de Queiroz, R.L. “Lapped Transforms.”
The Transforms and Applications Handbook: Second Edition.
Ed. Alexander D. Poularikas
Boca Raton: CRC Press LLC, 2000

14

Lapped Transforms

Ricardo L. deQueiroz
Xerox Corporation

14.1 Introduction

In this chapter, an effort will be made to cover the basic aspects of lapped transforms. It is a subject that has been extensively studied, making available a large number of papers and books. This is mostly true because of the direct correspondence among lapped transforms, filter banks, wavelets, and time-frequency transformations. Some of those topics are well covered in other chapters in this handbook. In any case it will be 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 chapters on wavelet and time-frequency transforms in this handbook, as well as References 50, 20, 53, and 44 for a more detailed treatment of filter banks.

We expect the reader to have a background in digital signal processing. An introductory chapter in this handbook on signals and systems, the chapter on Z-transforms, and the chapter on the discrete cosine transform are certainly useful.

14.1.1 Notation

In terms of notation, our conventions are: $\mathbf{I}_n$ is the $n \times n$ identity matrix. $\mathbf{O}_n$ is the $n \times n$ null matrix, while $\mathbf{O}_{n \times m}$ 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:

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

J reverses the order of elements of a vector. $[\ ]^T$ means transposition. $[\ ]^H$ means transposition combined with conjugation, where this combination is usually called the Hermitian 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 **a** represents a (column) vector while **A** represents a matrix.

14.1.2 Brief History

In the early 80's transform coding was maturing itself and the discrete cosine transform (DCT)[38] 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 at images which were compressed at low bit rates using block transforms. To resolve this problem, the idea of a lapped transform (LT, for short) was developed in the early 1980s at MIT. 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, but the new ingredient to overlapping blocks would be the fact that the number of transform coefficients would be the same as if there was no overlap, and that the transform would maintain orthogonality. Cassereau[3] introduced the lapped orthogonal transform (LOT). However, it was Malvar[12-14] 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 later pointed by Malvar[16] the equivalence between a LOT and a multirate filter bank which is now a very popular signal processing tool[50]. Based on cosine modulated filter banks[27], modulated lapped transforms were designed.[15,40] Modulated transforms were generalized for an arbitrary overlap later, creating the class of extended lapped transforms (ELT).[18-21] Recently a new class of LTs with symmetric bases were developed yielding the class of generalized LOTs (GenLOTs).[29,31,34] The GenLOTs were made to have an arbitrary length (not a multiple of the block size),[46] extended to the non-orthogonal case[49] and even made to have filters of different lengths.[48] As we mentioned, 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[50] is considered an excellent text on filter banks, while Malvar's book[20] is a good reference to bridge the gap between lapped transforms and filter banks. We, however, refer to LTs for uniform FIR filter banks with fast implementation algorithms based on special factorizations of the basis functions, with particular design attention for signal (mainly image) coding.

14.1.3 Block Transforms

We assume a one-dimensional input sequence $x(n)$ which is transformed into several coefficients $y_i(n)$, where $y_i(n)$ would belong to the i-th subband. In traditional block-transform processing, such as in image and audio coding, the signal is divided into blocks of M samples, and each block is processed independently.[4,9,20,26,37-39] Let the samples in the m-th block be denoted as:

$$\mathbf{x}_m^T = [\, x_0(m), x_1(m), \ldots, x_{M-1}(m)\,], \tag{14.1}$$

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

$$\mathbf{y}_m^T = [\, y_0(m), y_1(m), \ldots, y_{M-1}(m)\,]. \tag{14.2}$$

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

$$\mathbf{y}_m = \mathbf{A}\mathbf{x}_m \,, \tag{14.3}$$

and

$$\mathbf{x}_m = \mathbf{A}^T y_m \,. \tag{14.4}$$

The rows of **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.[39] The transform vector coefficients $[y_0(m), y_1(m), \ldots, y_{M-1}(m)]$ represent the corresponding weights of vector $\mathbf{x}_m$ with respect to this basis.

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

$$\mathbf{H} = diag\{\ldots, \mathbf{A}, \mathbf{A}, \mathbf{A}, \ldots\} \,, \tag{14.5}$$

where, of course, **H** is an orthogonal matrix if **A** is also an orthogonal matrix. In summary, a signal is transformed by segmentation into blocks followed by transformation, which amounts to transforming the signal with a sparse matrix. Also, it is well known that the signal energy is preserved under an orthogonal transformation,[9,38] assuming stationary signals, i.e.,

$$M\sigma_x^2 = \sum_{i=0}^{M-1} \sigma_i^2 \,, \tag{14.6}$$

where $\boldsymbol{\sigma}_i^2$ is the variance of $y_i(m)$ and $\boldsymbol{\sigma}_x^2$ is the variance of the input samples.

14.1.4 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 by a product of plane (Givens) rotations.[8] Let **A** be an $M \times M$ real orthogonal matrix and let $\boldsymbol{\Theta}(i, j, \theta_n)$ be a matrix with entries Θ_{kl} which is like the identity matrix $\mathbf{I}_M$ except for four entries:

$$\Theta_{ii} = \cos(\theta_n) \;\; \Theta_{jj} = \cos(\theta_n) \;\; \Theta_{ij} = \sin(\theta_n) \;\; \Theta_{ji} = -\sin(\theta_n) \,, \tag{14.7}$$

i.e., $\boldsymbol{\Theta}(i, j, \theta_n)$ corresponds to a plane rotation along the i-th and the j-th axes by an angle θ_n). Then, **A** can be factorized as:

$$\mathbf{A} = \mathbf{S}\prod_{i=0}^{M-Z}\prod_{j=i+1}^{M-1}\Theta(i,j,\theta_n) \tag{14.8}$$

where n is increased by one for every matrix and **S** is a diagonal matrix with entries ± 1 to correct for any sign error.[8] This correction is not necessary in most cases and is not required if we could apply variations of the rotation matrix defined in (14.7) as:

$$\Theta_{ii} = \cos(\theta_n) \;\; \Theta_{jj} = -\cos(\theta_n) \;\; \Theta_{ij} = \sin(\theta_n) \;\; \Theta_{ji} = \sin(\theta_n) \,. \tag{14.9}$$

All combinations of pairs of axes shall be used for a complete factorization. Figure 14.1(a) shows an example of the factorization of a 4×4 orthogonal matrix into plane rotations (the order differs from

that in (14.8), but the factorization is also complete). If the matrix is not orthogonal, we can always decompose the matrix using singular value decomposition (SVD).[8] **A** is decomposed through SVD as:

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{V} \tag{14.10}$$

where **U** and **V** are orthogonal matrices and **Λ** is a diagonal matrix containing the singular values of **A**. While **Λ** is already a sparse matrix, we can further decompose the orthogonal matrices using (14.8), i.e.,

$$A = S\left(\prod_{i=0}^{M-2}\prod_{j=i+1}^{M-1}\Theta(i,j,\theta_n^U)\right)\Lambda\left(\prod_{i=0}^{M-1}\prod_{j=i+1}^{M-1}\Theta(i,j,\theta_n^V)\right) \tag{14.11}$$

where θ_n^U and θ_n^V compose the set of angles for **U** and **V**, respectively. Figure 14.1(c) illustrates the factorization for a 4×4 non-orthogonal matrix, where α_i are the singular values.

The factorization is an invaluable tool for the design of block and lapped transforms as we will explain later. In the orthogonal case the angles are all the degrees of freedom. In an $M \times M$ orthogonal matrix, there are $M(M-1)/2$ angles, and by spanning all the angle spaces (0 to 2π for each one) one spans the space of all $M \times M$ orthogonal matrices. The idea is to span the angles in order to design orthogonal matrices through unconstrained optimization. In the general case, there are M^2 degrees of freedom either by utilizing the matrix entries directly or by using the SVD decomposition. However, we are mainly concerned with invertible matrices. Using the SVD-based method, one can design invertible matrices by freely spanning the angles, with the only mild constraint to assure that all singular values are not zero. The author commonly uses unconstrained non-linear optimization based on simplex search provided by MATLAB™ to span all angles and possibly singular values as well.

14.1.5 Discrete MIMO Linear Systems

Let a multi-input multi-output (MIMO)[50] discrete linear FIR system have M input and M output sequences with respective Z-transforms $X_i(z)$ and $Y_i(z)$, for $0 \le i \le M-1$. Then, $X_i(z)$ and $Y_i(z)$ are related by:

$$\begin{bmatrix} Y_0(z) \\ Y_1(z) \\ \vdots \\ Y_{M-1}(z) \end{bmatrix} = \begin{bmatrix} E_{0,0}(z) & E_{0,1}(z) & \cdots & E_{0,M-1}(z) \\ E_{1,0}(z) & E_{1,1}(z) & \cdots & E_{1,M-1}(z) \\ \vdots & \vdots & \ddots & \vdots \\ E_{M-1,0}(z) & E_{M-1,1}(z) & \cdots & E_{M-1,M-1}(z) \end{bmatrix} \begin{bmatrix} X_0(z) \\ X_1(z) \\ \vdots \\ X_{M-1}(z) \end{bmatrix} \tag{14.12}$$

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 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, causal FIR filters are assumed.

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

$$\mathbf{E}^H(e^{j\omega})\mathbf{E}(e^{j\omega}) = \mathbf{E}(e^{j\omega})\mathbf{E}^H(e^{j\omega}) = \mathbf{I}_M \,. \tag{14.13}$$

Furthermore:

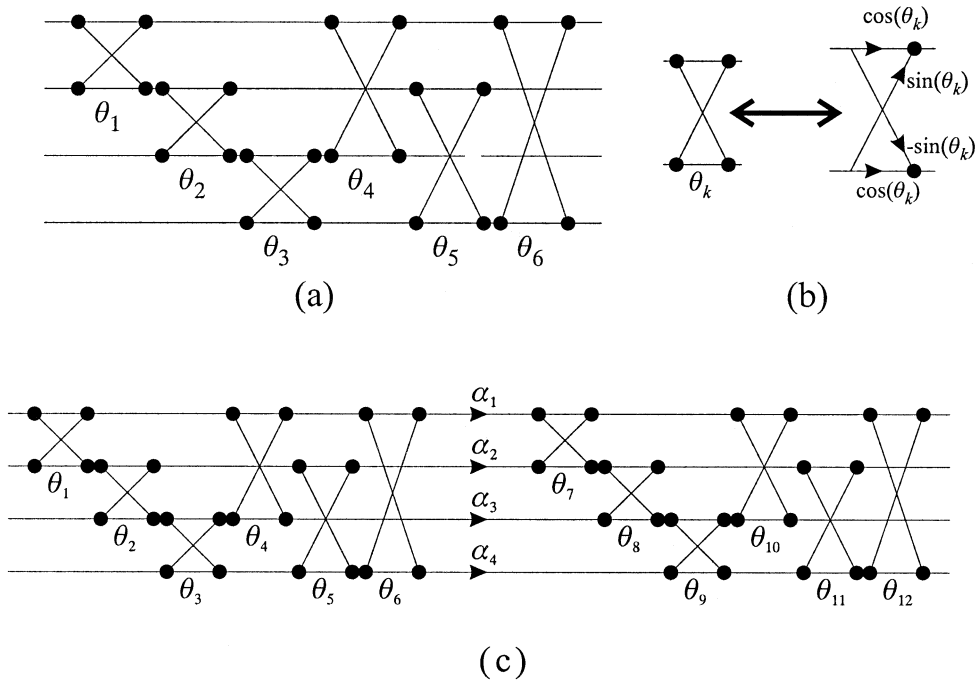

FIGURE 14.1 Factorization of a 4×4 matrix. (a) Orthogonal factorization into Givens rotations. (b) Detail of the rotation element. (c) Factorization of a nonorthogonal matrix through SVD with the respective factorization of SVD's orthogonal factors into rotations.

$$\mathbf{E}^{-1}(z) = \mathbf{E}^{T}(z^{-1}) . \tag{14.14}$$

For causal inverses of paraunitary systems,

$$\mathbf{E}'(z) = z^{-n}\mathbf{E}^{T}(z^{-1}) \tag{14.15}$$

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,[50] 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.[50] In fact, an orthogonal matrix is one where all $E_{ij}(z)$ are constant for all z.

We also have interest in invertible, although non-paraunitary, transfer matrices. In this case, it is required that the matrix be invertible in the unit circle, i.e., for all $z = e^{j\omega}$, ω real. Non-paraunitary systems are also called bi-orthogonal or perfect reconstruction (PR).[50]

14.1.6 Block Transform as a MIMO System

The sequences $x_i(m)$ in (14.1) are called the polyphase components of the input signal $x(n)$. In the other hand, the sequences $y_i(m)$ in (14.2) 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 Figure 14.2. 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 output 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

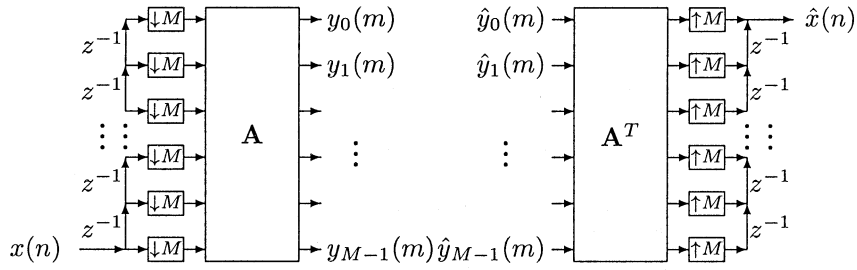

FIGURE 14.2 The signal samples are parallelized into polyphase components through a sequence of delays and decimators ($\downarrow M$ means subsampling by a factor of M). Effectively the signal is "blocked" and each block is transformed by system **A** into M subband samples (transformed samples). Inverse 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 ($\uparrow M$ means subsampling by a factor of M, padding the signal with $M-1$ zeros) and delays.

0, i.e., $\mathbf{E}(z) = \mathbf{A}$, and if **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).

14.2 Lapped Transforms

The motivation for a transform with overlap as we mentioned in the introduction was to try to improve the performance of block (non-overlapped) transforms for image and signal compression. Compression commonly implies signal losses due to quantization.[9] 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 in each side of the border may cause an artificial "edge" between blocks, the so-called *blocking* effect. In Figure 14.3 is shown an example signal which 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). The left side of Figure 14.3 shows translated versions of the first basis of the DCT, in order to account for all the different blocks. In the same figure, on the right, is shown 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.

Although Figure 14.3 shows just one basis for either DCT or LT, there are M of them. An example of the bases for $M = 8$ is shown in Figure 14.4. It shows the bases for the DCT and for the LOT, which is a particular LT that will be discussed later. The reader may note that not only are the LOT bases longer, but they are also smoother than their DCT counterparts. Figure 14.5(a) shows an example of an image compressed using the standard JPEG baseline coder,[26] where the reader can readily perceive the blocking artifacts at the boundaries of 8×8 pixel blocks. By replacing the DCT with the LOT at the same compression ratio, we obtain the image shown in Figure 14.5(b), where blocking is largely reduced. This brief introduction to the motivation behind the development of LTs helps to illustrate the overall problem, without detail on how to apply LTs. In this section we will develop the LT framework.

