Chapter 4

Transform Coding

In data compression, there is often a significant amount of correlation or redundance existing in the source data. A source symbol and its temporal or spatial neighborhood are often highly correlated. For example, in the *Lena* image as shown in Fig. 4.1, pixels in the box on the hat have similar values. One of the central tasks in efficient data compression design is to explore the source correlation and remove the redundant data so as to improve the rate-distortion performance of the compression system. In Chapter 3, we have discussed that vector quantization is able to exploit the source correlation. However, it computational complexity and implementation cost increases exponentially with the vector size. In this chapter, we introduce an efficient important approach to exploit source correlation: *transform coding*.

4.1 Block-Based Transform Coding

We consider a block-based transform coding system as illustrated in Fig. 4.2. The input is a vector of source symbols, denoted by $\mathbf{X} = [X_1, X_2, \cdots, X_N]^t$. If we assume that a matrix-based linear transform is used and the output vector is $\mathbf{Y} = [Y_1, Y_2, \cdots, Y_N]^t$, then

$$\mathbf{Y} = \mathbf{T}\mathbf{X}, \tag{4.1}$$

where $\mathbf{T}$ is the transform matrix. The transform ouptut Y_i is then quantized by a scalar quantizer and the reconstruction level is denoted by $\hat{Y}_i$, $1 \leq i \leq N$. The quantization outputs are entropy encoded and their codewords are mutiplexed into a bit stream. The problem in transform coding is: given a correlated input source, find an optimal transform matrix $\mathbf{T}$ such that the rate-distortion performance of the lossy data compression system is maximized.

4.1.1 Karhune-Loeve Transform (KLT)

Let $\mathbf{C}_X$ be the correlation matrix of the input vector $\mathbf{X} = [X_1, X_2, \cdots, X_N]^t$,

$$\mathbf{C}_X = E[\mathbf{X}\mathbf{X}^t] = [E(X_i X_j)]_{1 \leq i,j \leq N}\,. \tag{4.2}$$

Figure 4.1: The *Lena* image.

The correlation matrix of the transform output $\mathbf{Y} = [Y_1, Y_2, \cdots, Y_N]^t$, denoted by $\mathbf{C}_Y$, is

$$\begin{aligned} \mathbf{C}_Y &= E[\mathbf{Y} \cdot \mathbf{Y}^t] && (4.3)\\ &= E[\mathbf{TX} \cdot (\mathbf{TX})^t] && (4.4)\\ &= \mathbf{T} \cdot E[\mathbf{XX}] \cdot \mathbf{T}^t && (4.5)\\ &= \mathbf{T} \cdot \mathbf{C}_X \cdot \mathbf{T}^t. && (4.6) \end{aligned}$$

We choose the transform matrix $\mathbf{T}$ in such a way that transform outputs $\{y_i\}$ are independent of each other. In other words, $E(Y_i Y_j) = 0$, if $i \neq j$. This implies that $\mathbf{C}_Y$ is a diagonal matrix. To this end, we find the singular value decomposition of $\mathbf{C}_X$,

$$\mathbf{C}_X = \mathbf{U} \cdot \mathbf{\Lambda} \cdot \mathbf{U}^t, \tag{4.7}$$

where $\mathbf{U}$ is an orthonormal $N \times N$ matrix, and $\mathbf{\Lambda}$ is a diagonal matrix. Let

$$\mathbf{T} = \mathbf{U}^t. \tag{4.8}$$

We have

$$\mathbf{C}_Y = \mathbf{T} \cdot \mathbf{C}_X \cdot \mathbf{T}^t = \mathbf{U}^t\mathbf{U} \cdot \mathbf{\Lambda} \cdot \mathbf{U}^t\mathbf{U} = \mathbf{\Lambda}. \tag{4.9}$$

This says that, if the input source has a correlation matrix $\mathbf{C}_X$, we can choose a transform matrix $\mathbf{T} = \mathbf{U}^t$, such that the transform outputs are indepedent. This special transform is named the KLT transform.

4.1.2 Optimum Bit Allocation

Before we proceed to study the rate-distortion performance of transform coding, we introduce the optimum bit allocation. Considering a bank of N lossy data compression systems which encode

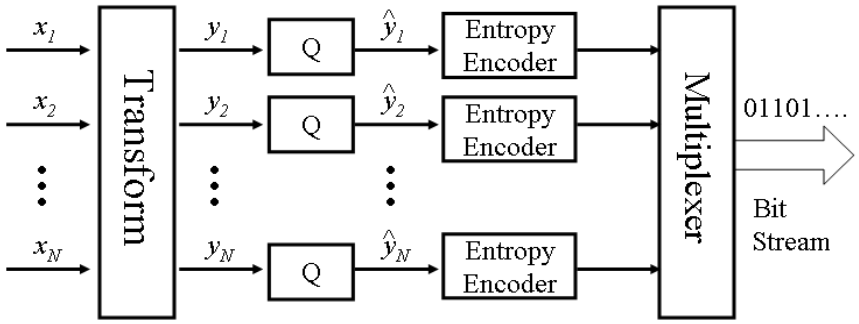

Figure 4.2: A lossy data compression system based on transform coding.

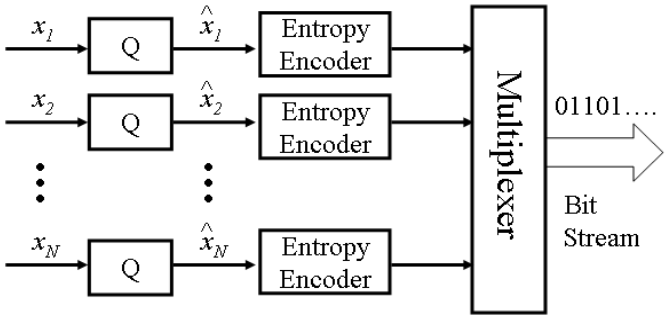

Figure 4.3: A lossy data compression system based using seperate coding without transform.

the input sources separately, as illustrated in Fig. 4.3. We assume that the input source to each branch is *i.i.d.* Let R_i be the coding bit rate of X_i. According to Chapter 3, the coding distortion is given be

$$D(R_i) = \sigma_i^2 2^{-2R_i}, \tag{4.10}$$

where σ^2 is the variance of X_i. The total bit rate is

$$R_T = \sum_{i=1}^{N} R_i, \tag{4.11}$$

and the overall distortion is

$$D = \sum_{i=1}^{N} D(R_i) = \sum_{i=1}^{N} \sigma_i^2 2^{-2R_i}. \tag{4.12}$$

In optimal bit rate allocation, we need to determine the bit rate for each source R_i such that the overall distortion D is minimized under the total bit rate constraint. This problem can be formulated as follows:

$$\min_{\{R_i\}} \quad D = \sum_{i=1}^{N} \sigma_i^2 2^{-2R_i} \tag{4.13}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} R_i \le R_T. \tag{4.14}$$

