CHAPTER

Filter Banks and Wavelet Compression

6

CHAPTER OUTLINE

6.1 Introduction to multiscale processing . . . 172
6.1.1 The short-time Fourier transform and the Gabor transform . . . 173
6.1.2 So what is a wavelet? . . . 174
6.1.3 Wavelet and filter bank properties . . . 177
6.2 Perfect reconstruction filter banks . . . 178
6.2.1 Filter and decomposition requirements . . . 178
6.2.2 The 1-D filter bank structure . . . 178
6.3 Multirate filtering . . . 182
6.3.1 Upsampling . . . 183
6.3.2 Downsampling . . . 183
6.3.3 System transfer function . . . 186
6.3.4 Perfect reconstruction . . . 189
6.3.5 Spectral effects of the two-channel decomposition . . . 191
6.4 Useful filters and filter banks . . . 191
6.4.1 Quadrature mirror filters . . . 192
6.4.2 Wavelet filters . . . 194
6.4.3 Multistage (multiscale) decompositions . . . 197
6.4.4 Separability and extension to 2-D . . . 200
6.4.5 Finite length sequences, edge artifacts, and boundary extension . . . 202
6.4.6 Wavelet compression performance . . . 203
6.5 Coefficient quantization and bit allocation . . . 204
6.5.1 Bit allocation and zonal coding . . . 204
6.5.2 Hierarchical coding . . . 207
6.6 JPEG2000 . . . 207
6.6.1 Overview . . . 207
6.6.2 Architecture—bit planes and scalable coding . . . 207
6.6.3 Coding performance . . . 208
6.6.4 Region of interest coding . . . 208
6.6.5 Benefits and status . . . 210

Communicating Pictures. http://dx.doi.org/10.1016/B978-0-12-405906-1.00006-4

6.7 Summary 210
References 210

Wavelets are mathematical functions that can be used as an alternative to transforms such as the DCT to split a signal into different frequency bands (subbands) prior to processing and quantization. Wavelets are usually implemented as banks of filters that enable each subband to be analyzed at a resolution matched to its scale. This, as we will see later, is normally achieved by combining filtering operations with sub-sampling, which also has the benefit of maintaining a constant overall sample rate at all stages in the decomposition. This approach has significant advantages over Fourier-based methods, especially when signals are not harmonically related, are of short duration with discontinuities or are non-stationary. It also has benefits in terms of subband coefficient quantization since the artifacts produced tend to be perceptually less annoying than the blocking artifacts associated with the DCT.

This chapter firstly examines the basic two-channel filter bank structure and then investigates the filter properties that enable perfect reconstruction in the absence of quantization. We then go on to consider the more general case of multirate filtering, showing how, through the appropriate combination of filters and up- and downsampling operations, we can achieve critical sampling. We next examine specific cases of useful subband and wavelet filters and then go on to extend the basic architecture to the case of a multistage (multiscale) decomposition and to two dimensions as required for image compression applications. Finally, we examine bit allocation strategies and scalability in the context of the JPEG2000 still image coding standard.

6.1 Introduction to multiscale processing

As the name suggests, wavelets are small waves with compact support. They can be designed to have specific and useful properties and, in the same way as Fourier or DCT bases, can be linearly combined to synthesize a signal. The difference with wavelets is that the basis functions at different scales are related through translation and dilation of the basic building block functions. Wavelets can thus be localized in both spatial (or temporal) and frequency domains and, as such, offer important multiresolution properties. They provide much more flexibility in tiling the time–frequency or spatial frequency plane than conventional transforms, offering a means of analyzing signals at different scales and with different resolutions. For example, the tiles can vary in size to better match the characteristics of the human visual system for image coding (or the auditory system in the case of speech coding), thus creating a basis for improved quantization strategies and, hence, improved compression performance.

Historically, development of the theory of wavelets (although they were not originally called wavelets) preceded that of filter banks due to the early work of Haar over 100 years ago [1]. The area was then dormant for many years until the emergence of subband coding methods for speech, audio, and image processing in the 1980s. One

For colour versions of Figures 6.24, 6.27 and 6.31 please refer to the electronic version or the website.

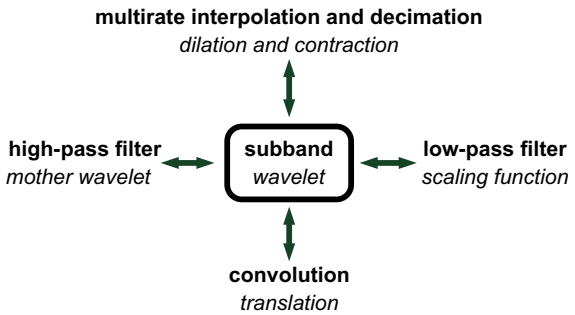


FIGURE 6.1

Equivalences between subband filtering and wavelet nomenclature.

of the major challenges identified at that time was the design of appropriate channel splitting filters, and the earliest contributions in this respect were by Crosier et al., who introduced the concept of the quadrature mirror filter (QMF) [2]. This then led to the development of perfect reconstruction filters with significant contributions from Smith and Barnwell [3,4], Vetterli [5], Woods and O'Neill [6], and Vaidyanathan [7].

Work on wavelets developed in parallel, perhaps starting with the contribution in 1984 from Grossman and Morlet [8], who were the first to use the term "wavelet". This initiated a major body of work on the topic, with major contributions from Daubechies [9] and several others. Wavelet theories and their associated design methods were developed into a comprehensive framework by Mallat [10,11] in 1989.

Although we will primarily adopt a subband filtering rather than a wavelet oriented nomenclature here, it is worth comparing the terms commonly used in each domain. This is illustrated in Figure 6.1. Many excellent texts exist on the topics of wavelets and subband filtering [12–14] and the reader is referred to these for further information.

6.1.1 The short-time Fourier transform and the Gabor transform

One of the main problems with conventional transforms (such as the Fourier transform) is that they are constructed around a set of infinite length basis functions such as sinusoids or complex exponentials. This means that any temporally (or spatially) localized changes or features are spread over the whole frequency domain. This is a particular problem for non-stationary signals such as those typical of images and videos.

One approach to overcome this has been to divide the input signal into shorter time duration sections. While this solves the issue of non-stationarity, it introduces additional problems due to boundary effects and edge distortions. As we saw in Chapter 5, these can distort the underlying spectrum and, in the context of compression, make perfect reconstruction of the original signal impossible. To reduce the influence of these boundary effects it has been conventional in spectrum analysis to window each block of data. The window function reduces to zero at the edges of the block, hence eliminating temporal discontinuities and any associated spectral ringing effects. If we

consider the case of the short-time Fourier transform (STFT), this is defined by equation (6.1) where, if the window function $w(t)$ is a Gaussian function, then this is referred to as the Gabor transform.

$$X(\omega, \tau) = \int_{-\infty}^{\infty} x(t)w(t-\tau)\mathrm{e}^{-j\omega t}\mathrm{d}t \tag{6.1}$$

A primary issue with the STFT is that the window size is fixed, so any variation in the temporal extent of any non-stationarities cannot be adequately represented. Also, no good basis functions exist for the STFT. Another problem is that the window function distorts the underlying signal which means that perfect reconstruction is not possible unless overlapping windows are used.

Clearly, because of the uncertainty principle, we cannot have both arbitrarily high temporal resolution and frequency resolution. We can however trade off temporal and frequency resolution by altering the window duration used and this is considered next.