14.2.1 Orthogonal Lapped Transforms

For lapped transforms,[20] the basis vectors can 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[20] 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 **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:

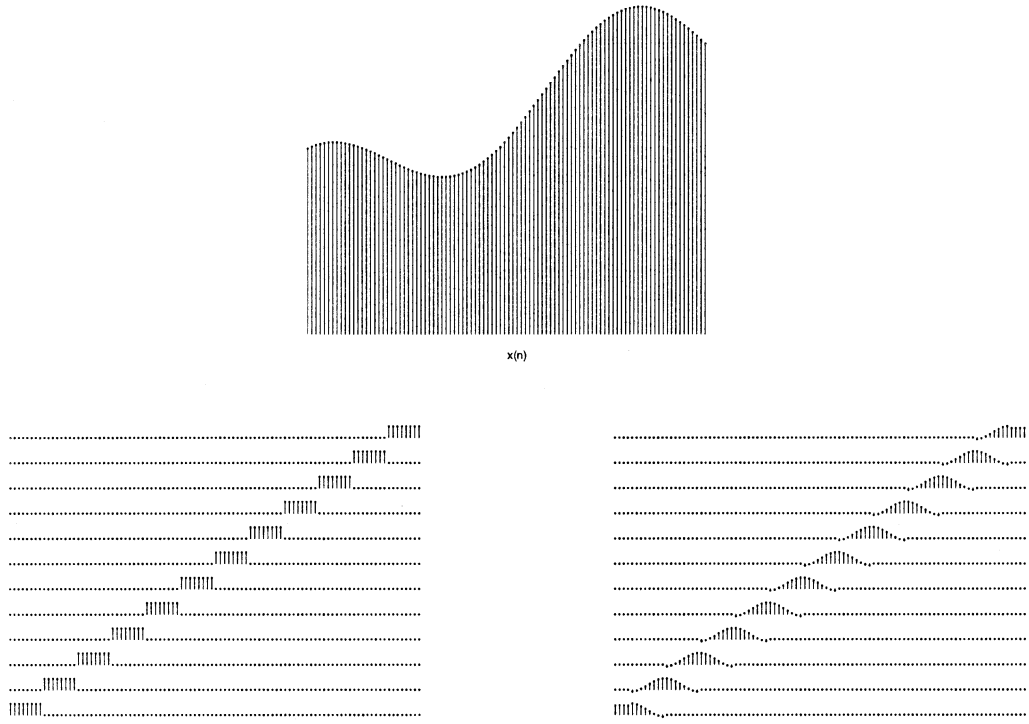

FIGURE 14.3 The example discrete signal on top is to be projected into a number of bases. Left: spatially displaced versions of the first DCT basis. Right: spatially displaced versions of the first basis of a typical short LT.

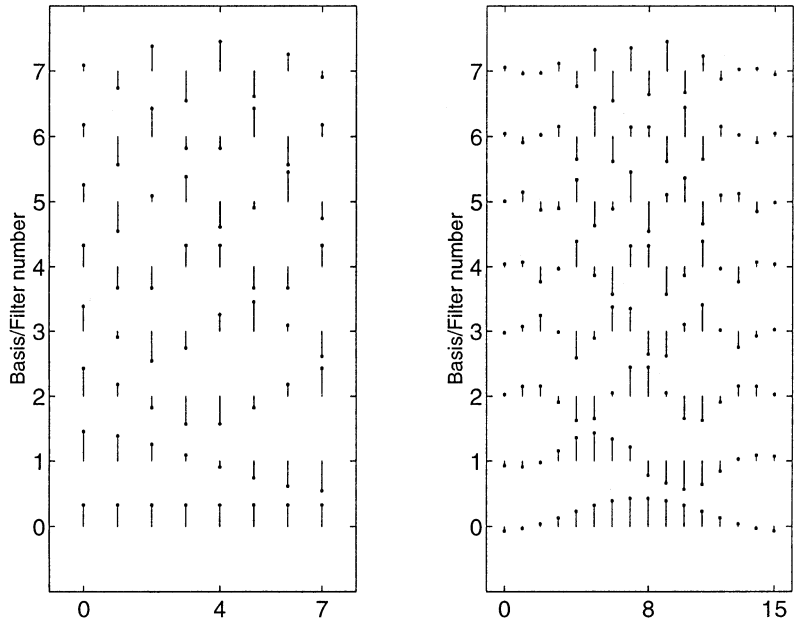

FIGURE 14.4 Bases for the 8-point DCT ($M = 8$) (left) and for the LOT (right) with $M = 8$. The LOT is a particular LT which will be explained later.

FIGURE 14.5 Zoom of image compressed using JPEG at 0.5 bit per pixel. (a) DCT, (b) LOT.

$$\mathbf{P} = [\mathbf{P}_0, \mathbf{P}_1, \cdots, \mathbf{P}_{N-1}] \,. \tag{14.16}$$

The orthogonality property does not hold because **P** is no longer a square matrix and it is replaced by the perfect reconstruction (PR) property[20], defined by:

$$\sum_{i=0}^{N-1-l} \mathbf{P}_i \mathbf{P}_{i+l}^T = \sum_{i=0}^{N-1-l} \mathbf{P}_{i+l}^T \mathbf{P}_i = \delta(l)\mathbf{I}_M \,, \tag{14.17}$$

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 we will see later (14.17) 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 $\mathbf{y}_m$ such as in (14.1) and (14.2). These blocks are not used by LTs in a straightforward manner. The actual vector which 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. These samples are chosen by picking $(L - M)/2$ samples at each side of the block $\mathbf{x}_m$, as shown in Figure 14.6, for $N = 2$. However, the number 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:

$$\mathbf{v}_m^T = \left[x\left(mM - (N-1)\frac{M}{2} \right) \cdots x\left(mM + (N+1)\frac{M}{2} - 1 \right) \right] . \tag{14.18}$$

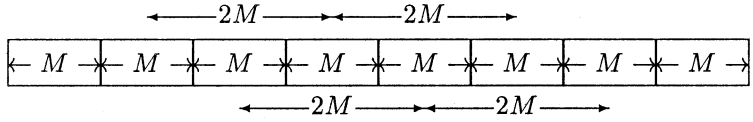

FIGURE 14.6 The signal samples are divided into blocks of M samples. The lapped transform uses neighboring block samples, as in this example for $N = 2$, i.e., $L = 2M$, yielding an overlap of $(L - M)/2 = M/2$ samples on either side of a block.

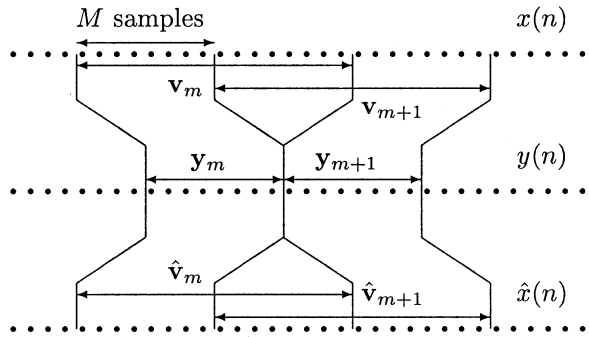

FIGURE 14.7 Illustration of a lapped transform with $N = 2$ applied to signal $x(n)$, yielding transform domain signal $y(n)$. The input L-tuple as vector $\mathbf{v}_m$ is obtained by a sliding window advancing M samples, generating $\mathbf{y}_m$. This sliding is also valid for the synthesis side.

Then, we have:

$$\mathbf{y}_m = \mathbf{P}\mathbf{v}_m . \tag{14.19}$$

The inverse transform is not direct as in the case of block transforms, i.e., with the knowledge of $\mathbf{y}_m$ we do not know the samples in the support region of $\mathbf{v}_m$, or in the support region of $\mathbf{x}_m$. We can reconstruct a vector $\hat{\mathbf{v}}_m$ from $\mathbf{y}_m$, as:

$$\hat{\mathbf{v}}_m = \mathbf{P}^T \mathbf{y}_m , \tag{14.20}$$

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 $\hat{\mathbf{v}}_m$, such that $x(n)$ was included in the region of support of the corresponding $\mathbf{v}_m$. This additional complication comes from the fact that $\mathbf{P}$ is not a square matrix.[20] However, the whole analysis-synthesis system (applied to the entire input vector) is orthogonal, assuring the PR property using (14.20).

We can also describe the process using a sliding rectangular window applied over the samples of $x(n)$. As an M-sample block $\mathbf{y}_m$ is computed using $\mathbf{v}_m$, $\mathbf{y}_{m+1}$ is computed from $\mathbf{v}_{m+1}$ which is obtained by shifting the window to the right by M samples, as shown in Figure 14.7.

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 fact later and assume infinite-length signals until then, or assume the length is very large and the borders of the signal are far enough from the region to which we are focusing our attention.

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 $\mathbf{y} = \mathbf{H}\mathbf{x}$ and $\hat{\mathbf{x}} = \mathbf{H}^T\mathbf{y}$. In this case, we have:

$$\mathbf{H} = \begin{bmatrix} \ddots & & & & 0 \\ & \mathbf{P} & & & \\ & & \mathbf{P} & & \\ & & & \mathbf{P} & \\ 0 & & & & \ddots \end{bmatrix} . \tag{14.21}$$

where the displacement of the matrices **P** obeys the following:

$$\mathbf{H} = \begin{bmatrix} \ddots & \ddots & & \ddots & & & 0 \\ & \mathbf{P}_0 & \mathbf{P}_1 & \cdots & \mathbf{P}_{N-1} & & \\ & & \mathbf{P}_0 & \mathbf{P}_1 & \cdots & \mathbf{P}_{N-1} & \\ 0 & & & \ddots & \ddots & & \ddots \end{bmatrix}. \tag{14.22}$$

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

Let the rows of **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:

$$y_i(m) = \mathbf{p}_i^T \mathbf{v}_m \,. \tag{14.23}$$

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 (14.17), the energy is preserved, such that (14.6) 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 (14.23):

$$E[y_i(m)] = \mathbf{p}_i^T E[\mathbf{v}_m] = \mathbf{p}_i^T \mathbf{0}_{L\times 1} = 0 \,, \tag{14.24}$$

so that:

$$\sigma_i^2 = E[y_i^2(m)] = \mathbf{p}_i^T E[\mathbf{v}_m \mathbf{v}_m^T] \mathbf{p}_i = \mathbf{p}_i^T \mathbf{R}_{xx} \mathbf{p}_i \,, \tag{14.25}$$

i.e., the output variance is easily computed from the input autocorrelation matrix for a given basis **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, then, from (14.21), we have:

$$\mathbf{y} = \mathbf{H}\mathbf{x} \tag{14.26}$$

and, if **H** is orthogonal,

$$\mathbf{x} = \mathbf{H}^T \mathbf{y} \,. \tag{14.27}$$

Note that **H** is orthogonal if and only if (14.17) is satisfied. Thus, the meaning for (14.17) becomes clear, as it forces the transform operating over the entire input-output signals to be orthogonal. So, the 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 (14.17), 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., no overlap. In fact, block transforms are a special case of lapped transforms and can be easily padded with zeroes. Similarly, basis functions can be increased by zero-padding as long as (14.17) is respected.

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:

$$\mathbf{v}_m^T = [\mathbf{x}_{m-N+1}^T, \ldots, \mathbf{x}_{m-1}^T, \mathbf{x}_m^T] , \tag{14.28}$$

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:

$$\mathbf{y}_m = \mathbf{P}\mathbf{v}_m = \sum_{i=0}^{N-1} \mathbf{P}_{N-1-i}\mathbf{x}_{m-i} . \tag{14.29}$$

Similarly, $\hat{\mathbf{v}}_m$ can be reconstructed as in (14.20) 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.

14.2.2 Nonorthogonal Lapped Transforms

So far, we have discussed orthogonal LTs. In those, a segment of the 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:

$$\mathbf{Q} = [\mathbf{Q}_0, \mathbf{Q}_1, \cdots, \mathbf{Q}_{N-1}] , \tag{14.30}$$

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 (14.20) is replaced by:

$$\hat{\mathbf{v}}_m = \mathbf{Q}^T\mathbf{y}_m . \tag{14.31}$$

We also define another transform matrix as:

$$\mathbf{H}' = \begin{bmatrix} \ddots & \ddots & & \ddots & & & 0 \\ & \mathbf{Q}_0 & \mathbf{Q}_1 & \cdots & \mathbf{Q}_{N-1} & & \\ & & \mathbf{Q}_0 & \mathbf{Q}_1 & \cdots & \mathbf{Q}_{N-1} & \\ 0 & & & \ddots & \ddots & & \ddots \end{bmatrix} . \tag{14.32}$$

The forward and inverse transformation are now:

$$\mathbf{y} = \mathbf{H}_F\mathbf{x}, \qquad \mathbf{x} = \mathbf{H}_I\mathbf{y} . \tag{14.33}$$

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:

$$\mathbf{H}'^T\mathbf{H} = \mathbf{I}_\infty . \tag{14.34}$$

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

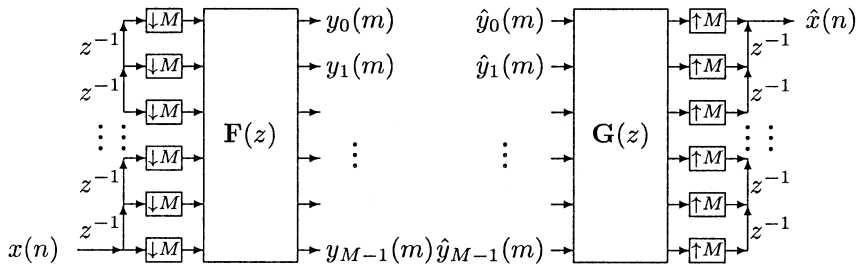

FIGURE 14.8 The filter bank represented as a MIMO system is applied to the polyphase components of the signal. The matrices $\mathbf{F}(z)$ and $\mathbf{G}(z)$ are called polyphase transfer matrices. For a PR system both must be inverses of each other and for paraunitary filter banks they must be paraunitary 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 $\mathbf{G}(z) = z^{-(N-1)}\,\mathbf{F}^{T}(z^{-1})$.

$$\sum_{k=0}^{N-1-m} \mathbf{Q}_k^T \mathbf{P}_{k+m} = \sum_{k=0}^{N-1-m} \mathbf{Q}_{k+m}^T \mathbf{P}_k = \delta(m)\mathbf{I}_M \ , \tag{14.35}$$

which establish general necessary and sufficient conditions for the perfect reconstruction of the signal by using **P** as a forward LT and **Q** as an inverse LT. Unlike the orthogonal case in (14.17), here both sets are necessary conditions, i.e., a total of $2N-1$ matrix equations.