This constrained optimization problem can solved with the Lagrange multiplier approach, which converts the constrained optimization problem into a non-constrained optimization problem by incorporating the constraint into the objective function as a penalty term. Note that the minimum distortion is achieved when $\sum_{i=1}^{N} R_i = R_T$. Let

$$J(R_1, R_2, \cdots, R_N) = \sum_{i=1}^{N} \sigma_i^2 2^{-2R_i} + \lambda(\sum_{i=1}^{N} R_i - R_T). \tag{4.15}$$

At the minimum solution, the derivative of J with respect to R_i should be 0:

$$\frac{\partial J(R_1, R_2, \cdots, R_N)}{\partial R_i} = \sigma_i^2 2^{-2R_i}(-2\ln 2) + \lambda = 0. \tag{4.16}$$

We have

$$\lambda = 2\ln 2 \cdot \sigma_i^2 2^{-2R_i}, \quad i = 1, 2, \cdots, N. \tag{4.17}$$

Multiplying these N equations together, we have

$$\lambda^N = (2\ln 2)^N \cdot \prod_{i=1}^{N} \sigma_i^2 \cdot \prod_{i=1}^{N} 2^{-2R_i} \tag{4.18}$$

$$= (2\ln 2)^N \cdot \prod_{i=1}^{N} \sigma_i^2 \cdot 2^{-2R_T}, \tag{4.19}$$

which yields

$$\lambda = 2\ln 2 \left(\prod_{i=1}^{N} \sigma_i^2\right)^{\frac{1}{N}} 2^{-2R_T/N} \tag{4.20}$$

According to (4.17), the optimal bit allocation for source X_i is given by

$$R_i^* = \frac{R_T}{N} + \frac{1}{2}\log_2 \frac{\sigma_i^2}{\left(\prod_{i=1}^{N} \sigma_i^2\right)^{\frac{1}{N}}}, \tag{4.21}$$

and the minimum distortion is given by

$$D^* = \left(\prod_{i=1}^{N} \sigma_i^2\right)^{\frac{1}{N}} \cdot 2^{-2R_T/N}. \tag{4.22}$$

Let

$$\sigma_{GM}^X = \left(\prod_{i=1}^{N} \sigma_i^2\right)^{\frac{1}{N}}, \tag{4.23}$$

which is the geometric mean of the input variance $\{\sigma_i^2\}$. We have

$$D^*(R) = \sigma_{GM}^X \cdot 2^{-2R}, \tag{4.24}$$

where $R = R_T/N$ is the average bit rate of the transform coding system. This equation is the overall rate-distortion function of the transform coding system. We can see that this rate-distortion

function $D^*(R)$ shares the same expression as the rate-distortion function for each individual encoder.

The above optimal rate allocation has achieved the following two objectives. First, it find the best minimum distortion or the best compression quality that the transform coding system can achieve for a given bit rate. Second, it gets rid of the intermediate variables, i.e., the coding bit rates $\{R_1, R_2, \cdots, R_N\}$ for individual encoders using an optimization process and derives the overall rate-distortion function for the whole transform coding system. This allows us to study the contribution of the transform to the overall compression performance of the transform coding system which is measured by its rate-distortion function.

4.1.3 Rate-Distortion Analysis of Transform Coding

In this section, we study the rate-distortion performance of transform coding using the KLT transform. The major result is that transforming the input source using the KLT matrix is able to improve the rate-distortion performance of a lossy data compression system. In addition, we are going to show that KLT is the optimal transform that maximizes the rate-distortion performance.

4.1.4 Transform Coding Gain

We compare two lossy data compression schemes: KLT-based transform coding as shown in Fig. 4.2, and separate coding or scalar coding as shown in Fig. 4.3. The scalar coding system passes the input vector directly to individual encoders. For fair comparison, we apply optimal rate allocation to both transform coding and scalar coding systems. The only difference between them is the use of transform. In this was, we can measure the contribution of transform by studying the difference between rate-distortion performance of these two systems.

For the scalar coding system, We perform optimal bit allocation among the input sources $\{X_i\}$. According to (4.22), the minimum coding distortion in scalar coding is

$$D_{SC} = \left(\prod_{i=1}^{N} \sigma_{X_i}^2\right)^{\frac{1}{N}} \cdot 2^{-2R} \tag{4.25}$$

$$= \sigma_{GM}^X \cdot 2^{-2R}. \tag{4.26}$$

For the transform coding system, let $\sigma_{Y_i}^2 = E[Y_i Y_i]$ be the variance of Y_i. We perform optimum bit allocation on the transform outputs $\{Y_i\}$. For the same bit rate R, the minimum distortion of the transform coding system is given by

$$D_{TC} = \left(\prod_{i=1}^{N} \sigma_{Y_i}^2\right)^{\frac{1}{N}} \cdot 2^{-2R} \tag{4.27}$$

$$= \sigma_{GM}^Y \cdot 2^{-2R}. \tag{4.28}$$

To measure the contribution of the transform, we define the ratio between these two distortion

D_{SC} and D_{TC} The

$$G_{TC} = \frac{D_{SC}}{D_{TC}} = \frac{\sigma_{GM}^X}{\sigma_{GM}^Y} \tag{4.29}$$

$$= \frac{\left(\prod_{i=1}^{N} \sigma_{X_i}^2\right)^{\frac{1}{N}}}{\left(\prod_{i=1}^{N} \sigma_{Y_i}^2\right)^{\frac{1}{N}}}. \tag{4.30}$$

In the following, we will introduce two assumption to simplify the above expression for the transform coding gain G_{TC}.

4.1.5 Assumption 1: Stationary Input Sources

First, we observe that the input sources $(X_1, X_2, \cdots, X_N)$ are stationary, or they have the same statistical distribution. This is fairly reasonable in practice. For example, from an image, we take a block of N consecutive pixels as the input vector $[x_1, x_2, \cdots, x_N]$ to the transform coding system. We then have a large number of blocks. The random variable X_i represents the i-th pixels in these blocks. Note that these i-th pixels of these blocks should have similar distribution as the j-th pixels of these blocks. In other words, the random variable X_i and X_j have similar distributions. Specifically, they should have the same variance

$$\sigma_{X_1}^2 = \sigma_{X_1}^2 = \cdots = \sigma_{X_N}^2, \tag{4.31}$$

In this case, their geometric mean σ_{GM}^X should be equal to their arithmetic mean σ_{AM}^X since they are constant. Specifically, we have

$$G_{TC} = \frac{\sigma_{GM}^X}{\sigma_{GM}^Y} = \frac{\sigma_{AM}^X}{\sigma_{GM}^Y}, \tag{4.32}$$

