b) or (c).

To better understand how connecting an LT to the tree can achieve the exchange between time and frequency resolutions, see Fig. 5.13 which plots the basis functions resulting from two similar tree-structured HLTs.

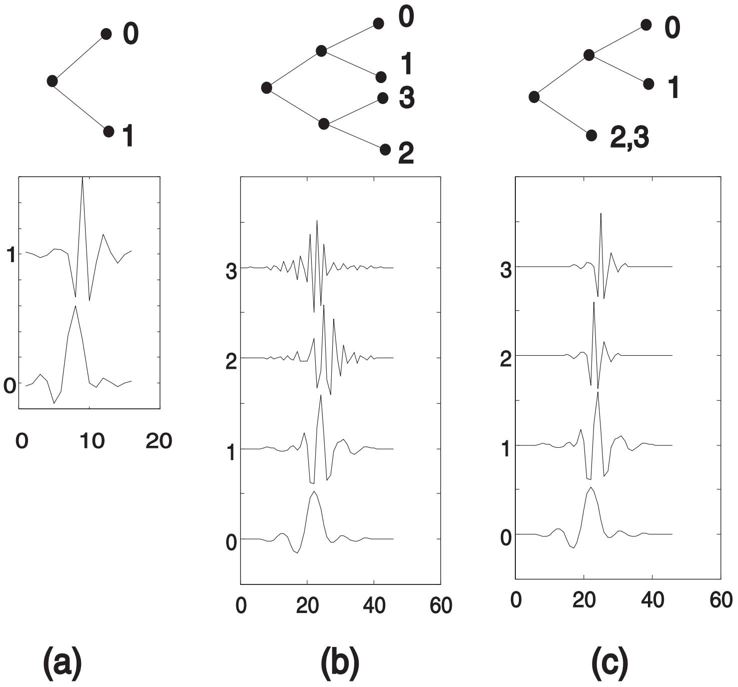


FIGURE 5.13
Two HLTs and resulting bases. (a) The 2-channel 16-tap-bases LT, showing low- and high-frequency bases, $f_0(n)$ and $f_1(n)$, respectively. (b) Resulting basis functions of a 2-stage HLT based on (a), given by $f_0(n)$ through $f_3(n)$. Its respective tree diagram is also shown. (c) Resulting HLT, by pruning one high-frequency branch in (b). Note that the two high-frequency basis functions are identical to the high-frequency basis function of (a), and, instead of having two distinct bases for high frequencies, occupying distinct spectral slots, the two bases are, now shifted in time. Thus, better time localization is attainable, at the expense of frequency resolution.

5.5.3 Variable-Length LTs

In the tree-structured method to cascade LTs, every time an LT is added to the structure, more subbands are created by further subdividing previous subbands, so the overall TF diagram of the decomposition is altered. There is a useful alternative to the tree structure in which the number of subbands does not change. See Fig. 5.14, where the "blocking" part of the diagram corresponds to the chain of delays and decimators (as in Fig. 5.8) that parallelizes the signal into polyphase components. System $\mathbf{A}(z)$ of M bases of length $N_A M$ is postprocessed by system $\mathbf{B}(z)$ of K bases

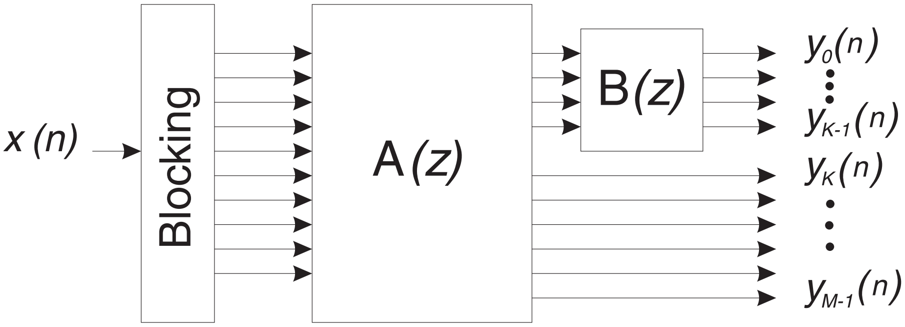


FIGURE 5.14

Cascadeof PTMsA (z) of M channelsandB (z) of K channels. Thetotalnumber of subbandsdoes notchange;however, some of A (z) basesare increasedin length andorder

of length $N_B K$. Clearly, entries in $\mathbf{A}(z)$ have order $N_A - 1$, and entries in $\mathbf{B}(z)$ have order $N_B - 1$. Without loss of generality we associate system $\mathbf{B}(z)$ to the first K output subbands of $\mathbf{A}(z)$. The overall PTM is given by

$$\mathbf{F}(z) = \begin{bmatrix} \mathbf{B}(z) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{M-K} \end{bmatrix} \mathbf{A}(z) \, , \qquad (5.62)$$

where $\mathbf{F}(z)$ has K bases of order $N_A + N_B - 2$ and $M - K$ bases of order $N_A - 1$. As the resulting LT has M channels, the final orders dictate that the first K bases have length $(N_A + N_B - 1)M$ while the others still have length $N_A M$. In other words, the effect of cascading $\mathbf{A}(z)$ and $\mathbf{B}(z)$ was only to modify K bases, so the length of the modified bases is equal to or larger than the length of the initial bases. An example is shown in Fig. 5.15. We start with the bases corresponding to $\mathbf{A}(z)$, shown in Fig. 5.15(a). There are 8 bases of length 16 so that $\mathbf{A}(z)$ has order 1. $\mathbf{A}(z)$ is postprocessed by $\mathbf{B}(z)$ which is a 4×4 PTM of order 3, whose corresponding bases are shown in Fig. 5.15(b). The resulting LT is shown in Fig. 5.15(c). There are 4 bases of length 16, and 4 of length 40. The shorter ones are identical to those in Fig. 5.15(a), while the longer ones have order which is the sum of the orders of $\mathbf{A}(z)$ and $\mathbf{B}(z)$, i.e., order 4, and the shape of the longer bases in $\mathbf{F}(z)$ is very different than the corresponding ones in $\mathbf{A}(z)$.

The effect of postprocessing some bases is a means to construct a new LT with larger bases from an initial one. In fact it can be shown that variable length LTs can be factorized using postprocessing stages [60, 59]. A general factorization of LTs is depicted in Fig. 5.16. Assume a variable-length $\mathbf{F}(z)$ whose bases are arranged in

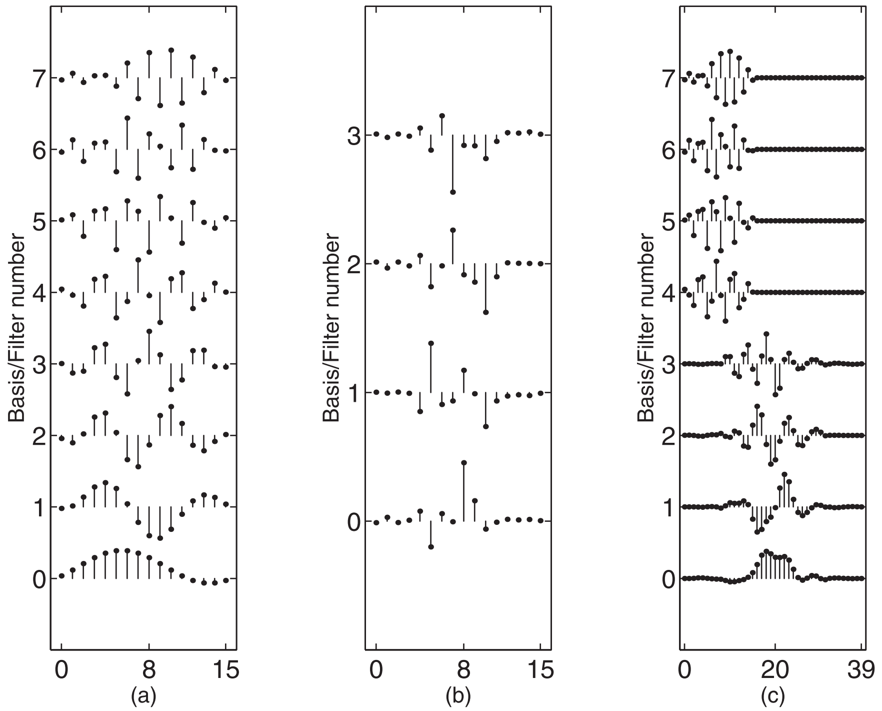


FIGURE 5.15

Example of constructing variable-length bases through cascading LTs. (a) The basis corresponding to $\mathbf{A}(z)$: an LT with 8 bases of length 16 (order 1). (b) The basis corresponding to $\mathbf{B}(z)$: an LT with 4 bases of length 16 (order 3). (c) The basis corresponding to $\mathbf{F}(z)$: 4 of the 8 bases have order 1, i.e., length 16, while the remaining 4 have order 4, i.e., length 40.

decreasing length order. Such a PTM can be factorized as

$$\mathbf{F}(z) = \prod_{i=0}^{M-2} \begin{bmatrix} \mathbf{B}_i(z) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_i \end{bmatrix} \tag{5.63}$$

where $\mathbf{I}_0$ is understood to be nonexisting and $\mathbf{B}_i(z)$ has size $(M-i) \times (M-i)$. The factors $\mathbf{B}_i$ can have individual orders K_i and can be factorized differently into factors $\mathbf{B}_{ik}(z)$ for $0 \leq k < K_i$. Hence,

$$\mathbf{F}(z) = \prod_{i=0}^{M-2} \prod_{k=0}^{K_i-1} \begin{bmatrix} \mathbf{B}_{ik}(z) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_i \end{bmatrix} . \tag{5.64}$$

A later section presents a very useful LT based on the factorization principles of Eq. (5.64).

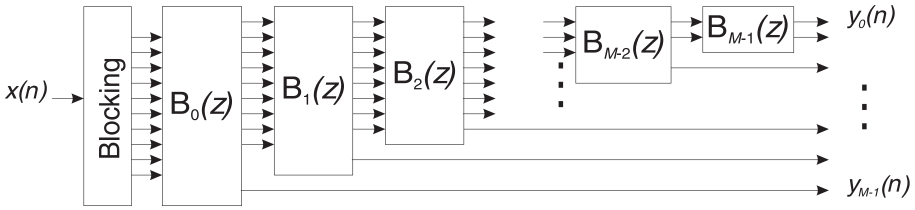


FIGURE 5.16
General factorization of a variable-length LT.

5.6 Practical Symmetric LTs

We have discussed LTs in a general sense as a function of several parameters, such as matrix entries, orthogonal, or invertible factors. The design of an LT suitable for a given application is the single most important step in the study of LTs. In order to do that, one may factorize the LT to facilitate optimization techniques.

An LT with symmetric bases is commonly used in image/video processing and compression applications. By symmetric bases we mean that the entries p_{ij} of $\mathbf{P}$ obey

$$p_{i,j} = (\pm 1)\ p_{i,L-1-j}\ . \tag{5.65}$$

The bases can be either symmetric or antisymmetric. In terms of the PTM, this constraint is given by

$$\mathbf{F}(z) = z^{-(N-1)}\mathbf{SF}\left(z^{-1}\right)\mathbf{J}_M\ , \tag{5.66}$$

where $\mathbf{S}$ is a diagonal matrix whose diagonal entries s_{ii} are ± 1, depending on whether the i-th basis is symmetric (+1) or anti-symmetric (-1). Note that we require that all bases share the same center of symmetry.

5.6.1 The Lapped Orthogonal Transform: LOT

The lapped orthogonal transform (LOT) [18, 19, 20] was the first useful LT with a well-defined factorization. Malvar developed the fast LOT based on the work by Cassereau [5] to provide not only a factorization, but a factorization based on the DCT. The DCT is attractive for many reasons, among them, fast implementation and near-optimal performance for block transform coding [45]. Also, since it is a popular transform, it has a reduced cost and is easily available in either software or hardware. The DCT matrix $\mathbf{D}$ is defined as having entries

$$d_{ij} = \sqrt{\frac{2}{M}}k_i \cos\left(\frac{(2j+1)i\pi}{2M}\right) \tag{5.67}$$

where $k_0 = 1$ and $k_i = 1/\sqrt{2}$, for $1 \leq i \leq M-1$.

The LOT as defined by Malvar [20] is orthogonal. Then, according to our notation, $\mathbf{P} = \mathbf{Q}$ and $\mathbf{H}^{-1} = \mathbf{H}^T$. It is also a symmetric LT with M even. The LT matrix is given by

$$\mathbf{P}_{LOT} = \begin{bmatrix} \mathbf{I}_M & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_R \end{bmatrix} \begin{bmatrix} \mathbf{D}_e - \mathbf{D}_o & \mathbf{J}_{M/2}\left(\mathbf{D}_e - \mathbf{D}_o\right) \\ \mathbf{D}_e - \mathbf{D}_o & -\mathbf{J}_{M/2}\left(\mathbf{D}_e - \mathbf{D}_o\right) \end{bmatrix} \tag{5.68}$$

where $\mathbf{D}_e$ is the $M/2 \times M$ matrix with the even-symmetric basis functions of the DCT, and $\mathbf{D}_o$ is the matrix of the same size with the odd-symmetric ones. In our notation, $\mathbf{D}_e$ also corresponds to the even numbered rows of $\mathbf{D}$, and $\mathbf{D}_o$ corresponds to the odd numbered rows of $\mathbf{D}$. $\mathbf{V}_R$ is an $M/2 \times M/2$ orthogonal matrix, which according to Malvar and Staelin [20] and Malvar [26] should be approximated by $M/2 - 1$ plane rotations as

$$\mathbf{V}_R = \prod_{i=\frac{M}{2}-2}^{0} \mathbf{\Theta}\left(i, i+1, \theta_i\right) \tag{5.69}$$

where $\mathbf{\Theta}$ is defined in Section 5.1.4. Suggestions of rotation angles that were designed to yield a good transform for image compression are [26]

$$M = 4 \quad \rightarrow \quad \theta_0 = 0.1\pi \tag{5.70}$$

$$M = 8 \quad \rightarrow \quad \{\theta_0, \theta_1, \theta_2\} = \{0.13, 0.16, 0.13\} \times \pi \tag{5.71}$$

$$\begin{aligned} M = 16 \quad \rightarrow \quad & \{\theta_0, \ldots, \theta_7\} \\ & = \{0.62, 0.53, 0.53, 0.50, 0.44, 0.35, 0.23, 0.11\} \times \pi \,. \end{aligned} \tag{5.72}$$

For $M \geq 16$ it is suggested to use

$$\mathbf{V}_R = \mathbf{D}_{IV}^T \mathbf{D}^T \tag{5.73}$$

where $\mathbf{D}_{IV}$ is the DCT type IV matrix [45] whose entries are

$$d_{ij}^{IV} = \sqrt{\frac{2}{M}} \cos\left(\frac{(2j+1)(2i+1)\pi}{4M}\right) \,. \tag{5.74}$$

A block diagram for the implementation of the LOT is shown in Fig. 5.17 for $M = 8$.

5.6.2 The Lapped Bi-Orthogonal Transform: LBT

The LOT is a great improvement over the DCT for image compression mainly because it reduces the so-called blocking effects. Although largely reduced, blocking is not eliminated because the format of the low frequency bases of LOT. In image compression, only a few bases are used to reconstruct the signal. From Fig. 5.4, one can see that the "tails" of the lower frequency bases of the LOT do not decay exactly

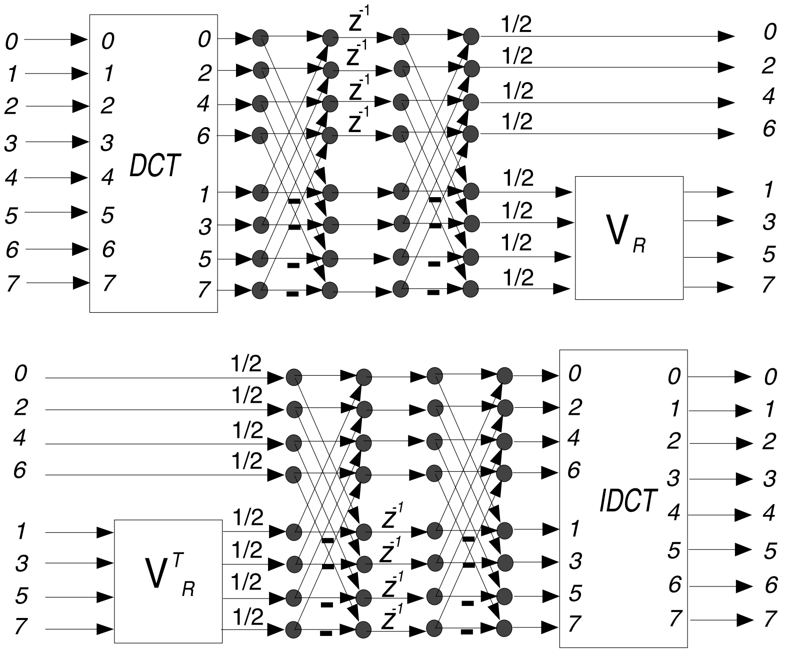


FIGURE 5.17
Implementation of the LOT for $M = 8$. Top: forward transform; bottom: inverse transform.

to zero. For this reason, there is some blocking effect left in images compressed using the LOT at lower bit rates.

To help resolve this problem, Malvar recently proposed to modify LOT, creating the lapped bi-orthogonal transform (LBT) [28]. (Bi-orthogonal is a common term in the filter banks community to designate PR transforms and filter banks that are not orthogonal.) In any case, the factorization of the LBT is almost identical to that of the LOT. However,

$$\mathbf{P}_{\mathrm{LBT}} = \begin{bmatrix} \mathbf{I}_M & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_R \end{bmatrix} \begin{bmatrix} \mathbf{D}_e - \boldsymbol{\Upsilon}\mathbf{D}_o & \mathbf{J}_{M/2}\left(\mathbf{D}_e - \boldsymbol{\Upsilon}\mathbf{D}_o\right) \\ \mathbf{D}_e - \boldsymbol{\Upsilon}\mathbf{D}_o & -\mathbf{J}_{M/2}\left(\mathbf{D}_e - \boldsymbol{\Upsilon}\mathbf{D}_o\right) \end{bmatrix} \tag{5.75}$$

where $\boldsymbol{\Upsilon}$ is the $M/2 \times M/2$ diagonal matrix given by $\boldsymbol{\Upsilon} = diag\{\sqrt{2}, 1, \ldots, 1\}$. Note that it implies only that one of the DCT's output is multiplied by a constant. The inverse is given by the LT $\mathbf{Q}_{LBT}$ which is found in an identical manner as in Eq. (5.75) except that the multiplier is inverted; $\boldsymbol{\Upsilon} = diag\{1/\sqrt{2}, 1, \ldots, 1\}$. The diagram for implementing an LBT for $M = 8$ is shown in Fig. 5.18.