14.3 LTs as MIMO Systems

As we discussed in Sec. 14.1.3 and Sec. 14.1.6, 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 some circumstances and since 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 (Sec. 14.1.6). The same is true for the inverse transform (from subbands $\hat{y}_i(m)$ to polyphase $\hat{\mathbf{x}}_i(m)$). Therefore, we can use the diagram shown in Figure 14.8 to represent the forward and inverse transforms. Note that Figure 14.8 is identical to Figure 14.2 except for the fact that the transforms have memory, i.e., depend not only on the present input vector, but also on past input vectors. 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 is performed by the chain of delays, upsamplers and downsamplers as shown in Figure 14.8. If we express the forward and inverse PTM as matrix polynomials:

$$\mathbf{F}(z) = \sum_{i=0}^{N-1} \mathbf{F}_i z^{-1} \ , \tag{14.36}$$

$$\mathbf{G}(z) = \sum_{i=0}^{N-1} \mathbf{G}_i z^{-1} \ , \tag{14.37}$$

then the forward and inverse transforms are given by:

$$\mathbf{y}_m = \sum_{i=0}^{N-1} \mathbf{F}_i \mathbf{x}_{m-i} \ , \tag{14.38}$$

$$\hat{\mathbf{x}}_m = \sum_{i=0}^{N-1} \mathbf{G}_i \hat{\mathbf{y}}_{m-i} \ . \tag{14.39}$$

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 that PR is possible if they are inverses of each other. Since the inverse of a causal FIR MIMO system may be non-causal, 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 (because N is the overlap factor of the equivalent LT), PR requires that:

$$\mathbf{G}(z)\mathbf{F}(z) = z^{-N+1}\mathbf{I}_M \rightarrow \mathbf{G}(z) = z^{-N+1}\mathbf{F}^{-1}(z) \ . \tag{14.40}$$

In this case, $\hat{\mathbf{x}} = \mathbf{x}_{m-N+1}$, i.e. 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 (14.38), (14.39) and (14.40) we can restate the PR conditions as:

$$\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \mathbf{G}_i \mathbf{F}_i z^{-i-j} = z^{-N+1}\mathbf{I}_M \ , \tag{14.41}$$

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

$$\sum_{k=0}^{N-1-m} \mathbf{G}_k \mathbf{F}_{k+m} = \sum_{k=0}^{N-1-m} \mathbf{G}_{k+m} \mathbf{F}_k = \delta(m)\mathbf{I}_M \ . \tag{14.42}$$

The reader should note the striking similarity of the above equation against (14.35). In fact, the simple comparison of the transformation process in space domain notation (14.33) against the MIMO system notation in (14.38) and (14.39) would reveal the following relations:

$$\mathbf{F}_k = \mathbf{P}_{N-1-k} \qquad \mathbf{G}_k = \mathbf{Q}_k^T \tag{14.43}$$

for $0 \le k < N$. In fact, the conditions imposed in (14.34), (14.35), (14.40), and (14.42) are equivalent and each one of them implies the others. This is a powerful tool in the design of lapped transforms. As an LT, the matrix is non-square 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 are a class of MIMO systems of interest. Let $\mathbf{E}(z)$ be a paraunitary PTM so that $\mathbf{E}^{-1}(z) = \mathbf{E}^T(z^{-1})$, and let:

$$\mathbf{F}(z) = \mathbf{E}(z), \qquad \mathbf{G}(z) = z^{-(N-1)}\mathbf{E}^T(z^{-1}) \ . \tag{14.44}$$

As a result, the reader can verify that it implies that $\mathbf{P}_i = \mathbf{Q}_i$ and that:

$$\sum_{i=0}^{N-1-l} \mathbf{P}_i \mathbf{P}_{i+l}^T = \sum_{i=0}^{N-1-l} \mathbf{P}_i^T \mathbf{P}_{i+l} = \delta(l)\mathbf{I}_M \ , \tag{14.45}$$

$$\mathbf{H}\mathbf{H}^T = \mathbf{H}^T\mathbf{H} = \mathbf{I}_\infty \ . \tag{14.46}$$

In other words, if the system's PTM is paraunitary, then the LT (**H**) is orthogonal and vice-versa.

14.4 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.[6,51] 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,

$$\mathbf{E}(z) = \mathbf{B}_0 \prod_{i=1}^{N_z} (\Upsilon(z)\mathbf{B}_i) \tag{14.47}$$

where $\Upsilon(z) = diag\{z^{-1}, 1, 1, \ldots, 1\}$, 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 $\mathbf{B}_{Nz}$ have only $M-1$ degrees of freedom.[52]

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, i.e., 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[29] 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 LTs whose bases have length up to $L = NM$, i.e., $\mathbf{E}(z)$ of order $N-1$.

In fact (14.47) may be able to implement all lapped transforms (orthogonal or not) whose degree is N_z. For that it is only required that all the multiplicative factors that compose the PTM are invertible. Let us consider a more particular factorization:

$$\mathbf{F}(z) = \prod_{i=0}^{(N-1)/(K-1)} \mathbf{B}_i(z) \tag{14.48}$$

where $\mathbf{B}_i(z) = \Sigma_{k=0}^{K-1} \mathbf{B}_{ik} z^{-k}$ 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:

$$\mathbf{G}(z) = z^{-N+1}\mathbf{F}^T(z^{-1}) = \prod_{i=(N-1)/(K-1)}^{0} \left(\sum_{k=0}^{K-1} \mathbf{B}_{ik}^T z^{-(K-1-k)} \right). \tag{14.49}$$

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

$$\mathbf{C}_i(z)\mathbf{B}_i(z) = z^{-K+1}\mathbf{I}_M . \tag{14.50}$$

Being that the case, the inverse PTM is simply given by:

$$\mathbf{G}(z) = \prod_{i=(N-1)/(K-1)}^{0} \mathbf{C}_i(z) . \tag{14.51}$$

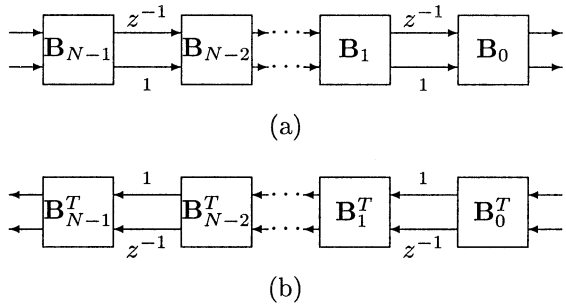

FIGURE 14.9 Flow graph for implementing an LT where $\mathbf{F}(z)$ can be factorized using symmetric delays and N stages. Signals $x(n)$ and $y(n)$ are segmented and processed using blocks of M samples, all branches carry $M/2$ samples, and blocks $\mathbf{B}_i$ are $M \times M$ orthogonal or invertible matrices. (a) Forward transform section; (b) inverse transform section.

With factorization, the design of $\mathbf{F}(z)$ is broken down in 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 allowing inverse. Even more desirable is to factorize the PTM as:

$$\mathbf{F}(z) = \mathbf{B}_0 \prod_{i=0}^{N-1} \mathbf{\Lambda}(z)\mathbf{B}_i \tag{14.52}$$

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:

$$\mathbf{G}(z) = \left(\prod_{i=N-1}^{0} \mathbf{B}_i^T \tilde{\mathbf{\Lambda}}(z) \right) \mathbf{B}_0^T \tag{14.53}$$

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

$$\mathbf{G}(z) = \left(\prod_{i=N-1}^{0} \mathbf{B}_i^{-1} \tilde{\mathbf{\Lambda}}(z) \right) \mathbf{B}_0^{-1}, \tag{14.54}$$

i.e., the design can be simplified by only applying 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,

$$\mathbf{\Lambda}(z) = \begin{bmatrix} z^{-1}\mathbf{I}_{M/2} & 0 \\ 0 & \mathbf{I}_{M/2} \end{bmatrix}, \qquad \tilde{\mathbf{\Lambda}}(z) = \begin{bmatrix} \mathbf{I}_{M/2} & 0 \\ 0 & z^{-1}\mathbf{I}_{M/2} \end{bmatrix}. \tag{14.55}$$

The flow graph for implementing an LT which can be parameterized using SDF is shown in Figure 14.9.

If we are given the SDF matrices instead of the basis coefficients, one can easily reconstruct the LT matrix. For this, start with the last stage and recur the structure in (14.52) using (14.55). Let $\mathbf{P}^{(i)}$ be the partial reconstruction of $\mathbf{P}$ after including up to the i-th stage. Then,

$$\mathbf{P}^{(0)} = \mathbf{B}_{N-1} \tag{14.56}$$

$$\mathbf{P}^{(i)} = \mathbf{B}_{N-1-i} \begin{bmatrix} \mathbf{I}_{M/2} & \mathbf{0}_{M/2} & \mathbf{0}_{M/2} & \mathbf{0}_{M/2} \\ \mathbf{0}_{M/2} & \mathbf{0}_{M/2} & \mathbf{0}_{M/2} & \mathbf{I}_{M/2} \end{bmatrix} \begin{bmatrix} \mathbf{P}^{(i-1)} & 0_M \\ 0_M & \mathbf{P}^{(i-1)} \end{bmatrix} \tag{14.57}$$

$$\mathbf{P} = \mathbf{P}^{(N-1)}. \tag{14.58}$$

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

14.5 Hierarchical Connection of LTs: 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 those questions.

The subject has been intensively studied and a large number of publications are available. Our intent, however, is just to provide a basic introduction, while leaving more detailed analysis to the references. The relation between filter banks and discrete wavelets[42,50,52] is well-known. Under conditions that are easily satisfied,[50] an infinite cascade of filter banks will generate a set of continuous orthogonal wavelet bases. In general, if only the low-pass subband is connected to another filter bank, for a finite number of stages, we call the resulting filter bank a discrete wavelet transform (DWT).[50,52] A free cascading of filter banks, however, is better known as a discrete wavelet packet (DWP).[5,54,28,42] As LTs and filter banks are equivalent in most senses, the same relations apply to LTs and wavelets. The system resulting from the hierarchical association of several LTs will be called here a hierarchical lapped transform (HLT).[17]

14.5.1 Time-Frequency Diagram

The description of the cascaded connection of LTs is better carried 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.[1,2,25] 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 References 1, 2, and 25 for details on TF signal representation, and other chapters in this handbook for the DFT). The TF representation is obtained by expressing the signal $x(n)$ with respect to bases that are functions of both frequency and time. For example, the size-r DFT of a sequence extracted from $x(n)$ (from $x(n)$ to $x(n + r - 1)$)[25] can be

$$\alpha(k,n) = \sum_{i=0}^{r-1} x(i+n) \exp\left(-\frac{j2\pi ki}{r}\right). \tag{14.59}$$

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

$$\alpha(k,n) = \sum_{i=-\infty}^{\infty} x(i) w(i) \exp\left(-\frac{jk(i-n)2\pi}{r}\right). \tag{14.60}$$

For more general bases we may write:

$$\alpha(k,n) = \sum_{i=-\infty}^{\infty} x(i) \phi(n-i,k) \tag{14.61}$$

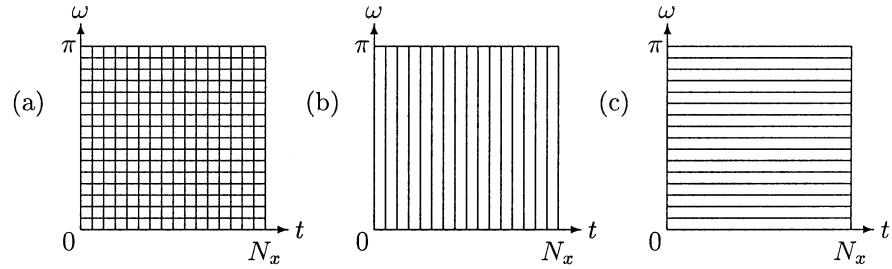

FIGURE 14.10 Examples of rectangular partitions of the time-frequency plane for a signal which has N_x samples. (a) Spectrogram with a N_x-length window, resulting in N_x^2 TF samples; (b) Input signal, no processing; (c) A transform such as the DCT or DFT is applied to all N_x samples.

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

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 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 $r = N_x$. In this case, the diagram for the spectrogram is given in Figure 14.10(a). We call such diagrams TF diagrams, because they only indicate 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 would represent a distinct region in a TF diagram. Note that in such representation, the signal is represented by N_x^2 TF coefficients. We are looking for 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, i.e., N_x TF coefficients will be generated. Also, we require that all N_x samples of $x(n)$ can be reconstructed from the N_x TF coefficients. If we use less than N_x samples in the TF plane, we clearly cannot reconstruct all possible combinations of samples in $x(n)$, from the TF coefficients, solely using linear relations.

Under these assumptions, Figure 14.10(b) shows the TF diagram for the original signal (only resolution in the time axis) for $N_x = 16$. Also, for $N_x = 16$, Figure 14.10(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.

14.5.2 Tree-Structured Hierarchical Lapped Transforms

The tree diagram is helpful to describe the hierarchical connection of filter banks. In this diagram we represent an M-band LT by nodes and branches of an M-ary tree. Figure 14.11(a) shows an M-band LT, where all the M subband signals have sampling rates M times smaller than that of $x(n)$. Figure 14.11(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 Figure 14.10, the equivalent TF diagram for an M-band LT is shown in Figure 14.11(c), for a 16-sample signal and for $M = 4$. Note that the TF diagram of Figure 14.11(c) resembles that of Figure 14.10(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 Figure 14.10(b), Figure 14.11(c) implies an exchange of resolution from time to frequency domain achieved by the LT.

The exchange of resolution in the TF diagram is obtained by 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, Figure 14.12, shows some examples of HLTs given

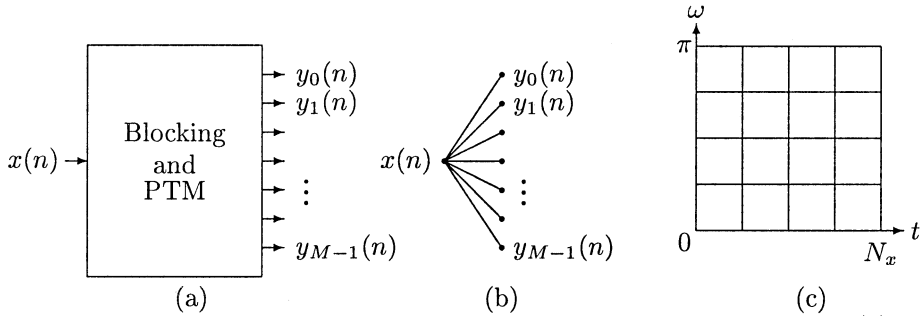

FIGURE 14.11 Representation of an M-channel LT as tree nodes and branches. (a) Forward section of an LT, including the blocking device. (b) Equivalent notation for (a) using an M-branch single-stage tree. (c) Equivalent TF diagram for (a) or (b) assuming $M = 4$ and $N_x = 16$.

by their tree diagrams and respective TF diagrams. Figure 14.12(a) shows 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. In Figure 14.12(b), a more generic hierarchical connection of 2-channel LTs is shown. First the signal is split into low- and high-pass. Each output branch is further connected to another 2-channel LT. In the third stage only the most low-pass subband signal is connected to another 2-channel LT. Figure 14.12(c) shows a 2-stage HLT obtaining the same TF diagram as Figure 14.12(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. Figure 14.12(d) shows the TF diagram corresponding to Figure 14.12(a), while Figure 14.12(e) shows the TF diagram corresponding to Figure 14.12(b,c). 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.

To better understand how connecting an LT to the tree can achieve the exchange between time and frequency resolutions, Figure 14.13 shows the basis functions resulting from two similar tree-structured HLTs. The difference between them is one tree cell which is applied or not to a terminal branch of the tree.