where

$$\sigma_{AM}^X = \frac{1}{N} \sum_{i=1}^{N} \sigma_{X_i}^2. \tag{4.33}$$

4.1.6 Assumption 2: Orthonormal Transforms

Second, we assume that the transform matrix T is orthonormal,

$$T \cdot T^t = T^t \cdot T = I. \tag{4.34}$$

With $\mathbf{Y} = \mathbf{WX}$, we have

$$\sum_{i=1}^{N} \sigma_{Y_i}^2 = \sum_{i=1}^{N} E[Y_i \cdot Y_i] \tag{4.35}$$

$$= E[Y^t \cdot Y] \tag{4.36}$$

$$= E[X^t T^t \cdot T\mathbf{X}] \tag{4.37}$$

$$= E[X^t \cdot X] \tag{4.38}$$

$$= \sum_{i=1}^{N} \sigma_{X_i}^2. \tag{4.39}$$

Therefore,

$$\sigma_{AM}^X = \sigma_{AM}^Y. \tag{4.40}$$

In other words, if the transform is orthonormal, then, the energy of the transform outputs is equal to the energy of the inputs. With this, the transform coding gain becomes

$$G_{TC} = \frac{\sigma_{AM}^X}{\sigma_{GM}^Y} = \frac{\sigma_{AM}^Y}{\sigma_{GM}^Y} \tag{4.41}$$

where σ_{AM}^Y and σ_{GM}^Y are the arithmetic and geometric means of the transform output variance $\{\sigma_{Y_i}^2\}$, respectively. It is well known that for any set of non-negative values, their arithmetic mean is always larger than their geometric mean. Therefore, the transform coding gain in (4.41) is always larger or equal to one. This implies that, if the input source is stationary, transforming the input source with an orthonormal matrix will always improve (at least won't degrade) the rate-distortion performance of the data compression system.

4.1.7 Optimal Orthonormal Transform

The transform coding gain defines the contribution of the transform. Next, we will demonstrate that the KLT transform maxmize the transform coding gain G_{TC}. Or, the KLT is the optimal transform. More specifically,

Theorem: Among all orthonormal transforms, the KLT transform maximizes the transform coding gain in (4.41).

Proof: Note that in the transform coding gain expression in (4.41), the nominator is always equal to the input source energy $\sum_{i=1}^{N} \sigma_{X_i}^2$, which is the same as the output energy $\sum_{i=1}^{N} \sigma_{Y_i}^2$, since the orthonormal transform does not change the energy. Now, we only need to show that the KLT transform minimizes the denominator. To this end, we need to use the following inequality: for any symmetric positive matric $A = [a_{ij}]_{1 \le i,j \le N}$, we have

$$\det[A] \le \prod_{i=1}^{N} a_{ii}. \tag{4.42}$$

Suppose an arbitrary orthonormal transform $\mathbf{T}$ is used. Let Y_i^T be the transform outputs and $\mathbf{C}_Y^T$ the correlation matrix. Note that $\mathbf{C}_Y^T$ is a symmetric positive matrix, according to (4.42), we have

$$\prod_{i=1}^{N} \sigma_{Y_i^T}^2 \ge \det[\mathbf{C}_Y^T] = \det[\mathbf{C}_X]. \tag{4.43}$$

The second equality is because orthonormal transforms do not change the determination. Suppose the KLT transform is used. Let Y_i^K be the transform outputs and $\mathbf{C}_Y^K$ the correlation matrix. Since $\mathbf{C}_Y^K$ is diagonal, we have

$$\prod_{i=1}^{N} \sigma_{Y_i^K}^2 = \det[\mathbf{C}_Y^K] = \det[\mathbf{C}_X]. \tag{4.44}$$

Therefore,

$$\prod_{i=1}^{N} \sigma^2_{Y_i^T} \geq \det[\mathbf{C}_Y^T] \geq \prod_{i=1}^{N} \sigma^2_{Y_i^K} = \det[\mathbf{C}_Y^K], \tag{4.45}$$

which implies that the KLT transform yields the smallest denominator in (4.41), hence the highest transform coding gain.

This theorem suggests that the most efficient way to transform the input source symbols is to make them independent, as in the KLT transform. This also explains why decorrelating the source data is crtical in achieving efficient data compression.

4.1.8 Energy Compaction of KLT

4.1.9 Limitations of KLT in Practical Data Compression

Although the KLT is optimal in terms of its capability in maximizing the transform coding, it has a number of limitations in practical data compression. First, the KLT assumes that the source is stationary with known statistics, i.e., the correlation matrix. However, in practical data compression, the input sources, such as images or videos, are often not stationary. Its source correlation varies temporally and / or spatially. This requires us to frequently update the source correlation matrix and re-compute KLT matrix, which is computationally intensive. Second, to reconstruct the source data, the decoder needs to know the KLT matrix. Therefore, the encoder has to spend a lot of bits to encode the KLT matrix and send it to the decoder once it is updated. This introduces a significant amount of overhead in data compression. Therefore, it is often desirable to use a signal-independent transform. It has been found that the discrete cosine transform (DCT) has a coding performance very close that of KLT. Therefore, DCT is used in many practical data compression systems, such as image and video compression.

4.2 Discrete Cosine Transform

4.2.1 Fourier Cosine Transform

We start from the definition of Fourier transform. Given a function $x(t)$, $-\infty < t < +\infty$, its forward Fourier transform is defined as

$$X(w) = \mathcal{F}[x(t)] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} x(t) e^{-jwt}\, dt, \tag{4.46}$$

and the inverse Fourier transform is

$$x(t) = \mathcal{F}^{-1}[X(w)] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} X(w) e^{jwt}\, dw. \tag{4.47}$$

If $x(t)$ is an even function, we have

$$\begin{aligned} \mathcal{F}[x(t)] &= \frac{1}{\sqrt{2\pi}} \int_{0}^{+\infty} x(t) e^{-jwt}\, dt + \frac{1}{\sqrt{2\pi}} \int_{0}^{+\infty} x(-t) e^{jwt}\, dt & (4.48) \\ &= \sqrt{\frac{2}{\pi}} \int_{0}^{+\infty} x(t) \cos wt\, dt. & (4.49) \end{aligned}$$

Note that

$$X(-w) = X(w). \tag{4.50}$$

Similarly, the inverse Fourier transform becomes

$$\begin{aligned} \mathcal{F}[X(w)] &= \frac{1}{\sqrt{2\pi}} \int_0^{+\infty} X(W) e^{jwt}\, dw + \frac{1}{\sqrt{2\pi}} \int_0^{+\infty} X(-w) e^{-jwt}\, dw & (4.51) \\ &= \sqrt{\frac{2}{\pi}} \int_0^{+\infty} X(w) \cos wt\, dw. & (4.52) \end{aligned}$$