6.1.2 So what is a wavelet?

Ideally we would like a set of basis functions that can isolate localized high frequency components in a signal while, at the same time, capturing lower frequency components of longer duration. Consider the example in Figure 6.2, where two signals—a sinusoid and an impulse—are analyzed in the frequency and temporal domains. The time

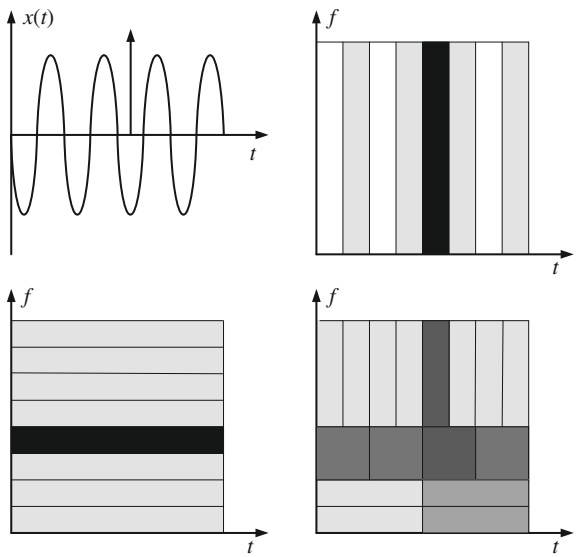


FIGURE 6.2

Comparison of time domain (top right), frequency domain (bottom left), and time–frequency analysis (bottom right) applied to the combination of a sinusoidal signal and an impulse function (top left). (Example adapted from Vetterli and Kovacevic [12].)

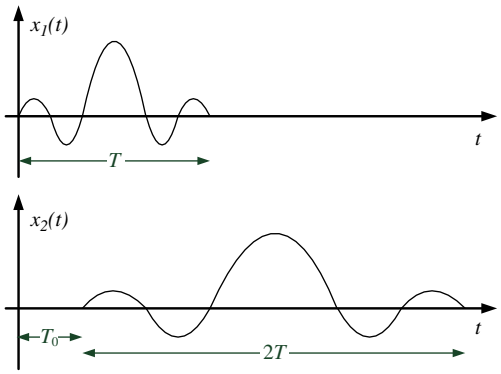


FIGURE 6.3

Wavelet translation and dilation.

domain analysis clearly picks out the impulse, whereas the frequency domain analysis localizes the sine wave. In contrast, frequency domain analysis cannot localize the impulse and time domain analysis cannot localize the sine wave. If we were able to tile the time–frequency plane more flexibly, as shown in the bottom right subfigure, then it would be possible to pick out both short duration high frequency signal components as well as longer duration low frequency components.

In practice, we can do this by combining short high frequency basis functions with longer lower frequency ones. Wavelets achieve this in a very elegant way, through the dilation and translation of a single prototype function. Consider Figure 6.3, where a basic function of duration T is dilated by a factor of 2 and translated by an amount T_0. Clearly the relationship between $x_1(t)$ and $x_2(t)$ in the figure is:

$$x_2(t) = x_1\left(\frac{t - T_0}{2}\right)$$

We can imagine a linear combination of weighted versions of these translated and dilated functions being used to approximate a signal of interest. However, unlike conventional basis functions constructed from cosine terms or complex exponentials, these exhibit compact support as well as oscillatory behavior. They thus offer the potential to capture the local features as well as the large scale characteristics of a signal.

The continuous wavelet transform (CWT)

We can define a single general prototype function (equation (6.2)) that can be stretched and shifted to form new functions which can be linearly combined to synthesize a signal of interest. This is the basic wavelet transform as given in equation (6.3).

$$h_{a,b}(t) = \frac{1}{\sqrt{a}} h\left(\frac{t - b}{a}\right) \tag{6.2}$$

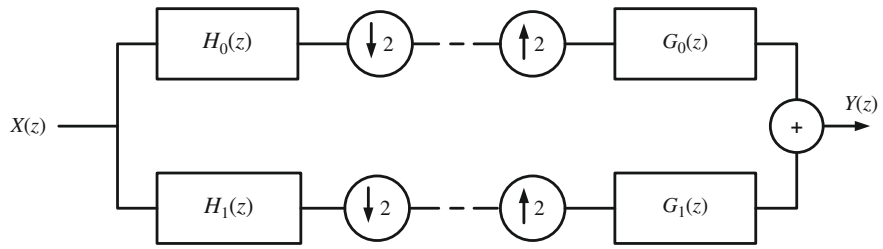


FIGURE 6.4

Two-channel perfect reconstruction filter bank.

$$X(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} h\left(\frac{t-b}{a}\right) x(t)\mathrm{d}t \tag{6.3}$$

Here a is the scale parameter, b is the translation parameter, and $h(t)$ is called the mother wavelet (i.e. the prototype function from which all the other bases are formed).

This ability to deal with local variations as well as big image features in the same framework is clearly attractive for image compression. However, in order to apply it, we need to be able to evaluate it in discrete form. The continuous time wavelet transform is highly redundant, so the scale space can be sampled dyadically using a wavelet series approach. This however is not a true discrete transform, but fortunately one does exist and we will consider that next.

The discrete wavelet transform (DWT)

Although the CWT can be computed discretely as described above, this is still only a sampled version of the CWT and not a proper discrete transform. The discrete wavelet transform (DWT) is instead based on a set of discrete functions or filters that are convolved with the input signal to effect translation and that are effectively dilated by sub-sampling the signal as it passes through each scale. It is normal to use octave band decompositions with dyadic sampling. This means that the wavelet filters form a set of band-pass responses that provide constant-Q filtering. As the width of the octave bands reduce, more localized high frequency details are captured by the representation. To achieve this we therefore, in equation (6.3), constrain the scale and translation terms to $a = 2^j$ and $b = k2^j$ respectively, where j and k are integers.

The wavelet decomposition is usually realized in the form of a filter bank as shown (for the case of a simple two-band split) in Figure 6.4. The input signal is spectrally decomposed into distinct bands in an analysis section which uses carefully designed filters in conjunction with downsampling[1] to split the signal without increasing the effective sample rate. After analysis, the information in each band can be processed (e.g. quantized) independently according to the characteristics of the source and/or the application. The signal is then passed though a matched synthesis section that

[1]Downsampling is sometimes referred to as decimation.

combines filtering with upsampling to reconstruct the original signal or (in the case of lossy compression) an approximation to it.

The wavelet transform and its filter bank realization do not simply divide a signal into multiple frequency bands, but it does this iteratively, normally only iterating the low-pass output at each scale. It thus produces a series of band-pass outputs which are, in fact, the wavelet transform coefficients. The role of the wavelet is played by the high-pass filter and the cascade of low-pass filters followed by a high-pass filter. This will be discussed in more detail later in Section 6.4.3 and is illustrated in Figure 6.19.

As indicated above, this framework has been formalized by Mallat and others, and the reader is referred to Refs. [10,12] for further details.

6.1.3 Wavelet and filter bank properties

As we have discussed above, the DWT is normally implemented as a filter bank and, in this respect, it is identical to the conventional subband filtering structure. Wavelet filter banks can be considered as a subclass of subband filterbanks where the filters used have some specific properties associated with regularity. This means that, as the filters are iterated, their characteristic shape converges to a regular function, i.e. one that is continuous or continuously differentiable. A necessary condition in this respect is that the filter should have sufficient zeros at the half sampling frequency (i.e. at $z = -1$). This ensures that the spectral images introduced during each filtering stage are attenuated. An excellent review of the requirements for convergence is provided by Rioul and Vetterli [15].

Before we look in detail at the construction of wavelet filter banks, let us summarize some of their advantages. Wavelet coding offers a number of potential benefits over other transforms. These include:

- **Constant relative bandwidth processing:** As illustrated in Figure 6.2, this facilitates more flexible and perceptually accurate tiling of the spatial frequency (or time–frequency) plane.
- **Improved perceptual quality:** As discussed previously, the main disadvantage of transform coding is the subjective degradation at lower bit rates where the viewer is able to perceive the outline of the transform blocks. Subband approaches are however (generally) free of block distortions. By decomposing the signal into a number of individual (possibly overlapping) frequency bands, each band may be quantized differently according to subjective HVS or energy distribution criteria.
- **Scalability:** As well as improved perceptual performance, wavelet methods also provide a basis for scalability,[2] both for images—using for example Set Partitioning (SPIHT) [16] or Embedded Block Truncation Coding (EBCOT) [17], and for video using 3-D wavelets and motion compensated temporal filtering (MCTF) [18].