14.5.3 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 that the overall TF diagram of the decomposition

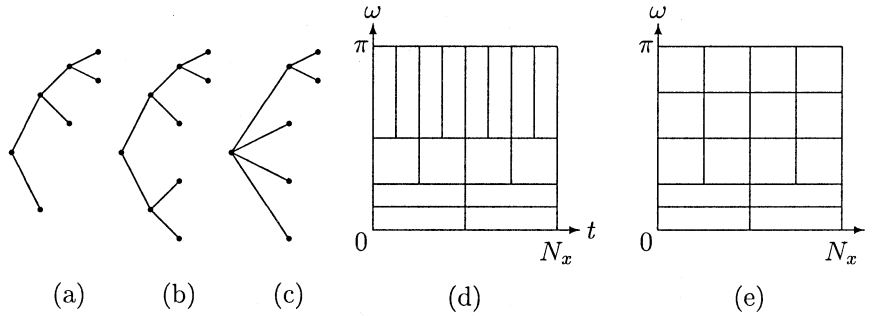

FIGURE 14.12 Tree and TF diagrams. (a) The 3-stages DWT binary-tree diagram, where only the low-pass subband is submitted to further LT stages. (b) A more generic 3-stages tree diagram. (c) A 2-stages tree-diagram resulting in the same TF diagram as (b). (d) TF diagram for (a). (3) TF diagram for (b) or (c).

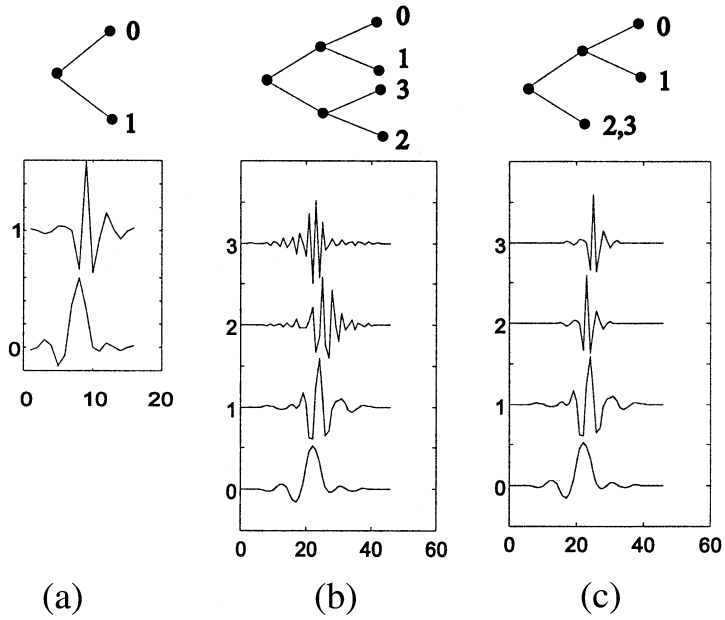

FIGURE 14.13 Two HLTs and resulting bases. (a) The 2-channel 16-tap-bases LT, showing low- and high-frequency bases, $f_0(n)$ and $f_1(n)$, respectively. (b) Resulting basis functions of a 2-stage HLT based on (a), given by $f_0(n)$ through $f_3(n)$. Its respective tree diagram is also shown. (c) Resulting 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.

is altered. There is a useful alternative to the tree structure in which the number of subbands does not change. We refer to Figure 14.14, where the "blocking" part of the diagram corresponds to the chain of delays and decimators (as in Figure 14.8) that parallelizes the signal into polyphase components. System $\mathbf{A}(z)$ of M bases of length N_AM is post-processed by system $\mathbf{B}(z)$ of K bases of length N_BK. Clearly, entries in $\mathbf{A}(z)$ have order $N_A - 1$ and entries in $\mathbf{B}(z)$ have order $N_B - 1$. Without loss generality, we associate system $\mathbf{B}(z)$ to the first K output subbands of $\mathbf{A}(z)$. The overall PTM is given by:

$$\mathbf{F}(x) = \begin{bmatrix} \mathbf{B}(z) & 0 \\ 0 & \mathbf{I}_{M-K} \end{bmatrix} \mathbf{A}(z) \,, \tag{14.62}$$

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_AM. In other words the effect of cascading $\mathbf{A}(z)$ and $\mathbf{B}(z)$ was only to modify K bases, so that the length of the modified bases is equal or larger than the length of the initial bases. An example is shown in Figure 14.15. We start with the bases corresponding to $\mathbf{A}(z)$ shown in Figure 14.15(a). There are 8 bases of length 16 so that $\mathbf{A}(z)$ has order 1. $\mathbf{A}(z)$ is post-processed by $\mathbf{B}(z)$ which is a 4×4 PTM of order 3 whose corresponding bases are shown in Figure 14.15(b). The resulting LT is shown in Figure 14.15(c). There are 4 bases of length 16 and 4 of length 40. The shorter ones are identical to those in Figure 14.15(b), while the longer ones have orders which are the sums 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 from the corresponding ones in $\mathbf{A}(z)$.

The effect of post-processing few 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 post-processing

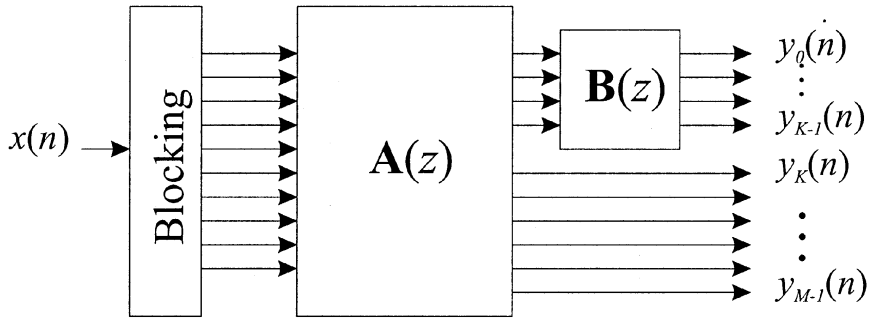

FIGURE 14.14 Cascade of PTMs $\mathbf{A}(z)$ of M channels and $\mathbf{B}(z)$ of K channels. The total number of subbands does not change, however some of $\mathbf{A}(z)$ bases are increased in length and order.

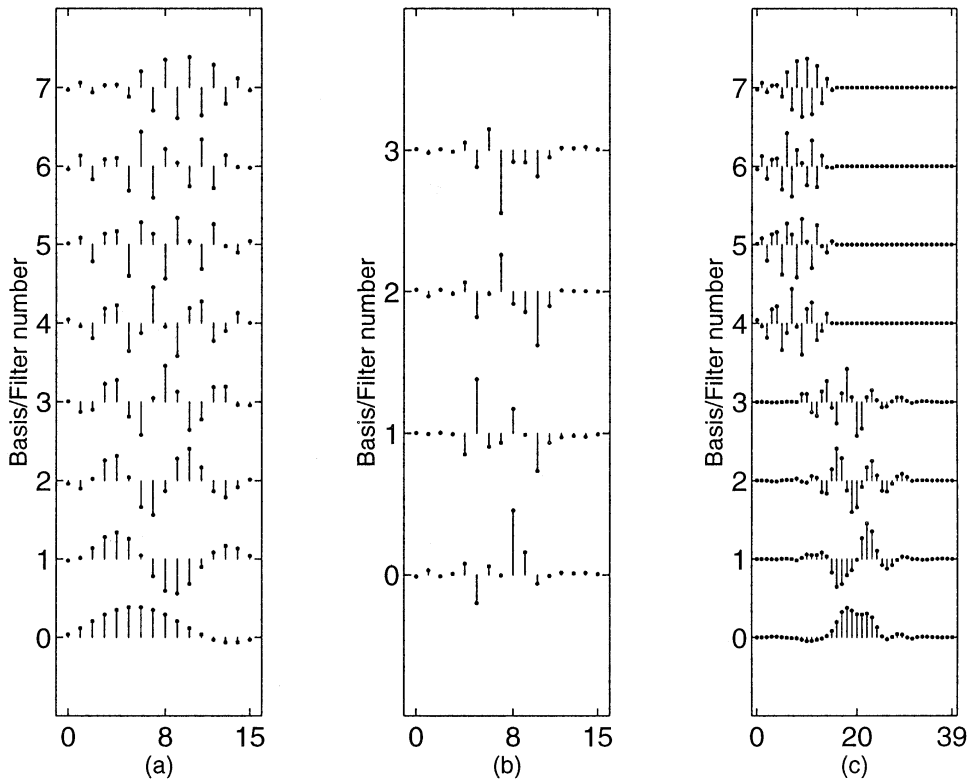

FIGURE 14.15 Example of constructing variable-length bases through cascading LTs. (a) The basis corresponding to $\mathbf{A}(z)$: an LT with 8 bases of length 16 (order 1). (b) The bases corresponding to $\mathbf{B}(z)$: an LT with 4 bases of length 16 (order 3). (c) The bases corresponding to $\mathbf{F}(z)$: 4 of the 8 bases have order 1, i.e., length 16, while the remaining 4 have order 4, i.e., length 40.

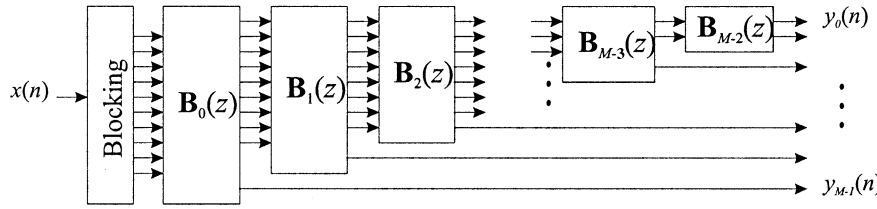

FIGURE 14.16 General factorization of a variable-length LT.

stages.[47,48] A general factorization of LTs is as shown in Figure 14.16. Assume a variable-length $\mathbf{F}(z)$ whose bases are arranged in decreasing length order. Such a PTM can be factorized as:

$$\mathbf{F}(z) = \prod_{i=0}^{M-2} \begin{bmatrix} \mathbf{B}_i(z) & 0 \\ 0 & \mathbf{I}_i \end{bmatrix} \tag{14.63}$$

where $\mathbf{I}_0$ is understood to be non-existing 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 $0 \le k < K_i$. Hence,

$$\mathbf{F}(z) = \prod_{i=0}^{M-2} \prod_{k=0}^{K_i-1} \begin{bmatrix} \mathbf{B}_{ik}(z) & 0 \\ 0 & \mathbf{I}_i \end{bmatrix}. \tag{14.64}$$

In a later section we will show a very useful LT which is based on the factorization principles of (14.64).

14.6 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, etc. 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 processing and compression applications. By symmetric bases we mean that:

$$p_{i,j} = (\pm 1)\ \ p_{i,L-1-j}\ . \tag{14.65}$$

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

$$\mathbf{F}(z) = z^{-(N-1)}\mathbf{S}\mathbf{F}(z^{-1})\mathbf{J}_M\ , \tag{14.66}$$

where $\mathbf{S}$ is a diagonal matrix whose diagonal entries s_{ii} are ± 1, depending 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.

14.6.1 The Lapped Orthogonal Transform: LOT

Lapped orthogonal transform (LOT)[12-14] was the first useful LT with a well defined factorization. Malvar developed the fast LOT based on the work by Cassereau[3] 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.[38] Also, since it is a popular transform, it has a reduced cost and is easily available in either software or hardware. The DCT matrix **D** is defined as having entries:

$$d_{ij} = \sqrt{\frac{2}{M}} k_i \cos\left(\frac{(2j+1)i\pi}{2M}\right) \tag{14.67}$$

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

The LOT as defined by Malvar 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:

$$\mathbf{P}_{LOT} = \begin{bmatrix} \mathbf{I}_M & 0 \\ 0 & \mathbf{V}_R \end{bmatrix} \begin{bmatrix} \mathbf{D}_e - \mathbf{D}_o & \mathbf{J}_{M/2}(\mathbf{D}_e - \mathbf{D}_o) \\ \mathbf{D}_e - \mathbf{D}_o & -\mathbf{J}_{M/2}(\mathbf{D}_e - \mathbf{D}_o) \end{bmatrix} \tag{14.68}$$

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 **D** and $\mathbf{D}_o$ corresponds to the odd numbered rows of **D**. $\mathbf{V}_R$ is an $M/2 \times M/2$ orthogonal matrix, which according to References 15 and 21 should be approximated by $M/2 - 1$ plane rotations as:

$$\mathbf{V}_R = \prod_{i=M/2-2}^{0} \mathbf{\Theta}(i, i+1, \theta_i) \tag{14.69}$$

where $\mathbf{\Theta}$ is defined in Sec. (14.1.4). Suggestions of rotation angles which were designed to yield a good transform for image compression are[20]

$$M = 4 \rightarrow \theta_0 = 0.1\pi \tag{14.70}$$

$$M = 8 \rightarrow \{\theta_0, \theta_1, \theta_2\} = \{0.13, 0.16, 0.13\} \times \pi \tag{14.71}$$

$$M = 16 \rightarrow \{\theta_0, \ldots, \theta_7\} = \{0.62, 0.53, 0.53, 0.50, 0.44, 0.35, 0.23, 0.11\} \times \pi\,. \tag{14.72}$$

For $M \ge 16$ it is suggested to use:

$$\mathbf{V}_R = \mathbf{D}_{IV}^T \mathbf{D}^T \tag{14.73}$$

where $\mathbf{D}_{IV}$ is the DCT type IV matrix[38] whose entries are

$$d_{ij}^{IV} = \sqrt{\frac{2}{M}} \cos\left(\frac{(2j+1)(2i+1)\pi}{4M}\right). \tag{14.74}$$

A block diagram for the implementation of the LOT is shown in Figure 14.17 for $M = 8$.