Therefore, we can define the forward and inverse Fourier cosine transform as follows:

$$\begin{aligned} X_c(w) &= \mathcal{F}_c[x(t)] = \sqrt{\frac{2}{\pi}} \int_0^{+\infty} x(t) \cos wt\, dt, & (4.53) \\ x(t) &= \mathcal{F}_c^{-1}[X_c(w)] = \sqrt{\frac{2}{\pi}} \int_0^{+\infty} X(w) \cos wt\, dw. & (4.54) \end{aligned}$$

4.2.2 One-Dimensional Discrete Cosine Transform

The discrete cosine transform (DCT) is the discritized version of the Fourier cosine transform. In a 1-D DCT, an input vector of size N, $\mathbf{X} = [x_0, x_1, \cdots, x_{N-1}]^t$, is mapped into a vector $\mathbf{Y} = [y_0, y_1, \cdots, y_{N-1}]^t$ as follows,

$$\mathbf{Y} = \mathbf{T}_c \cdot \mathbf{X}, \tag{4.55}$$

where $\mathbf{T_c}$ is the DCT tranform matrix given by

$$\mathbf{T}_c = [t_{kn}]_{0 \le k,n \le N-1}, \quad t_{kn} = C(k) \cos \frac{(2n+1)k\pi}{2N}, \tag{4.56}$$

where

$$C(k) = \begin{cases} \sqrt{\frac{1}{N}} & k = 0, \\ \sqrt{\frac{2}{N}} & k = 1, 2, \cdots, N-1. \end{cases} \tag{4.57}$$

It can be shown that $\mathbf{T_c}$ is unitary, i.e.,

$$\mathbf{T}_c \cdot \mathbf{T}_c^t = \mathbf{T}_c^t \cdot \mathbf{T}_c = \mathbf{I}. \tag{4.58}$$

Therefore, the inverse DCT is given by

$$\mathbf{X} = \mathbf{T}_c^t \cdot \mathbf{Y}. \tag{4.59}$$

The forward and inverse N-point DCT can be written into the following summation forms:

$$\begin{aligned} y_k &= \sum_{n=0}^{N-1} C(k) \cos \frac{(2n+1)k\pi}{2N} x_n, \quad 1 \le k \le N-1, & (4.60) \\ x_n &= \sum_{k=0}^{N-1} C(k) \cos \frac{(2n+1)k\pi}{2N} y_k, \quad 1 \le n \le N-1. & (4.61) \end{aligned}$$

We can also consider the DCT transform as a vector representation: representing the input vector $\mathbf{X}$ by a linear combination of basis vectors of an N-dimensional space. Let

$$\mathbf{T}_c^t = [\mathbf{t}_0, \mathbf{t}_1, \cdots, \mathbf{t}_{N-1}], \tag{4.62}$$

where $\mathbf{t}_k$ is the k-th column vector of matrix $\mathbf{T}_c^t$, or the k-th row vector of $\mathbf{T}_c$. Since $\mathbf{T}_c$ is unitary, $\{\mathbf{t}_k\}$ forms an orthonormal basis vector set of the N-dimensional Euclid space. The inverse DCT in (4.59) can be re-written as

$$\mathbf{X} = \sum_{k=0}^{N-1} y_k \mathbf{t}_k, \tag{4.63}$$

which represents the input signal $\mathbf{X}$ using a linear combination of the basis vectors $\{\mathbf{t}_k\}$. Similarly, the forward DCT in (4.60) can re-written as

$$y_k = (\mathbf{X}, \mathbf{t}_k), \tag{4.64}$$

where $(\cdot, \cdot)$ represents the inner product of two vectors. Fig. 4.4 depicts the basis vector set $\{\mathbf{t}_k\}$, $1 \le k \le N-1$ used in a 16-point DCT. It can be seen that as k increases, the frequency of change in $\{\mathbf{t}_k\}$ becomes larger and larger. Therefore, we often refer to k as the frequency component index. Since the output y_k is the inner product between the input and the k-th basis vector, we refer to y_k as the k-th frequency component of the input signal. Here, y_0 represents the DC component of the input. Those y_k with lower values of k are called low-frequency components. Accordingly, those with higher values are called high-frequency components. Fig. 4.5 shows one line of 16 pixels $\{x_n\}$, $0 \le n \le 15$, taken from the Lena image. Fig. 4.6 shows the DCT output $\{y_k\}$, $0 \le k \le 15$. We can see that only the first few DCT coefficients have relatively large values, but the rest are very small, nearly zeros. From the energy-compaction perspective, the DCT compacts the input signal energy into few low-frequence components.

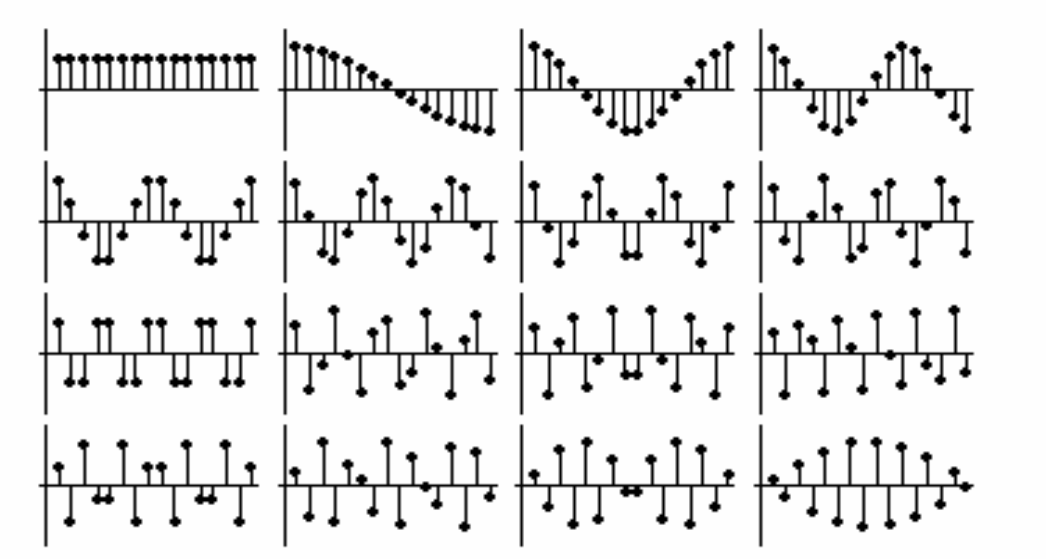

Figure 4.4: Basis functions of a 16-point 1-D DCT.