[2]Scalability refers to the property where, after encoding, a compressed bitstream can be manipulated to reduce its bit rate without decoding. This is useful in order to adapt to prevailing channel conditions without the need for explicit transcoding.

- **Error resilience:** A single error during transmission can be modeled as an impulse that excites the synthesis filters to produce a weighted impulse response, which is superimposed on the correct response. With smooth short filters, this produces a localized ringing effect which is perceptually preferable to the blocking errors common with the DCT. Of course error propagation due to prediction loops and variable length coding is still possible. Techniques such as wavelet-based Pyramid Vector Quantization (PVQ) [19] overcome this rather successfully by using fixed length codes (see Chapter 11).

Although not selected for incorporation in standards such as JPEG, MPEG-2/4, H.263/4 or HEVC, subband (wavelet) decomposition techniques are widely acknowledged to offer improved performance and flexibility (with similar implementation complexity) compared with their block-transform counterparts, such as the DCT. This is particularly true for image coding, especially at lower bit rates. These benefits have been acknowledged through the adoption of wavelet coding methods in the most recent still image coding standard, JPEG2000 [20,21].

6.2 Perfect reconstruction filter banks

6.2.1 Filter and decomposition requirements

A simple two-channel filter bank is shown in Figure 6.4. Here, as we will shortly see, $H_0(z)$ and $H_1(z)$ are the low- and high-pass analysis filters and $G_0(z)$ and $G_1(z)$ are the corresponding synthesis filters. The circles in Figure 6.4 represent the downsampling (decimation) and upsampling (interpolation) operators. These are non-linear operators that:

1. **Ensure critical sampling:** The effective sampling rate throughout the filter bank is kept constant. This is important in compression applications as otherwise there is an expansion in the amount of data being processed.
2. **Are responsible for introducing aliasing and imaging:** This, as we will see later, must be managed through the careful design of the filters used.

In designing filters for filter bank use, we must therefore consider two important issues:

1. The filters used in practice do not have perfect transition characteristics and so there will be spectral cross-talk across the bands.
2. The sample rate changing operations will introduce aliasing and spectral imaging that must be eliminated during reconstruction if perfect reconstruction (a property essential for lossless compression) is to be achieved.

6.2.2 The 1-D filter bank structure

The fundamental building block of subband or wavelet filtering is the two-channel filter bank of Figure 6.4. We will examine this structure and its properties in detail

later. In the meantime, let us consider how it comes about using a very simple approach to filtering and sampling.

Intuitive development of the two-channel filter bank

The basic idea of the two-band split is to divide a signal into high-pass and low-pass components such that they can be analyzed or quantized independently in a manner related to signal scale. Consider a simple signal that comprises a steadily changing function with some superimposed random fluctuations, for example that given below:

$$x[n] = \{2, 4, 8, 8, 10, 14, 12, 6, 8, 4\}$$

We could estimate the underlying function through low-pass filtering and isolate the noise through high-pass filtering. Perhaps the simplest low-pass filter is the two-tap moving average as given by equation (6.4). The residual signal, $r[n]$, remaining after applying equation (6.4) is the high-pass component and can be characterized using equation (6.5).

$$l[n] = \frac{x[n] + x[n-1]}{2} \tag{6.4}$$

$$r[n] = x[n] - l[n] = \frac{x[n] - x[n-1]}{2} \tag{6.5}$$

Let us examine the frequency domain characteristics of these two simple filters in more detail, firstly the moving average or low-pass filter. From equation (6.4) we can see that:

$$L(z) = 0.5\left(X(z) + X(z)z^{-1}\right)$$

and thus:

$$H_l(z) = 0.5\left(\frac{z+1}{z}\right) \tag{6.6}$$

This, as expected, provides a simple low-pass filter response due to its pole at the origin in the z plane and its zero at $\Omega = \pi$. Similarly, the difference operation or high-pass filter is given by equation (6.7) which also has a pole at the origin and a zero at $\Omega = 0$.

$$\begin{aligned} R(z) &= 0.5\left(X(z) - X(z)z^{-1}\right) \\ H_r(z) &= 0.5\left(1 - z^{-1}\right) = 0.5\left(\frac{z-1}{z}\right) \end{aligned} \tag{6.7}$$

The frequency responses of this complementary pair are shown in Figure 6.5. These responses together form an all-pass function—which is not entirely surprising since their time domain equivalents are also complementary, adding to exactly reconstruct the original signal.

So we now begin to see a relationship between perfect reconstruction and a pair of complementary filters that cross at $\Omega = \pi/2$. It can be seen that the filter responses

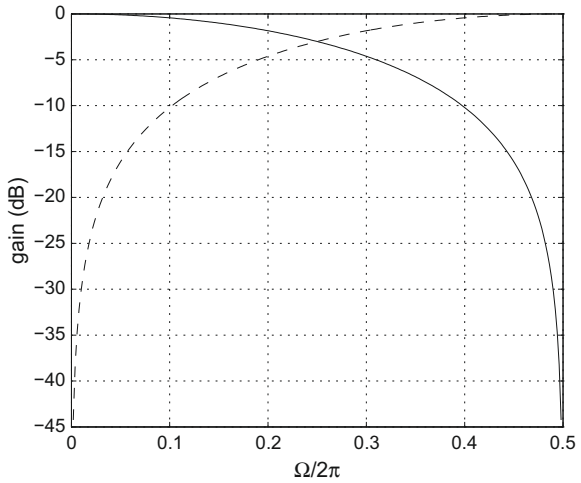


FIGURE 6.5

Complementary half-band filters from equations (6.4) (solid) and (6.5) (dotted).

are symmetrical, crossing with −3 dB attenuation at $\Omega = \pi/2$. Filters with this characteristic are often referred to as half-band filters or, more specifically, quadrature mirror filters (QMF). We will explore this relationship further when we examine filter design and performance later.

We can apply these simple filters to our example data sequence, $x[n]$, to produce the following outputs (assuming that $x[-1] = 0$):

$$l[n] = \{1, 3, 6, 8, 9, 12, 13, 9, 7, 6\}$$
$$r[n] = \{1, 1, 2, 0, 1, 2, -1, -3, 1, -2\}$$

The signals $l[n]$ and $r[n]$ together completely characterize the signal $x[n]$ and could be used as a representation for $x[n]$ that emphasizes its low-pass and high-pass (trend and difference) components. However, one obvious problem arises: $l[n]$ and $r[n]$ together contain twice as many samples as $x[n]$ alone. Doubling the size of the input signal is clearly not a great starting point for compression!

So what can we do about this? Firstly, there is obviously redundancy in the $\{l[n], r[n]\}$ combination—this must be the case as we have doubled the size without introducing any new information. We can observe this redundancy by splitting $l[n]$ and $r[n]$ into even and odd sequences thus: $\{l[2n-1], l[2n]\}$ and $\{r[2n-1], r[2n]\}$. Considering, for example, only the even sequences, we can reconstruct the original sequence $x[n]$ as follows:

$$l[2n] = \frac{x[2n] + x[2n-1]}{2}$$
$$r[2n] = \frac{x[2n] - x[2n-1]}{2}$$

Thus:

$$x[2n] = l[2n] + r[2n] \tag{6.8}$$

$$x[2n-1] = l[2n] - r[2n] \tag{6.9}$$

Hence, we can perfectly reconstruct the original sequence $x[n]$, using only half of the data produced by the filters, by combining the downsampled sequence $x[2n-1]$ with the one-sample delayed version, $x[2n]$. The system diagram for this process is given in Figure 6.6. This shows, on the left of the diagram, the analysis stage which produces the filtered outputs from equations (6.4) and (6.5) and downsamples these by a factor of 2 in order to maintain a constant overall bit rate. On the right side of the diagram is the synthesis section. This recombines the filtered and downsampled samples, through upsampling and reconstruction based on equations (6.8) and (6.9).