14.6.2 The Lapped Bi-orthogonal Transform: LBT

The LOT is a large improvement over the DCT for image compression mainly because it reduces the so-called blocking effects. Although there is a very large reduction, blocking is not eliminated. The reason

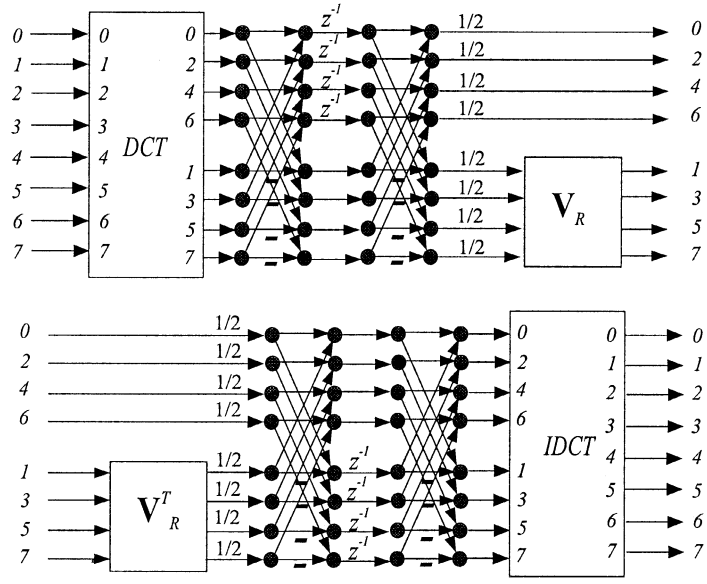

FIGURE 14.17 Implementation of the LOT for $M = 8$. Top: forward transform, bottom: inverse transform.

for that lies in the format of the low frequency bases of LOT. In image compression, few bases are used to reconstruct the signal. From Figure 14.4, one can see that the "tails" of the lower frequency bases of the LOT do not exactly decay to zero. For that 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 the LOT, creating the lapped bi-orthogonal transform (LBT).[22] (Bi-orthogonal is a jargon used in the filter banks community to designate transforms and filter banks which are not orthogonal.) In any case, the factorization of the LBT is almost identical to that of the LOT. However:

$$\mathbf{P}_{LBT} = \begin{bmatrix} \mathbf{I}_M & 0 \\ 0 & \mathbf{V}_R \end{bmatrix} \begin{bmatrix} \mathbf{D}_e - \Upsilon\mathbf{D}_o & \mathbf{J}_{M/2}(\mathbf{D}_e - \Upsilon\mathbf{D}_o) \\ \mathbf{D}_e - \Upsilon\mathbf{D}_o & -\mathbf{J}_{M/2}(\mathbf{D}_e - \Upsilon\mathbf{D}_o) \end{bmatrix} \tag{14.75}$$

where Υ is the $M/2 \times M/2$ diagonal matrix given by $\Upsilon = diag\{\sqrt{2}, 1, \ldots, 1\}$. Note that it only implies 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 (14.75) except that the multiplier is inverted, i.e. $\Upsilon = diag\{1/\sqrt{2}, 1, \ldots, 1\}$. The diagram for implementing an LBT for $M = 8$ is shown in Figure 14.18.

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 the LOT into the new bases of the forward LBT, as it can be seen in Figure 14.19. In Figure 14.19 the reader can note the reduction in the amplitude of the boundary samples of the LBT and an enlargement of the same samples in the inverse LBT. This simple "trick" improved noticeably 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 the LBT instead of the LOT whenever a non-orthogonal LT can be used.

14.6.3 The Generalized LOT: GenLOT

The formulation for the LOT[14] which is shown in (14.68), is not the most general there is for this kind of LT. In fact it can be generalized to become:

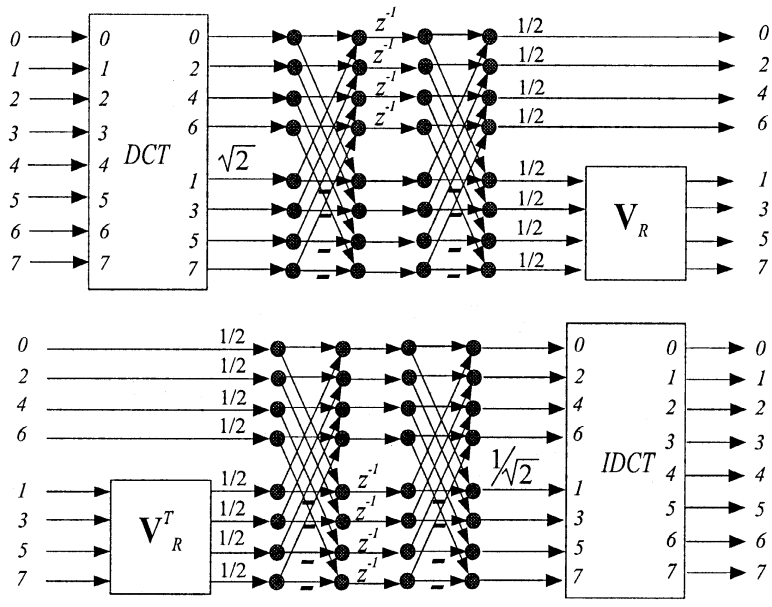

FIGURE 14.18 Implementation of the LBT for $M = 8$. Top: forward transform, bottom: inverse transform. Note that there is only one extra multiplication as compared to the LOT.

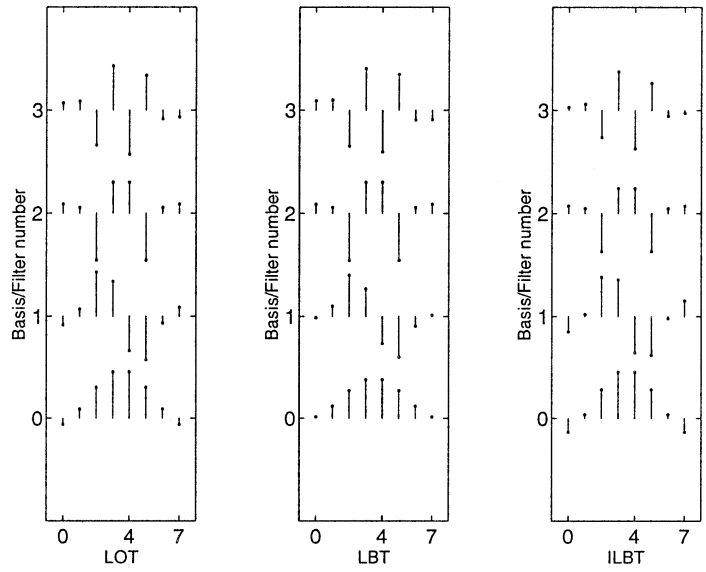

FIGURE 14.19 Comparison of bases for the LOT ($\mathbf{P}_{LOT}$), inverse LBT ($\mathbf{Q}_{LBT}$) and forward LBT ($\mathbf{P}_{LBT}$). The extreme samples of the lower frequency bases of the LOT are larger than those of the inverse LBT. This is an advantage for image compression.

$$\mathbf{P} = \begin{bmatrix} \mathbf{U} & 0 \\ 0 & \mathbf{V} \end{bmatrix} \begin{bmatrix} \mathbf{D}_e - \mathbf{D}_o & \mathbf{J}_{M/2}(\mathbf{D}_e - \mathbf{D}_o) \\ \mathbf{D}_e - \mathbf{D}_o & -\mathbf{J}_{M/2}(\mathbf{D}_e - \mathbf{D}_o) \end{bmatrix}. \tag{14.76}$$

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

$$\mathbf{W} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I}_{M/2} & \mathbf{I}_{M/2} \\ \mathbf{I}_{M/2} & -\mathbf{I}_{M/2} \end{bmatrix}, \tag{14.77}$$

$$\mathbf{\Phi}_i = \begin{bmatrix} \mathbf{U}_i & 0_{M/2} \\ 0_{M/2} & \mathbf{V}_i \end{bmatrix}, \tag{14.78}$$

$$\mathbf{\Lambda}(z) = \begin{bmatrix} \mathbf{I}_{M/2} & 0_{M/2} \\ 0_{M/2} & z^{-1}\mathbf{I}_{M/2} \end{bmatrix}, \tag{14.79}$$

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

$$\mathbf{F}(z) = \mathbf{\Phi}_1 \mathbf{W} \mathbf{\Lambda}(z) \mathbf{W} \mathbf{D} \ . \tag{14.80}$$

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 Figure 14.20.

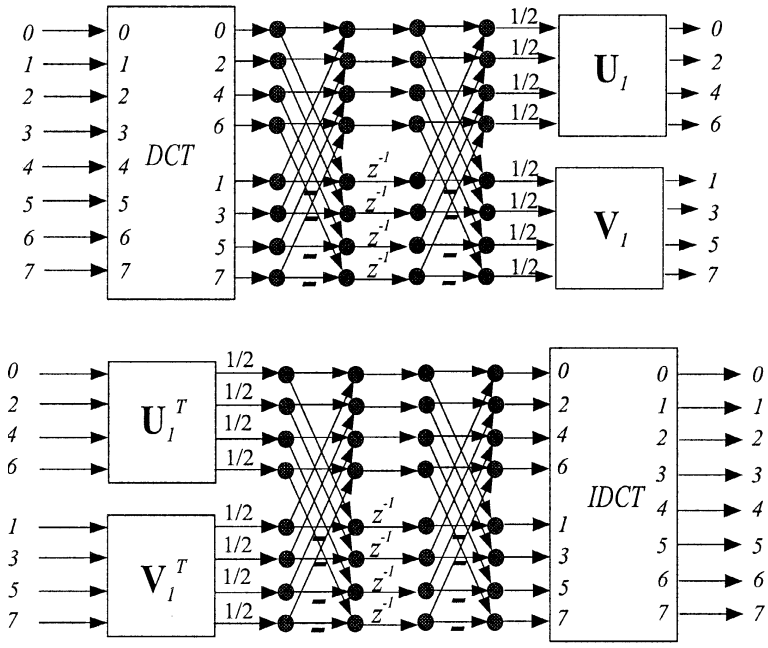

FIGURE 14.20 Implementation of a more general version of the LOT for $M = 8$. Top: forward transform, bottom: inverse transform.

From this formulation along with other results it was realized[34] that all orthogonal symmetric LTs can be expressed as:

$$\mathbf{F}(z) = \mathbf{K}_{N-1}(z)\mathbf{K}_{N-2}(z)\cdots\mathbf{K}_1(z)\mathbf{K}_0 \tag{14.81}$$

where

$$\mathbf{K}_i(z) = \mathbf{\Phi}_i \mathbf{W} \mathbf{\Lambda}(z) \mathbf{W} \ , \tag{14.82}$$

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

$$\mathbf{G}(z) = \mathbf{K}_0^T \mathbf{K}_1'(z)\mathbf{K}_2'(z)\cdots\mathbf{K}_{N-1}'(z) \tag{14.83}$$

where:

$$\mathbf{K}_i'(z) = z^{-1}\mathbf{W}\mathbf{\Lambda}(z^{-1})\mathbf{W}\mathbf{\Phi}_i^T \ . \tag{14.84}$$

From that perspective, the generalized LOT (GenLOT) is defined as the orthogonal LT as in (14.81) in which $\mathbf{K}_0 = \mathbf{D}$, i.e.,

$$\mathbf{F}(z) = \mathbf{K}_{N-1}(z)\cdots\mathbf{K}_1(z)\mathbf{D} \ . \tag{14.85}$$

A diagram for implementing a GenLOT for even M is shown in Figure 14.21. In this diagram, the scaling parameters are $\beta = 2^{-(N-1)}$ and account for the terms $1/\sqrt{2}$ in the definition of 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 Sec. 14.1.4 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 non-linear unconstrained optimization. Rotation angles are searched to minimize some cost function. GenLOT examples are given elsewhere[34] and we present two examples, for $M = 8$, in Tables 14.1 and 14.2, which are also plotted in Figure 14.22.

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

$$\mathbf{F}(z) = \mathbf{K}_{(N-1)/2}(z)\cdots\mathbf{K}_1(z)\mathbf{D} \ . \tag{14.86}$$

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

$$\mathbf{K}_i(z) = \mathbf{\Phi}_{2i}^o \mathbf{W}^o \mathbf{\Lambda}^{o1}(z) \mathbf{W}^o \mathbf{\Phi}_{2i-1}^o \mathbf{W}^o \mathbf{\Lambda}^{o2}(z) \mathbf{W}^o \tag{14.87}$$

and where:

$$\mathbf{\Phi}_{2i}^o = \begin{bmatrix} \mathbf{U}_{2i} & 0 \\ 0 & \mathbf{V}_{2i} \end{bmatrix}, \tag{14.88}$$

$$\mathbf{\Phi}_{2i-1}^o = \begin{bmatrix} \mathbf{U}_{2i-1} & & 0 \\ & 1 & \\ 0 & & \mathbf{V}_{2i-1} \end{bmatrix}, \tag{14.89}$$

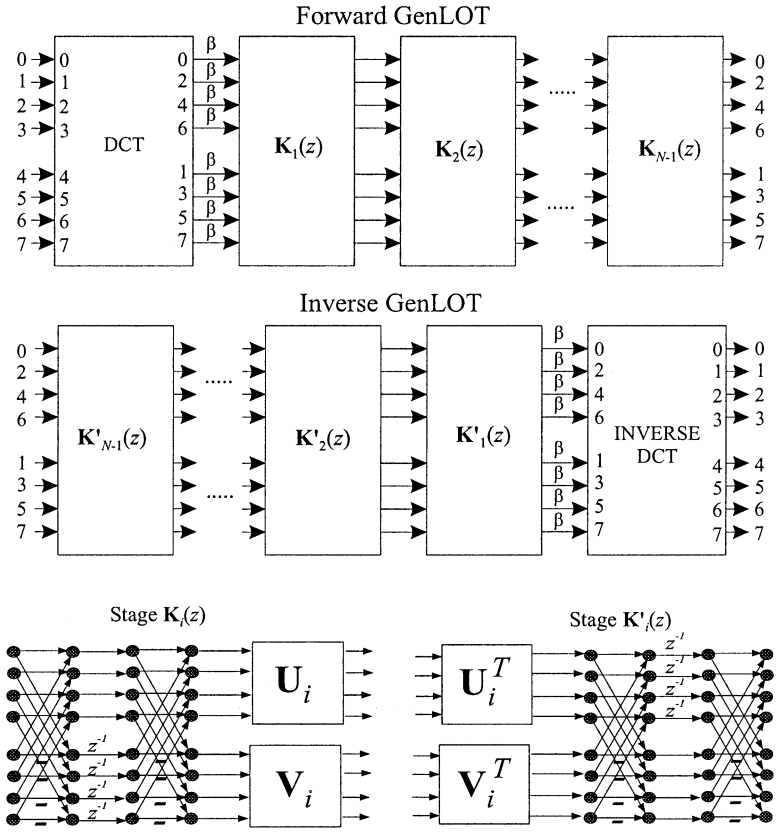

FIGURE 14.21 Implementation of a GenLOT for even M, ($M = 8$). Forward and inverse transforms are shown along with details of each stage. $\beta = 2^{-(N-1)}$ accounts for all terms of the form $1/\sqrt{2}$ which make the butterflies (W) orthogonal.

TABLE 14.1 GenLot example for $N = 4$. The even bases are symmetric while the odd ones are anti-symmetric, so that 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

TABLE 14.2 GenLOT example for $N = 6$. The even bases are symmetric while the odd ones are anti-symmetric, so that 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

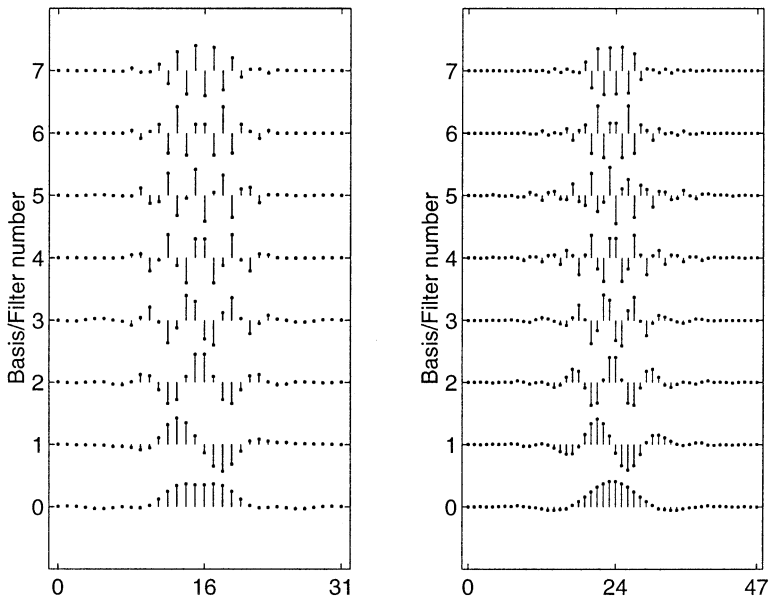

FIGURE 14.22 Example of optimized GenLOT bases for $M = 8$ and for $N = 4$ (left) and $N = 6$ (right).