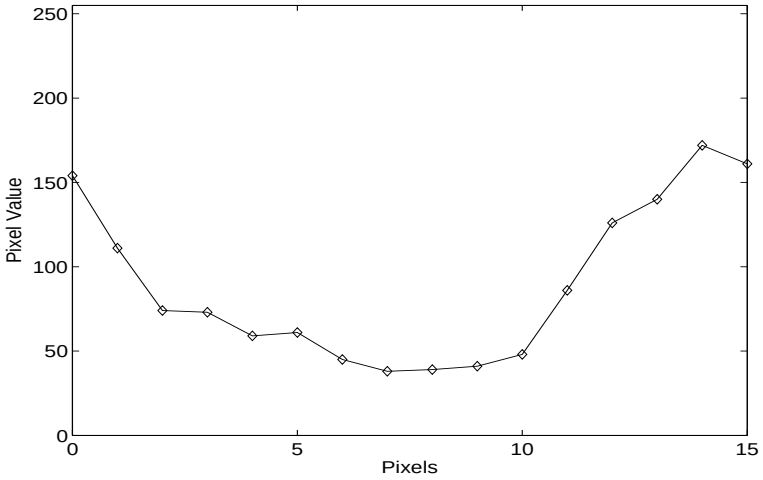

Figure 4.5: A line of 16 pixels in the Lena image.

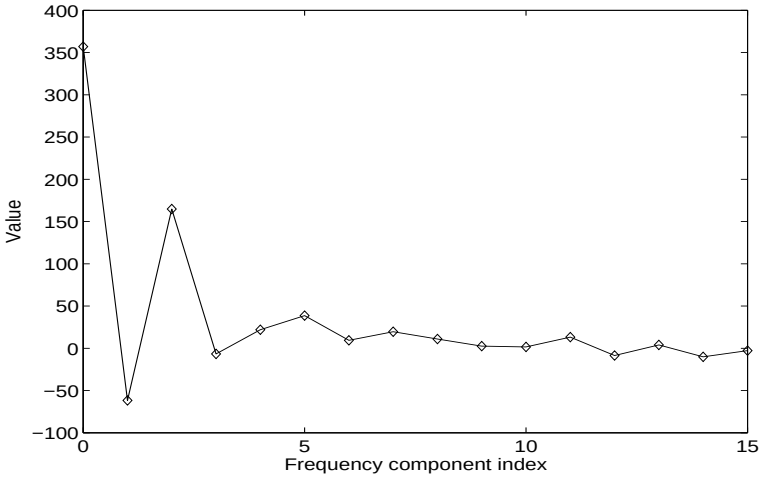

Figure 4.6: DCT output.

4.2.3 Relationship between DCT and KLT

Suppose the input source sequence $\mathbf{X} = [\mathbf{X}_0, \mathbf{X}_1, \cdots, \mathbf{X}_{N-1}]$ is a first-order Markov process and its auto-correlation matrix is

$$\mathbf{C}_\mathbf{X} = [E(\mathbf{X}_k \mathbf{X}_l)] = \rho^{|k-l|}, \quad 0 \leq k, l \leq N-1. \tag{4.65}$$

then the DCT basis functions are very close to those of the KLT. For example, Fig. 4.7 shows the basis functions for the KLT of a first-order Markov source with $\rho = 0.95$. We can see that they are very close to the DCT basis functions shown in Fig. 4.4. In general, the DCT is less efficient than the KLT. However, it should be also the DCT is source-independent, while the KLT depends on the source statistis and has several limitations as discussed in Section 4.1.9, DCT has been extensively used in transform coding of images and videos.

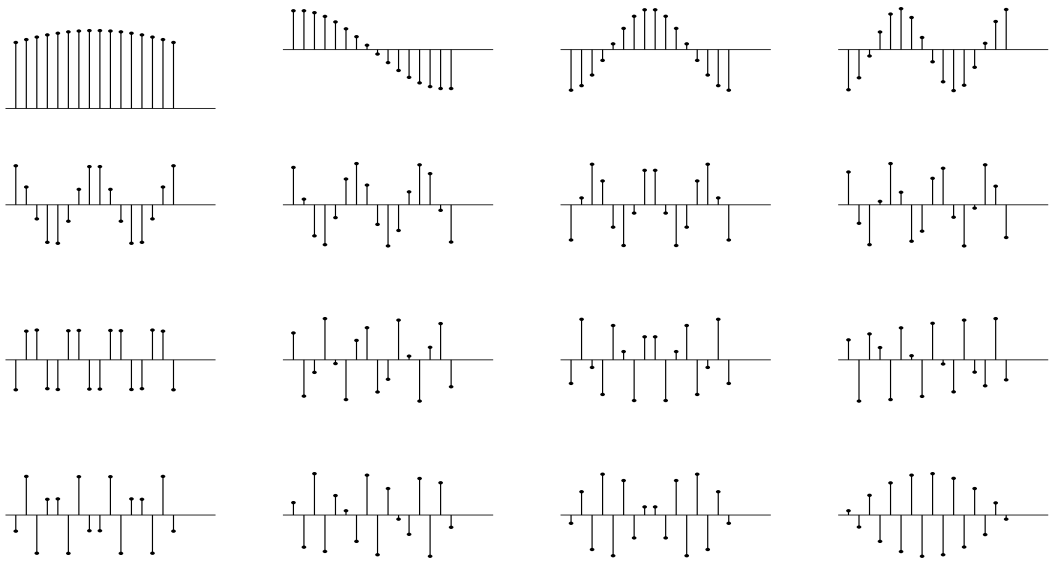

Figure 4.7: The basis functions for the KLT with $\rho = 0.95$.

4.2.4 2-D DCT

The 1-D DCT can be extended to the 2-D case. The input to a 2-D DCT is a $M \times N$ block of data, denoted by $\mathbf{X} = [x_{mn}]$, $0 \le m \le M$, $0 \le n \le N-1$, and the output is a 2-D block $\mathbf{Y} = [y_{kl}]_{0 \le k,l \le N-1}$, where

$$y_{kl} = \sqrt{\frac{2}{N}}\sqrt{\frac{2}{M}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \Lambda(m)\Lambda(n) \cos\left[\frac{(2m+1)k\pi}{2M}\right] \cos\left[\frac{(2n+1)l\pi}{2N}\right] \cdot x_{mn}. \tag{4.66}$$

with

$$\Lambda(i) = \begin{cases} \frac{1}{\sqrt{2}}, & i = 0, \\ 1 & \text{otherwise.} \end{cases} \tag{4.67}$$

We call this transform as an $M \times N$ 2-D DCT. $\{y_{kl}\}$ are called DCT coefficients of $\mathbf{X}$. The 2-D DCT is seperable. It can be achieved by applying an N-point 1-D DCT to every row of the input block $\mathbf{X}$ followed by an M-point 1-D DCT column-wise. Fig. 4.8 shows the 2-D basis functions for the 2-D DCT. Fig. 4.9(b) shows the 512×512 2-D DCT coefficients of the Lena image in Fig. 4.9(a). We can see that the energy of the input image is compacted into a small number of DCT coefficients in the upper-left corner while the rest DCT coefficients are very close to zeros.