Because of the multiplicative factor, the LT is no longer orthogonal. However, the factor is very easily inverted. The result is a reduction of amplitude of lateral samples of the first bases of LOT into the new bases of the forward LBT, as seen in Fig. 5.19. In Fig. 5.19 the reader can note the reduction in the amplitude of the boundary samples

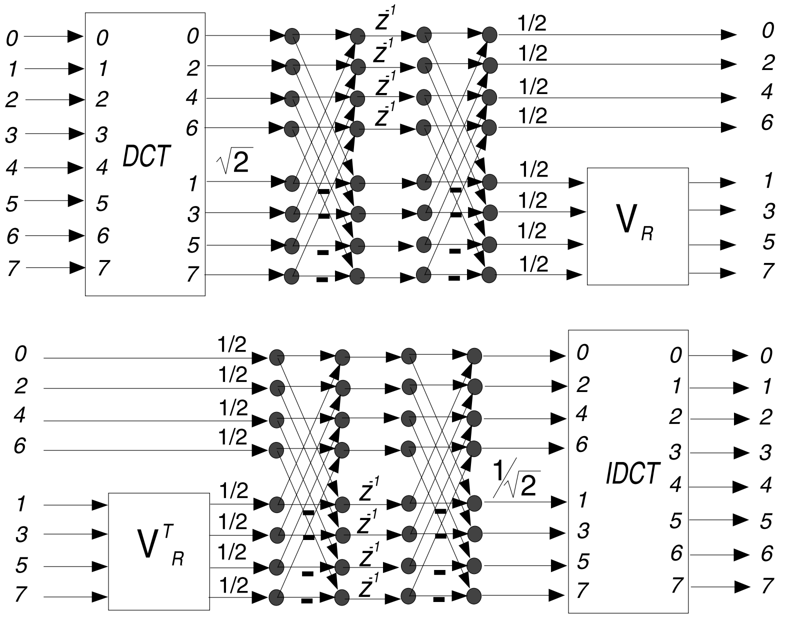


FIGURE 5.18
Implementation of the LBT for $M = 8$. Top: forward transform; bottom: inverse transform. Note that there is only one extra multiplication as compared to LOT.

of the inverse LBT and an enlargement of the same samples in the forward LBT. This simple "trick" noticeably improves the performance of the LOT/LBT for image compression at negligible overhead. Design of the other parameters of the LOT are not changed. It is recommended to use LBT instead of LOT whenever orthogonality is not a crucial constraint.

Another LBT with high practical value is LiftLT [62]. Instead of parameterizing the orthogonal matrix $\mathbf{V}_R$ in Eq. (5.69) by rotation angles, a series of dyadic lifting steps are used to construct $\mathbf{V}_R$, as shown in Fig. 5.20. The $\mathbf{V}_R$ matrix is still invertible, but no longer orthogonal. Hence, the LiftLT is a bi-orthogonal LT. The $\sqrt{2}$ factor in the Malvar's LBT can be replaced by a rational number to facilitate finite-precision implementations. A good rational scaling factor is $\frac{25}{16}$ for the forward transform, and $\frac{16}{25}$ for the inverse transform. The LiftLT offers a VLSI-friendly implementation using integer (even binary) arithmetic. Yet, it does not sacrifice anything in coding performance. It achieves 9.54 dB coding gain (a popular objective measure of transform performance to be described later), compared to LOT's 9.20 dB and LBT's 9.52 dB. It is the first step towards LTs that can map integers to integers and multiplierless LTs that can be implemented using only shift-and-add operations.

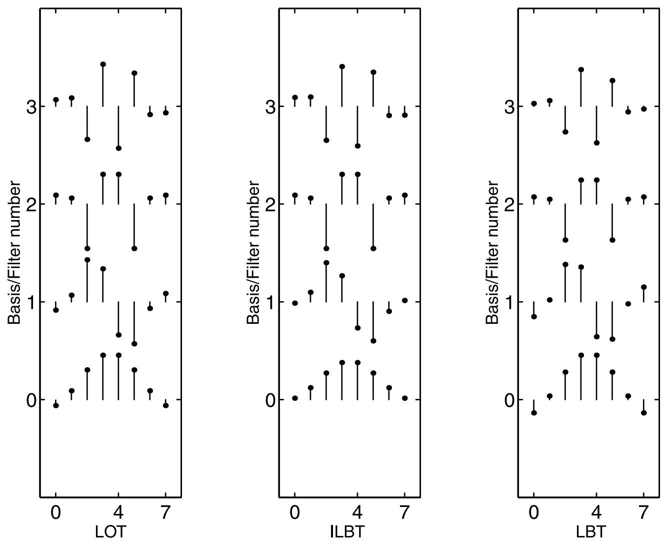


FIGURE 5.19
Comparison of bases for the LOT ($\mathbf{P}_{LOT}$), inverse LBT ($\mathbf{Q}_{LBT}$), and forward LBT ($\mathbf{P}_{LBT}$). The extreme samples of the lower frequency bases of the LOT are larger than those of the inverse LBT. This is an advantage for image compression.

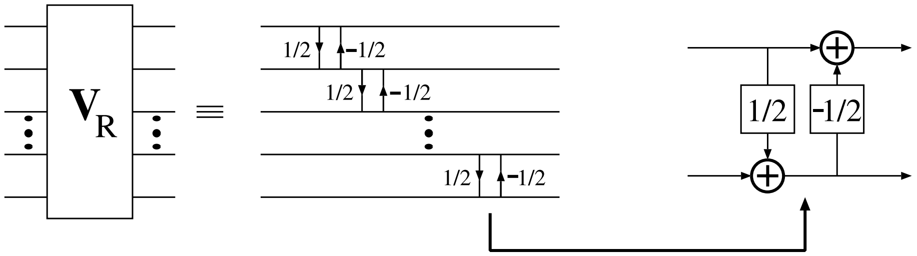


FIGURE 5.20
Parameterization of the $\mathbf{V}_R$ matrix using dyadic lifting steps in the LiftLT.

5.6.3 The Generalized LOT: GenLOT

The formulation for LOT [20] that is shown in Eq. (5.68) is not the most general there is for this kind of LT. In fact, it can be generalized to become

$$\mathbf{P} = \begin{bmatrix} \mathbf{U} & \mathbf{0} \\ \mathbf{0} & \mathbf{V} \end{bmatrix} \begin{bmatrix} \mathbf{D}_e - \mathbf{D}_o & \mathbf{J}_{M/2}\left(\mathbf{D}_e - \mathbf{D}_o\right) \\ \mathbf{D}_e - \mathbf{D}_o & -\mathbf{J}_{M/2}\left(\mathbf{D}_e - \mathbf{D}_o\right) \end{bmatrix} . \qquad (5.76)$$

As long as **U** and **V** remain orthogonal matrices, LT is orthogonal. In terms of the PTM, $\mathbf{F}(z)$ can be similarly expressed. Let

$$\begin{aligned} \mathbf{W} &= \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I}_{M/2} & \mathbf{I}_{M/2} \\ \mathbf{I}_{M/2} & -\mathbf{I}_{M/2} \end{bmatrix}, & (5.77) \\ \mathbf{\Phi}_i &= \begin{bmatrix} \mathbf{U}_i & \mathbf{0}_{M/2} \\ \mathbf{0}_{M/2} & \mathbf{V}_i \end{bmatrix}, & (5.78) \\ \mathbf{\Lambda}(z) &= \begin{bmatrix} \mathbf{I}_{M/2} & \mathbf{0}_{M/2} \\ \mathbf{0}_{M/2} & z^{-1}\mathbf{I}_{M/2} \end{bmatrix}, & (5.79) \end{aligned}$$

and let **D** be the $M \times M$ DCT matrix. Then, for the general LOT,

$$\mathbf{F}(z) = \mathbf{\Phi}_1 \mathbf{W} \mathbf{\Lambda}(z) \mathbf{W} \mathbf{D} \, . \tag{5.80}$$

Where $\mathbf{U}_1 = \mathbf{U}$ and $\mathbf{V}_1 = -\mathbf{V}$. Note that the regular LOT is the case where $\mathbf{U}_1 = \mathbf{I}_{M/2}$ and $\mathbf{V}_1 = -\mathbf{V}_R$. The implementation diagram for $M = 8$ is shown in Fig. 5.21.

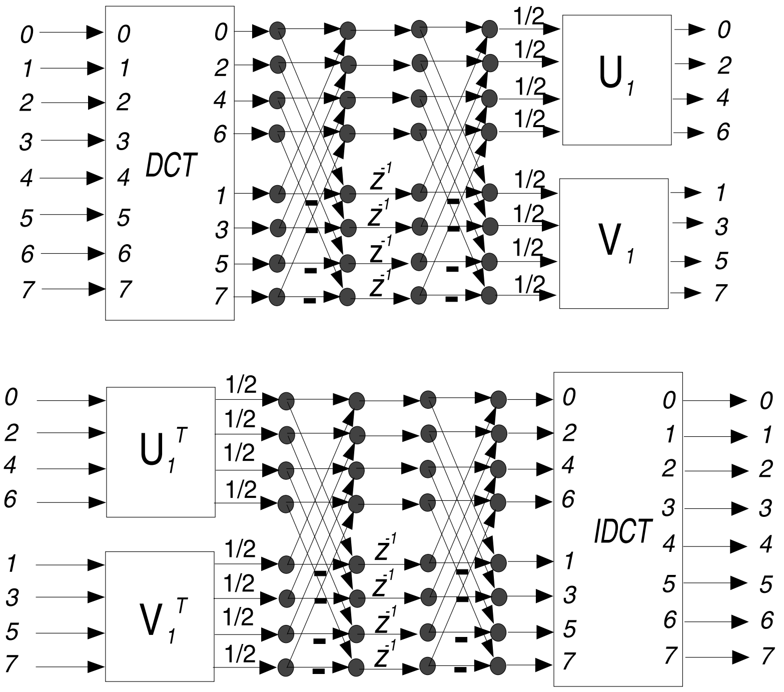


FIGURE 5.21
Implementation of a more general version of the LOT for $M = 8$. Top: forward transform; bottom: inverse transform.

From this formulation, along with other results, it was realized [40] that all orthogonal symmetric LTs can be expressed as

$$\mathbf{F}(z) = \mathbf{K}_{N-1}(z)\mathbf{K}_{N-2}(z) \cdots \mathbf{K}_1(z)\mathbf{K}_0 \tag{5.81}$$

where

$$\mathbf{K}_i(z) = \mathbf{\Phi}_i \mathbf{W} \mathbf{\Lambda}(z) \mathbf{W} \; , \tag{5.82}$$

and where $\mathbf{K}_0$ is any orthogonal symmetric matrix. The inverse is given by

$$\mathbf{G}(z) = \mathbf{K}_0^T \mathbf{K}_1'(z) \mathbf{K}_2'(z) \cdots \mathbf{K}_{N-1}'(z) \tag{5.83}$$

where

$$\mathbf{K}_i'(z) = z^{-1} \mathbf{W} \mathbf{\Lambda} \left(z^{-1} \right) \mathbf{W} \mathbf{\Phi}_i^T \; . \tag{5.84}$$

From this perspective, the generalized LOT (GenLOT) is defined as the orthogonal LT, as in Eq. (5.81) in which $\mathbf{K}_0 = \mathbf{D}$; i.e.,

$$\mathbf{F}(z) = \mathbf{K}_{N-1}(z) \cdots \mathbf{K}_1(z) \mathbf{D} \; . \tag{5.85}$$

A diagram for implementing a GenLOT for even M is shown in Fig. 5.22. In this diagram, the scaling parameters are $\beta = 2^{-(N-1)}$ and account for the terms $1/\sqrt{2}$ in the definition of $\mathbf{W}$.

The degrees of freedom of a GenLOT are the orthogonal matrices $\mathbf{U}_i$ and $\mathbf{V}_i$. There are $2(N-1)$ matrices to optimize, each of size $M/2 \times M/2$. From Section 5.1.4 we know that each one can be factorized into $M(M-2)/8$ rotations. Thus, the total number of rotations is $(L-M)(M-2)/4$, which is less than the initial number of degrees of freedom in a symmetric $M \times L$ matrix, $LM/2$. However, it is still a large number of parameters to design. In general, GenLOTs are designed through nonlinear unconstrained optimization. Rotation angles are searched to minimize some cost function. GenLOT examples are given elsewhere [40], and we present two examples, for $M = 8$, in Tables 5.1 and 5.2, which are also plotted in Fig. 5.23.

In the case when M is odd, the GenLOT is defined as

$$\mathbf{F}(z) = \mathbf{K}_{(N-1)/2}(z) \cdots \mathbf{K}_1(z) \mathbf{D} \tag{5.86}$$

where the stages $\mathbf{K}_i$ have necessarily order 2 as:

$$\mathbf{K}_i(z) = \mathbf{\Phi}_{2i}^o \mathbf{W}^o \mathbf{\Lambda}^{o1}(z) \mathbf{W}^o \mathbf{\Phi}_{2i-1}^o \mathbf{W}^o \mathbf{\Lambda}^{o2}(z) \mathbf{W}^o \tag{5.87}$$

and where

$$\mathbf{\Phi}_{2i}^o = \begin{bmatrix} \mathbf{U}_{2i} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{2i} \end{bmatrix} , \tag{5.88}$$

$$\mathbf{\Phi}_{2i-1}^o = \begin{bmatrix} \mathbf{U}_{2i-1} & & \mathbf{0} \\ & 1 & \\ \mathbf{0} & & \mathbf{V}_{2i-1} \end{bmatrix} , \tag{5.89}$$

$$\mathbf{W}^o = \begin{bmatrix} \mathbf{I}_{(M-1)/2} & \mathbf{0}_{(M-1)/2 \times 1} & \mathbf{I}_{(M-1)/2} \\ \mathbf{0}_{1 \times (M-1)/2} & 1 & \mathbf{0}_{1 \times (M-1)/2} \\ \mathbf{I}_{(M-1)/2} & \mathbf{0}_{(M-1)/2 \times 1} & -\mathbf{I}_{(M-1)/2} \end{bmatrix} , \tag{5.90}$$

$$\mathbf{\Lambda}^{o1}(z) = diag \left\{ \underbrace{1, 1, \ldots, 1}_{(M+1)/2 \times 1's}, \underbrace{z^{-1}, \ldots, z^{-1}}_{(M-1)/2 \times z^{-1}} \right\} , \tag{5.91}$$

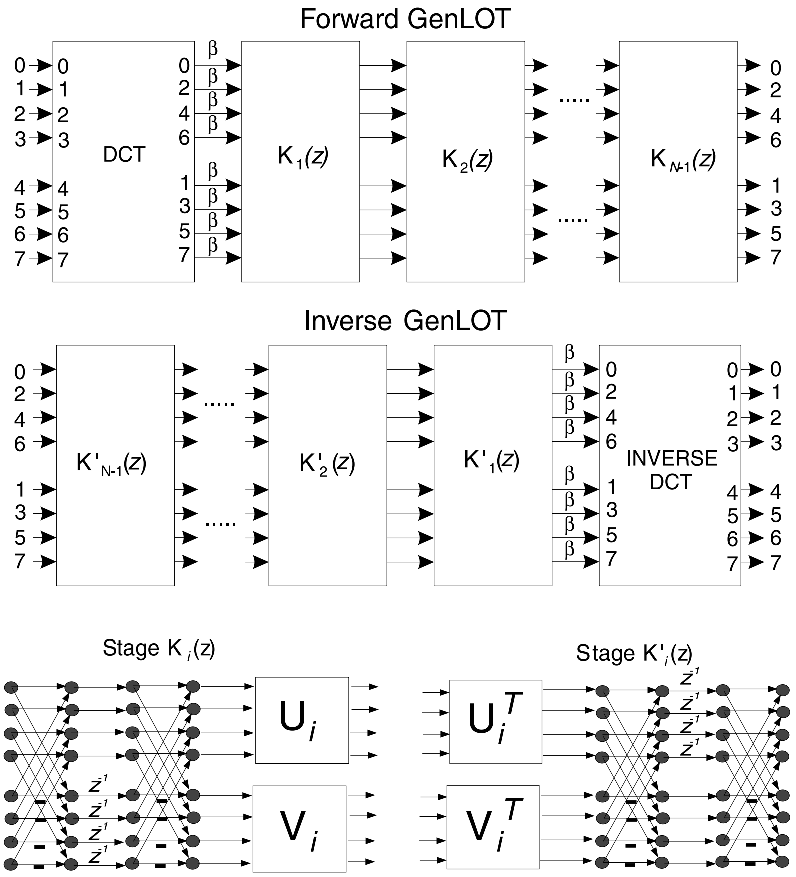


FIGURE 5.22
Implementation of a GenLOT for even M, ($M = 8$). Forward and inverse transforms are shown along with details of each stage. $\beta = 2^{-(N-1)}$ accounts for all terms of the form $1/\sqrt{2}$ which make the butterflies (W) orthogonal.

$$\mathbf{\Lambda}^{o2}(z) \quad = \quad diag\left\{\underbrace{1,1,\ldots,1}_{(M-1)/2\times 1's},\underbrace{z^{-1},\ldots,z^{-1}}_{(M+1)/2\times z^{-1}}\right\} . \qquad (5.92)$$

Although it may seem that the formulation of the odd-channel case is more complex than the one for the even-M case, the implementation is very similar in complexity as shown in Fig. 5.24. The main difference is that two stages have to be connected. The inverse transform is accomplished in the same way as for the even channel case:

$$\mathbf{G}(z) = \mathbf{D}^T\mathbf{K}'_1(z)\mathbf{K}'_2(z)\cdots\mathbf{K}'_{N-1}(z) \qquad (5.93)$$

Table 5.1 GenLOT Example for $N = 4$. The Even Bases are Symmetric while the Odd Ones are Anti-Symmetric, so Only their First Half is Shown

p_{0n}	p_{1n}	p_{2n}	p_{3n}	p_{4n}	p_{5n}	p_{6n}	p_{7n}
0.004799	0.004829	0.002915	−0.002945	0.000813	−0.000109	0.000211	0.000483
0.009320	−0.000069	−0.005744	−0.010439	0.001454	0.003206	0.000390	−0.001691
0.006394	−0.005997	−0.011121	−0.010146	0.000951	0.004317	0.000232	−0.002826
−0.011794	−0.007422	−0.001800	0.009462	−0.001945	−0.001342	−0.000531	0.000028
−0.032408	−0.009604	0.008083	0.031409	−0.005262	−0.007504	−0.001326	0.003163
−0.035122	−0.016486	0.001423	0.030980	−0.005715	−0.006029	−0.001554	0.001661
−0.017066	−0.031155	−0.027246	0.003473	−0.003043	0.005418	−0.000789	−0.005605
0.000288	−0.035674	−0.043266	−0.018132	−0.000459	0.013004	−0.000165	−0.010084
−0.012735	−0.053050	0.007163	−0.083325	0.047646	0.011562	0.048534	0.043066
−0.018272	−0.090207	0.131531	0.046926	0.072761	−0.130875	−0.089467	−0.028641
0.021269	−0.054379	0.109817	0.224818	−0.224522	0.136666	0.022488	−0.025219
0.126784	0.112040	−0.123484	−0.032818	−0.035078	0.107446	0.147727	0.109817
0.261703	0.333730	−0.358887	−0.379088	0.384874	−0.378415	−0.339368	−0.216652
0.357269	0.450401	−0.292453	−0.126901	−0.129558	0.344379	0.439129	0.317070
0.383512	0.369819	0.097014	0.418643	−0.419231	0.045807	−0.371449	−0.392556
0.370002	0.140761	0.478277	0.318691	0.316307	−0.433937	0.146036	0.427668

where the inverse factors are

$$\mathbf{K}_i'(z) = z^{-2}\mathbf{K}_i^T\left(z^{-1}\right) , \tag{5.94}$$

whose structure is evident from Fig. 5.24.