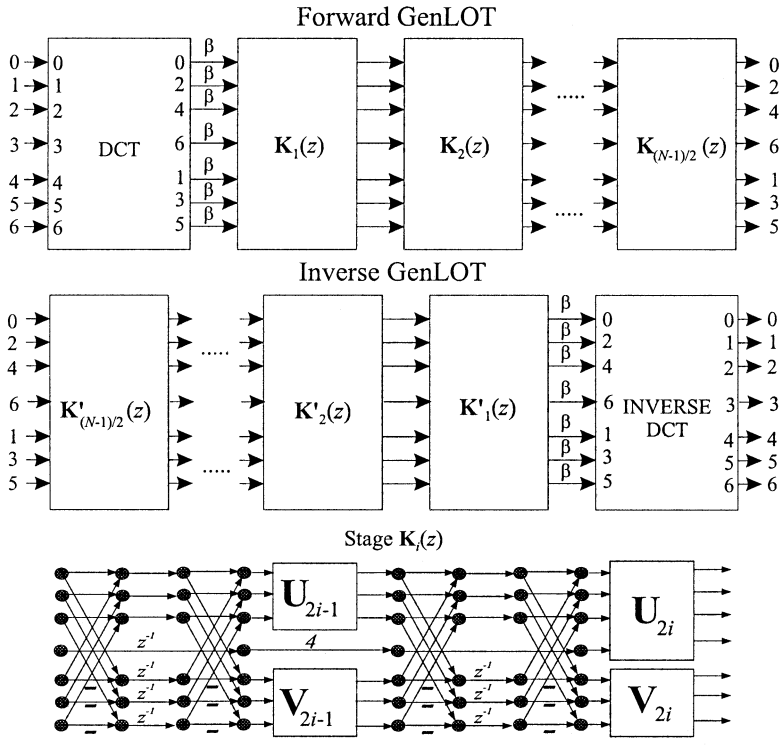

FIGURE 14.23 Implementation of a GenLOT for M odd. Forward and inverse transforms are shown along with details of each stage and $\beta = 2^{-(N-1)}$.

$$\mathbf{W}^{o} = \begin{bmatrix} \mathbf{I}_{(M-1)/2} & \mathbf{0}_{(M-1)/2\times 1} & \mathbf{I}_{(M-1)/2} \\ \mathbf{0}_{1\times(M-1)/2} & 1 & \mathbf{0}_{1\times(M-1)/2} \\ \mathbf{I}_{(M-1)/2} & \mathbf{0}_{(M-1)/2\times 1} & -\mathbf{I}_{(M-1)/2} \end{bmatrix}, \tag{14.90}$$

$$\mathbf{\Lambda}^{o1}(z) = diag\{\underbrace{1,1,\ldots,1}_{(M+1)/2\times 1's}, \underbrace{z^{-1},\ldots,z^{-1}}_{(M-1)/2\times z^{-1}}\}, \tag{14.91}$$

$$\mathbf{\Lambda}^{o2}(z) = diag\{\underbrace{1,1,\ldots,1}_{(M-1)/2\times 1's}, \underbrace{z^{-1},\ldots,z^{-1}}_{(M+1)/2\times z^{-1}}\}, \tag{14.92}$$

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 Figure 14.23. The main difference is that two stages have to be connected together. The inverse transform is accomplished in the same way as for the even channel case:

$$\mathbf{G}(z) = \mathbf{D}^{T}\mathbf{K}'_1(z)\mathbf{K}'_2(z)\cdots\mathbf{K}'_{N-1}(z) \tag{14.93}$$

where the inverse factors are:

$$\mathbf{K}'_i(z) = z^{-2}\mathbf{K}_i^{T}(z^{-1}), \tag{14.94}$$

whose structure is evident from Figure 14.23.

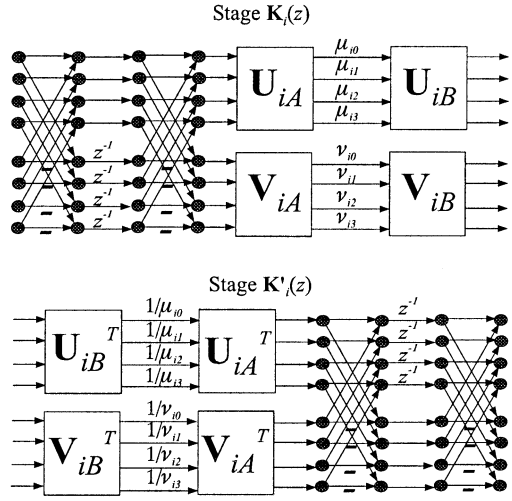

FIGURE 14.24 Implementation of the factors of the general factorization (GLBT) for M even. Top: factor of the forward transform: $\mathbf{K}_i(z)$. Bottom: factor of the inverse transform: $\mathbf{K}'_i(z)$.

14.6.4 The General Factorization: GLBT

The general factorization for all symmetric LTs[49] can be viewed either as an extension of GenLOTs or as a generalization of the LBT. It can be shown that for M even, all LTs obeying (14.65) or (14.66) can be factorized as in (14.81), where the $\mathbf{K}_i(z)$ factors are given in (14.82) with the matrices $\mathbf{U}_i$ and $\mathbf{V}_i$ (which compose $\mathbf{\Phi}_i$) being only required to be general invertible matrices. From Sec. 14.1.4, each factor can be decomposed as:

$$\mathbf{U}_i = \mathbf{U}_{iB}\mathbf{U}_{id}\mathbf{U}_{iA}, \qquad \mathbf{V}_i = \mathbf{V}_{iB}\mathbf{V}_{id}\mathbf{V}_{iA}, \tag{14.95}$$

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 non-zero diagonal entries.

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

$$\mathbf{K}_0 = \mathbf{\Phi}_0\mathbf{W}, \tag{14.96}$$

where $\mathbf{\Phi}_i$ is given as in (14.78), and factors $\mathbf{U}_0$ and $\mathbf{V}_0$ are only required to be invertible. The general factorization can be viewed as a generalized LBT (GLBT) and its implementation flow graph for M even is shown in Figure 14.24.

The inverse GLBT is similar to the GenLOT case, where:

$$\mathbf{K}'_i(z) = z^{-1}\mathbf{W}\mathbf{\Lambda}(z)\mathbf{W}\mathbf{\Phi}_i^{-1}. \tag{14.97}$$

and:

$$\mathbf{\Phi}_i^{-1} = \begin{bmatrix} \mathbf{U}_i^{-1} & 0_{M/2} \\ 0_{M/2} & \mathbf{V}_i^{-1} \end{bmatrix} = \begin{bmatrix} \mathbf{U}_{iA}^T\mathbf{U}_{id}^{-1}\mathbf{U}_{iB}^T & 0_{M/2} \\ 0_{M/2} & \mathbf{V}_{iA}^T\mathbf{V}_{id}^{-1}\mathbf{V}_{iB}^T \end{bmatrix} \tag{14.98}$$

while

$$\mathbf{K}_0^{-1} = \mathbf{W}\mathbf{\Phi}_0^{-1}. \tag{14.99}$$

TABLE 14.3 Forward GLBT bases example for $M = 8$ and $N = 2$. The even bases are symmetric while the odd ones are anti-symmetric, so that 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

TABLE 14.4 Inverse GLBT bases example for $M = 8$ and $N = 2$. The even bases are symmetric while the odd ones are anti-symmetric, so that 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

The diagram for the implementation of the inverse stages of the GLBT is shown in Figure 14.24.

Examples of bases for the GLBT of particular interest to image compression are given in Tables 14.3 and 14.4.

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

$$\mathbf{F}(z) = \mathbf{K}_{(N-1)/2}(z)\cdots\mathbf{K}_1(z)\mathbf{K}_0 \tag{14.100}$$

where the stages $\mathbf{K}_i$ are as in (14.87) 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 non-zero 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 Figure 14.25.

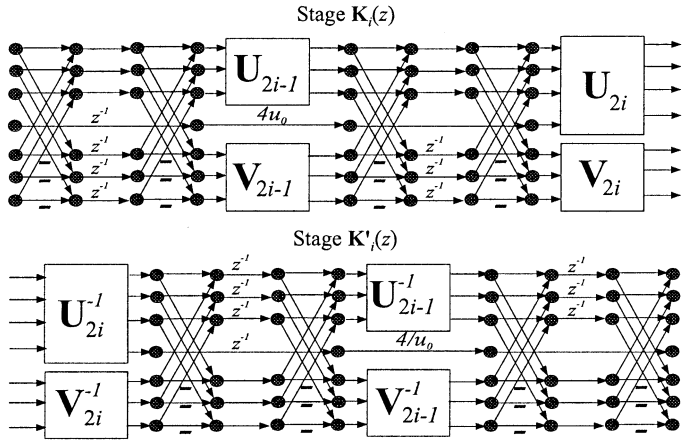

FIGURE 14.25 Implementation of the factors of the general factorization (GLBT) for M odd. Top: factor of the forward transform: $\mathbf{K}_i(z)$. Bottom: factor of the inverse transform: $\mathbf{K}'_i(z)$.

14.7 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 and, yet, to provide some advantage over a block transform. For that we use the principles of Sec. 14.5.3 and define the FLT as the LT whose PTM is given by:

$$\mathbf{F}(z) = \begin{bmatrix} \mathbf{E}(z) & 0 \\ 0 & \mathbf{I}_{M-K} \end{bmatrix} \mathbf{D}_M \tag{14.101}$$

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:

$$\mathbf{G}(z) = \mathbf{D}_M^T \begin{bmatrix} \mathbf{E}'(z) & 0 \\ 0 & \mathbf{I}_{M-N} \end{bmatrix}, \tag{14.102}$$

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:

$$\mathbf{E}(z) = \mathbf{\Psi}(z)\mathbf{D}_K^T \tag{14.103}$$

where $\mathbf{\Psi}(z)$ is a known LT and $\mathbf{D}_K$ is the $K \times K$ DCT matrix, i.e., 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 Figure 14.26. In that example, the first case where $K = 2$, direct optimization is recommended, for which the values $\{\alpha_{00}, \alpha_{01}, \alpha_{10}, \alpha_{10}, \alpha_{20}, \alpha_{21}\} = \{1.9965, 1.3193, 0.4388, 0.7136, 0.9385, 1.2878\}$ yield an excellent FLT for image compression. In the middle of Figure 14.26 the case K = 4 can be optimized by optimizing 2 invertible matrices. In the case where we use the method in (14.103) and the LBT as the K channel post-processing stage, we can see that the LBT's DCT stage is cancelled 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 Figure 14.27. 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).[38]

14.8 Modulated LTs

Cosine modulated LTs[50] use a low-pass prototype to modulate a cosine sequence. By a proper choice of the phase of the cosine sequence, Malvar developed the modulated lapped transform (MLT),[15] which led to the so-called extended lapped transforms (ELT).[18-21] The 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 low-pass prototype choice and the phase of the cosine sequence.[11,15,19,20,24,27,40,44,45,49]

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:

$$p_{k,n} = h(n)\cos\left[\left(k+\frac{1}{2}\right)\left(\left(n-\frac{L-1}{2}\right)\frac{\pi}{M}+(N+1)\frac{\pi}{2}\right)\right] \tag{14.104}$$

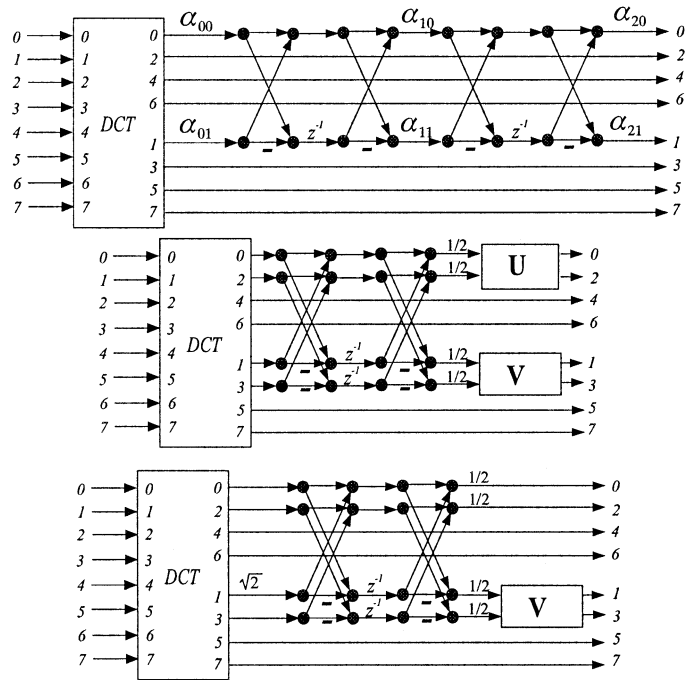

FIGURE 14.26 Implementation of examples of the FLT. On top, $K = 2$; middle, case $K = 4$; bottom, case $K = 4$ where $\mathbf{\Psi}(z)$ is the LBT, thus having its DCT stage cancelled.

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 low-pass 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 ELTs with other values of K will be referred as ELT-K.

The ELTs have as their major plus a 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[38] orthogonal transform in the last stage, which has fast implementation algorithms. The lattice-style algorithm is shown in Figure 14.28 for an ELT with generic overlap factor K. In Figure 14.28 each branch carries $M/2$

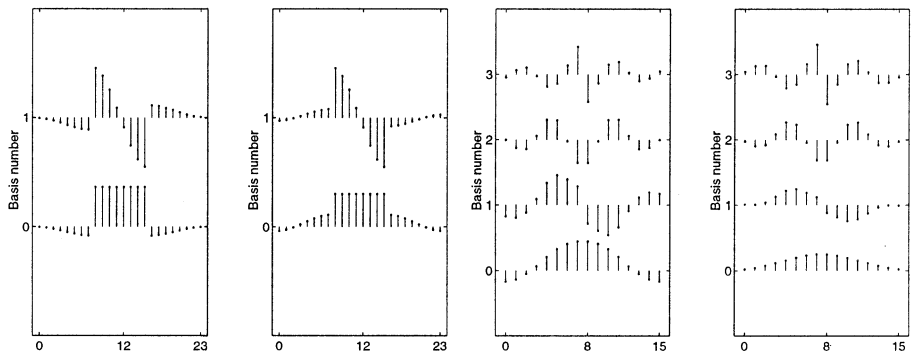

FIGURE 14.27 Bases of the FLT in the case $M = 8$ for forward and inverse LTs. From left to right: forward transform bases for the case $K = 2$, inverse transform bases for the case $K = 2$, forward transform bases for the case $K = 4$, inverse transform bases for the case $K = 4$. The remaining bases, not shown are the regular bases of the DCT and have length 8.