A refined system diagram, obtained through reconfiguration of the synthesis section to yield a more recognizable filter bank structure, is given in Figure 6.7. Let us now validate the operation of this system through an example (Example 6.1).

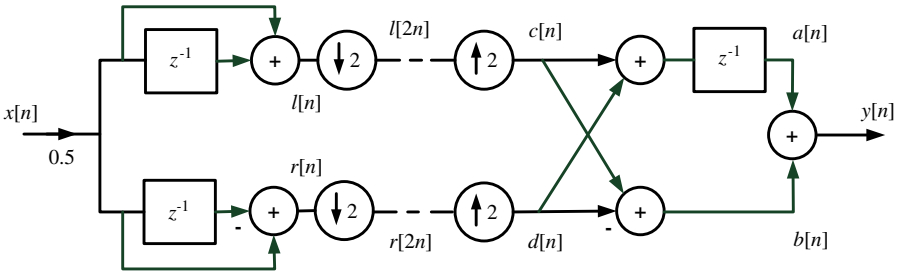


FIGURE 6.6

Basic system diagram for the two-channel filter bank.

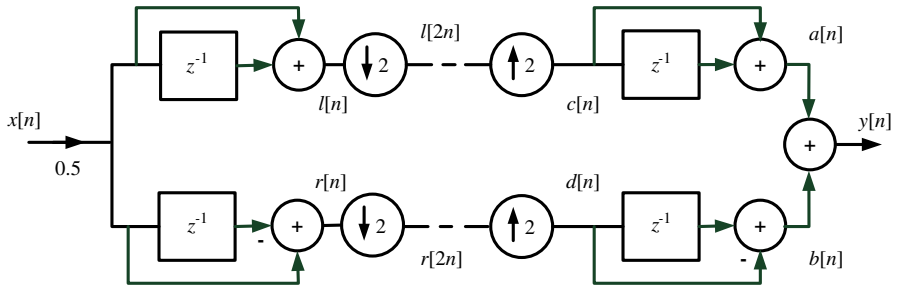


FIGURE 6.7

The two-channel filter bank.

Example 6.1 (Operation of the simple two-channel perfect reconstruction filter bank)
Verify the operation of the filter bank in Figure 6.7 using the example data sequence: $x[n] = \{2, 4, 8, 8, 10, 14, 12, 6, 8, 4\}$.

Solution. The operation of the system for this data is captured in the table below. This illustrates that the filter bank does indeed provide perfect reconstruction, with the output delayed by one sample with respect to the input.

n	0	1	2	3	4	5	6	7	8	9
$x[n]$	2	4	8	8	10	14	12	6	8	4
$l[n]$	1	3	6	8	9	12	13	9	7	6
$r[n]$	1	1	2	0	1	2	−1	−3	1	−2
$l[2n]$	1	6	9	13	7	0	0	0	0	0
$r[2n]$	1	2	1	−1	1	0	0	0	0	0
$c[n]$	1	0	6	0	9	0	13	0	7	0
$d[n]$	1	0	2	0	1	0	−1	0	1	0
$a[n]$	1	1	6	6	9	9	13	13	7	7
$b[n]$	−1	1	−2	2	−1	1	1	−1	−1	1
$y[n]$	0	2	4	8	8	10	14	12	6	8

6.3 Multirate filtering

Subband decomposition techniques are based on multirate processing. Multirate theory enables a signal to be represented in the form of multiple band-pass components whilst maintaining a constant overall bit rate. Consider, for example, the two-band split. Since the convolution of a signal with a filter impulse response does not affect bit rate,[3] for an input bit rate of k bps, the overall output bit rate without sub-sampling will be 2 kbps. Multirate theory however enables us to sub-sample the filter outputs in order to maintain a constant bit rate while preserving overall signal content (this is known as critical decimation or critical sampling).

The two primary operations used in multirate systems are interpolation and decimation. Decimation is normally employed in the encoder (or analysis stage) to reduce the sampling rate by a factor of M at each node relative to the input rate, whereas interpolation is normally used in the decoder (or synthesis stage) to increase it during reconstruction. These operations facilitate the removal of redundant information and hence help to manage the processing requirements.

[3] Assuming an infinite length sequence.

6.3.1 Upsampling

Increasing the sample rate of a signal is often referred to as interpolation and is shown diagrammatically in Figure 6.8. The operation of the upsampler is characterized according to its upsampling ratio, L, and is defined by equation (6.10):

$$x_u[n] = \begin{cases} x\left[\frac{n}{L}\right] & n = mL \\ 0 & \text{otherwise} \end{cases} \tag{6.10}$$

So, for example with $L = 2$, we insert $L - 1 = 1$ zeros between each of the original samples. Thus if $x[n] = \{1, 2, 3, 4, 5, 6\}$ then $x_u[n] = \{1, 0, 2, 0, 3, 0, 4, 0, 5, 0, 6\}$. The z transform of the upsampled signal is then given by:

$$X_u(z) = \sum_{n=-\infty}^{\infty} x_u[n] z^{-n} = \sum_{n=-\infty}^{\infty} x\left[\frac{n}{L}\right] z^{-n}; \quad n/\text{L integer} \tag{6.11}$$

Substituting $m = \frac{n}{L}$ in equation (6.11):

$$= \sum_{m=-\infty}^{\infty} x_u[m] z^{-mL} = X\left(z^L\right) \tag{6.12}$$

Let us consider what the implications of this are in the frequency domain. In the case of $L = 2$, the mapping $X\left(z^2\right)$ provides a factor of 2 "compression" of the spectrum in the frequency domain. This can be verified by evaluating equation (6.12) on the unit circle in the z plane, i.e.:

$$X\left(z^2\right)\Big|_{z=e^{j\Omega}} = X\left(e^{j2\Omega}\right)$$

and plotting the result, as illustrated in Figure 6.9. This figure shows that an additional spectral image is introduced at $\Omega = \pi$ as a result of upsampling. This must be removed using a subsequent interpolation filter, as shown in Figure 6.10. An example signal showing the spectral effect of upsampling is shown in Figure 6.11.

6.3.2 Downsampling

Decreasing the sample rate of a signal is often referred to as decimation (although strictly this should refer only to downsampling by a factor of 10). The downsampling operation is shown diagrammatically in Figure 6.12. The operation of the

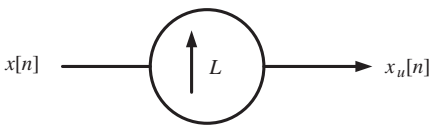


FIGURE 6.8

Upsampling by a factor L.