4.3 Discrete Wavelet Transform and Subband Decomposition

4.3.1 Contineous Wavelet Transform

"*Wave*" means that a function should have a mean of 0 and *wave* around 0. "let" means that the function has to be well localized. Fourier transform and Fourier cosine transform are localized in

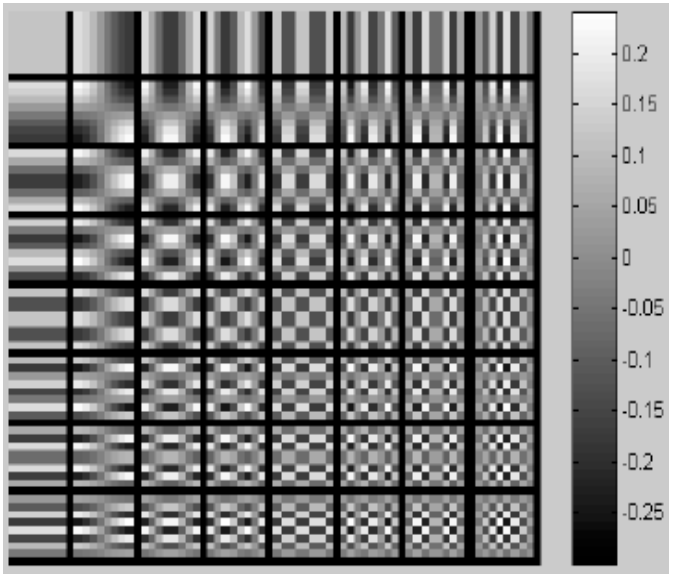

Figure 4.8: Basis functions of 2-D 8×8 DCT.

frequency, but not in time. The wavelet transform is localized in both frequency and in time.

$$\mathcal{W}[f](a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(t)\phi^*(\frac{t-b}{a})\, dt, \tag{4.68}$$

where a and b are called the scale and translation parameters, respectively, and $\phi(t)$ is the basis wavelet. Let $\Phi(w)$ be the Fourier transform of $\phi(t)$. Define

$$C_\phi = \int_{-\infty}^{+\infty} \frac{|\Phi(w)|^2}{w}\, dw. \tag{4.69}$$

The inverse wavelet transform is given by

$$f(t) = \frac{1}{C_\phi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \mathcal{W}[f](a,b) \cdot \phi(\frac{t-b}{a}) \frac{1}{a^2}\, da\, db. \tag{4.70}$$

4.3.2 Introduction to Multirate Signal Processing

Definition

Consider a discrete-time signal $x[n]$, $n = 0, \pm 1, \pm 1, \cdots$, its z-transform is given by

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n}, \tag{4.71}$$

and its discrete-time Fourier transform (DTFT) is given by

$$X(e^{jw}) = X(z)|_{z=e^{jw}} = \sum_{n=-\infty}^{\infty} x[n] e^{-jwn}. \tag{4.72}$$

The quantities $|X(e^{jw})|$ and $\theta(w) = \arg\{X(e^{jw})\}$ are called its magnitude and phase spectrum.

(a) (b)

Figure 4.9: (a) The *Lena* image and (b) its 2-D DCT coefficients.

In the time domain, we can perform down-sampling and up-sampling of a discrete-time signal. In down-sampling with a sampling rate of M, we take every M-th sample of the input sequence to generate the output sequence, as illustrated in Fig. 4.10(a). More specifically, the output sequence, denoted by $y[n]$, is given by

$$y_d[n] = x[nM], \quad n = 0, \pm 1, \pm 1, \cdots . \tag{4.73}$$

In up-sampling with a sampling rate of L, we insert $L-1$ zero-valued samples between two consecutive samples of the input sequence $x[n]$ and the ouptput sequence is

$$y_u[n] = \begin{cases} x[n/L], & n = 0, \pm L, \pm 2L, \cdots , \\ 0, & \text{otherwise.} \end{cases} \tag{4.74}$$

(a) (b)

Figure 4.10: Sampling of a discrete-time signal: (a) down-sampling; (b) up-sampling.

Frequency-Domain Analysis

We now derive the relationship between the input and output signals of the sampling operation in the frequency-domain. We start with the up-sampling operation. The z-transform of the up-

sampling output is given by

$$\begin{aligned} Y_u(z) &= \sum_{n=-\infty}^{\infty} y[n]z^{-n} \\ &= \sum_{k=-\infty}^{\infty} x[k]z^{-kL} \\ &= X(z^L). \end{aligned} \tag{4.75}$$

To derive the z-transform of the down-sampling output $y_d[n]$, we introduce the following "sampling" function:

$$S[n] = \begin{cases} 1, & n = 0, \pm M, \pm 2M, \cdots, \\ 0, & \text{otherwise.} \end{cases} \tag{4.76}$$

It can be shown that

$$S[n] = \frac{1}{M} \sum_{l=0}^{M-1} e^{j\frac{2\pi ln}{M}}. \tag{4.77}$$

Now, we can re-write the down-sampling output as

$$\begin{aligned} Y_d(z) &= \sum_{n=-\infty}^{\infty} y[n]z^{-n} \\ &= \sum_{n=-\infty}^{\infty} x[nM](z^{1/M})^{-nM} \\ &= \sum_{k=-\infty}^{\infty} x[k] \cdot S[k](z^{1/M})^{-k} \\ &= \frac{1}{M} \sum_{k=-\infty}^{\infty} x[k] \left(\sum_{l=0}^{M-1} e^{j\frac{2\pi lk}{M}} \right) (z^{1/M})^{-k} \\ &= \frac{1}{M} \sum_{l=0}^{M-1} \left[\sum_{k=-\infty}^{\infty} x[k] e^{j\frac{2\pi lk}{M}} (z^{1/M})^{-k} \right] \\ &= \frac{1}{M} \sum_{l=0}^{M-1} X(z^{1/M} e^{j\frac{2\pi l}{M}}). \end{aligned} \tag{4.78}$$

Example: If $M = 2$, according to (4.78), we have

$$Y_d(z) = \frac{1}{2}[X(z^{\frac{1}{2}}) + X(-z^{\frac{1}{2}})], \tag{4.79}$$

and its DTFT is

$$Y_d(e^{jw}) = \frac{1}{2}[X(e^{jw/2}) + X(-e^{jw/2})]. \tag{4.80}$$

4.3.3 Two-Channel Quadrature-Mirror Filter Bank

The two-channel quadrature-mirror filter (QMF) bank is often used in discrete wavelet transform and subband decomposition. As illustrated in Fig. 4.11, the input signal $x[n]$ is passed through two

analysis filters $H_0(z)$ and $H_1(z)$, which typically have lowpass and highpass frequency responses, respectively. The filtering outputs, denoted by $U_0[n]$ and $U_1[n]$, are called lowpass and highpass subbands, respectively. Then subband signals are then down-sampled by 2 and the outputs, denoted by $V_0[n]$ and $V_1[n]$, are encoded and transmitted to the decoder side. We assume that the subband signals are perfectly reconstructed at the decoder. At the decoder side, the subband signals are up-sampled by 2 and then passed through synthesis filters $G_0[z]$ and $G_1[z]$. The filter outputs are added together to form the output $y[n]$.