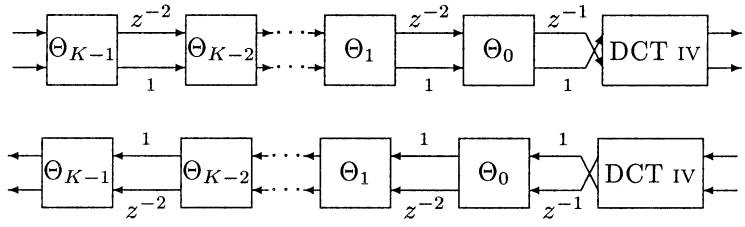

FIGURE 14.28 Flow graph for the direct (top) and inverse (bottom) ELT. Each branch carries $M/2$ samples.

samples and both analysis (forward transform) and synthesis (inverse transform) flow-graphs are shown. The plane rotation stages are of the form indicated in Figure 14.29 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:

$$\mathbf{\Theta}_i = \begin{bmatrix} -\mathbf{C}_i & \mathbf{S}_i\mathbf{J}_{M/2} \\ \mathbf{J}_{M/2}\mathbf{S}_i & \mathbf{J}_{M/2}\mathbf{C}_i\mathbf{J}_{M/2} \end{bmatrix},$$

$$\mathbf{C}_i = diag\{\cos(\theta_{0,i}), \cos(\theta_{0,i}), \ldots, \cos(\theta_{\frac{M}{2}-1,\,i})\} \tag{14.105}$$

$$\mathbf{S}_i = diag\{\sin(\theta_{0,i}), \sin(\theta_{1,i}), \ldots, \sin(\theta_{\frac{M}{2}-1,\,i})\}.$$

$\theta_{i,j}$ 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 and brings the total number of degrees of freedom to KM.

In general, there is no simple relation among the rotation angles and the window. Optimized angles for several values of M and K are presented in extensive tables in Reference 21. In the ELT-2 case, however, one can use a parameterized design.[19-21] In this design, we have:

$$\theta_{k,0} = -\frac{\pi}{2} + \mu_{M/2+k} \tag{14.106}$$

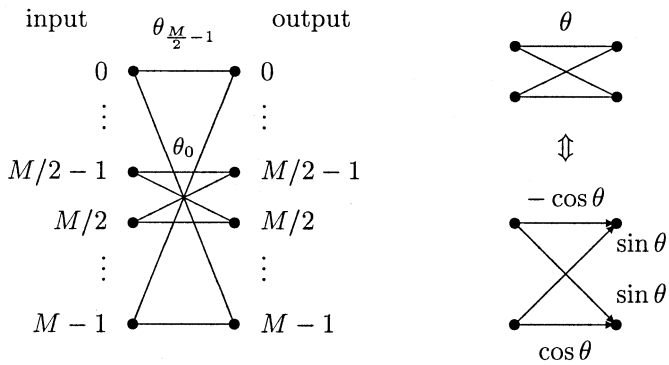

FIGURE 14.29 Implementation of plane rotations stage showing the displacement of the $M/2$ butterflies.

$$\theta_{k,1} = -\frac{\pi}{2} + \mu_{M/2-1-k} \tag{14.107}$$

where:

$$\mu_i = \left[\left(\frac{1-\gamma}{2M} \right) (2k+1) + \gamma \right] \tag{14.108}$$

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:

$$\begin{aligned} h(n) &= \cos(\theta_{n0}) \cos(\theta_{n1}) \\ h(M-1-n) &= \cos(\theta_{n0}) \sin(\theta_{n1}) \\ h(M+n) &= \sin(\theta_{n0}) \cos(\theta_{n1}) \\ h(2M-1-n) &= -\sin(\theta_{n0}) \sin(\theta_{n1}) \end{aligned} \tag{14.109}$$

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

$$\theta_{k0} = \frac{\pi}{2} - \frac{\pi}{2M} \left(k + \frac{1}{2} \right) \tag{14.110}$$

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

$$\begin{aligned} h(n) = h(2M-1-n) &= -\cos(\theta_{n0}) \\ h(M+n) = h(M-1-n) &= -\sin(\theta_{n0}) \end{aligned} \tag{14.111}$$

for $0 \leq n \leq (M/2) - 1$. The bases for the ELT using the suggested angles are shown in Figure 14.30. In this figure, the 8-channel examples are for $N = 2$ ($K = 1$) and for $N = 4$ ($K = 2$).

14.9 Finite-Length Signals

Since the LT matrices are not square, in order to obtain n transformed subband samples one has to evaluate more than n samples of the input signal. For the same reason, n subband samples would generate more than n signals samples after inverse transformation. All the 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;
- using time-varying bases for the boundary regions.

We will discuss the first method only. The second is just applicable to few transforms and filter banks and can be covered elsewhere. The subject of time-varying LTs is very rich and provides for solutions to

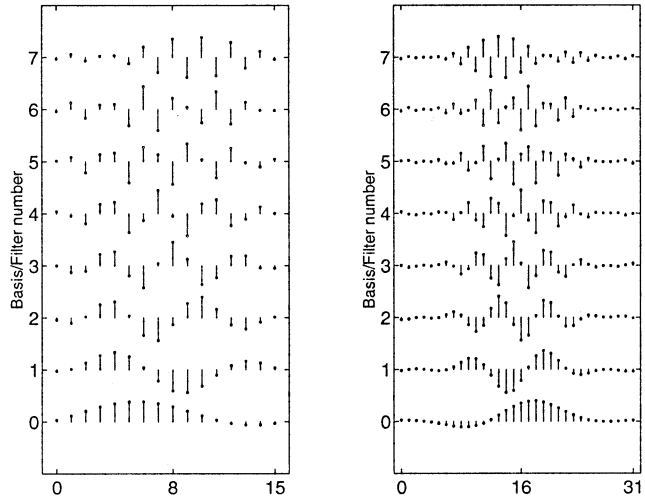

FIGURE 14.30 Example of ELT bases for the given angles design method for $M = 8$. Left: $K = 1$, $N = 2$; right: $K = 2$, $N = 4$.

several problems including the processing of boundary samples. We will not cover it in this chapter. The reader is referred to[7,28,29,32,41] and their references for further information on time-varying LTs.

14.9.1 Overall Transform

Here we assume the model of extension and windowing described in Figure 14.31.[33] 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:

$$\tilde{\mathbf{x}}^T = [\mathbf{x}_{e,l}^T, \mathbf{x}^T, \mathbf{x}_{e,r}^T] = [(\mathbf{R}_l x_l)^T, \mathbf{x}_l^T, \mathbf{x}_c^T, \mathbf{x}_r^T, (\mathbf{R}_r \mathbf{x}_r)^T]. \tag{14.112}$$

The extended sections are found by a linear transform of the boundary samples of $\mathbf{x}$ as shown in Figure 14.32, i.e.,

$$\mathbf{x}_{e,l} = \mathbf{R}_l \mathbf{x}_l, \qquad \mathbf{x}_{e,r} = \mathbf{R}_r \mathbf{x}_r \tag{14.113}$$

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.

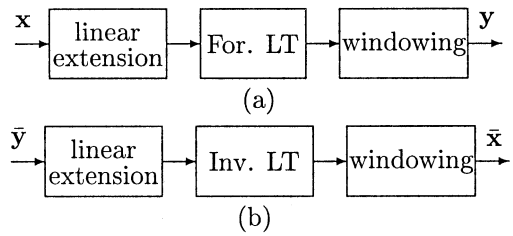

FIGURE 14.31 Extension and windowing in transformation of a finite-length signal using LTs. (a) Overall forward transform section. (b) Overall inverse transform section.

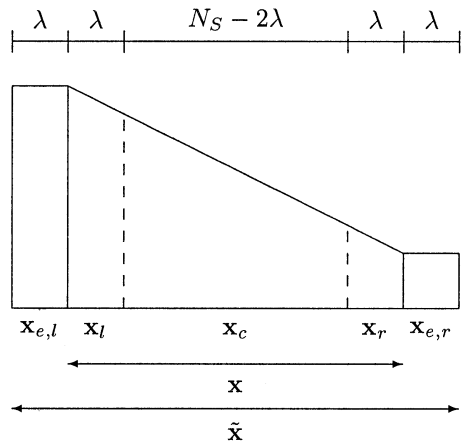

FIGURE 14.32 Illustration of signal extension of vector $\mathbf{x}$ into vector $\tilde{\mathbf{x}}$. In each border, $\lambda = (L - M)/2$ samples outside initial signal boundaries are found by linear relations applied to the λ boundary samples of $\mathbf{x}$, i.e., $\mathbf{x}_{e,l} = \mathbf{R}_l\mathbf{x}_l$ 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.

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 $\tilde{\mathbf{P}}$, i.e.,

$$\tilde{\mathbf{P}} = \begin{pmatrix} \ddots & & & \ddots & & & & 0 \\ & \mathbf{P}_0 & \mathbf{P}_1 & \cdots & \mathbf{P}_{N-1} & & & \\ & & \mathbf{P}_0 & \mathbf{P}_1 & \cdots & \mathbf{P}_{N-1} & & \\ & & & \mathbf{P}_0 & \mathbf{P}_1 & \cdots & \mathbf{P}_{N-1} & \\ 0 & & & & \ddots & & & \ddots \end{pmatrix}. \tag{14.114}$$

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 (14.21) 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 (14.32) 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:

$$\tilde{\mathbf{y}} = \tilde{\mathbf{P}}\tilde{\mathbf{x}}, \qquad \bar{\tilde{\mathbf{x}}} = \tilde{\mathbf{Q}}^T\tilde{\mathbf{y}} \ . \tag{14.115}$$

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

$$\bar{\tilde{\mathbf{x}}} = \tilde{\mathbf{Q}}^T\tilde{\mathbf{y}} = \tilde{\mathbf{Q}}^T\tilde{\mathbf{P}}\tilde{\mathbf{x}} = \tilde{\mathbf{T}}\tilde{\mathbf{x}} \tag{14.116}$$

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 (14.35) we can easily show that the transform matrix is:

$$\tilde{\mathbf{T}} = \tilde{\mathbf{Q}}^T\tilde{\mathbf{P}} = \begin{bmatrix} \mathbf{T}_L & & 0 \\ & \mathbf{I}_{N_x - 2\lambda} & \\ 0 & & \mathbf{T}_R \end{bmatrix} \tag{14.117}$$

where $\mathbf{T}_l$ and $\mathbf{T}_r$ are some $2\lambda \times 2\lambda$ matrices. Thus, distortion is just incurred 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

$$\mathbf{y} = \mathbf{T}\mathbf{x}, \qquad \bar{\mathbf{x}} = \mathbf{T}^{-1}\bar{\mathbf{y}} \tag{14.118}$$

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 $\mathbf{T}$'s orthogonality.[32] We will not use time-varying LTs here but rather use extended signals and transform matrices.

14.9.2 Recovering Distorted Samples

Let:

$$[\mathbf{\Phi}_l \,|\, \mathbf{\Phi}_r] = \left[\begin{array}{cccc|cccc} \mathbf{P}_0 & \mathbf{P}_1 & \cdots & \mathbf{P}_{N-2} & \mathbf{P}_{N-1} & & & \mathbf{0} \\ & \mathbf{P}_0 & \mathbf{P}_1 & \cdots & \mathbf{P}_{N-2} & \mathbf{P}_{N-1} & & \\ & & \ddots & \ddots & & \ddots & \ddots & \\ \mathbf{0} & & & \mathbf{P}_0 & \mathbf{P}_1 & \cdots & \mathbf{P}_{N-2} & \mathbf{P}_{N-1} \end{array}\right], \tag{14.119}$$

$$[\mathbf{\Psi}_l \,|\, \mathbf{\Psi}_r] = \left[\begin{array}{cccc|cccc} \mathbf{Q}_0 & \mathbf{Q}_1 & \cdots & \mathbf{Q}_{N-2} & \mathbf{Q}_{N-1} & & & \mathbf{0} \\ & \mathbf{Q}_0 & \mathbf{Q}_1 & \cdots & \mathbf{Q}_{N-2} & \mathbf{Q}_{N-1} & & \\ & & \ddots & \ddots & & \ddots & \ddots & \\ \mathbf{0} & & & \mathbf{Q}_0 & \mathbf{Q}_1 & \cdots & \mathbf{Q}_{N-2} & \mathbf{Q}_{N-1} \end{array}\right]. \tag{14.120}$$

Hence,

$$\mathbf{T}_l = \mathbf{\Psi}_l^T \mathbf{\Phi}_l, \qquad \mathbf{T}_r = \mathbf{\Psi}_r^T \mathbf{\Phi}_r \,. \tag{14.121}$$

If we divide $\bar{\tilde{\mathbf{x}}}$ in the same manner as $\tilde{\mathbf{x}}$,

$$\bar{\tilde{\mathbf{x}}} = [\bar{\mathbf{x}}_{e,l}^T, \bar{\mathbf{x}}_l^T, \bar{\mathbf{x}}_c^T, \bar{\mathbf{x}}_r^T, \bar{\mathbf{x}}_{e,r}^T] \,, \tag{14.122}$$

then,

$$\begin{bmatrix} \bar{\mathbf{x}}_{e,l} \\ \bar{\mathbf{x}}_l \end{bmatrix} = \mathbf{T}_l \begin{bmatrix} \mathbf{x}_{e,l} \\ \mathbf{x}_l \end{bmatrix} = T_l \begin{bmatrix} \mathbf{R}_l \mathbf{x}_l \\ \mathbf{x}_l \end{bmatrix} = \mathbf{T}_l \begin{bmatrix} \mathbf{R}_l \\ \mathbf{I}_\lambda \end{bmatrix} \mathbf{x}_l = \mathbf{\Gamma}_l \mathbf{x}_l \tag{14.123}$$

where

$$\mathbf{\Gamma}_l = \mathbf{T}_l \begin{bmatrix} \mathbf{R}_l \\ \mathbf{I}_\lambda \end{bmatrix} \tag{14.124}$$

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

$$\mathbf{x}_l = \boldsymbol{\Gamma}_l^+ \begin{bmatrix} \bar{\mathbf{x}}_{e,l} \\ \bar{\mathbf{x}}_l \end{bmatrix} = (\boldsymbol{\Gamma}_l^T \boldsymbol{\Gamma}_l)^{-1} \boldsymbol{\Gamma}_l^T \begin{bmatrix} \bar{\mathbf{x}}_{e,l} \\ \bar{\mathbf{x}}_l \end{bmatrix}. \tag{14.125}$$

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

$$\mathbf{x}_r = \boldsymbol{\Gamma}_r^+ \begin{bmatrix} \bar{\mathbf{x}}_r \\ \bar{\mathbf{x}}_{e,r} \end{bmatrix} = (\boldsymbol{\Gamma}_r^T \boldsymbol{\Gamma}_r)^{-1} \boldsymbol{\Gamma}_r^T \begin{bmatrix} \bar{\mathbf{x}}_r \\ \bar{\mathbf{x}}_{e,r} \end{bmatrix}, \tag{14.126}$$

where:

$$\boldsymbol{\Gamma}_r = \mathbf{T}_r \begin{bmatrix} \mathbf{I}_\lambda \\ \mathbf{R}_r \end{bmatrix} \tag{14.127}$$

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

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

- Select **P** and **Q** and identify their submatrices $\mathbf{P}_i$ and $\mathbf{Q}_i$
- Find $\boldsymbol{\Phi}_l$, $\boldsymbol{\Phi}_r$, $\boldsymbol{\Psi}_l$, $\boldsymbol{\Psi}_r$, from (14.119)(14.120)
- Find $\mathbf{T}_l$ and $\mathbf{T}_r$ from (14.121)
- Find $\boldsymbol{\Gamma}_l$ and $\boldsymbol{\Gamma}_r$ from (14.124)(14.127)
- Test rank of $\boldsymbol{\Gamma}_l$ and $\boldsymbol{\Gamma}_r$
- If ranks are λ, obtain $\boldsymbol{\Gamma}_l^+, \boldsymbol{\Gamma}_r^+$ and reconstruct $\mathbf{x}_l$ and $\mathbf{x}_r$

This is an extension of Reference 33 to non-orthogonal LTs, with the particular concern to test whether the pseudo inverses exist.