5.6.4 The General Factorization: GLBT

The general factorization for all symmetric LTs [61] can be viewed either as an extension of GenLOTs or as a generalization of the LBT. For M even, all LTs obeying Eq. (5.65) or Eq. (5.66) can be factorized as in Eq. (5.81), where the $\mathbf{K}_i(z)$ factors are given in Eq. (5.82) with the matrices $\mathbf{U}_i$ and $\mathbf{V}_i$ (which compose $\mathbf{\Phi}_i$) being required only to be general invertible matrices. From Section 5.1.4, each factor can be decomposed as

$$\mathbf{U}_i = \mathbf{U}_{iB}\mathbf{U}_{id}\mathbf{U}_{iA} \quad , \quad \mathbf{V}_i = \mathbf{V}_{iB}\mathbf{V}_{id}\mathbf{V}_{iA} \; , \tag{5.95}$$

where $\mathbf{U}_{iA}$, $\mathbf{U}_{iB}$, $\mathbf{V}_{iA}$, and $\mathbf{V}_{iB}$ are general $M/2 \times M/2$ orthogonal matrices, while $\mathbf{U}_{id}$ and $\mathbf{V}_{id}$ are diagonal matrices with nonzero diagonal entries.

The first factor $\mathbf{K}_0$ is given by

$$\mathbf{K}_0 = \mathbf{\Phi}_0\mathbf{W} \; , \tag{5.96}$$

where $\mathbf{\Phi}_i$ is given as in Eq. (5.78), and factors $\mathbf{U}_0$ and $\mathbf{V}_0$ are required only to be invertible. The general factorization can be viewed as a generalized LBT (GLBT) and its implementation flow graph for M is even shown in Fig. 5.25.

Table 5.2 GenLOT Example for $N = 6$. The Even Bases are Symmetric while the Odd Ones are Anti-Symmetric, so Only their First Half is Shown

p_{0n}	p_{1n}	p_{2n}	p_{3n}	p_{4n}	p_{5n}	p_{6n}	p_{7n}
−0.000137	−0.000225	0.000234	0.000058	−0.000196	−0.000253	0.000078	0.000017
−0.000222	−0.000228	0.000388	0.000471	0.000364	0.000163	−0.000220	−0.000283
0.001021	0.000187	0.002439	0.001211	−0.000853	−0.002360	0.000157	−0.000823
0.000536	0.000689	0.000029	0.000535	0.000572	0.000056	0.000633	0.000502
−0.001855	0.000515	−0.006584	−0.002809	0.003177	0.006838	−0.000886	0.001658
0.001429	0.001778	−0.000243	0.000834	0.000977	−0.000056	0.001687	0.001429
0.001440	0.001148	0.000698	0.000383	0.000109	−0.000561	−0.000751	−0.001165
0.001056	0.001893	0.002206	0.005386	0.005220	0.001676	0.001673	0.000792
0.009734	0.002899	0.018592	0.004888	−0.006600	−0.018889	−0.000261	−0.006713
−0.005196	−0.013699	−0.008359	−0.021094	−0.020406	−0.009059	−0.012368	−0.005263
−0.000137	−0.001344	−0.027993	−0.028046	0.026048	0.024169	−0.001643	−0.000402
−0.007109	−0.002130	0.002484	0.013289	0.013063	0.002655	−0.002180	−0.006836
−0.011238	−0.002219	0.033554	0.062616	−0.058899	−0.031538	−0.001404	0.004060
−0.020287	−0.006775	0.003214	0.019082	0.018132	0.004219	−0.006828	−0.019040
−0.028214	−0.018286	−0.059401	−0.023539	0.024407	0.056646	0.009849	0.021475
−0.034379	−0.055004	−0.048827	−0.052703	−0.051123	−0.048429	−0.049853	−0.031732
−0.029911	−0.106776	0.070612	−0.088796	0.086462	−0.066383	0.097006	0.031014
−0.004282	−0.107167	0.197524	0.049701	0.051188	0.193302	−0.104953	−0.006324
0.058553	−0.026759	0.144748	0.241758	−0.239193	−0.143627	0.020370	−0.048085
0.133701	0.147804	−0.123524	0.026563	0.025910	−0.125263	0.147501	0.130959
0.231898	0.330343	−0.376982	−0.365965	0.366426	0.377886	−0.332858	−0.228016
0.318102	0.430439	−0.312564	−0.174852	−0.174803	−0.314092	0.431705	0.317994
0.381693	0.368335	0.061832	0.393949	−0.395534	−0.060887	−0.369244	−0.384842
0.417648	0.144412	0.409688	0.318912	0.319987	0.411214	0.145256	0.419936

The inverse GLBT is similar to the GenLOT case, where

$$\mathbf{K}_i'(z) = z^{-1}\mathbf{W}\boldsymbol{\Lambda}(z)\mathbf{W}\boldsymbol{\Phi}_i^{-1} \tag{5.97}$$

and

$$\boldsymbol{\Phi}_i^{-1} = \begin{bmatrix} \mathbf{U}_i^{-1} & \mathbf{0}_{M/2} \\ \mathbf{0}_{M/2} & \mathbf{V}_i^{-1} \end{bmatrix} = \begin{bmatrix} \mathbf{U}_{iA}^T\mathbf{U}_{id}^{-1}\mathbf{U}_{iB}^T & \mathbf{0}_{M/2} \\ \mathbf{0}_{M/2} & \mathbf{V}_{iA}^T\mathbf{V}_{id}^{-1}\mathbf{V}_{iB}^T \end{bmatrix} \tag{5.98}$$

while

$$\mathbf{K}_0^{-1} = \mathbf{W}\boldsymbol{\Phi}_0^{-1} \, . \tag{5.99}$$

The diagram for the implementation of the inverse stages of the GLBT is shown in Fig. 5.25. Examples of bases for the GLBT of particular interest to image compression are given in Tables 5.3 and 5.4.

For the odd case, the GLBT can be similarly defined. It follows the GenLOT factorization:

$$\mathbf{F}(z) = \mathbf{K}_{(N-1)/2}(z) \cdots \mathbf{K}_1(z)\mathbf{K}_0 \tag{5.100}$$

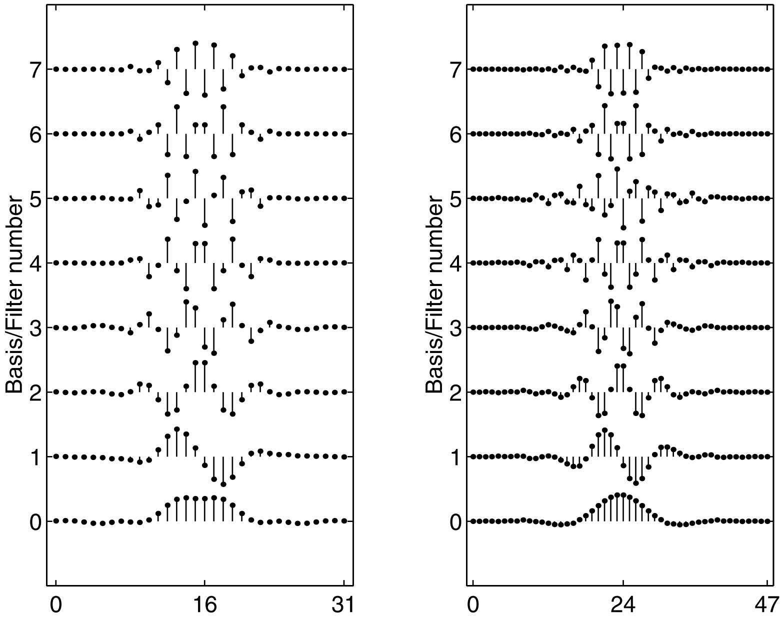


FIGURE 5.23
Example of optimized GenLOT bases for $M = 8$ and for $N = 4$ (left) and $N = 6$ (right).

Table 5.3 Forward GLBT Bases Example for $M = 8$ and $N = 2$. The Even Bases are Symmetric while the Odd Ones are Anti-Symmetric, so Only their First Half is Shown

p_{0n}	p_{1n}	p_{2n}	p_{3n}	p_{4n}	p_{5n}	p_{6n}	p_{7n}
−0.21192	−0.18197	0.00011	−0.09426	0.03860	−0.03493	0.04997	0.01956
−0.13962	−0.19662	0.16037	0.05334	0.09233	0.12468	−0.09240	−0.03134
−0.03387	−0.09540	0.17973	0.25598	−0.24358	−0.12311	0.01067	−0.01991
0.09360	0.10868	−0.06347	−0.01332	−0.05613	−0.10218	0.16423	0.11627
0.23114	0.34101	−0.36293	−0.39498	0.42912	0.36084	−0.35631	−0.22434
0.35832	0.46362	−0.35056	−0.16415	−0.13163	−0.31280	0.47723	0.31907
0.46619	0.42906	0.00731	0.42662	−0.45465	−0.07434	−0.40585	−0.38322
0.53813	0.22604	0.42944	0.36070	0.32595	0.43222	0.15246	0.39834

where the stages $\mathbf{K}_i$ are as in Eq. (5.87) with the following differences: (i) all factors $\mathbf{U}_i$ and $\mathbf{V}_i$ are only required to be invertible; (ii) the center element of $\mathbf{\Phi}_{2i-1}$ is a nonzero constant u_0 and not 1. Again $\mathbf{K}_0$ is a symmetric invertible matrix. Forward and inverse stages for the odd-channel case are illustrated in Fig. 5.26.

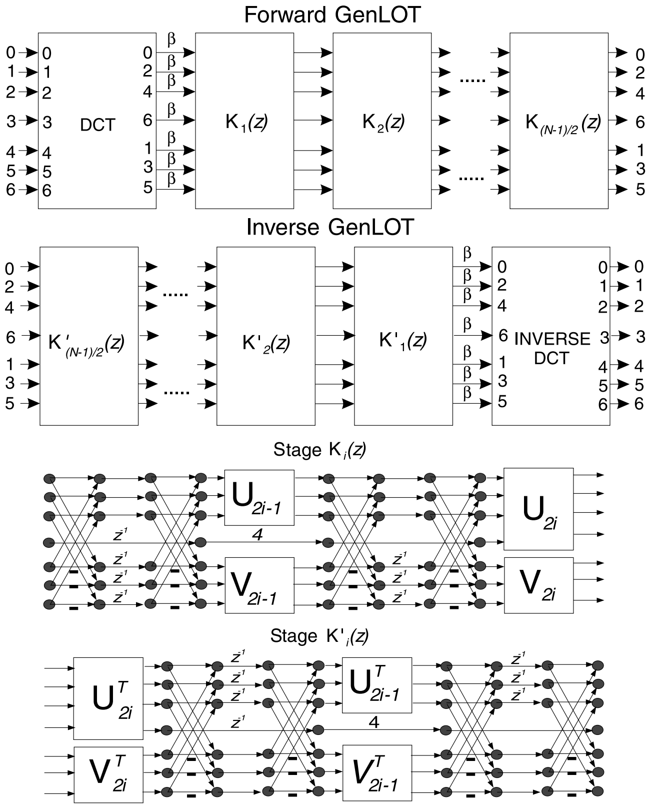


FIGURE 5.24
Implementation of a GenLOT for M odd. Forward and inverse transforms are shown along with details of each stage and $\beta = 2^{-(N-1)}$.

5.7 The Fast Lapped Transform: FLT

The motivation behind the fast lapped transform (FLT) is to design an LT with minimum possible complexity compared to a block transform, yet to provide some advantages over a block transform. For that we use the principles of Section 5.5.3 and define the FLT as the LT whose PTM is given by

$$\mathbf{F}(z) = \begin{bmatrix} \mathbf{E}(z) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{M-K} \end{bmatrix} \mathbf{D}_M \tag{5.101}$$

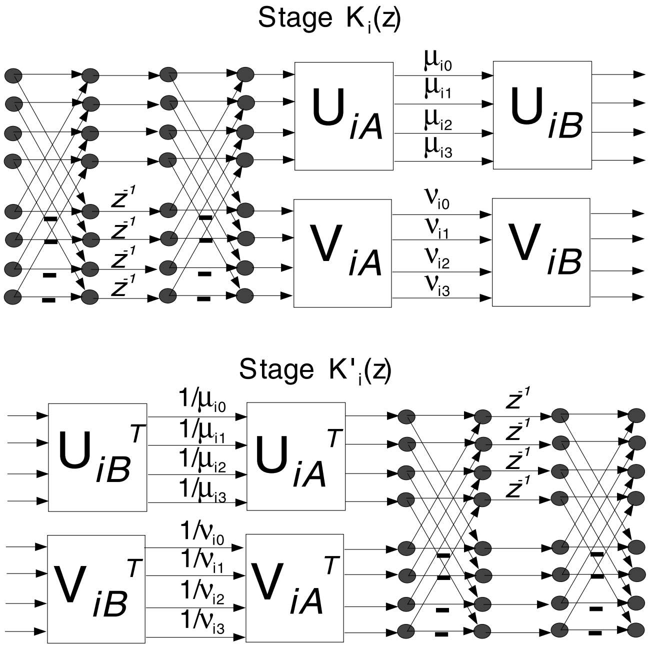


FIGURE 5.25

Implementation of the factors of the general factorization (GLBT) for M even. Top: factor of the forward transform, $\mathbf{K}_i(z)$. Bottom: factor of the inverse transform, $\mathbf{K}'_i(z)$.

where $\mathbf{E}(z)$ is a $K \times K$ PTM and $\mathbf{D}_M$ is the $M \times M$ DCT matrix. The PTM for the inverse LT is given by

$$\mathbf{G}(z) = \mathbf{D}_M^T \begin{bmatrix} \mathbf{E}'(z) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{M-K} \end{bmatrix}, \tag{5.102}$$

where $\mathbf{E}'(z)$ is the inverse of $\mathbf{E}(z)$.

The design of $\mathbf{E}(z)$ can be done in two basic ways. Firstly, one can use direct optimization. Secondly, one can design $\mathbf{E}(z)$ as

$$\mathbf{E}(z) = \mathbf{\Psi}(z)\mathbf{D}_K^T \tag{5.103}$$

where $\mathbf{\Psi}(z)$ is a known LT and $\mathbf{D}_K$ is the $K \times K$ DCT matrix; in other words, we perform an inverse DCT followed by a known LT. For example, if $\mathbf{\Psi}(z)$ is the LOT, GenLOT, or LBT of K channels, the first stage ($\mathbf{D}_K$) cancels the inverse DCT. Examples of FLT are given in Fig. 5.27. In that example, the first case where $K = 2$, direct optimization is recommended, for which the values $\{\alpha_{00}, \alpha_{01}, \alpha_{10}, \alpha_{11}, \alpha_{20}, \alpha_{21}\} = \{1.9965, 1.3193, 0.4388, 0.7136, 0.9385, 1.2878\}$ yield an excellent FLT for image compression. In the middle of Fig. 5.27 the case $K = 4$ can be optimized by optimizing two invertible matrices. In the case where we use the method in Eq. (5.103)

Table 5.4 Inverse GLBT Bases Example for $M = 8$ and $N = 2$. The Even Bases are Symmetric while the Odd Ones are Anti-Symmetric, so Only their First Half is Shown

p_{0n}	p_{1n}	p_{2n}	p_{3n}	p_{4n}	p_{5n}	p_{6n}	p_{7n}
0.01786	−0.01441	0.06132	0.01952	0.05243	0.05341	0.04608	0.08332
0.05692	−0.01681	0.16037	0.12407	0.04888	0.16065	−0.09042	−0.02194
0.10665	0.06575	0.12462	0.24092	−0.21793	−0.13556	0.02108	−0.00021
0.16256	0.20555	−0.12304	−0.03560	−0.02181	−0.08432	0.13397	0.12747
0.22148	0.34661	−0.38107	−0.35547	0.36530	0.39610	−0.30170	−0.23278
0.27739	0.40526	−0.32843	−0.12298	−0.12623	−0.35462	0.41231	0.34133
0.32711	0.33120	0.03939	0.38507	−0.38248	−0.08361	−0.35155	−0.40906
0.36617	0.13190	0.44324	0.30000	0.28191	0.45455	0.13232	0.41414

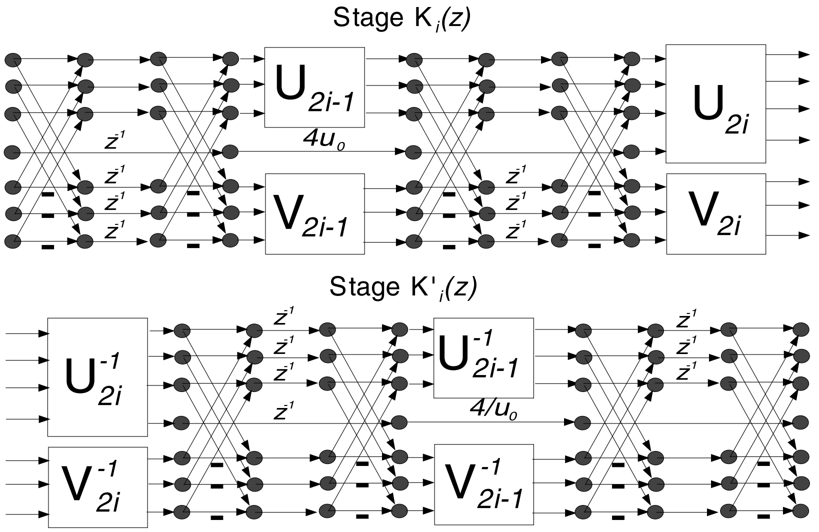


FIGURE 5.26
Implementation of the factors of the general factorization (GLBT) for M odd. Top: factor of the forward transform, $\mathbf{K}_i(z)$. Bottom: factor of the inverse transform, $\mathbf{K}'_i(z)$.

and the LBT as the K channel postprocessing stage, we can see that the LBT's DCT stage is canceled, yielding a very simple flow graph. The respective bases for forward and inverse transforms for the two FLTs ($K = 2$ with the given parameters, and $K = 4$ using the LBT) are shown in Fig. 5.28. Both bases are excellent for image coding, virtually eliminating ringing, despite the minimal complexity added to the DCT (which by itself can be implemented in a very fast manner [45]).