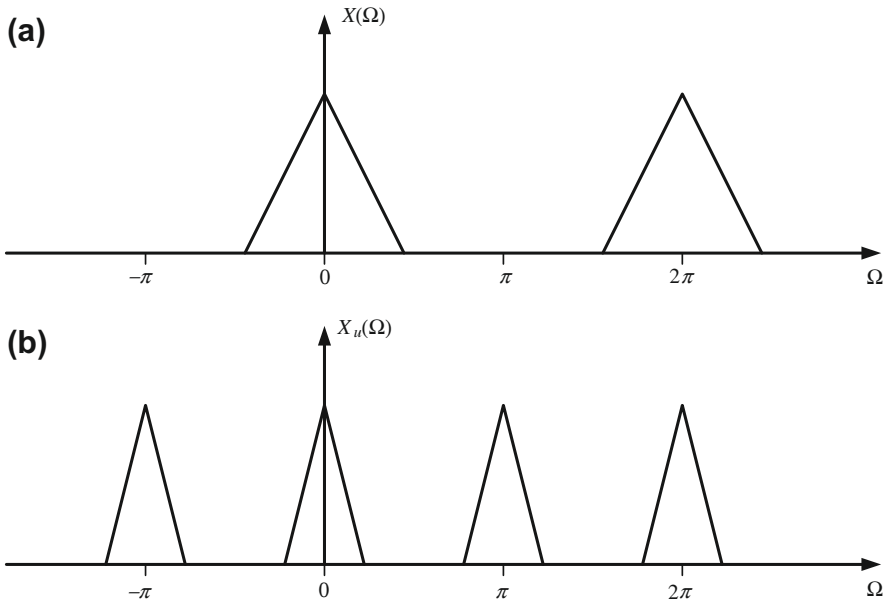


FIGURE 6.9

Spectral effect of upsampling: (a) input; (b) output.

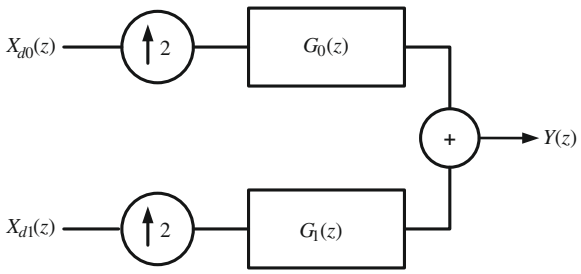


FIGURE 6.10

Two-channel synthesis filter bank.

downsampler is characterized according to its downsampling ratio, M, and is defined by equation (6.13):

$$x_d[n] = x[nM] \tag{6.13}$$

So, for example with $M = 2$, we delete every Mth sample in the original sequence. Thus if $x[n] = \{1, 2, 3, 4, 5, 6\}$ then $x_d[n] = \{1, 3, 5\}$. The z transform of the downsampled signal is then given by:

$$X_d(z) = \sum_{n=-\infty}^{\infty} x_d[n]z^{-n} = \sum_{n=-\infty}^{\infty} x[nM]z^{-n} \tag{6.14}$$

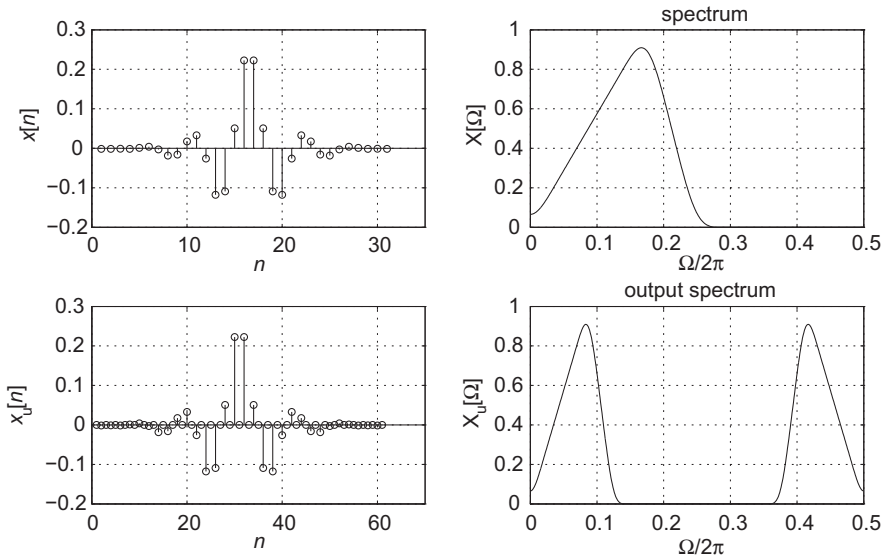


FIGURE 6.11

Spectral effects of upsampling where $L = 2$.

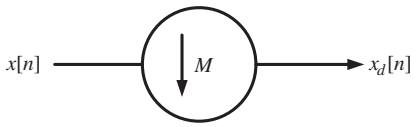


FIGURE 6.12

Downsampling by a factor of M.

Substituting $m = nM$ in equation (6.14):

$$X_d(z) = \sum_{m=-\infty}^{\infty} x'[m]z^{-m/M} = X'\left(z^{1/M}\right) \tag{6.15}$$

Note that $x'[n]$ is not the same as $x[n]$. They can however be related by $x'[n] = c[n]x[n]$, where:

$$c[n] = \begin{cases} 1 & n = 0, \pm M, \pm 2M \ldots \\ 0 & \text{otherwise} \end{cases}$$

where $c[n]$ can be represented, using the DFT, as:

$$c[n] = \frac{1}{M} \sum_{k=0}^{M-1} W_M^{nk} \tag{6.16}$$

So, for example, in the case of $M = 2$, we have:

$$c[n] = \frac{1}{2}\left(1 + (-1)^n\right)$$

Now the z-transform of $x'[n]$ is given by:

$$X'(z) = \frac{1}{M}\sum_{n=-\infty}^{\infty}\left(\sum_{k=0}^{M-1} W_M^{nk}\right)x[n]z^{-n} = \frac{1}{M}\sum_{k=0}^{M-1}\left(\sum_{n=-\infty}^{\infty} x[n]W_M^{nk}z^{-n}\right) \quad (6.17)$$

$$= \frac{1}{M}\sum_{k=0}^{M-1} X\left(zW_M^{-k}\right) \quad (6.18)$$

Equation (6.18) can now be substituted into equation (6.15) to give:

$$X_d(z) = \frac{1}{M}\sum_{k=0}^{M-1} X\left(z^{1/M}W_M^{-k}\right) \quad (6.19)$$

Consider the case where $M = 2$:

$$X_d(z) = \frac{1}{2}\left(X\left(z^{1/2}\right) + X\left(-z^{1/2}\right)\right) \quad (6.20)$$

Evaluating this on the unit circle in the z-plane gives the frequency response:

$$X_d\left(e^{j\Omega}\right) = \frac{1}{2}\left(X\left(e^{j\Omega/2}\right) + X\left(-e^{j\Omega/2}\right)\right) \quad (6.21)$$

or equivalently:

$$X_d\left(e^{j\Omega}\right) = \frac{1}{2}\left(X\left(e^{j\Omega/2}\right) + X\left(e^{j(\Omega/2+\pi)}\right)\right) \quad (6.22)$$

A plot of equation (6.22) is shown in Figure 6.13. This shows that the original spectrum is expanded by a factor of 2 in the frequency domain, but also that an additional spectral image is introduced at $\Omega = 2\pi$ as a result of the second term in equation (6.22). It can be seen that, in order to avoid aliasing which would occur if the original spectrum extended beyond $\Omega = \pi/2$, the input signal must be anti-alias filtered prior to downsampling. This is achieved using the analysis filter in Figure 6.14. An example of downsampling with an actual signal is shown in Figure 6.15.