In QMF design, we need to design the analysis and synthesis filters such that that the output $y[n]$ is the same as (or very close to) the input $x[n]$. This condition is also called perfect (or near perfect) reconstruction. We analysis the behavior of the QMF bank in the z-domain. We have

$$U_k(z) = H_k(z)X(z), \tag{4.81}$$

$$V_k(z) = \frac{1}{2}[U_k(z^{1/2}) + U_k(-z^{1/2})], \tag{4.82}$$

$$W_k(z) = V_k(z^2), \tag{4.83}$$

for $k = 0, 1$, which yield

$$W_k(z) = \frac{1}{2}[U_k(z) + U_k(-z)] = \frac{1}{2}[H_k(z)X(z) + H_k(-z)X(-z)]. \tag{4.84}$$

The output of the filter bank is given by

$$\begin{aligned} Y(z) &= G_0(z)W_0(z) + G_1(z)W_1(z) \\ &= \frac{1}{2}[H_0(z)G_0(z) + H_1(z)G_1(z)]X(z) \\ &+ \frac{1}{2}[H_0(-z)G_0(z) + H_1(-z)G_1(z)]X(-z) \qquad (4.85) \\ &= T(z)X(z) + A(z)X(-z), \qquad (4.86) \end{aligned}$$

where

$$T(z) = \frac{1}{2}[H_0(z)G_0(z) + H_1(z)G_1(z)] \tag{4.87}$$

is called the *distortion transfer function* and

$$A(z) = \frac{1}{2}[H_0(-z)G_0(z) + H_1(-z)G_1(z)]. \tag{4.88}$$

Here, $A(z)X(-z)$ is called the aliasing term.

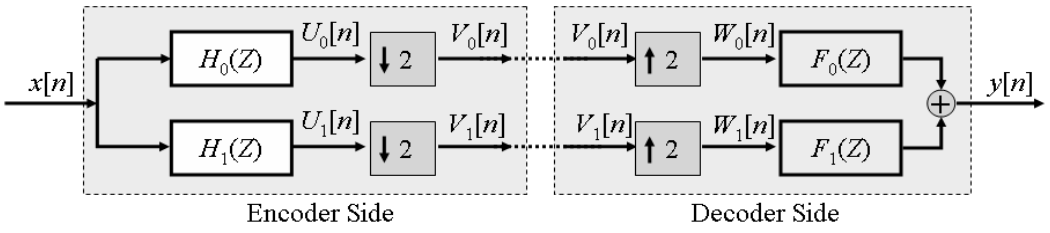

Figure 4.11: The two-channel QMF filter bank.

4.3.4 Perfection Reconstruction of Two-Channel FIR Filter Banks

We re-write the filter bank ouptut $Y(z)$ in the following matrix form,

$$Y(z) = \frac{1}{2}[G_0(z)\ G_1(z)] \begin{bmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{bmatrix} \begin{bmatrix} X(z) \\ X(-z) \end{bmatrix} \tag{4.89}$$

We have

$$Y(-z) = \frac{1}{2}[G_0(-z)\ G_1(-z)] \begin{bmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{bmatrix} \begin{bmatrix} X(z) \\ X(-z) \end{bmatrix} \tag{4.90}$$

Combining the above two equations we have

$$\begin{aligned} \begin{bmatrix} Y(z) \\ Y(-z) \end{bmatrix} &= \frac{1}{2} \begin{bmatrix} G_0(z) & G_1(z) \\ G_0(-z) & G_1(-z) \end{bmatrix} \begin{bmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{bmatrix} \begin{bmatrix} X(z) \\ X(-z) \end{bmatrix} \\ &= \frac{1}{2}\mathbf{G}(z)\mathbf{H}(z)^t \begin{bmatrix} X(z) \\ X(-z) \end{bmatrix}, \end{aligned} \tag{4.91}$$

where

$$\mathbf{G}(z) = \begin{bmatrix} G_0(z) & G_1(z) \\ G_0(-z) & G_1(-z) \end{bmatrix}, \tag{4.92}$$

and

$$\mathbf{H}(z) = \begin{bmatrix} H_0(z) & H_1(z) \\ H_0(-z) & H_1(-z) \end{bmatrix}. \tag{4.93}$$

To achieve perfect reconstruction, we need to have

$$Y(z) = z^{-l}X(z), \tag{4.94}$$

and correspondingly,

$$Y(-z) = z^{-l}X(-z), \tag{4.95}$$

where l is an integer, representing the processing delay. This implies that

$$\frac{1}{2}\mathbf{G}(z)\mathbf{H}(z)^t = \begin{bmatrix} z^{-l} & 0 \\ 0 & (-z)^{-l} \end{bmatrix}. \tag{4.96}$$

If l is an odd postive integer, solving the equation for $G_0(z)$ and $G_1(z)$, we have

$$G_0(z) = \frac{2z^{-l}}{\det[\mathbf{H}(z)]} \cdot H_1(-z), \tag{4.97}$$

and

$$G_1(z) = -\frac{2z^{-l}}{\det[\mathbf{H}(z)]} \cdot H_0(-z), \tag{4.98}$$

where

$$\det[\mathbf{H}(z)] = H_0(z)H_1(-z) - H_0(-z)H_1(z). \tag{4.99}$$

For FIR analysis filters $H_0(z)$ and $H_1(z)$, the synthesis filters $G_0(z)$ and $G_1(z)$ will also be FIR if

$$\det[\mathbf{H}(z)] = cz^{-k}, \tag{4.100}$$

where k is a positive integer. In this case, we have

$$G_0(z) = \frac{2}{c} z^{-(l-k)} H_1(-z), \tag{4.101}$$

$$G_1(z) = -\frac{2}{c} z^{-(l-k)} H_0(-z). \tag{4.102}$$

Eq. (4.100) is called the perfect reconstruction condition. In the following, we discuss two types of filter bank designs which satisfy this perfect reconstruction condition.