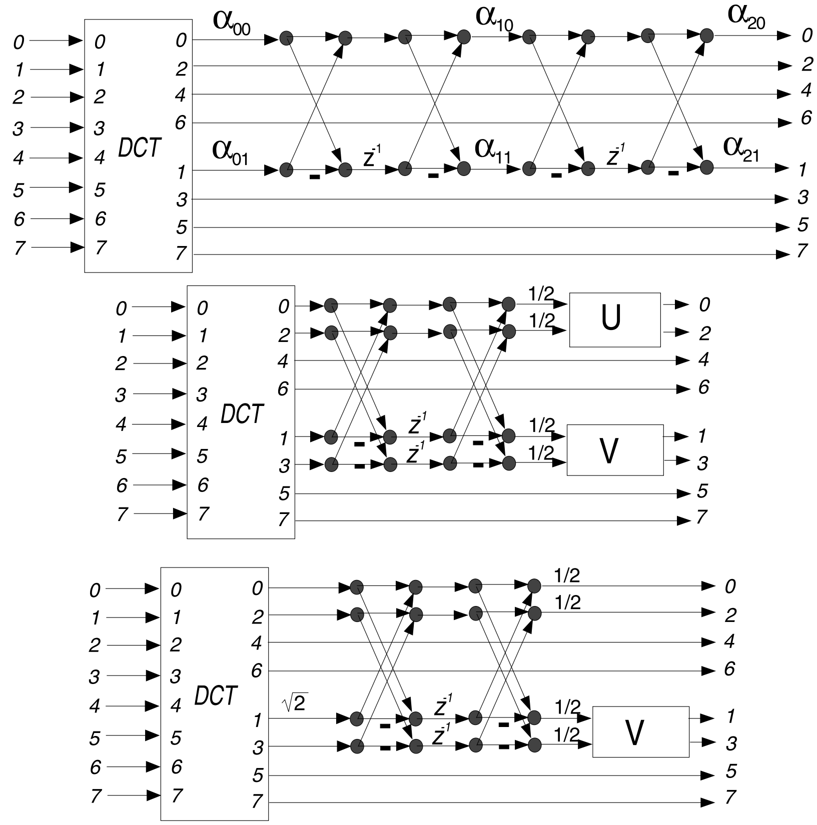


FIGURE 5.27
Implementation of examples of the FLT. On top, $K = 2$; middle, case $K = 4$; bottom, case $K = 4$ where $\Psi(z)$ is the LBT, thus having its DCT stage canceled.

5.8 Modulated LTs

Cosine modulated LTs or filter banks [63] use a lowpass prototype to modulate a cosine sequence. By a proper choice of the phase of the cosine sequence, Malvar developed the modulated lapped transform (MLT) [21], which led to the so-called extended lapped transforms (ELT) [24]–[27]. ELT allows several overlapping factors, generating a family of orthogonal cosine modulated LTs. Both designations (MLT and ELT) are frequently applied to this class of filter banks. Other cosine-modulation approaches have also been developed and the most significant difference among them is the lowpass prototype choice and the phase of the cosine sequence [17, 21, 25, 26, 30, 33, 48, 49, 55, 56, 63].

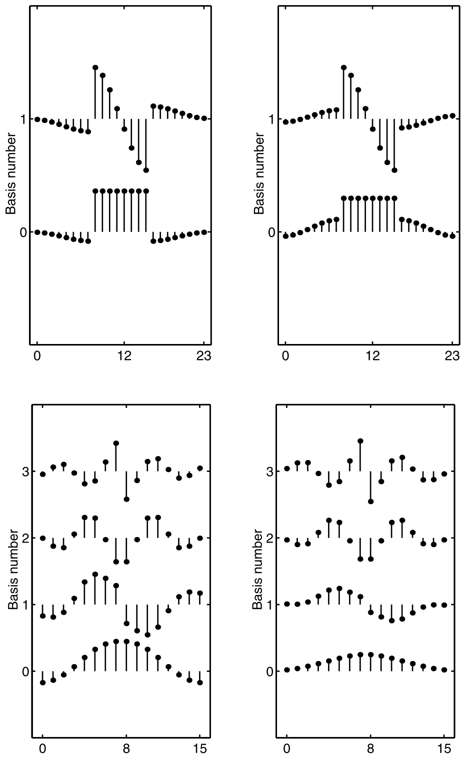


FIGURE 5.28
Bases of the FLT in the case $M = 8$ for forward and inverse LTs. From left to right: forward transform bases for the case $K = 2$; inverse transform bases for the case $K = 2$; forward transform bases for the case $K = 4$; inverse transform bases for the case $K = 4$. The remaining bases, not shown, are the regular bases of the DCT and have length 8.

In the ELTs, the filters' length L is basically an even multiple of the block size M, as $L = NM = 2KM$. Thus, K is referred to as the *overlap factor* of the ELT. The MLT-ELT class is defined by

$$p_{k,n} = h(n) \cos\left[\left(k + \frac{1}{2}\right)\left(\left(n - \frac{L-1}{2}\right)\frac{\pi}{M} + (N+1)\frac{\pi}{2}\right)\right] \qquad (5.104)$$

for $k = 0, 1 \ldots, M - 1$ and $n = 0, 1, \ldots, L - 1$. $h(n)$ is a symmetric window modulating the cosine sequence and the impulse response of a lowpass prototype (with cutoff frequency at $\pi/2M$), which is translated in frequency to M different frequency slots in order to construct the LT. A very useful ELT is the one with $K = 2$, which will be designated as ELT-2, while ELT with $K = 1$ will be referred to as the MLT.

A major plus of the ELTs is its fast implementation algorithm. The algorithm is based on a factorization of the PTM into a series of plane rotation stages and delays and a DCT type IV [45] orthogonal transform in the last stage, which also has fast implementation algorithms. The lattice-style algorithm is shown in Fig. 5.29 for an ELT with generic overlap factor K. In Fig. 5.29, each branch carries $M/2$ samples, and both analysis (forward transform) and synthesis (inverse transform) flow graphs are shown. The plane rotation stages are of the form indicated in Fig. 5.30 and contain $M/2$ orthogonal butterflies to implement the $M/2$ plane rotations. The stages Θ_i contain the plane rotations and are defined by

$$\begin{aligned}
\Theta_i &= \begin{bmatrix} -\mathbf{C}_i & \mathbf{S}_i\mathbf{J}_{M/2} \\ \mathbf{J}_{M/2}\mathbf{S}_i & \mathbf{J}_{M/2}\mathbf{C}_i\mathbf{J}_{M/2} \end{bmatrix}, \qquad (5.105)\\
\mathbf{C}_i &= diag\left\{\cos\left(\theta_{0,i}\right),\ \cos\left(\theta_{1,i}\right), \ldots, \cos\left(\theta_{\frac{M}{2}-1,i}\right)\right\}\ ,\\
\mathbf{S}_i &= diag\left\{\sin\left(\theta_{0,i}\right),\ \sin\left(\theta_{1,i}\right), \ldots, \sin\left(\theta_{\frac{M}{2}-1,i}\right)\right\}\ ,
\end{aligned}$$

where $\theta_{i,j}$ are rotation angles. These angles are the free parameters in the design of an ELT because they define the modulating window $h(n)$. Note that there are KM angles, while $h(n)$ has $2KM$ samples; however, $h(n)$ is symmetric which brings the total number of degrees of freedom to KM.

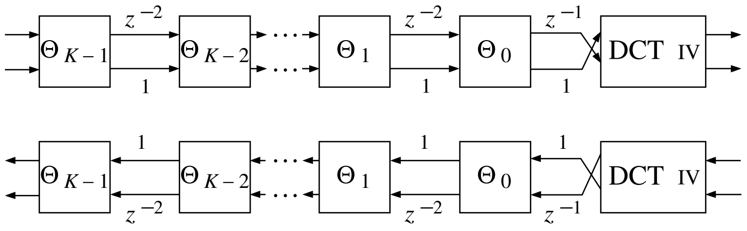


FIGURE 5.29
Flow graph for the direct (top) and inverse (bottom) ELT. Each branch carries $M/2$ samples.

In general, there is no simple relationship among the rotation angles and the window. Optimized angles for several values of M and K are presented in the extensive tables in [26]. In the ELT-2 case, however, one can use a parameterized design [25]–[27]. In this design, we have

$$\theta_{k,0} = -\frac{\pi}{2} + \mu_{M/2+k} \qquad (5.106)$$

$$\theta_{k,1} = -\frac{\pi}{2} + \mu_{M/2-1-k} \qquad (5.107)$$

where

$$\mu_i = \left[\left(\frac{1-\gamma}{2M}\right)(2k+1) + \gamma\right] \qquad (5.108)$$

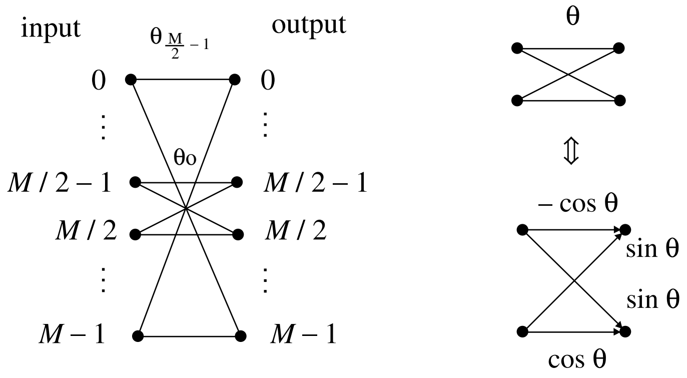


FIGURE 5.30
Implementation of the plane rotations stage showing the displacement of the $M/2$ butterflies.

and γ is a control parameter, for $0 \leq k \leq (M/2) - 1$. In general, although suboptimal for individual applications, $\gamma = 0.5$ provides a balanced trade-off of stopband attenuation and transition range for the equivalent filters (which are the bases of the LT viewed as a filter bank). The equivalent modulating window $h(n)$ is related to the angles as

$$\begin{aligned} h(n) &= \cos(\theta_{n0})\ \cos(\theta_{n1}) \\ h(M-1-n) &= \cos(\theta_{n0})\ \sin(\theta_{n1}) \\ h(M+n) &= \sin(\theta_{n0})\ \cos(\theta_{n1}) \\ h(2M-1-n) &= -\sin(\theta_{n0})\ \sin(\theta_{n1}) \end{aligned} \tag{5.109}$$

for $0 \leq n \leq (M/2) - 1$. In the $K = 1$ case, some example angles are

$$\theta_{k0} = \frac{\pi}{2} - \frac{\pi}{2M}\left(k + \frac{1}{2}\right) \tag{5.110}$$

for $0 \leq k \leq (M/2) - 1$. The corresponding modulating window $h(n)$ is

$$\begin{aligned} h(n) = h(2M-1-n) &= -\cos(\theta_{n0}) \\ h(M+n) = h(M-1-n) &= -\sin(\theta_{n0}) \end{aligned} \tag{5.111}$$

for $0 \leq n \leq (M/2) - 1$. The bases for the ELT using the suggested angles are shown in Fig. 5.31. In this figure, the 8-channel examples are for $N = 2$ ($K = 1$) and for $N = 4$ ($K = 2$).

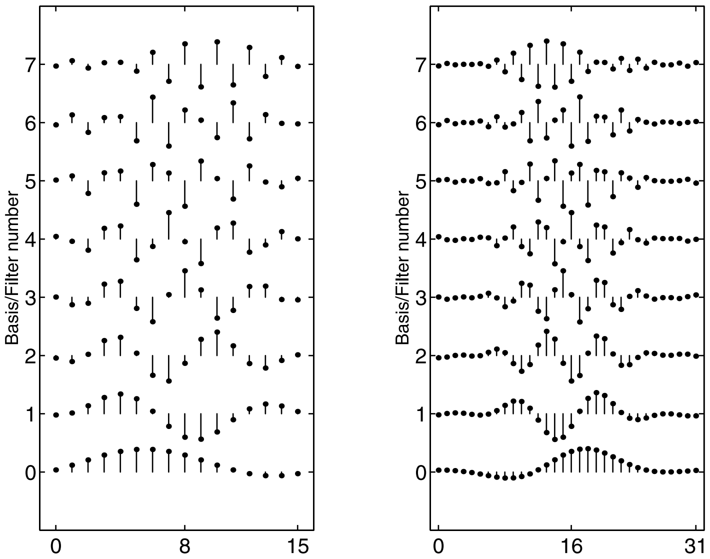


FIGURE 5.31
Example of ELT bases for the given angles design method for $M = 8$. Left: $K = 1, N = 2$, right: $K = 2, N = 4$.

5.9 Finite-Length Signals

Since the LT matrices are not square, in order to obtain n transformed subband samples one has to process more than n samples of the input signal. For the same reason, n subband samples generate more than n signal samples after inverse transformation. Our analysis so far has assumed infinite-length signals. Processing finite-length signals, however, is not trivial. Without proper consideration, there will be a distortion in the reconstruction of the boundary samples of the signal. There are basically three methods to process finite-length signals with LTs:

- signal extension and windowing of subband coefficients;
- same as above but using different extensions for different bases;
- time-varying bases for the boundary regions.

We discuss only the first method. The second is applicable only to a few transforms and filter banks. The subject of time-varying LTs is very rich and provides solutions to several problems, including the processing of boundary samples; we do not cover it in this chapter, but the reader is referred to [34, 9, 38, 52, 35] and their references for further information on time-varying LTs.

5.9.1 Overall Transform

Here we assume the model of extension and windowing described in Fig. 5.32 [39]. The input vector $\mathbf{x}$ is assumed to have $N_x = N_B M$ samples and is divided into 3 sections: $\mathbf{x}^T = [\mathbf{x}_l^T, \mathbf{x}_c^T, \mathbf{x}_r^T]$, where $\mathbf{x}_l$ and $\mathbf{x}_r$ contain the first and last λ samples of $\mathbf{x}$, respectively. Following the signal extension model, $\mathbf{x}$ is extended into $\tilde{\mathbf{x}}$ as

$$\tilde{\mathbf{x}}^T = \left[\mathbf{x}_{e,l}^T, \mathbf{x}^T, \mathbf{x}_{e,r}^T\right] = \left[(\mathbf{R}_l x_l)^T, \mathbf{x}_l^T, \mathbf{x}_c^T, \mathbf{x}_r^T, (\mathbf{R}_r \mathbf{x}_r)^T\right] . \quad (5.112)$$

In other words, the extended sections are found by a linear transform of the boundary samples of $\mathbf{x}$, as shown in Fig. 5.33; i.e.,

$$\mathbf{x}_{e,l} = \mathbf{R}_l \mathbf{x}_l \quad , \quad \mathbf{x}_{e,r} = \mathbf{R}_r \mathbf{x}_r \quad (5.113)$$

and $\mathbf{R}_l$ and $\mathbf{R}_r$ are arbitrary $\lambda \times \lambda$ "extension" matrices. For example, $\mathbf{R}_l = \mathbf{R}_r = \mathbf{J}_\lambda$ yields a symmetric extension.

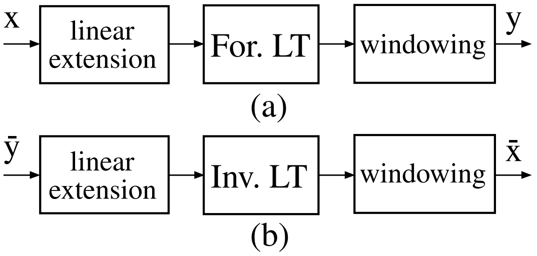


FIGURE 5.32
Extension and windowing in transformation of a finite-length signal using LTs. (a) Overall forward transform section. (b) Overall inverse transform section.