6.3.3 System transfer function

We are now in a position to derive a transfer function for the two-channel filter bank that was introduced in Figure 6.4. This combines the analysis stage of Figure 6.14 and the synthesis stage of Figure 6.10. Considering the top path in the analysis stage only at first, we have:

$$X_0(z) = X(z)H_0(z) \quad (6.23)$$

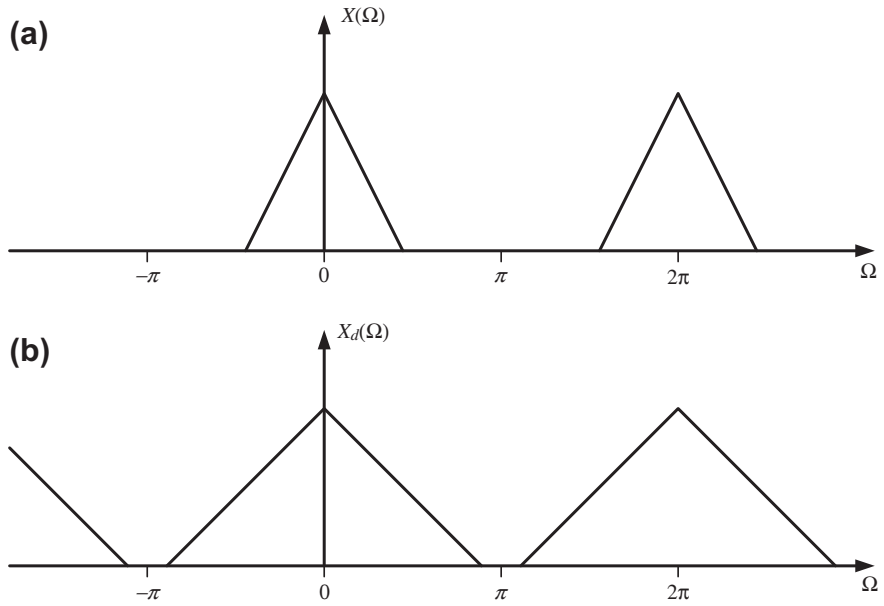


FIGURE 6.13

Spectral effects of downsampling: (a) input; (b) output.

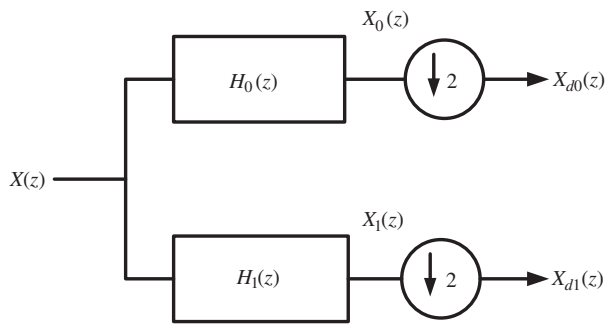


FIGURE 6.14

Two-channel analysis filter bank.

and

$$X_{d0}(z) = \frac{1}{2}\left(X_0\left(z^{1/2}\right) + X_0\left(-z^{1/2}\right)\right) \tag{6.24}$$

Substituting equation (6.23) in equation (6.24) gives:

$$X_{d0}(z) = \frac{1}{2}\left(X\left(z^{1/2}\right)H_0\left(z^{1/2}\right) + X\left(-z^{1/2}\right)H_0\left(-z^{1/2}\right)\right) \tag{6.25}$$

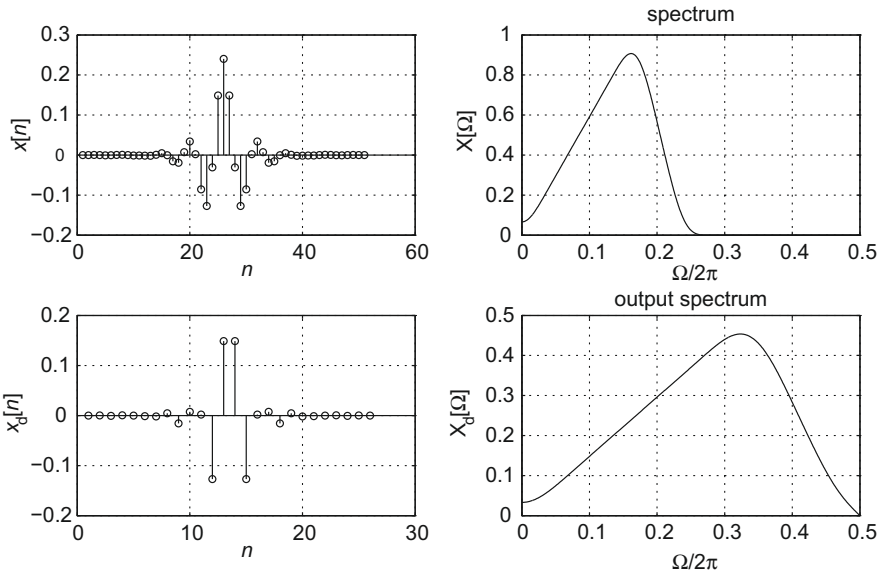


FIGURE 6.15

Spectral effects of downsampling with $M = 2$.

Now considering the synthesis stage:

$$X_{ud0}(z) = X_{d0}\left(z^2\right)$$

$$X_{ud0}(z) = \frac{1}{2}\left(X(z)H_0(z) + X(-z)H_0(-z)\right)$$

Defining the filtered output on the upper path as $Y_0(z)$:

$$Y_0(z) = X_{ud0}(z)G_0(z) \tag{6.26}$$

$$Y_0(z) = \frac{1}{2}\left(X(z)H_0(z)G_0(z) + X(-z)H_0(-z)G_0(z)\right) \tag{6.27}$$

Similarly for the filtered output on the lower path, we have:

$$Y_1(z) = \frac{1}{2}\left(X(z)H_1(z)G_1(z) + X(-z)H_1(-z)G_1(z)\right) \tag{6.28}$$

Finally, combining equations (6.27) and (6.28) at the output of the filter bank gives:

$$Y(z) = \frac{1}{2}X(z)\left[H_0(z)G_0(z) + H_1(z)G_1(z)\right] + \frac{1}{2}X(-z)\left[H_0(-z)G_0(z) + H_1(-z)G_1(z)\right] \tag{6.29}$$

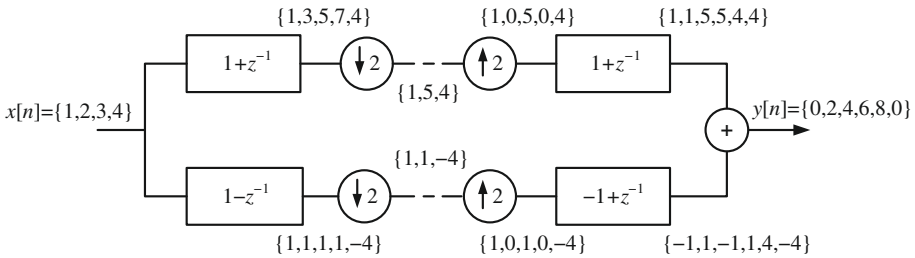
So we now have the ability to analyze the end-to-end operation of the two-channel filter bank. Let us consider this further in Example 6.2.

Example 6.2 (A simple filter bank transfer function)
Consider the two-band split in Figure 6.4, where the filters are given as follows:

$$H_0(z) = 1 + z^{-1} \quad H_1(z) = 1 - z^{-1}$$
$$G_0(z) = 1 + z^{-1} \quad G_1(z) = -1 + z^{-1}$$

These filters are identical to those defined in our earlier example, except that the multiplicative factor of 0.5 in the analysis stage has been omitted for numerical convenience. Using equation (6.29), compute the output of the filter bank for an input $x[n] = \{1, 2, 3, 4\}$.

Solution. The effects of analysis filtering, downsampling, upsampling, and synthesis filtering can be seen in the diagram below, yielding an output that is the same as the input, but scaled by a factor of 2 and delayed by one sample.