4.3.5 Orthogonal Filter Bank Design

To satisfy the perfect reconstruction condition in (4.100), we can let

$$H_1(z) = z^{-N} H_0(-z^{-1}). \tag{4.103}$$

We then have

$$\det[\mathbf{H}(z)] = -z^{-N}[H_0(z)H_0(z^{-1}) + H_0(-z)H_0(-z^{-1})]. \tag{4.104}$$

If we choose $H_0(z)$ such that

$$H_0(z)H_0(z^{-1}) + H_0(-z)H_0(-z^{-1}) = 1, \tag{4.105}$$

we have

$$\det[\mathbf{H}(z)] = -z^{-N}, \tag{4.106}$$

which satisfies the perfect reconstruction condition. Note that, let $z = e^{jw}$, Eq. (4.105) becomes

$$\begin{aligned} H_0(z)H_0(z^{-1}) &+ H_0(-z)H_0(-z^{-1}) = H_0(e^{jw})H_0(e^{-jw}) + H_0(-e^{jw})H_0(-e^{-jw}) \\ &= H_0(e^{jw})[H_0(e^{jw})]^* + H_0(e^{j(\pi-w)})[H_0(e^{j(\pi-w)})]^* \\ &= |H_0(e^{jw})|^2 + |H_0(e^{j(\pi-w)})|^2 = 1. \end{aligned} \tag{4.107}$$

Therefore, the condition of Eq. (4.105) is called the *power symmetric* condition. A filter satisfying this condition is called a power symmetric filter. Now, the perfect reconstruction filter band design reduces to finding a power symmetric FIR filter $H_0(z)$. A perfect reconstruction power-symmetric filter bank is also called *orthogonal filter bank*.

Example: It can be verified that the following filter

$$H_0(z) = -0.3415(1 + z^{-1})^2[1 - (2 - \sqrt{3})z^{-1}] \tag{4.108}$$

is power-symmetric, i.e., satisfies Eq. (4.105).

4.3.6 Biorthongal Filter Bank Design

Before introducing the second approach to perfect reconstruction filter bank design, we define some special filters.

Definition: A filter $H(z)$ is called linear-phase if its phase spectrum $\theta(w) = \arg\{H(e^{jw})\}$ satisfies

$$\theta(w) = -d \cdot w, \tag{4.109}$$

where d is called the *phase delay.*

It can be shown that a causal FIR filter $h[n]$ of length $N+1$ has a linear phase if $h[n]$ is either symmetric, i.e.,

$$h[n] = h[N-n], \quad 0 \leq N \leq N, \tag{4.110}$$

or $h[n]$ is anti-symmetric, i.e.,

$$h[n] = -h[N-n], \quad 0 \leq N \leq N, \tag{4.111}$$

Definition: A filter $H(z)$ is called half-band if

$$H(z) + H(-z) = 1. \tag{4.112}$$

In the frequency-domain, this implies that

$$H(e^{jw}) + H(-e^{jw}) = H(e^{jw}) + H(e^{\pi - jw}) = 1. \tag{4.113}$$

Example: $H_0(z) = 0.5 + z^{-1}E(z^2)$, where $E(z)$ is a FIR filter, is a half-band filter.

Now, we are ready to introduce the second type of perfect reconstruction filter banks, *biorthogonal filter banks.* The analysis filters $H_0(z)$ and $H_1(z)$ in biorthogonal filter banks satisfy the following conditions: (1) $H_0(z)$ and $H_1(z)$ are linear-phase filters. (2) Let $F(z) = H_0(z)H_1(-z)z^N$ is a half-band filter Under these two conditions, we have

$$\det[\mathbf{H}(z) = H_0(z)H_1(-z) - H_0(-z)H_1(z) = z^{-N}[F(z) + F(-z)] = z^{-N}. \tag{4.114}$$

which satisfies the perfection reconstruction condition in Eq. (4.100). Therefore, to design an biorthogonal filter bank, we need to design a half-band filter $F(z)$ such that $z^{-N}F(z)$ can be factorized as follow:

$$z^{-N}F(z) = H_0(z)H_1(-z), \tag{4.115}$$

where $H_0(z)$ and $H_1(z)$ are linear-phase filters. The two synthesis filters are given by

$$G_0(z) = H_1(-z), \quad G_1(z) = -H_0(-z). \tag{4.116}$$

Example: Daubechies (5, 3) filter bank. It can be verified that

$$F(z) = \frac{1}{16}(-z^3 + 9z + 16 + 9z^{-1} - z^{-3}) \tag{4.117}$$

is a half-band filter, and $z^{-3}F(z)$ can be factorized as

$$z^{-3}F(z) = H_0(z)H_1(-z), \tag{4.118}$$

where

$$H_0(z) = \frac{1}{8}(-1 + 2z^{-1} + 6z^{-2} + 2z^{-3} - z^{-4}), \tag{4.119}$$

$$H_1(z) = \frac{1}{2}(-1 + 2z^{-1} - z^{-2}). \tag{4.120}$$

Table 4.1: Debauchies (9, 7) filter bank.

n	$h_0[n]$	$h_1[n]$	$g_0[n]$	$g_1[n]$
0	+0.6029490182363579	+1.115087052456994	+1.115087052456994	+0.6029490182363579
±1	+0.2668641184428723	-0.5912717631142470	+0.5912717631142470	-0.2668641184428723
±2	-0.07822326652898785	-0.05754352622849957	-0.05754352622849957	-0.07822326652898785
±3	-0.01686411844287495	+0.09127176311424948	-0.09127176311424948	+0.01686411844287495
±4	+0.02674875741080976			+0.02674875741080976

The synthesis filters are then given by

$$G_0(z) = \frac{1}{2}(1 + 2z^{-1} + z^{-2}) \tag{4.121}$$

$$G_1(z) = \frac{1}{8}(-1 - 2z^{-1} + 6z^{-2} - 2z^{-3} - z^{-4}). \tag{4.122}$$

Another commonly used biorthogonal filter bank is Debauchies (9, 7) filter bank, whose filter coefficients are list in Table 4.1.

Problems

Problem 1. Design a 16-point KLT transform for images. More specifically, using the training images, such as *Lena* and *Barbara*, to determine the correlation matrix $\mathbf{C_X}$ of the input source $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_{16}]$. Then, find the SVD decomposition of $\mathbf{C_X}$ and determine the KLT matrix according to (4.8). Plot the basis functions of the KLT transform, i.e., the column vectors of the KLT transform matrix.

Problem 2. Show that

$$S[n] = \frac{1}{M} \sum_{l=0}^{M-1} e^{j\frac{2\pi ln}{M}}, \tag{4.123}$$

where $S[n]$ is given by (4.76).

Problem 3. Write a Matlab for C code to implement and evaluate the Debauchies (5, 3) filter bank, including both analysis and synthesis modules. Use this filter bank to perform subband decomposition of images, for example, the *Peppers* image. Compute the mean squared error between the original image and the reconstructed one.