The transformation from the $N_x + 2\lambda$ samples in $\tilde{\mathbf{x}}$ to vector $\mathbf{y}$ with $N_B M = N_x$ subband samples is achieved through the block-banded matrix $\tilde{\mathbf{P}}$:

$$\tilde{\mathbf{P}} = \left(\begin{array}{ccccccc} \ddots & & \ddots & & & & 0 \\ & \mathbf{P}_0 & \mathbf{P}_1 & \cdots & \mathbf{P}_{N-1} & & \\ & & \mathbf{P}_0 & \mathbf{P}_1 & \cdots & \mathbf{P}_{N-1} & \\ & & & \mathbf{P}_0 & \mathbf{P}_1 & \cdots & \mathbf{P}_{N-1} \\ 0 & & & & \ddots & & \ddots \end{array} \right) . \quad (5.114)$$

Note that there are N_B block rows and that $\lambda = (N-1)M/2$. The difference between $\tilde{\mathbf{P}}$ and $\mathbf{H}$, defined in Eq. (5.21), is that $\mathbf{H}$ is assumed to be infinite and $\tilde{\mathbf{P}}$ is assumed to have only N_B block rows. We can use the same notation for $\tilde{\mathbf{Q}}$ with respect to $\mathbf{Q}_i$, so that, again, the difference between $\tilde{\mathbf{Q}}$ and $\mathbf{H}'$ defined in Eq. (5.32) is that $\mathbf{H}'$ is assumed to be infinite and $\tilde{\mathbf{Q}}$ is assumed to have only N_B block rows. The forward and inverse transform systems are given by

$$\tilde{\mathbf{y}} = \tilde{\mathbf{P}}\tilde{\mathbf{x}} \quad , \quad \tilde{\bar{\mathbf{x}}} = \tilde{\mathbf{Q}}^T \tilde{\mathbf{y}} . \quad (5.115)$$

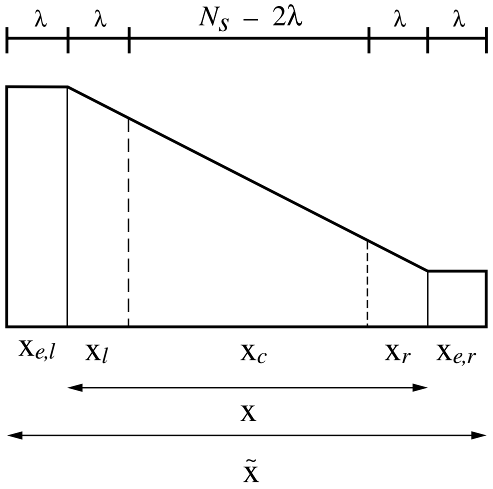


FIGURE 5.33
Illustration of signal extension of vector x into vector x̃. At each border, $\lambda = (L - M)/2$ samples outside the initial signal boundaries are found by linear relations applied to the λ boundary samples of x, i.e., $\mathbf{x}_{e,l} = \mathbf{R}_l \mathbf{x}_l$ and $\mathbf{x}_{e,r} = \mathbf{R}_r \mathbf{x}_r$. As only λ samples are affected across the signal boundaries, it is not necessary to use an infinite-length extension. Also, $\mathbf{x}_l$ and $\mathbf{x}_r$ contain the samples possibly affected by the border distortions after the inverse transformation.

In the absence of quantization or processing of the subband signals, $\tilde{\mathbf{y}} = \mathbf{y}$ and

$$\bar{\tilde{\mathbf{x}}} = \tilde{\mathbf{Q}}^T \tilde{\mathbf{y}} = \tilde{\mathbf{Q}}^T \tilde{\mathbf{P}} \tilde{\mathbf{x}} = \tilde{\mathbf{T}} \tilde{\mathbf{x}} \tag{5.116}$$

where $\bar{\tilde{\mathbf{x}}}$ is the reconstructed vector in the absence of quantization, and $\tilde{\mathbf{T}} = \tilde{\mathbf{Q}}^T \tilde{\mathbf{P}}$ is the transform matrix between $\bar{\tilde{\mathbf{x}}}$ and $\tilde{\mathbf{x}}$. Note that $\tilde{\mathbf{T}}$ has size $(N_x + \lambda) \times (N_x + \lambda)$ because it maps two extended signals. From Eq. (5.35) we can easily show that the transform matrix is

$$\tilde{\mathbf{T}} = \tilde{\mathbf{Q}}^T \tilde{\mathbf{P}} = \left[\begin{array}{c|c|c} \mathbf{T}_l & & 0 \\ \hline & \mathbf{I}_{N_x - 2\lambda} & \\ \hline 0 & & \mathbf{T}_r \end{array} \right] \tag{5.117}$$

where $\mathbf{T}_l$ and $\mathbf{T}_r$ are some $2\lambda \times 2\lambda$ matrices. Thus, distortion is incurred only to the λ boundary samples in each side of $\mathbf{x}$ (2λ samples in each side of $\tilde{\mathbf{x}}$).

In another view of the process, regardless of the extension method, there is a transform $\mathbf{T}$ such that

$$\mathbf{y} = \mathbf{T}\mathbf{x} \quad , \quad \bar{\mathbf{x}} = \mathbf{T}^{-1}\bar{\mathbf{y}} \tag{5.118}$$

without resorting to signal extension. The key is to find $\mathbf{T}$ and to invert it. If $\mathbf{T}$ is made orthogonal, one can easily invert it by applying transposition. This is the concept behind the use of time-varying LTs for correcting boundary distortions. For example, the LT can be changed near the borders to ensure the orthogonality of **T** [38]. We will not use time-varying LTs here but rather use extended signals and transform matrices.

5.9.2 Recovering Distorted Samples

Let

$$\left[\boldsymbol{\Phi}_l \middle| \boldsymbol{\Phi}_r\right] = \left[\begin{array}{cccc|ccccc} \mathbf{P}_0 & \mathbf{P}_1 & \cdots & \mathbf{P}_{N-2} & \mathbf{P}_{N-1} & & & & 0 \\ & \mathbf{P}_0 & \mathbf{P}_1 & \cdots & \mathbf{P}_{N-2} & \mathbf{P}_{N-1} & & & \\ & & \ddots & \ddots & & & \ddots & \ddots & \\ 0 & & & \mathbf{P}_0 & \mathbf{P}_1 & & \cdots & \mathbf{P}_{N-2} & \mathbf{P}_{N-1} \end{array}\right], \tag{5.119}$$

$$\left[\boldsymbol{\Psi}_l \middle| \boldsymbol{\Psi}_r\right] = \left[\begin{array}{cccc|ccccc} \mathbf{Q}_0 & \mathbf{Q}_1 & \cdots & \mathbf{Q}_{N-2} & \mathbf{Q}_{N-1} & & & & 0 \\ & \mathbf{Q}_0 & \mathbf{Q}_1 & \cdots & \mathbf{Q}_{N-2} & \mathbf{Q}_{N-1} & & & \\ & & \ddots & \ddots & & & \ddots & \ddots & \\ 0 & & & \mathbf{Q}_0 & \mathbf{Q}_1 & & \cdots & \mathbf{Q}_{N-2} & \mathbf{Q}_{N-1} \end{array}\right]. \tag{5.120}$$

Hence,

$$\mathbf{T}_l = \boldsymbol{\Psi}_l^T \boldsymbol{\Phi}_l \quad , \quad \mathbf{T}_r = \boldsymbol{\Psi}_r^T \boldsymbol{\Phi}_r \, . \tag{5.121}$$

If $\bar{\tilde{\mathbf{x}}}$ is divided in the same manner as $\tilde{\mathbf{x}}$, i.e.,

$$\bar{\tilde{\mathbf{x}}} = \left[\bar{\mathbf{x}}_{e,l}^T, \bar{\mathbf{x}}_l^T, \bar{\mathbf{x}}_c^T, \bar{\mathbf{x}}_r^T, \bar{\mathbf{x}}_{e,r}^T\right], \tag{5.122}$$

then,

$$\left[\begin{array}{c} \bar{\mathbf{x}}_{e,l} \\ \bar{\mathbf{x}}_l \end{array}\right] = \mathbf{T}_l \left[\begin{array}{c} \mathbf{x}_{e,l} \\ \mathbf{x}_l \end{array}\right] = \mathbf{T}_l \left[\begin{array}{c} \mathbf{R}_l \mathbf{x}_l \\ \mathbf{x}_l \end{array}\right] = \mathbf{T}_l \left[\begin{array}{c} \mathbf{R}_l \\ \mathbf{I}_\lambda \end{array}\right] \mathbf{x}_l = \boldsymbol{\Gamma}_l \, \mathbf{x}_l \tag{5.123}$$

where

$$\boldsymbol{\Gamma}_l = \mathbf{T}_l \left[\begin{array}{c} \mathbf{R}_l \\ \mathbf{I}_\lambda \end{array}\right] \tag{5.124}$$

is a $2\lambda \times \lambda$ matrix. If and only if $\boldsymbol{\Gamma}_l$ has rank λ, then $\mathbf{x}_l$ can be recovered through the pseudo-inverse of $\boldsymbol{\Gamma}_l$ as

$$\mathbf{x}_l = \boldsymbol{\Gamma}_l^+ \left[\begin{array}{c} \bar{\mathbf{x}}_{e,l} \\ \bar{\mathbf{x}}_l \end{array}\right] = \left(\boldsymbol{\Gamma}_l^T \boldsymbol{\Gamma}_l\right)^{-1} \boldsymbol{\Gamma}_l^T \left[\begin{array}{c} \bar{\mathbf{x}}_{e,l} \\ \bar{\mathbf{x}}_l \end{array}\right]. \tag{5.125}$$

For the other ("right") border, the identical result is trivially found to be

$$\mathbf{x}_r = \boldsymbol{\Gamma}_r^+ \left[\begin{array}{c} \bar{\mathbf{x}}_r \\ \bar{\mathbf{x}}_{e,r} \end{array}\right] = \left(\boldsymbol{\Gamma}_r^T \boldsymbol{\Gamma}_r\right)^{-1} \boldsymbol{\Gamma}_r^T \left[\begin{array}{c} \bar{\mathbf{x}}_r \\ \bar{\mathbf{x}}_{e,r} \end{array}\right], \tag{5.126}$$

where

$$\mathbf{\Gamma}_r = \mathbf{T}_r \begin{bmatrix} \mathbf{I}_\lambda \\ \mathbf{R}_r \end{bmatrix} \tag{5.127}$$

is also assumed to have rank λ. It is necessary, but not sufficient that $\mathbf{\Phi}_l, \mathbf{\Phi}_r, \mathbf{\Psi}_l$, and $\mathbf{\Psi}_r$ have rank λ since rank can be reduced by the matrix products. It is also possible to express the conditions in more detail. However, without any useful analytical solution, numerical rank checking is the best approach.

To summarize, the steps to achieve PR for given $\mathbf{R}_l$ and $\mathbf{R}_r$ are

1. Select $\mathbf{P}$ and $\mathbf{Q}$ and identify their submatrices $\mathbf{P}_i$ and $\mathbf{Q}_i$.
2. Find $\mathbf{\Phi}_l, \mathbf{\Phi}_r, \mathbf{\Psi}_l, \mathbf{\Psi}_r$, from Eqs. (5.119) and (5.120).
3. Find $\mathbf{T}_l$ and $\mathbf{T}_r$ from Eq. (5.121).
4. Find $\mathbf{\Gamma}_l$ and $\mathbf{\Gamma}_r$ from Eqs. (5.124) and (5.127).
5. Test ranks of $\mathbf{\Gamma}_l$ and $\mathbf{\Gamma}_r$.
6. If ranks are λ, obtain $\mathbf{\Gamma}_l^+$, $\mathbf{\Gamma}_r^+$ and reconstruct $\mathbf{x}_l$ and $\mathbf{x}_r$.

This is an extension of de Queiroz and Rao [39] to nonorthogonal LTs, with particular concern to test whether the pseudo-inverses exist.

The model in Fig. 5.32 and the proposed method are not applicable for some LTs. The notable classes of LTs include those whose bases have different lengths and different symmetries. Examples are (i) two-channel nonorthogonal LTs with odd-length (2-channel biorthogonal filter banks [55]); (ii) FLT; and (iii) other composite systems, i.e., cascaded systems such as those used in de Queiroz [41]. For the first example, it is trivial to use symmetric extensions but different symmetries for different bases [55]. The second example has the same reasoning. However, an FLT can be efficiently implemented by applying the method just described to each of the stages of the transformation (i.e., first apply the DCT and then use the method above for the second part). The reason for rank deficiency is that different filters would require different extensions during the forward transformation process; therefore, the model in Fig. 5.32 is not applicable.

The above method works very well for M-channel filter banks whose filters have the same length. The phase of the filters and the extensions can be arbitrary, and the method has been shown to be consistent for all uniform-length filter banks of interest tested.

5.9.3 Symmetric Extensions

In case the LT is symmetric and obeys Eqs. (5.65) and (5.66), there is a much simpler method to implement the LT over a finite-length signal of N_B blocks of M samples.

In the forward transform section, we perform symmetric extensions as described, applied to the last $\lambda = (L - M)/2$ samples on each border, resulting in a signal $\{\tilde{x}(n)\}$ with $N_x + 2\lambda = N_x + L - M$ samples,

$$x(\lambda - 1), \ldots, x(0), x(0), \ldots, x\left(N_x - 1\right), x\left(N_x - 1\right), \ldots, x\left(N_x - \lambda\right). \quad (5.128)$$

The signal is processed by the PTM $\mathbf{F}(z)$ as a clocked system, without concern for border locations. The internal states of the system $\mathbf{F}(z)$ can be initialized in any way. So, the $N_B + N - 1$ blocks of the extended signal are processed yielding an equal number of blocks of subband samples. Discarding the first $N - 1$ output blocks, obtain N_B transform-domain blocks corresponding to N_B samples of each subband.

The general strategy to achieve perfect reconstruction, without great increase in complexity or change in the implementation algorithm, is to extend the samples in the subbands, generating more blocks to be inverse transformed, in such a way that after inverse transformation (assuming no processing of the subband signals) the signal recovered is identical to the original at the borders. The extension of the k-th subband signal depends on the symmetry of the k-th basis. Let $p_{kn} = v_k p_{k,L-1-n}$ for $0 \leq k \leq M - 1$ and $0 \leq n \leq L - 1$, i.e., $v_k = 1$ if p_{kn} is symmetric and $v_k = -1$ if p_{kn} is anti-symmetric. Before inverse transformation, for each subband signal $\{\bar{y}_k(m)\}$ of N_B samples, fold the borders of $\{\bar{y}_k(m)\}$ (as in the analysis section) in order to find a signal $\{\tilde{\bar{y}}_k(m)\}$, and invert the sign of the extended samples if p_{kn} is anti-symmetric. For s samples reflected about the borders, the k-th subband signal will have samples

$$\begin{gathered} v_k \hat{y}_k(s - 1), \ldots, v_k \hat{y}_k(0), \hat{y}_k(0), \cdots \hat{y}_k(N_B - 1), \\ v_k \hat{y}_k\left(N_B - 1\right), \ldots, v_k \hat{y}_k\left(N_B - s\right). \end{gathered}$$

The inverse transformation can be performed as

- *N odd* — Reflect $s = (N - 1)/2$ samples about each border, thus getting $N_B + N - 1$ blocks of subband samples to be processed. To obtain the inverse transformed samples $\{\hat{x}(n)\}$, initialize the internal states in any way, run the system $\mathbf{G}(z)$ over the $N_B + N - 1$ blocks, and discard the first $N - 1$ reconstructed blocks, retaining the $N_x = N_B M$ remaining samples.

- *N even* — Reflect $s = N/2$ samples about each border, thus getting $N_B + N$ blocks to be processed. To obtain the inverse transformed samples $\{\hat{x}(n)\}$, initialize the internal states in any way and run the system $\mathbf{G}(z)$ over the $N_B + N$ blocks. Discard the first $N - 1$ reconstructed blocks and the first $M/2$ samples of the N-th block. Include in the reconstructed signal the last $M/2$ samples of the N-th block and the subsequent $(N_B - 1)M$ samples. In the last block, include the first $M/2$ samples in the reconstructed signal and discard the rest.

This approach will assure the perfect reconstruction property and orthogonality of the overall transformation if the LT is orthogonal [38]. The price paid is running the algorithm over extra N or $N - 1$ blocks. As it is common to have $N_B \gg N$, the computational increase is only marginal.

5.10 Design Issues for Compression

Block transform coding and subband coding have been two dominant techniques in existing image compression standards and implementations. Both methods actually exhibit many similarities: relying on a certain transform to convert the input image to a more decorrelated representation, then utilizing the same basic building blocks such as bit allocator, quantizer, and entropy coder to achieve compression.

Block transform coders enjoyed success due to their low complexity in implementation and their reasonable performance. The most popular block transform coder led to the current image compression standard JPEG [32] which utilizes the 8 × 8 DCT at its transformation stage. At high bit rates, JPEG offers almost visually lossless reconstruction image quality. However, when more compression is needed (i.e., at lower bit rates), annoying blocking artifacts show up for two reasons: (i) the DCT bases are short, nonoverlapped, and have discontinuities at the ends; and (ii) JPEG processes each image block independently. So, interblock correlation is not taken into account.

The development of lapped transforms helps solve the blocking problem by borrowing pixels from the adjacent blocks to produce the transform coefficients of the current block. In other words, lapped transforms are block transforms with overlapping basis functions. Compared to the traditional block transforms such as DCT, DST, and the Walsh-Hadamard transform [46], lapped transforms offer two main advantages: (i) from the analysis viewpoint, they take into account interblock correlation and, hence, provide better energy compaction that leads to more efficient entropy coding of the coefficients; and (ii) from the synthesis viewpoint, their basis functions decay asymptotically to zero at the ends, reducing blocking discontinuities drastically.

All of the lapped transforms presented in the previous sections are designed to have high practical value. They all have perfect reconstruction. Some of them even have real and symmetric basis functions. However, for the transforms to achieve high coding performance, several other properties are also needed. Transforms can be obtained using unconstrained nonlinear optimization where some of the popular cost criteria are coding gain, DC leakage, attenuation around mirror frequencies, and stopband attenuation. In the particular field of image compression, all of these criteria are well-known desired properties in yielding the best reconstructed image quality [44, 55]. The cost function in the optimization process can be a weighted linear combination of these measures as follows

$$\begin{aligned} C_{\text{overall}} &= \alpha_1\, C_{\text{coding gain}} + \alpha_2\, C_{\text{DC}} + \alpha_3\, C_{\text{mirror}} \\ &\quad + \alpha_4\, C_{\text{analysis stopband}} + \alpha_5\, C_{\text{synthesis stopband}}\,. \end{aligned} \tag{5.129}$$

Coding Gain

The coding gain of a transform 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, $\sigma_{x_i}^2$ as the variance of the i-th subband, and $||q_i||^2$ as the $\mathcal{L}^2$ norm of the i-th bases of the inverse LT $\mathbf{Q}$. With several assumptions including scalar quantization and a sufficiently large bit rate, the generalized coding gain can be formulated [12, 15, 28] as

$$C_{\text{coding gain}} = 10 \log_{10} \frac{\sigma_x^2}{\left(\prod_{i=0}^{M-1} \sigma_{x_i}^2 \, \|q_i\|^2 \right)^{\frac{1}{M}}} . \tag{5.130}$$

The coding gain can be thought of as an approximate measure of the transform's energy compaction capability. Among the listed criteria, higher coding gain correlates most consistently with higher objective performance (measured in MSE or PSNR). Transforms with higher coding gain compact more signal energy into a fewer number of coefficients, leading to more efficient entropy coding.