Rather than compute the filter bank response in the time domain, as shown above, we can instead verify its operation in the z-domain using equation (6.29). Thus we have:

$$\begin{aligned} Y(z) &= \frac{1}{2}\left(1 + 2z^{-1} + 3z^{-2} + 4z^{-3}\right)\left[\left(1 + 2z^{-1} + z^{-2}\right) + \left(-1 + 2z^{-1} - z^{-2}\right)\right] \\ &\quad + \cdots + \frac{1}{2}\left(1 - 2z^{-1} + 3z^{-2} - 4z^{-3}\right)\left[\left(1 - z^{-2}\right) + \left(-1 + z^{-2}\right)\right] \\ &= \frac{1}{2}\left(1 + 2z^{-1} + 3z^{-2} + 4z^{-3}\right)\left[\left(4z^{-1}\right)\right] + \frac{1}{2}\left(1 - 2z^{-1} + 3z^{-2} - 4z^{-3}\right)[(0)] \\ &= \left(2z^{-1} + 4z^{-2} + 6z^{-3} + 8z^{-4}\right) \end{aligned}$$

Note that the leading and trailing zeros shown in the filter bank output are redundant and due to the expansive nature of the convolution operation.

6.3.4 Perfect reconstruction

Equation (6.29) can be rewritten as:

$$Y(z) = F_0(z)X(z) + F_1(z)X(-z) \tag{6.30}$$

where

$$F_0(z) = \frac{1}{2}\left[H_0(z)G_0(z) + H_1(z)G_1(z)\right]$$

and

$$F_1(z) = \frac{1}{2}\left[H_0(-z)G_0(z) + H_1(-z)G_1(z)\right]$$

$F_1(z)$ represents the alias components produced by the overlapping filter frequency responses. If $F_1(z) = 0$ then the filter bank is alias-free. $F_0(z)$ represents the quality

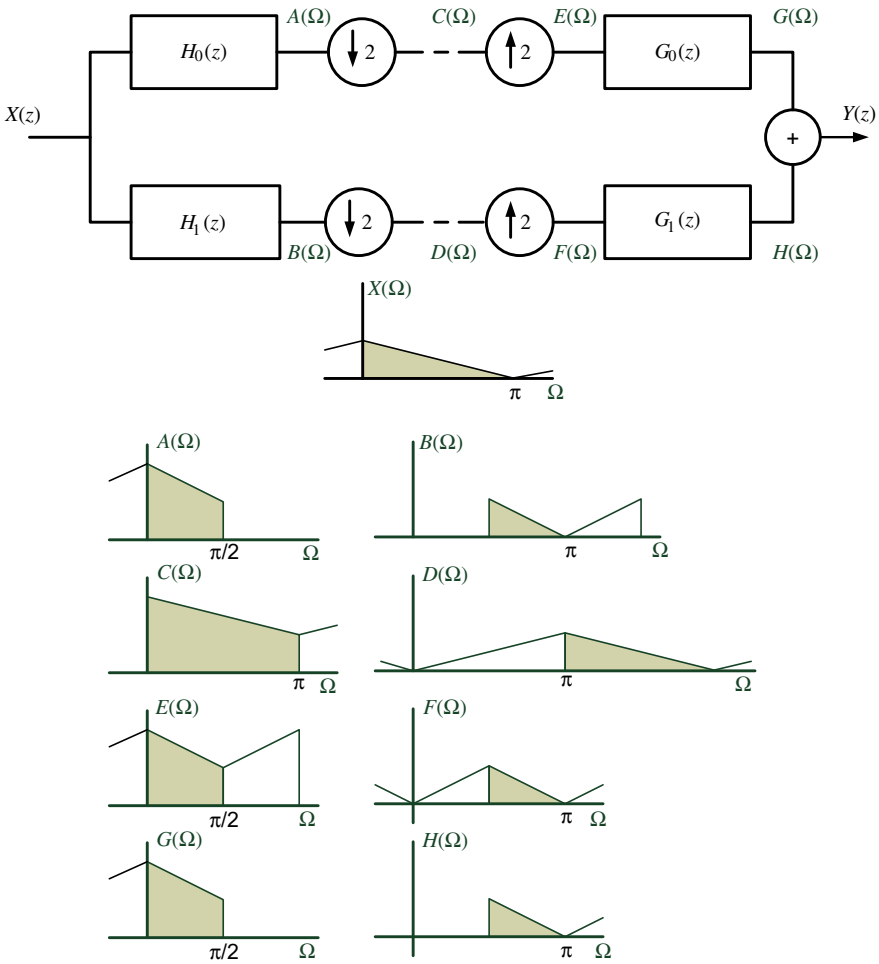


FIGURE 6.16

Spectral relationships within the two-channel perfect reconstruction filter bank. The spectral characteristics at signal points *A*–*H* are shown.

of the reconstruction. If $\left|F_0\left(e^{j\Omega}\right)\right| = 1$ for all frequencies, then the filter bank is free of amplitude distortion. If $F_0\left(e^{j\Omega}\right)$ is linear phase then the filter bank is free of phase distortion. In general we say that, if $F_1(z) = 0$ and $F_0(z) = cz^{-k}$ (where c and k are constants), then the filter bank provides perfect reconstruction (PR).

6.3.5 Spectral effects of the two-channel decomposition

A perhaps more intuitive visualization of the operation of the two-channel PR filter bank is provided by Figure 6.16. This shows the spectral characteristics of a signal, $x[n]$, which is bandlimited to the Nyquist frequency, as it passes through the filter bank. For simplicity in the diagram, we assume that the filters have brickwall responses with no leakage between bands. However, as we have already seen, with appropriate filter design, this is not a requirement for perfect reconstruction.

Working across the system from left to right we can see firstly that $H_0(z)$ and $H_1(z)$ select the low-pass and high-pass bands respectively. In order to preserve critical sampling, these signals are then downsampled by a factor of 2. The results of this operation, $C(\Omega)$ and $D(\Omega)$, present spectra that are expanded by a factor of 2 and that also include aliases as expected. The signals would then typically be quantized for transmission, but we ignore this step at the moment and move straight to the decoder.

At the decoder, the signals are upsampled in preparation for recombination. The effect of this is to "compress" the spectra by a factor of 2 along the frequency axis as shown by $E(\Omega)$ and $F(\Omega)$. Finally these signals are passed through the interpolation filters $G_0(z)$ and $G_1(z)$, which remove the aliases by low-pass and high-pass filtering respectively, to yield $G(\Omega)$ and $H(\Omega)$. As can be observed, $G(\Omega)$ and $H(\Omega)$ sum at the output to provide a perfect reconstruction of the input signal.

6.4 Useful filters and filter banks

For compression applications it is desirable for the filters to exhibit certain characteristics. These are:

1. **Perfect reconstruction:** To enable distortion-free reconstruction of the original signal at the output of the filter bank, in the absence of quantization.
2. **Compact support:** To ensure that distortions due to quantization and transmission loss are localized.
3. **Smoothness:** To better match the characteristics of natural images.
4. **Regularity:** To ensure convergence and maintain smoothness when iterated.
5. **Orthogonality and orthonormality:** To enable matching of analysis and synthesis sections and to remove transform redundancy.[4]

Some commonly used filters are examined below.

[4] *Note:* The imposition of orthogonality conditions places severe constraints on the filters (or basis functions) used in the filter bank. An alternative condition, known as *biorthogonality*, can however be satisfied more generally, where the analysis filters are independent but do not form an orthogonal pair. This means that the inverse or synthesis transform is no longer given by the transpose of the analysis transform. In the case of biorthogonality, a dual basis can be found where projection on to an orthogonal set is possible.

6.4.1 Quadrature mirror filters

Quadrature mirror filters (QMF) are an important class of alias-cancelation filters that were used in early subband processing applications [2].