The model in Figure 14.31 and the proposed method are not applicable for some LTs. The notable classes of LTs include those LTs whose bases have different length and different symmetries. Examples are: (1) some two-channel nonorthogonal LTs with odd-length; (2) the FLT; (3) other composite systems, i.e. cascaded systems such as those used in Reference 35. For the first example, it is trivial to use symmetric extensions, but different symmetries for different bases.[44] 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 problems is that different filters would require different extensions during the forward transformation process, therefore, the model in Figure 14.31 is not applicable.

The above method works very well for *M*-channel filter banks whose filters have 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.

14.9.3 Symmetric Extensions

In case the LT is symmetric and obeys (14.65) and (14.66), 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 extension as described, applied to the last $\lambda = (L - M)/2$ samples on each border, resulting in a signal $\tilde{\mathbf{x}}(n)$ with $N_x + 2\lambda = N_x + L - M$ samples, as:

$$x(\lambda-1),\ldots, x(0), x(0),\ldots, x(N_x-1), x(N_x-1),\ldots, x(N_x-\lambda)\ . \tag{14.128}$$

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. Discard the first $N - 1$ output blocks, obtaining 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 $p_{kn} = \nu_k p_{k,L-1-n}$ for $0 \le k \le M-1$ and $0 \le n \le L-1$, i.e., $\nu_k = 1$ if p_{kn} is symmetric and $\nu_k = -1$ 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 $\bar{\bar{y}}_k(m)$, and invert the sign of the extended samples if p_{kn} is anti-symmetric. For s samples reflected around the borders, then the k-th subband signal will have samples:

$$\nu_k\hat{y}_k(s-1),\ldots,\nu_k\hat{y}_k(0),\hat{y}_k(0),\ldots\hat{y}_k(N_B-1),\nu_k\hat{y}_k(N_B-1),\ldots,\nu_k\hat{y}_k(N_B-s)\ .$$

The inverse transformation can be performed as:

- *N odd* – Reflect $s = (N-1)/2$ samples around each border, getting, thus, $N_B + N - 1$ blocks with 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 around each border, getting, thus, $N_B + N$ 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.

The approach will assure the perfect reconstruction property and orthogonality of the overall transformation if the LT is orthogonal.[32] The price paid is to run 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.

14.10 Conclusions

We hope this material will serve as an introduction to lapped transforms and give some insight into their nature. This 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 overlap of the basis functions. It was also shown that the analyses of MIMO systems, mainly their factorizations, are invaluable tools for the design of useful lapped transforms. That was the case in the design of transforms such as the LOT, LBT, GenLOT, GLBT, FLT, MLT and ELT. Those practical LTs were presented by not only presenting the general factorization, but by also plotting bases and describing in detail how to construct at least a good design example, either by printing basis entries or by providing all parameters necessary to construct the bases. Even if the particular examples are not ideal for a particular application the reader might have in mind, they may provide an experimental example, from which one can build up on, by exploring the references and performing customized optimization.

It is worth pointing that we intentionally avoided viewing the transforms as filter banks, so that the bases were not discussed as impulse responses of filters and their frequency response was not analyzed.

Nevertheless the framework is the same and so is the analysis of MIMO systems. Therefore, this chapter should give insight in such a vast field which is based on the study of multirate systems.

Good luck in the field of lapped transforms!

References

1. Boashash, B., Ed., 1992. *Time-Frequency Signal Analysis,* New York, NY: John Wiley & Sons.
2. Bordreaux-Bartels, G. F. "Mixed time-frequency signal transformations," in this Handbook.
3. Cassereau, P. 1985. *A New Class of Optimal Unitary Transforms for Image Processing,* Master's Thesis, Mass. Inst. Tech., Cambridge, MA.
4. Clarke, R. J. 1985. *Transform Coding of Images,* Orlando, FL: Academic Press.
5. Coifman, R., Meier, Y., Quaker, D. and Wickerhauser, V. 1991. "Signal processing and compression using wavelet packets," Technical Report, Dept of Mathematics, Yale Univ.
6. Doganata, Z., Vaidyanathan, P. P., and Nguyen, T. Q. 1988. "General synthesis procedures for FIR lossless transfer matrices, for perfect reconstruction multirate filter banks applications," *IEEE Trans. Acoust., Speech, Signal Processing,* Vol. 36, No. 10, pp. 1561–1574.
7. Herley, C., Kovacevic, J., Ramchandran, K. and Vetterli, M. 1993. "Tilings of the time-frequency plane: construction of arbitrary orthogonal bases and fast tiling algorithms," *IEEE Trans. on Signal Processing,* Vol. 41, pp. 3341–3359.
8. Hohn, F. E. 1964. *Elementary Matrix Algebra.* 2nd ed., New York, NY: MacMillan.
9. Jayant, N. S. and Noll, P. 1984. *Digital Coding of Waveforms.* Englewood Cliffs, NJ: Prentice-Hall.
10. Jozawa, H. and Watanabe, H. Sept. 4–6, 1991. "Intrafield/Interfield adaptive lapped transform for compatible HDTV coding," *4th International Workshop on HDTV and Beyond,* Torino, Italy.
11. Koilpillai, R. D. and Vaidyanathan, P. P. 1992. "Cosine modulated FIR filter banks satisfying perfect reconstruction," *IEEE Trans. Signal Processing,* Vol. 40, pp. 770–783.
12. Malvar, H. S. 1986. *Optimal Pre- and Post-Filtering in Noisy Sampled-Data Systems,* Ph.D. Dissertation. Mass. Inst. Tech., Cambridge, MA.
13. Malvar, H. S. 1988. "Reduction of blocking effects in imaging coding with a lapped orthogonal transform," *Proc. of Intl. Conf. on Acoust., Speech, Signal Processing,* Glasgow, Scotland, pp. 781–784.
14. Malvar, H. S. and Staelin, D. H. 1989. "The LOT: transform coding without blocking effects," *IEEE Trans. Acoust., Speech, Signal Processing,* ASSP-37, pp. 553–559.
15. Malvar, H. S. 1990. "Lapped transforms for efficient transform/subband coding," *IEEE Trans. Acoust., Speech, Signal Processing,* ASSP-38, pp. 969–978.
16. Malvar, H. S. 1988. "The LOT: a link between block transform coding and multirate filter banks," *Proc. Intl. Symp. Circuits and Systems,* Espoo, Finland, pp. 835–838.
17. Malvar, H. S. 1990. "Efficient signal coding with hierarchical lapped transforms," *Proc. of Intl. Conf. on Acoust., Speech, Signal Processing,* Albuquerque, NM, pp. 761–764.
18. Malvar, H. S. 1990. "Modulated QMF filter banks with perfect reconstruction," *Elect. Letters,* Vol. 26, pp. 906–907.
19. Malvar, H. S. 1991. "Extended lapped transform: fast algorithms and applications," *Proc. of Intl. Conf. on Acoust., Speech, Signal Processing,* Toronto, Canada, pp. 1797–1800.
20. Malvar, H. S. 1992. *Signal Processing with Lapped Transforms.* Norwood, MA: Artech House.
21. Malvar, H. S. 1992. "Extended lapped transforms: properties, applications and fast algorithms," *IEEE Trans. Signal Processing,* Vol. 40, pp. 2703–2714.
22. Malvar, H. S. 1998. "Biorthogonal and nonuniform lapped transforms for transform coding with reduced blocking and ringing artifacts," *IEEE Trans. on Signal Processing,* Vol. 46, pp. 1043–1053.
23. Nayebi, K., Barnwell, T. P., and Smith, M. J. 1992. "The time domain filter bank analysis: a new design theory," *IEEE Trans. on Signal Processing,* Vol. 40, pp. 1412–1429.
24. Nguyen, T. Q. and Koilpillai, R. D. 1996. "Theory and design of arbitrary-length cosine-modulated filter banks and wavelets satisfying perfect reconstruction," *IEEE Trans. on Signal Processing,* Vol. 44, pp. 473–483.

25. Oppenheim, A. V. and Schafer, R. W. 1989. *Discrete-Time Signal Processing,* Englewoods Cliffs, NJ: Prentice-Hall.
26. Pennebaker, W. B. and Mitchell, J. L. 1993. *JPEG: Still Image Compression Standard,* New York, NY: Van Nostrand Reinhold.
27. Princen, J. P. and Bradley, A. B. 1986. "Analysis/synthesis filter bank design based on time domain aliasing cancellation," *IEEE Trans. Acoust., Speech, Signal Processing,* ASSP-34, pp. 1153–1161.
28. de Queiroz, R. L. and Rao, K. R. 1993. "Time-varying lapped transforms and wavelet packets," *IEEE Trans. on Signal Processing,* Vol. 41, pp. 3293–3305.
29. de Queiroz, R. L. 1996. *On Lapped Transforms,* Ph.D. Dissertation, The University of Texas at Arlington.
30. de Queiroz, R. L. and Rao, K. R. 1995. "The extended lapped transform for image coding," *IEEE Trans. on Image Processing,* Vol. 4, pp. 828–832.
31. de Queiroz, R. L., Nguyen, T. Q., and Rao, K. R. Jan. 1994. "The generalized lapped orthogonal transforms," *Electronics Letters,* Vol. 30, pp. 107–107.
32. de Queiroz, R. L. and Rao, K. R. 1995. "On orthogonal transforms of images using paraunitary filter banks," *Journal on Visual Communications and Image Representation,* Vol. 6, No. 2, pp. 142–153.
33. de Queiroz, R. L. and Rao, K. R. 1995. "On reconstruction methods for processing finite-length signals with paraunitary filter banks," *IEEE Trans. on Signal Processing,* Vol. 43, pp. 2407–2410.
34. de Queiroz, R. L., Nguyen, T. Q., and Rao, K. R. 1996. "The GenLOT: generalized linear-phase lapped orthogonal transform," *IEEE Trans. on Signal Processing,* Vol. 44, pp. 497–507.
35. de Queiroz, R. L. 1997. "Uniform filter banks with non-uniform bands: post-processing design," *Proc. of Intl. Conf. Acoust., Speech, Signal Proc.,* Seattle, WA, Vol. III, pp. 1341–1344.
36. de Queiroz, R. L. and Eschbach, R. 1997. "Fast downscaled inverses for images compressed with M-channel lapped transforms," *IEEE Trans. on Image Processing,* Vol. 6, pp. 794–807.
37. Rabbani, M. and Jones, P. W. 1991. *Digital Image Compression Techniques,* SPIE Optical Engineering Press, Bellingham, WA.
38. Rao, K. R. and Yip, P. 1990. *Discrete Cosine Transform: Algorithms, Advantages, Applications,* San Diego, CA: Academic Press.
39. Rao, K. R. (ed.), 1985. *Discrete transforms and their applications,* New York, NY: Van Nostrand Reinhold.
40. Schiller, H. 1988. "Overlapping block transform for image coding preserving equal number samples and coefficients," *Proc. SPIE, Visual Communications and Image Processing,* Vol. 1001, pp. 834–839.
41. Sodagar, I., Nayebi, K., and Barnwell, T. P. 1993. "A class of time-varying wavelet transforms," *Proc. of Intl. Conf. on Acoust., Speech, Signal Processing,* Minneapolis, MN, Vol. III, pp. 201–204.
42. Soman, A. K. and Vaidyanathan, P. P. 1992. "Paraunitary filter banks and wavelet packets," *Proc. of Intl. Conf. on Acoust., Speech, Signal Processing,* Vol. IV, pp. 397–400.
43. Soman, A. K., Vaidyanathan, P. P., and Nguyen, T. Q. 1993. "Linear-phase paraunitary filter banks: theory, factorizations and applications," *IEEE Trans. on Signal Processing,* Vol. 41, pp. 3480–3496.
44. Strang, G. and Nguyen, T. 1996. *Wavelets and Filter Banks,* Wellesley, MA: Wellesley-Cambridge.
45. Temerinac, M. and Edler, B. 1992. "A unified approach to lapped orthogonal transforms," *IEEE Trans. Image Processing,* Vol. 1, pp. 111–116.
46. Tran, T. D. and Nguyen, T. Q. 1997. "On M-channel linear-phase FIR filter banks and application in image compression," *IEEE Trans. on Signal Processing,* Vol. 45, pp. 2175–2187.
47. Tran, T. D. 1998. *Linear Phase Perfect Reconstruction Filter Banks: Theory, Structure, Design, and Application in Image Compression,* Ph.D. thesis, University of Wisconsin, Madison, WI.
48. Tran, T. D., de Queiroz, R. L., and Nguyen, T. Q. 1998. "The variable-length generalized lapped biorthogonal transform," *Proc. Intl. Conf. Image Processing,* Chicago, IL, Vol. III, pp. 697–701.
49. Tran, T. D., de Queiroz, R., and Nguyen, T. Q. 1998. "The generalized lapped biorthogonal transform," *Proc. Intl. Conf. Acoust., Speech, Signal Proc.,* Seattle, Vol. III, pp. 1441–1444.
50. Vaidyanathan, P. P. 1993. *Multirate Systems and Filter Banks,* Englewood Cliffs, NJ: Prentice-Hall.

51. Vaidyanathan, P. P. and Hoang, P. 1988. "Lattice structures for optimal design and robust implementation of 2-channel PR-QMF banks," *IEEE Trans. Acoust., Speech, Signal Processing,* ASSP-36, pp. 81–94.
52. Vetterli, M. and Herley, C. 1992. "Wavelets and filter banks: theory and design," *IEEE Trans. Signal Processing,* Vol. 40, pp. 2207–2232.
53. Vetterli, M. and Kovacevic, J. 1995. *Wavelets and Subband Coding,* Englewood Cliffs, NJ: Prentice-Hall.
54. Wickerhauser, M. V. 1992. "Acoustical signal compression using wavelet packets," in *Wavelets: a Tutorial in Theory and Applications,* ed. C. K. Chui, Academic Press, San Diego, CA.
55. Young, R. W. and Kingsbury, N. G. 1993. "Frequency domain estimation using a complex lapped transform," *IEEE Trans. Image Processing,* Vol. 2, pp. 2–17.