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, which 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 [55]. The DC cost function can be defined as

$$C_{\text{DC}} = \sum_{i=1}^{M-1} \sum_{n=0}^{L-1} p_{in} \,, \tag{5.131}$$

where $\{p_{in}\}$ are entries of the LT matrix $\mathbf{P}$. The reader should note that all antisymmetric filters have a zero at DC (zero frequency). Therefore, the above formula needs to apply only to symmetric bases to reduce the complexity of the optimization process. It is interesting to note that the zero leakage condition is equivalent to having one vanishing moment — a necessary condition in the construction of wavelets.

Attenuation at Mirror Frequencies

Viewing the transform bases as filters, we can generalize C_{DC} to also encompass mirror frequency points. The concern is now at all aliasing frequencies $\omega_m = \frac{2\pi m}{M},\ m \in \mathcal{Z},\ 1 \le m \le \frac{M}{2}$. Ramstad, Aase, and Husoy [44] have shown that frequency attenuation at mirror frequencies is very important in the further reduction of blocking artifacts; the filter (basis function) response should be small at these mirror frequencies as well. The corresponding cost function is

$$C_{\text{mirror}} = \sum_{i=0}^{M-2} \left| P_i \left(e^{j\omega_m} \right) \right|^2, \quad \omega_m = \frac{2\pi m}{M}, \quad 1 \le m \le \frac{M}{2} \,, \tag{5.132}$$

where $P_i(e^{j\omega})$ is the Fourier transform of $\{p_{in}\}$. Low DC leakage and high attenuation near the mirror frequencies are not as essential to the coder's objective performance as coding gain. However, they do improve the visual quality of the reconstructed image significantly.

Stopband Attenuation

Stopband attenuation is a classical performance criterion in filter design. In the forward transform, the stopband attenuation cost helps in improving the signal decorrelation and decreasing the amount of aliasing. In meaningful images, we know a priori that most of the energy is concentrated in a low frequency region. Hence, high stopband attenuation in this part of the frequency spectrum becomes extremely desirable. In the inverse transform, the synthesis filters (basis functions) covering low-frequency bands need to have high stopband attenuation near and/or at $\omega = \pi$ to enhance their smoothness. The biased weighting can be enforced using two simple functions $W_i^a(e^{j\omega})$ and $W_i^s(e^{j\omega})$. For our purposes, the stopband attenuation criterion measures the sum of all of the filters' energy outside the designated passbands:

$$\begin{aligned} C_{\text{analysis stopband}} &= \sum_{i=0}^{M-1} \int_{\omega\in\Omega_{stopband}} W_i^a\left(e^{j\omega}\right) \left|P_i\left(e^{j\omega}\right)\right|^2 d\omega \\ C_{\text{synthesis stopband}} &= \sum_{i=0}^{M-1} \int_{\omega\in\Omega_{stopband}} W_i^s\left(e^{j\omega}\right) \left|Q_i\left(e^{j\omega}\right)\right|^2 d\omega\,, \end{aligned} \tag{5.133}$$

where $Q_i(e^{j\omega})$ is the Fourier transform of $\{q_{in}\}$, which are the entries of $\mathbf{Q}$.

5.11 Transform-Based Image Compression Systems

Transform coding is the single-most popular approach for image compression. The basic building blocks of a transform coder are illustrated in Fig. 5.34. The entropy coder is the step that actually performs any compression. The entropy of the symbols to be compressed is reduced by the quantizer which is the only building block which is not reversible; it is a lossy operator. The transform step neither causes losses nor performs compression, but it is the core of the compression system. It enables compression by compacting the energy into few coefficients, thus reducing the distortion caused by the quantization step.

A separable transformation is one where all rows are transformed, and then all columns are transformed. Let $\mathbf{X}$ be the matrix containing the image pixels and let $\mathbf{Y}$ be the transformed image; then from Eq. (5.33) we have

$$\mathbf{Y} = \mathbf{H}_F\mathbf{X}\mathbf{H}_F^T \qquad \mathbf{X} = \mathbf{H}_I^T\mathbf{Y}\mathbf{H}_I\ . \tag{5.134}$$

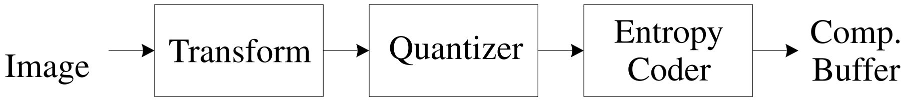


FIGURE 5.34
Basic building blocks of a transform coder. The image is transformed and quantized in order to submit the data to a lossless (entropy) coder.

$\mathbf{Y}$ is composed by $M \times M$ blocks $\mathbf{Y}_{ij}$, each block having a full set of transform coefficients (subband samples). Each of the blocks $\mathbf{Y}_{ij}$ is then quantized and encoded. In the case of a block transform, each transformed block $\mathbf{Y}_{ij}$ is related to only one image block of $M \times M$ pixels. For LTs, of course, each block $\mathbf{Y}_{ij}$ is related to several pixel blocks. We will discuss only the performance of LTs in the context of two popular image coders.

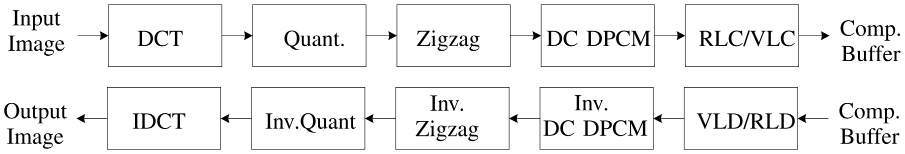


FIGURE 5.35
JPEG building blocks. The image is broken into blocks and each block is transformed using the DCT, quantized, and encoded. The decoder performs the inverse operations.

5.11.1 JPEG

Transform coding is the framework employed by the Joint Photographic Experts Group (JPEG) still image compression standard. The reader is referred to Pennebacker and Mitchell [32] for a detailed description of the JPEG image compression system; the publication not only covers JPEG well but it also includes a copy of the standard draft! The steps for JPEG compression are shown in Fig. 5.35. The image is broken into blocks of 8×8 pixels, which are transformed using a separable 8-channel DCT. The transformed block is quantized (effectively divided by an integer number and rounded) and then encoded. The quantized transformed samples in a block are scanned into a vector following a zigzag pattern starting from the lowest frequency band to the highest. The lowest frequency sample of a block is known as the DC coefficient (DCC). Before encoding, the quantized DCC of a block is actually replaced by the difference of itself and the DCC of a previous block (this is referred to as DPCM). Finally the scanned vector is fed into an entropy coder which uses a combination of run-length coding (RLC) and variable-length coding (VLC) to compress the data. The decoder runs the inverse of all the encoding steps in reverse order to reconstruct the image block from the compressed data.

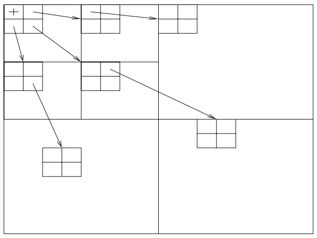
For our purposes, we compare the performance of LTs against the DCT. LTs can be very easily incorporated into JPEG by simply replacing the DCT. Even though the bases overlap, the subband samples are arranged in a block just like the DCT. The transformed block is fed to the rest of the JPEG coder (which does not care if the samples were found through the DCT) to be quantized and encoded. All results here for JPEG are found by merely replacing the DCT by an 8-channel ($M = 8$) LT while maintaining all the other parameters and settings unchanged.

5.11.2 Embedded Zerotree Coding

Embedded Zerotree Coding (EZC) is often associated with the dyadic wavelet transform. The multiresolution characteristics of the wavelet transform have created an intuitive foundation on which simple, yet sophisticated, methods of encoding the transform coefficients are developed. Exploiting the relationship between the parent and the offspring coefficients in a wavelet tree, the original Embedded Zerotree Wavelet (EZW) coder [50] and its variations [47, 71] can effectively order the coefficients by bit planes and transmit the more significant bits first. This coding scheme results in an embedded bit stream along with many other advantages, such as exact bit rate control and near-idempotency (perfect idempotency is obtained when the transform maps integers to integers). In these subband coders, global information is taken into account fully. In this section, we confirm that the embedded zerotree framework is not limited only to the wavelet transform but it can be utilized with various LTs as well. In fact, the combination of a LT and several levels of wavelet decomposition of the DC band can provide much finer frequency spectrum partitioning, leading to significant improvement over current embedded wavelet coders [58, 61].

The EZC approach relies on the fundamental idea that the most important information (defined here as what decreases a certain distortion measure the most) should be transmitted first. Assuming that the distortion measure is the mean-squared error (MSE), the transform is orthogonal, and transform coefficients C_{ij} are transmitted one by one, it is well known that the MSE decreases by $\frac{1}{N}|C_{ij}|^2$, where N is the total number of pixels. Therefore, large magnitude coefficients should always be transmitted first. If one bit is transmitted at a time, this approach can be generalized to ranking the coefficients by bit planes, and the most significant bits are transmitted first [43] as demonstrated in Fig. 5.36. The progressive transmission scheme results in an embedded bit stream (i.e., it can be truncated at any point by the decoder to yield the best corresponding reconstructed image). The algorithm can be thought of as an elegant combination of a scalar quantizer with power-of-two stepsizes and an entropy coder to encode wavelet coefficients.

Embedded algorithm relies on the hierarchical coefficients' tree structure called a *wavelet tree* — a set of wavelet coefficients from different scales that belong in the same spatial locality. This is demonstrated in Fig. 5.37, where the tree in the vertical direction is circled. All of the coefficients in the lowest frequency band make up the *DC band* or the *reference signal* (located at the upper left corner). Besides these DC coefficients, in a wavelet tree of a particular direction, each lower frequency



	sign	s	s	s	s	s	s	s	s	s	s	s	s	s
msb	5	1	1	0	0	0	0	0	0	0	0	0	0	0
	4	⋗	⋗	1	1	0	0	0	0	0	0	0	0	0
	3	⋗	⋗	⋗	⋗	1	1	1	1	0	0	0	0	0
	2	⋗	⋗	⋗	⋗	⋗	⋗	⋗	⋗	1	1	1	1	1
	1	⋗	⋗	⋗	⋗	⋗	⋗	⋗	⋗	⋗	⋗	⋗	⋗	⋗
lsb	0	⋗	⋗	⋗	⋗	⋗	⋗	⋗	⋗	⋗	⋗	⋗	⋗	⋗

FIGURE 5.36
Embedded zerotree coding as a bit-plane refinement scheme.

parent node has four corresponding higher frequency *offspring nodes.* All coefficients below a parent node in the same spatial locality is defined as its *descendants.* Define a coefficient C_{ij} to be *significant* with respect to a given threshold T if $|C_{ij}| \geq T$, and *insignificant* otherwise. Meaningful image statistics have shown that if a coefficient is insignificant, it is very likely that its offspring and descendants are insignificant as well. Exploiting this fact, sophisticated embedded wavelet coders can output a single marker to represent very efficiently a large, smooth image area (an insignificant tree). For more details on the algorithm, the reader is referred to references [47, 50], and [71].

LTs obtain a uniform spectrum partitioning whereas the wavelet transform has an octave-band signal decomposition. All LT subbands have the same size. A parent node would not have four offspring nodes as in the case of the wavelet representation. In order to use embedded zerotree algorithms to encode the LT coefficients, we have to modify the data structure. Investigating the analogy between the wavelet and the LT, as in Fig. 5.37, reveals that the parent, offspring, and descendants in a wavelet tree cover the same spatial locality, and so do the coefficients of a block of LT coefficients. In fact, a wavelet tree in an L-level decomposition is analogous to a 2^L-band LT's coefficient block. The difference lies at the bases that generate these coefficients.

Let $\mathcal{O}(i, j)$ be the set of coordinates of all offsprings of the node (i, j) in an M-band LT $(0 \leq i, j \leq M - 1)$; then $\mathcal{O}(i, j)$ can be represented as follows:

$$\mathcal{O}(i, j) = \{(2i, 2j), (2i, 2j + 1), (2i + 1, 2j), (2i + 1, 2j + 1)\} . \qquad (5.135)$$

All (0, 0) coefficients from all transform blocks form the DC band, which is similar to the wavelet transform's reference signal, and each of these nodes has only three offspring: (0, 1), (1, 0), and (1, 1). The complete tree is now available locally; we do not have to search for the offspring across the subbands anymore. The only requirement here is that the number of bands that M has be a power of two. Fig. 5.38 demonstrates, through a simple rearrangement of the LT coefficients, that the redefined tree structure in Eq. (5.135) does possess a wavelet-like multiscale representation. To decorrelate the DC band even more, several levels of wavelet decomposition can be used depending on the input image size. Besides the obvious increase in the coding efficiency of DC coefficients, thanks to deeper coefficient trees, wavelets provide variably longer bases for the signal's DC component, leading to smoother reconstructed images; in other words, blocking artifacts are further reduced. LT

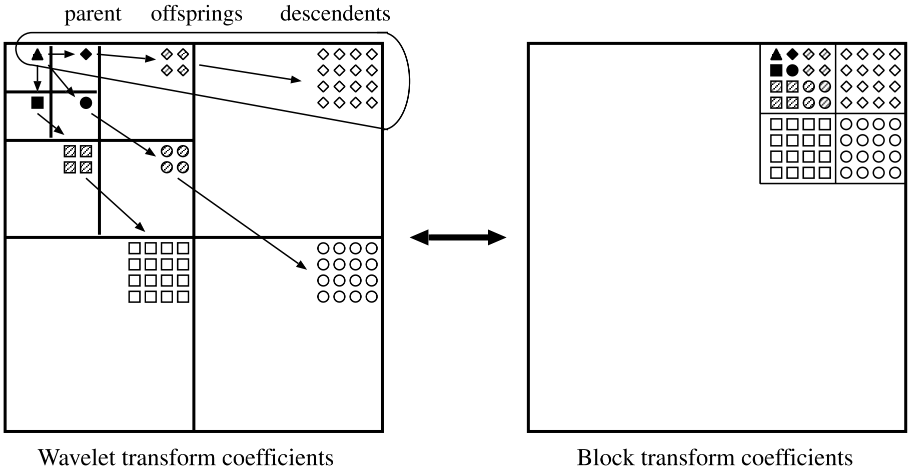


FIGURE 5.37
Wavelet and block transform analogy.

iteration (resulting in the HLTs or M-band wavelets) can be applied to the DC band as well.

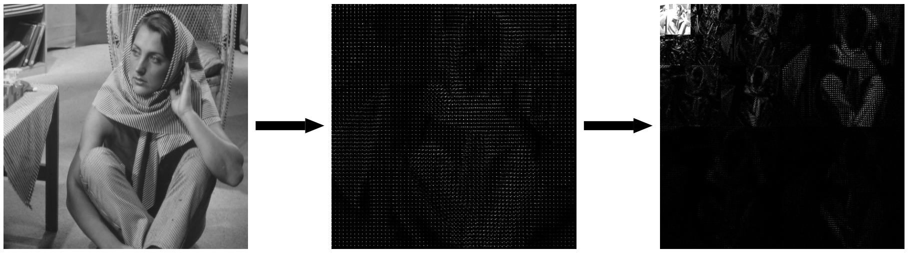


FIGURE 5.38
A demonstration of the analogy between block transform and wavelet representation. Reproduced by Special Permission of *Playboy* magazine. Copyright ©1972, 2000 by Playboy.

5.11.3 Other Coders

Although we use JPEG and EZC only as examples, LTs can be used to replace other transforms in a variety of coders. If the coder treats the subbands independently, encoding of the LT's output is not different from encoding the output of any other transform. If the subbands are not encoded independently, but the coder was designed for a nonhierarchical M-channel transform (e.g., DCT), then the LT can be used to immediately replace the transform, just like we do for JPEG. If the subbands are not encoded independently, and the coder was designed for a hierarchical transform (like wavelets), then one can use the same approach as in the previous section to incorporate the LTs. It is straightforward to utilize LTs in coders such as the new standard (JPEG 2000) [11] or other efficient coders based on optimized classification of subband samples [13]. Furthermore, LTs can be used to replace the wavelet transform in several efficient coders [1, 51, 67, 69]. Adaptive LTs have also been applied to image compression [16].

5.12 Performance Analysis

The performance of any compression system is measured by computing the distortion achieved by compressing a particular image at a certain compression ratio. For every compression attempt for a particular image and coding settings, one can compute the rate achieved R, often expressed in bits per pixel (or bpp), and the distortion is some measure of the difference between the original image and its reconstructed approximation after decompression. We use the peak signal-to-noise ratio measure which is given in decibels (dB) and is defined as

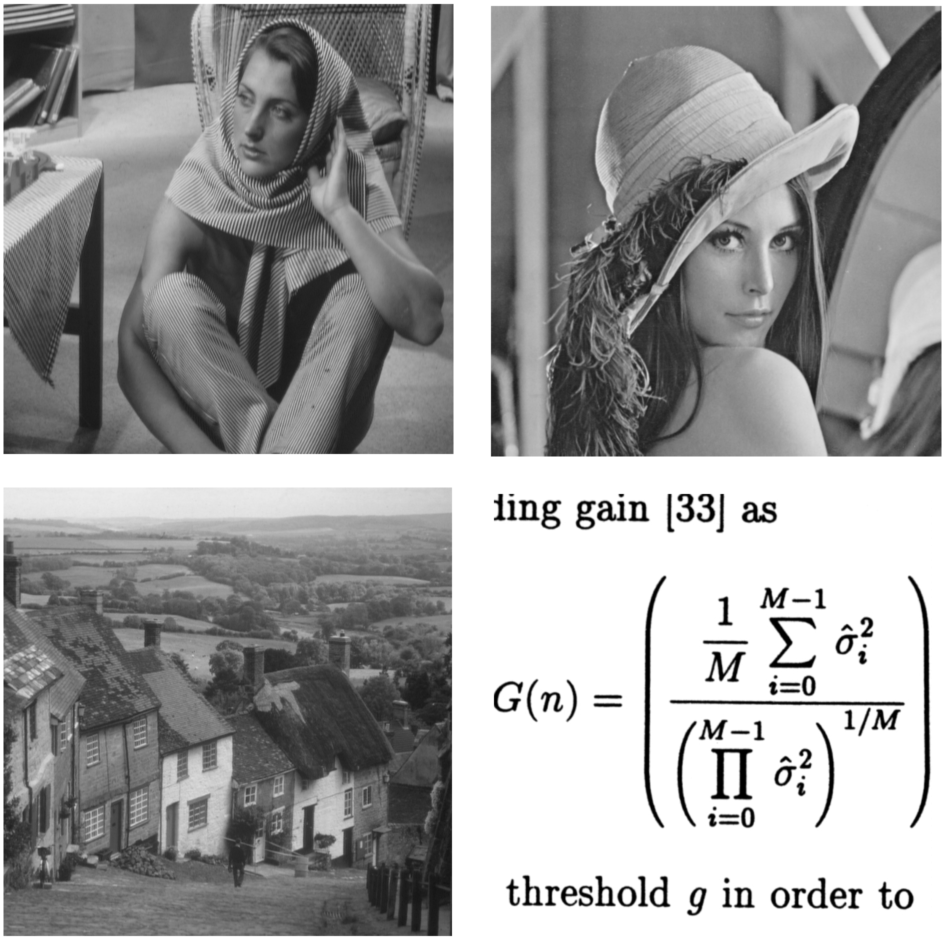
$$PSNR = 10 \log_{10} \left(\frac{(255)^2}{\frac{1}{N_{\text{pixels}}} \sum_{ij} [x_o(i,j) - x_r(i,j)]^2} \right) , \tag{5.136}$$

where $x_o(i,j)$ and $x_r(i,j)$ are the original and reconstructed pixels, respectively.

We used four 512×512-pixel test images shown in Fig. 5.39 for benchmarking the LTs in image compression. The image, "Barbara" has large detailed areas with high-frequency patterns, while "Lena" has mainly smooth areas with occasional edges and textures. "Goldhill" is a typical landscape image with many details, and the "text" image is basically composed of high contrast edges.

5.12.1 JPEG

For JPEG, we compared the following transforms for $M = 8$: ELT-2, MLT, DCT, LOT, GenLOT ($L = 48$), GLBT ($L = 16$), and FLT (two 24-tap bases along with six 8-tap DCT bases). For each image and transform, several rate-distortion points are obtained by compressing the image using JPEG's default (example) quantizer table [32] scaled by a multiplicative factor. Instead of providing the actual PSNR obtained with every experiment we compared every result to the PSNR obtained by compressing the same image at the same bit rate using the DCT. Objective results are shown in Fig. 5.40. The curves can be viewed as incremental PSNR plots, in which for each transform, image, and bit rate, it indicates the gain in PSNR (dB) obtained by replacing the DCT by a given LT. Note that the performance of the LTs is far superior to that of the DCT except for the "text" image. This image is not suitable



ıing gain [33] as

$$G(n) = \left(\frac{\frac{1}{M} \sum_{i=0}^{M-1} \hat{\sigma}_i^2}{\left(\prod_{i=0}^{M-1} \hat{\sigma}_i^2 \right)^{1/M}} \right)$$

threshold g in order to

FIGURE 5.39
Test images of 512×512 **pixels. From left to right and top to bottom: "Barbara," "Lena," "Goldhill," and "text." Reproduced by Special Permission of *Playboy* magazine. Copyright ©1972, 2000 by Playboy.**

for transform compression because it is mainly composed of sharp edges. Hence, the short bases of the DCT concentrate the artifacts (caused by compressing the sharp edges) into small areas. In any case, the GLBT and the GenLOT are always good performers.

Objective comparisons are not always the best. The FLT, for example, seems to have similar performance to DCT in terms of PSNR. However, it produces noticeably better images. It is virtually free of the ringing and blocking artifacts which are the main drawbacks of using block transforms like DCT. Reconstructed images are shown in Fig. 5.41.

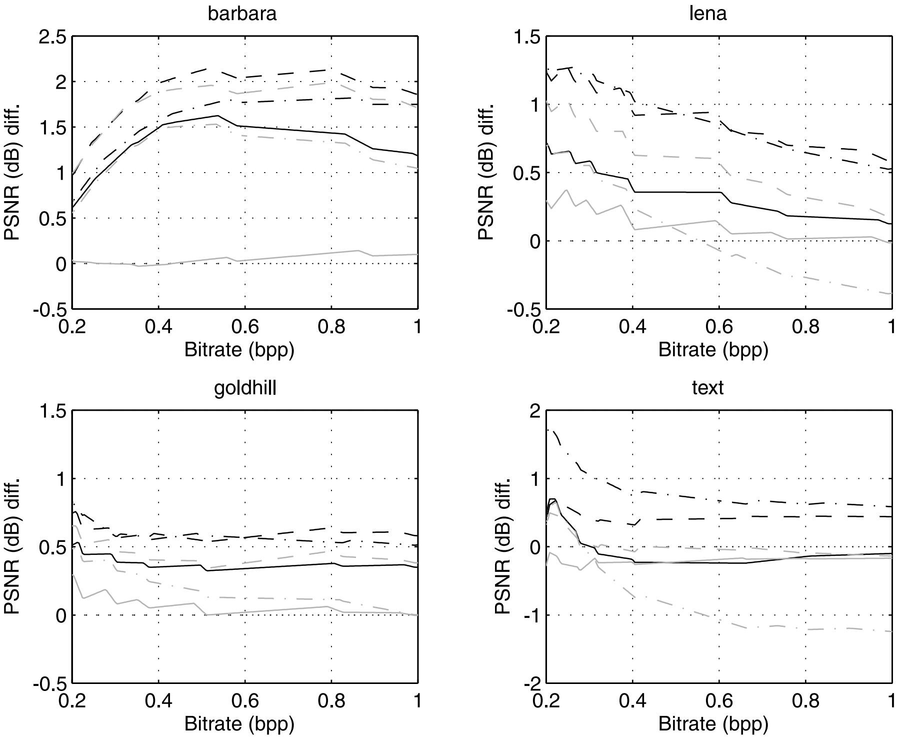


FIGURE 5.40

Comparison among transforms in JPEG for several images and bit rates. Incremental PSNR plots are shown, indicating the gain in PSNR (dB) obtained by replacing the DCT by the given transform. Curve line styles: black solid — LOT; black dashed — GenLOT ($L = 48$); black dash-dot — GLBT ($L = 16$); gray solid — FLT ($2 \times L = 24$ and $6 \times L = 8$); gray dashed — ELT-2 ($L = 32$); gray dash-dot — MLT ($L = 16$).

5.12.2 Embedded Zerotree Coding

The objective coding results (PSNR in dB) for the four test images ("Barbara," "Lena," "Goldhill," and "text") are tabulated in Table 5.5. The transforms in comparison are the ELT-2, MLT, DCT, LOT, GenLOT ($L = 40$), GLBT ($L = 16$), FLT (all with $M = 8$), and the 9/7-tap bi-orthogonal wavelet [2]. In the LT cases, we use three additional levels of 9/7 wavelet decomposition on the DC bands. All computed PSNR quotes in dB are obtained from a real compressed bit stream with all overheads included. Incremental rate-distortion plots are shown in Fig. 5.42 where the 9/7 wavelet serves as the performing benchmark.

The coding results clearly confirm the potential of LTs. For a smooth image such as "Lena," which the wavelet transform can sufficiently decorrelate, the 9/7 wavelet offers a comparable performance. However, for a highly-textured image such as

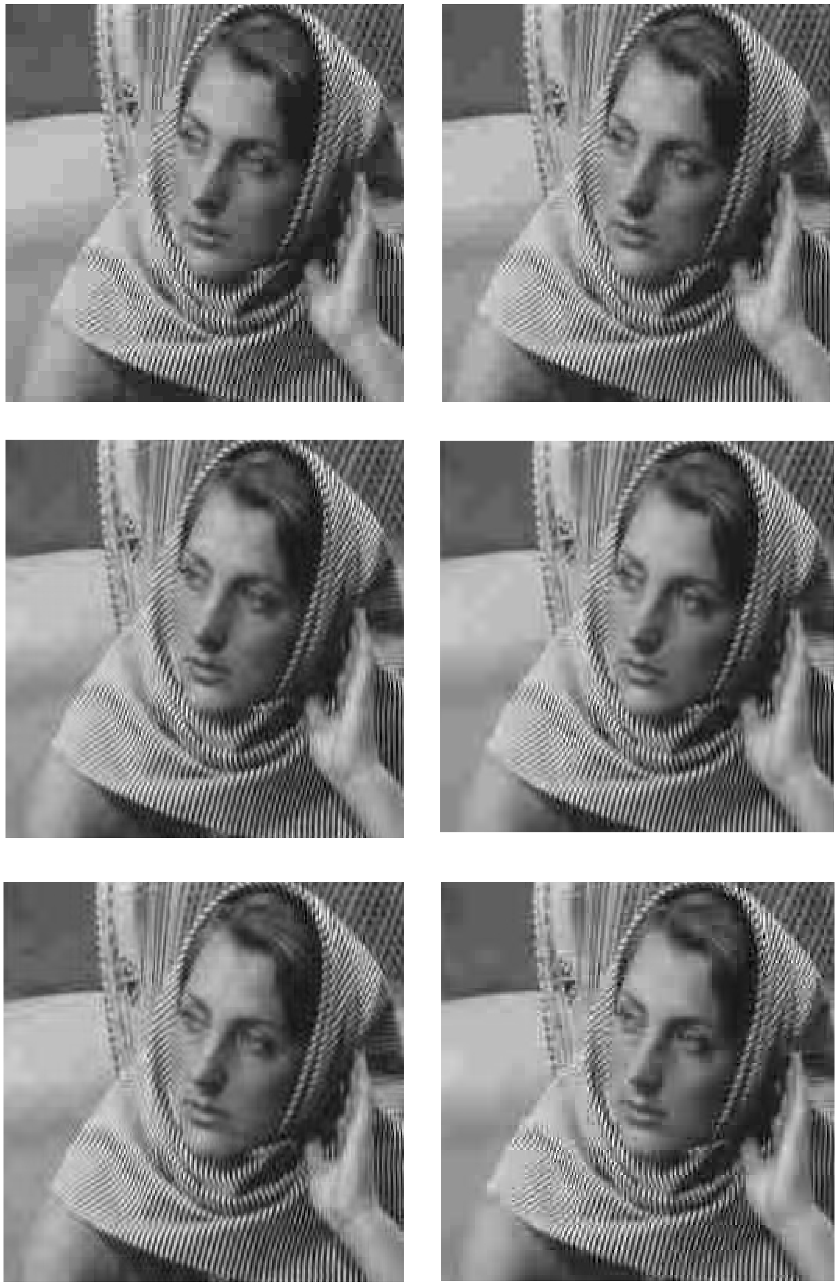


FIGURE 5.41

Enlarged portion of reconstructed images using JPEG at 0.3 bpp. Top left: DCT (25.67); top right: LOT (26.94); middle left: GenLOT, $L = 48$, (27.19); middle right: GLBT, $L = 16$, (26.88); bottom left: ELT-2 (27.24); bottom right: FLT, 2×24 and 6×8, (25.63). Number in parentheses indicates PSNR in dB. Reproduced by Special Permission of *Playboy* magazine. Copyright ©1972, 2000 by Playboy.

Table 5.5 Objective Coding Results (PSNR in dB). (a) Lena. (b) Goldhill. (c) Barbara. (d) Text

(a)

Lena	Transform							
Comp. Ratio	9 / 7 wavelet	8 x 8 DCT	8 x 16 LOT	8 x 40 GenLOT	2x24 6x8 FLT	8 x 16 GLBT	8 x 32 ELT	8 x 16 MLT
8:1	40.41	39.91	40.05	40.43	39.89	40.35	40.01	39.20
16:1	37.21	36.38	36.72	37.32	36.51	37.28	36.93	36.27
32:1	34.11	32.90	33.56	34.23	33.25	34.14	33.94	33.42
64:1	31.10	29.67	30.48	31.16	30.15	31.04	30.93	30.44
100:1	29.35	27.80	28.62	29.31	28.31	29.14	29.14	28.61

(b)

Goldhill	Transform							
Comp. Ratio	9 / 7 wavelet	8 x 8 DCT	8 x 16 LOT	8 x 40 GenLOT	2x24 6x8 FLT	8 x 16 GLBT	8 x 32 ELT	8 x 16 MLT
8:1	36.55	36.25	36.63	36.80	36.22	36.69	36.46	36.19
16:1	33.13	32.76	33.18	33.36	32.76	33.31	33.14	32.93
32:1	30.56	30.07	30.56	30.79	30.25	30.70	30.65	30.48
64:1	28.48	27.93	28.36	28.60	28.17	28.58	28.52	28.33
100:1	27.38	26.65	27.09	27.40	27.06	27.33	27.33	27.10

(c)

Barbara	Transform							
Comp. Ratio	9 / 7 wavelet	8 x 8 DCT	8 x 16 LOT	8 x 40 GenLOT	2x24 6x8 FLT	8 x 16 GLBT	8 x 32 ELT	8 x 16 MLT
8:1	36.41	36.31	37.43	38.08	36.22	37.84	37.53	37.07
16:1	31.40	31.11	32.70	33.47	31.12	33.02	33.12	32.59
32:1	27.58	27.28	28.80	29.53	27.42	29.04	29.33	28.74
64:1	24.86	24.58	25.70	26.37	24.86	26.00	26.30	25.65
100:1	23.76	23.42	24.34	24.95	23.74	24.55	24.90	24.28

(d)

Text	Transform							
Comp. Ratio	9 / 7 wavelet	8 x 8 DCT	8 x 16 LOT	8 x 40 GenLOT	2x24 6x8 FLT	8 x 16 GLBT	8 x 32 ELT	8 x 16 MLT
8:1	40.49	40.24	40.36	40.41	40.05	40.57	39.89	38.80
16:1	37.14	36.81	36.85	36.89	36.60	37.24	36.38	35.34
32:1	33.11	32.93	33.36	33.41	32.75	33.84	32.89	32.60
64:1	29.14	28.01	29.10	29.09	28.01	29.51	28.77	28.61
100:1	25.89	25.08	26.50	26.47	25.36	26.73	26.15	26.00

"Barbara," the 8 × 40 GenLOT and the 8 × 16 GLBT can provide a PSNR gain of more than 1.5 dB over a wide range of bit rates. Fig. 5.43 demonstrates the high level of reconstructed image quality as well. The ELT, GLBT, and GenLOT can completely eliminate blocking artifacts.

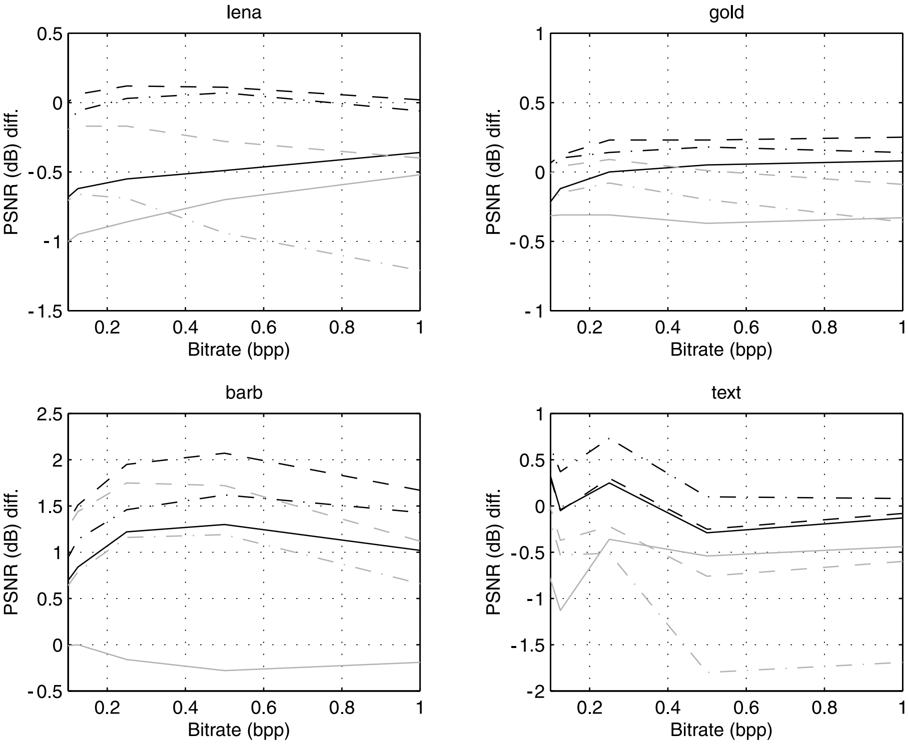


FIGURE 5.42

Comparison among transforms in EZC for several images and bit rates. Incremental PSNR plots are shown, indicating the gain (or loss) in PSNR (dB) obtained by replacing the 9/7 bi-orthogonal wavelet by the given transform. Curve lines: black solid — LOT; black dashed — GenLOT ($L = 40$); black dash-dot — GLBT ($L = 16$); gray solid — FLT ($2 \times L = 24$ and $6 \times L = 8$); gray dashed — ELT-2 ($L = 32$); gray dash-dot — MLT ($L = 16$).

5.13 Conclusions

We hope that this chapter serves as an eye-catching introduction to lapped transforms and their potentials in image/video compression. As for the theory of LTs, this

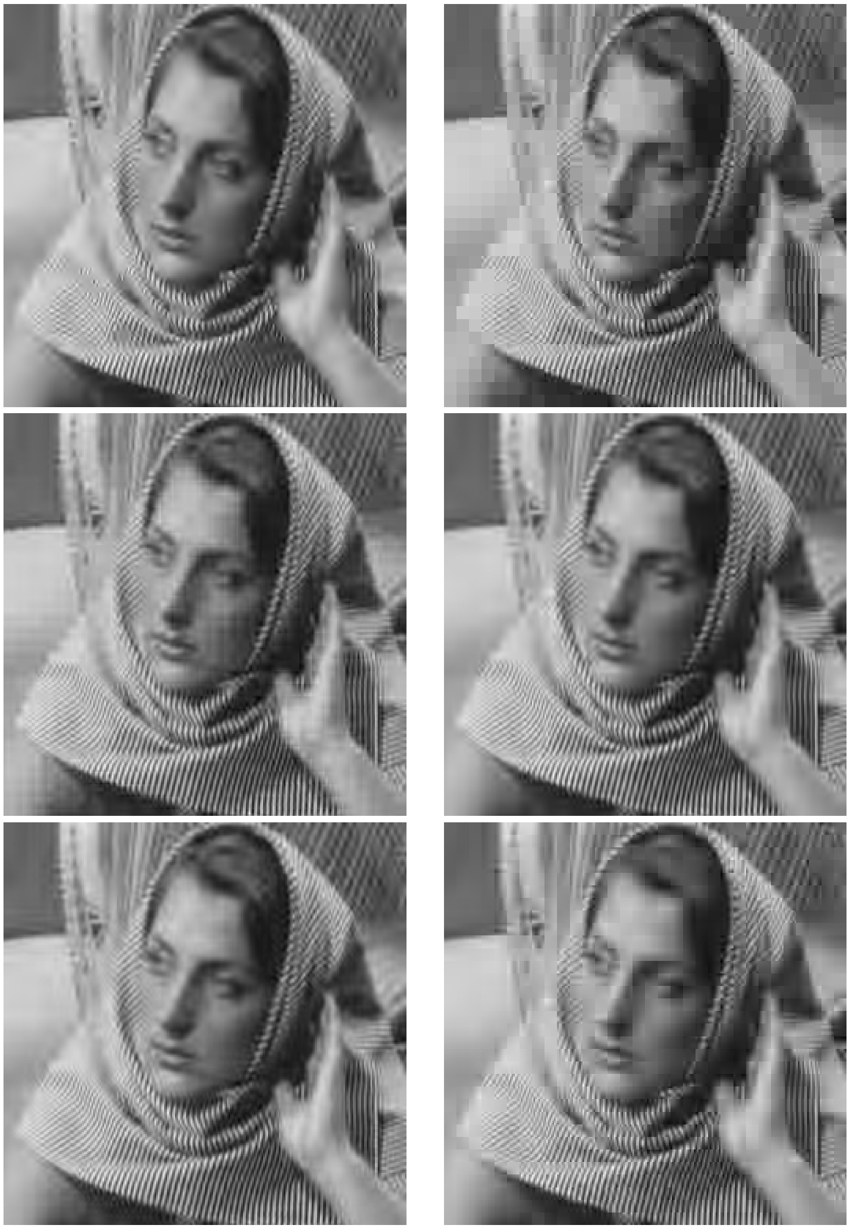


FIGURE 5.43

Perceptual coding comparison using EZC at a bit-rate of 0.25 bpp. The 9/7 wavelet transform is compared to various LTs and enlarged portions of the reconstructed “Barbara” image are shown. Top left: 9/7 bi-orthogonal wavelet (27.58); top right: DCT (27.28); middle left: LOT (28.80); middle right: GLBT, $L = 16$, (29.04); bottom left: GenLOT, $L = 40$, (29.53); bottom right: 2 × 24 and 6 × 8 FLT (27.42). Number in parentheses indicates PSNR in dB. Reproduced by Special Permission of *Playboy* magazine. Copyright ©1972, 2000 by Playboy.

chapter should be viewed as a first step, whereas the references and the references therein should give a more detailed treatment of the subject.

It was shown how lapped transforms can replace block transforms allowing an overlap of the basis functions. It was also shown that the analysis of MIMO systems, mainly their factorization, are invaluable tools for the design of useful lapped transforms. That was the case of transforms such as LOT, LBT, GenLOT, GLBT, FLT, MLT, and ELT. We presented these practical LTs by not only describing the general factorization, but also by plotting bases and discussing in detail how to construct at least a good design example. We made an effort in tabulating the bases entries or providing all parameters necessary to construct the bases. Even if the examples are not ideal for a particular application that the reader might have in mind, they may provide the basics upon which the reader can build by exploring the references and performing customized optimization. Invariably, the design examples presented here were tuned for image compression applications.

Some image compression methods were briefly described to serve as a comparison framework in which the LTs are applied for compression of typical images. Several LTs were compared to transforms such as DCT and wavelets, showing how truly promising LTs are for image compression. FLT and LBT with lifting steps require minimal computation apart from the DCT computation and are very attractive replacements for DCT, rivalling wavelet transforms at a lower implementation complexity. Buffering is also reduced since the transforms are not implemented hierarchically. Parallel computation and region-of-interest coding/decoding are also greatly facilitated.

It is worth noting that we intentionally avoided viewing the transforms as filter banks, so the bases were not discussed as impulse responses of filters, and their frequency response was not analyzed. Nevertheless, the framework is the same as is the analysis of MIMO systems. We trust that this chapter will give some insight into this vast field which is based on the study of multirate systems.

References

[1] Andrew, J., Simple and efficient hierarchical image coder, *Proc. of IEEE International Conference on Image Processing,* 3, 658–661, Santa Barbara, CA, Oct. 1997.

[2] Antonini, M., Barlaud, M., Mathieu, P., and Daubechies, I., Image coding using the wavelet transform, *IEEE Trans. on Image Processing,* 1, 205–220, 1992.

[3] Boashash, B., Ed., *Time-Frequency Signal Analysis,* John Wiley & Sons, New York, 1992.

[4] Bordreaux-Bartels, G.F., Mixed time-frequency signal transformations, *The Transforms and Applications Handbook,* Poularikas, A., Ed., CRC Press, Boca Raton, FL, 1996.

[5] Cassereau, P., *A New Class of Optimal Unitary Transforms for Image Processing,* Master's thesis, M.I.T., Cambridge, MA, May 1985.

[6] Clarke, R.J., *Transform Coding of Images,* Academic Press, Orlando, FL, 1985.

[7] Coifman, R., Meier, Y., Quaker, D., and Wickerhauser, V., Signal processing and compression using wavelet packets, Technical Report, Dept. of Mathematics, Yale Univ., New Haven, CT, 1991.

[8] Doğanata, Z., Vaidyanathan, P.P., and Nguyen, T.Q., General synthesis procedures for FIR lossless transfer matrices, for perfect reconstruction multirate filter banks applications, *IEEE Trans. Acoust., Speech, Signal Processing,* 36(10), 1561–1574, 1988.

[9] Herley, C., Kovacevic, J., Ramchandran, K., and Vetterli, M., Tilings of the time-frequency plane: construction of arbitrary orthogonal bases and fast tiling algorithms, *IEEE Trans. on Signal Processing,* 41, 3341–3359, 1993.

[10] Hohn, F.E., *Elementary Matrix Algebra,* 2nd ed., MacMillan, New York,1964.

[11] ISO/IEC JTC1/SC29/WG1, JPEG 2000 Committee, Working Draft 2.0, June 25, 1999.

[12] Jayant, N.S. and Noll, P., *Digital Coding of Waveforms,* Prentice-Hall, Englewood Cliffs, NJ, 1984.

[13] Joshi, R.L., Crump, V.J., and Fisher, T.R., Image subband coding using arithmetic coded trellis coded quantization, *IEEE Trans. Circuits and Systems for Video Technology,* 5, 515–523, 1995.

[14] Jozawa, H. and Watanabe, H., Intrafield/Interfield adaptive lapped transform for compatible HDTV coding, *4th International Workshop on HDTV and Beyond,* Torino, Italy, Sept. 4–6, 1991.

[15] Katto, J. and Yasuda, Y., Performance evaluation of subband coding and optimization of its filter coefficients, *SPIE Proc. Visual Comm. and Image Proc.,* 1991.

[16] Klausutis, T.J. and Madisetti, V.K., Variable block size adaptive lapped transform-based image coding, *Proc. of IEEE International Conference on Image Processing,* 3, 686–689, Santa Barbara, CA, Oct. 1997.

[17] Koilpillai, R.D. and Vaidyanathan, P.P., Cosine modulated FIR filter banks satisfying perfect reconstruction, *IEEE Trans. Signal Processing,* 40, 770–783, 1992.

[18] Malvar, H.S., *Optimal pre- and post-filtering in noisy sampled-data systems,* Ph.D. dissertation, M.I.T., Cambridge, MA, Aug. 1986.

[19] Malvar, H.S., Reduction of blocking effects in image coding with a lapped orthogonal transform, *Proc. of Intl. Conf. on Acoust., Speech, Signal Processing,* Glasgow, Scotland, 781–784, Apr. 1988.

[20] Malvar, H.S. and Staelin, D.H., The LOT: transform coding without blocking effects, *IEEE Trans. Acoust., Speech, Signal Processing,* ASSP-37, 553–559, 1989.

[21] Malvar, H.S., Lapped transforms for efficient transform/subband coding, *IEEE Trans. Acoust., Speech, Signal Processing,* ASSP-38, 969–978, 1990.

[22] Malvar, H.S., The LOT: a link between block transform coding and multirate filter banks, *Proc. Intl. Symp. Circuits and Systems,* Espoo, Finland, 835–838, June 1988.

[23] Malvar, H.S., Efficient signal coding with hierarchical lapped transforms, *Proc. of Intl. Conf. on Acoust., Speech, Signal Processing,* Albuquerque, NM, 761–764, 1990.

[24] Malvar, H.S., Modulated QMF filter banks with perfect reconstruction, *Elect. Letters,* 26, 906–907, 1990.

[25] Malvar, H.S., Extended lapped transform: fast algorithms and applications, *Proc. of Intl. Conf. on Acoust., Speech, Signal Processing,* Toronto, Canada, 1797–1800, 1991.

[26] Malvar, H.S., *Signal Processing with Lapped Transforms,* Artech House, Norwood, MA, 1992.

[27] Malvar, H.S., Extended lapped transforms: properties, applications and fast algorithms, *IEEE Trans. Signal Processing,* 40, 2703–2714, 1992.

[28] Malvar, H.S., Biorthogonal and nonuniform lapped transforms for transform coding with reduced blocking and ringing artifacts, *IEEE Trans. on Signal Processing,* 46, 1043–1053, 1998.

[29] Nayebi, K., Barnwell, T.P., and Smith, M.J., The time domain filter bank analysis: a new design theory, *IEEE Trans. on Signal Processing*, 40, 1412–1429, 1992.

[30] Nguyen, T.Q. and Koilpillai, R.D., Theory and design of arbitrary-length cosine-modulated filter banks and wavelets satisfying perfect reconstruction, *IEEE Trans. on Signal Processing,* 44, 473–483, 1996.

[31] Oppenheim, A.V. and Schafer, R.W., *Discrete-Time Signal Processing,* Prentice-Hall, Englewood Cliffs, NJ, 1989.

[32] Pennebaker, W.B. and Mitchell, J.L., *JPEG: Still Image Compression Standard,* Van Nostrand Reinhold, New York, 1993.

[33] Princen, J.P. and Bradley, A.B., Analysis/synthesis filter bank design based on time domain aliasing cancellation, *IEEE Trans. Acoust., Speech, Signal Processing,* ASSP-34, 1153–1161, 1986.

[34] de Queiroz, R.L. and Rao, K.R., Time-varying lapped transforms and wavelet packets, *IEEE Trans. on Signal Processing,* 41, 3293–3305, 1993.

[35] de Queiroz, R.L., *On Lapped Transforms,* Ph.D. dissertation, The University of Texas at Arlington, August 1996.

[36] de Queiroz, R.L. and Rao, K.R., The extended lapped transform for image coding, *IEEE Trans. on Image Processing,* 4, 828–832, 1995.

[37] de Queiroz, R.L., Nguyen, T.Q., and Rao, K.R., The generalized lapped orthogonal transforms, *Electronics Letters,* 30, 107–107, 1994.

[38] de Queiroz, R.L. and Rao, K.R., On orthogonal transforms of images using paraunitary filter banks, *J. Visual Communications and Image Representation,* 6(2), 142–153, 1995.

[39] de Queiroz, R.L. and Rao, K.R., On reconstruction methods for processing finite-length signals with paraunitary filter banks, *IEEE Trans. on Signal Processing,* 43, 2407–2410, 1995.

[40] de Queiroz, R.L., Nguyen, T.Q., and Rao, K.R., The GenLOT: generalized linear-phase lapped orthogonal transform, *IEEE Trans. on Signal Processing,* 44, 497–507, 1996.

[41] de Queiroz, R.L., Uniform filter banks with non-uniform bands: post-processing design, *Proc. of Intl. Conf. Acoust. Speech. Signal Proc.,* Seattle, WA, III, 1341–1344, May 1997.

[42] de Queiroz, R.L. and Eschbach, R., Fast downscaled inverses for images compressed with M-channel lapped transforms, *IEEE Trans. on Image Processing,* 6, 794–807, 1997.

[43] Rabbani, M. and Jones, P.W., *Digital Image Compression Techniques,* SPIE Optical Engineering Press, Bellingham, WA, 1991.

[44] Ramstad, T.A., Aase, S.O., and Husoy, J.H., *Subband Compression of Images: Principles and Examples,* Elsevier, New York, 1995.

[45] Rao, K.R. and Yip, P., *Discrete Cosine Transform: Algorithms, Advantages, Applications,* Academic Press, San Diego, CA, 1990.

[46] Rao, K.R., Ed., *Discrete Transforms and Their Applications,* Van Nostrand Reinhold, New York,1985.

[47] Said, A. and Pearlman, W.A., A new fast and efficient image codec based on set partitioning in hierarchical trees, *IEEE Trans. on Circuits Syst. Video Tech.,* 6, 243–250, 1996.

[48] Schiller, H., Overlapping block transform for image coding preserving equal number of samples and coefficients, *Proc. SPIE, Visual Communications and Image Processing,* 1001, 834–839, 1988.

[49] Schuller, G.D.T. and Smith, M.J.T., New framework for modulated perfect reconstruction filter banks, *IEEE Trans. on Signal Processing,* 44, 1941–1954, 1996.

[50] Shapiro, J.M., Embedded image coding using zerotrees of wavelet coefficients, *IEEE Trans. on Signal Processing,* 41, 3445–3462, 1993.

[51] Silva, E.A.B., Sampson, D.G., and Ghanbari, M., A successive approximation vector quantizer for wavelet transform image coding, *IEEE Trans. Image Processing,* 5, 299–310, 1996.

[52] Sodagar, I., Nayebi, K., and Barnwell, T.P., A class of time-varying wavelet transforms, *Proc. of Intl. Conf. on Acoust., Speech, Signal Processing,* Minneapolis, MN, III, 201–204, Apr. 1993.

[53] Soman, A.K. and Vaidyanathan, P.P., Paraunitary filter banks and wavelet packets, *Proc. of Intl. Conf. on Acoust., Speech, Signal Processing,* IV, 397–400, 1992.

[54] Soman, A.K., Vaidyanathan, P.P., and Nguyen, T.Q., Linear-phase paraunitary filter banks: theory, factorizations and applications, *IEEE Trans. on Signal Processing,* 41, 3480–3496, 1993.

[55] Strang, G. and Nguyen, T., *Wavelets and Filter Banks,* Wellesley-Cambridge, Wellesley, MA, 1996.

[56] Temerinac, M. and Edler, B., A unified approach to lapped orthogonal transforms, *IEEE Trans. Image Processing,* 1, 111–116, 1992.

[57] Tran, T.D. and Nguyen, T.Q., On M-channel linear-phase FIR filter banks and application in image compression, *IEEE Trans. on Signal Processing,* 45, 2175–2187, 1997.

[58] Tran, T.D. and Nguyen, T.Q., A progressive transmission image coder using linear phase uniform filter banks as block transforms, *IEEE Trans. on Image Processing,* in press.

[59] Tran, T.D., *Linear Phase Perfect Reconstruction Filter Banks: Theory, Structure, Design, and Application in Image Compression,* Ph.D. thesis, University of Wisconsin, Madison, WI, May 1998.

[60] Tran, T.D., de Queiroz, R.L., and Nguyen, T.Q., The variable-length generalized lapped biorthogonal transform, *Proc. Intl. Conf. Image Processing,* Chicago, IL, III, 697–701, 1998.

[61] Tran, T.D., de Queiroz, R.L., and Nguyen, T.Q., The generalized lapped biorthogonal transform, *Proc. Intl. Conf. Acoust., Speech Signal Proc.,* Seattle, III, 1441–1444, May 1998.

[62] Tran, T.D., The LiftLT: fast lapped transforms via lifting steps, *Proc. IEEE Int. Conf. on Image Processing,* Kobe, Japan, Oct. 1999.

[63] Vaidyanathan, P.P., *Multirate Systems and Filter Banks.* Prentice-Hall, Englewood Cliffs, NJ, 1993.

[64] Vaidyanathan, P.P. and Hoang, P., Lattice structures for optimal design and robust implementation of 2-channel PR-QMF banks, *IEEE Trans. Acoust., Speech, Signal Processing,* ASSP-36, 81–94, 1988.

[65] Vetterli, M. and Herley, C., Wavelets and filter banks: theory and design, *IEEE Trans. Signal Processing,* 40, 2207–2232, 1992.

[66] Vetterli, M. and Kovacevic, J., *Wavelets and Subband Coding,* Prentice-Hall, Englewood Cliffs, NJ, 1995.

[67] Wang, H. and Kuo, C.-C.J., A multi-threshold wavelet coder (MTWC) for high fidelity image compression, *Proc. of IEEE Int. Conf. on Image Processing,* 1, 652–655, Santa Barbara, CA, Oct. 1997.

[68] Wickerhauser, M.V., Acoustical signal compression using wavelet packets, in *Wavelets: A Tutorial in Theory and Applications,* Chui, C.K., Ed., Academic Press, San Diego, CA, 1992.

[69] Xiong, Z., Ramchandran, K., and Orchard, M.T., Space frequency quantization for wavelet image coding, *IEEE Trans. Image Processing,* 6, 677–693, 1997.

[70] Young, R.W. and Kingsbury, N.G., Frequency domain estimation using a complex lapped transform, *IEEE Trans. Image Processing,* 2, 2–17, 1993.

[71] Zandi, A., Allen, J., Schwartz, E., and Boliek, M., CREW: compression with reversible embedded wavelets, *Proc. IEEE Data Compression Conf.,* Snowbird, UT, 212–221, 1995.