Aliasing elimination

QMF filters start by eliminating the aliasing term in equation (6.30) by using the following relationships:

$$G_0(z) = H_1(-z); \quad G_1(z) = -H_0(-z)$$

QMF filters are based on a pair of analysis filters that are symmetrical about $\Omega = \pi/2$. This can be achieved if $H_0(z) = H(z)$ and $H_1(z) = H(-z)$. We can therefore base our QMF filter design on the following relationships:

$$\begin{aligned} H_0(z) &= H(z); \quad H_1(z) = H(-z) \\ G_0(z) &= H(z); \quad G_1(z) = -H(-z) \end{aligned}$$

If we substitute these relationships into equation (6.29), we can see that the aliasing term is zero, thus:

$$F_1(z) = \frac{1}{2}\left[H(-z)H(z) - H(z)H(-z)\right] = 0 \tag{6.31}$$

Amplitude distortion

The distortion term for the QMF filter bank is given by:

$$F_0(z) = \frac{1}{2}\left[H(z)H(z) - H(-z)H(-z)\right] \tag{6.32}$$

Furthermore, if the prototype filter is linear phase: $H(z) = A(z)z^{-(N-1)/2}$, where $A\left(e^{j\Omega}\right)$ is a real zero-phase frequency response, then:

$$F_0(z) = \left[A^2(z) + A^2(-z)\right] z^{-(N-1)/2}$$

The QMF filter bank will therefore exhibit amplitude distortion unless the magnitude of $F_0\left(e^{j\Omega}\right)$ is constant for all Ω. As we will see in Example 6.3, QMF filters can be designed that exhibit perfect reconstruction.

Example 6.3 (Simple QMF filters)
Consider the following example prototype filter:

$$H(z) = \frac{1}{\sqrt{2}}\left(1 + z^{-1}\right)$$

Generate the corresponding set of QMF filters and show that this filter bank exhibits perfect reconstruction.

Solution. This prototype yields the following set of filters, which hopefully the reader will recognize as the half-band filters from our previous example:

$$H_0(z) = \frac{1}{\sqrt{2}}\left(1 + z^{-1}\right); \quad H_1(z) = \frac{1}{\sqrt{2}}\left(1 - z^{-1}\right)$$
$$G_0(z) = \frac{1}{\sqrt{2}}\left(1 + z^{-1}\right); \quad G_1(z) = \frac{1}{\sqrt{2}}\left(-1 + z^{-1}\right)$$

We can see from equation (6.31) that this filter bank is alias-free. We can further show, from equation (6.32) that the system provides a constant delay with unity magnitude since:

$$F_0(z) = \frac{1}{4}\left[\left(1 + z^{-1}\right)\left(1 + z^{-1}\right) - \left(1 - z^{-1}\right)\left(1 - z^{-1}\right)\right] = z^{-1}$$

In fact these are the only causal FIR QMF analysis filters yielding exact perfect reconstruction.

Practical QMF filters

In practice, approximate QMF filters can be usefully designed where the characteristic does not exactly provide perfect reconstruction. In such cases, the amplitude distortion can be minimized using an optimization procedure to iteratively adjust the filter coefficients. We ideally want the magnitude response of the filter characteristic to have a sharp transition so that subbands are spectrally distinct while providing an overall system response that is approximately all-pass. This means that the filters have to be designed so that each branch delivers an attenuation of 6 dB (0.5) at the point where the low- and high-pass responses intersect—at one quarter of the sampling frequency. Each of the four filters must therefore have an attenuation of 3 dB at the half Nyquist frequency. Such filters have been designed by Johnson [22] and others. Johnson's method minimizes the residual amplitude distortion using an objective measure, based upon stop-band attenuation and the requirement to maintain the power symmetry property of the filters. Low amplitude distortions, typically less than 0.02 dB, result. Example Johnson filters are:

$$h_0[0] = -0.006443977 = h_0[11]; \quad h_0[1] = 0.02745539 = h_0[10];$$
$$h_0[2] = -0.00758164 = h_0[9]; \quad h_0[3] = -0.0013825 = h_0[8];$$
$$h_0[4] = 0.09808522 = h_0[7]; \quad h_0[5] = 0.4807962 = h_0[6]$$

The mirror responses of these filters are shown in Figure 6.17. It can be seen that the overall distortion characteristic, although small, is not zero. Hence these filters do not provide perfect reconstruction. In practice, for compression applications where significant quantization error is introduced prior to transmission of the subband coefficients, this small distortion may not be a major issue. However, we will see shortly how useful filters can be designed that do offer perfect reconstruction.

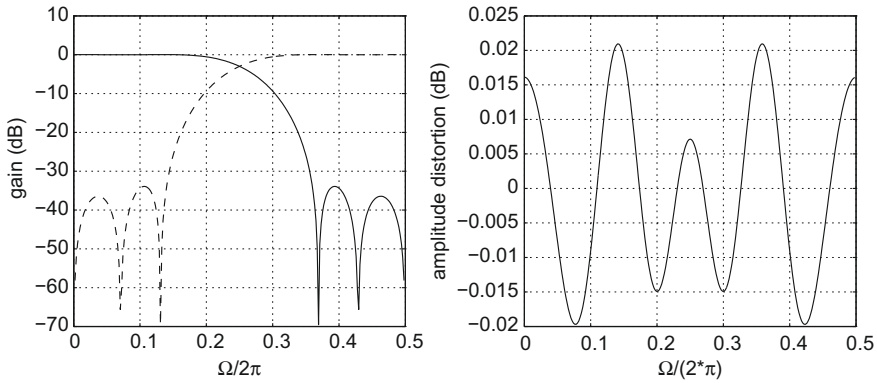


FIGURE 6.17

QMF filter pair frequency response.

6.4.2 Wavelet filters

The wavelet filter bank can form a perfect reconstruction system that is completely defined by the low-pass filters. The high-pass filters are obtained by shifting and modulating the low-pass filters, thus:

$$H_1(z) = z^{-l}G_0(-z) \text{ and } G_1(z) = z^l H_0(-z) \tag{6.33}$$

and provided that l is odd, then:

$$F_1(z) = \frac{1}{2}\left[H_0(-z)G_0(z) + (-z)^{-l}G_0(z)z^l H_0(-z)\right] = 0$$

i.e. the system is alias-free.

We can now address the distortion term $F_0(z)$. If $F_0(z) = \frac{1}{2}\left[H_0(z)G_0(z) + H_1(z)G_1(z)\right] = cz^{-k}$, then the filter bank will exhibit perfect reconstruction. If we let $P(z) = H_0(z)G_0(z)$, and we assume zero-phase response filters, then we can rewrite $F_0(z)$ in the form (absorbing the $\frac{1}{2}$ term into the constant, c):

$$P(z) + P(-z) = c \tag{6.34}$$

Since all terms in odd powers of z will cancel, for equation (6.34) to hold true, with the exception of the z^0 term, all the terms in even powers of z must sum to zero.

To design filters we can now factorize the polynomial $P(z)$ into the two low-pass filters, $H_0(z)$ and $G_0(z)$, resulting in a completely defined filter bank. The question is, how do we factorize it, and how do we ensure that we have good filter characteristics? We have already discussed the desirable property of regularity and the associated requirement to have multiple roots at $z = -1$. This gives us a good starting point. So if we have:

$$P(z) = \left(1 + 2z^{-1} + z^{-2}\right)$$

then this can be factorized into our familiar simple PR filter bank where both low-pass filters are identical, thus:

$$P(z) = \left(1 + z^{-1}\right)\left(1 + z^{-1}\right)$$

There are several filter combinations which (in the absence of quantization) offer perfect reconstruction. A performance comparison of these is provided by Villasenor [23] and a good overview of design methods is provided by Kingsbury [24]. We will consider two example sets of filters here—both of which have been selected for use in JPEG2000. These are the LeGall 5/3 wavelet and the Daubechies 9/7 wavelet. They are both compact, have symmetric responses (this helps to avoid boundary artifacts) and are highly regular (they have a maximum number of vanishing moments).

LeGall 5/3 filters

One example of a simple, yet effective, set of wavelet filters is the biorthogonal LeGall (5/3) pair [25]. These can be designed using the above approach and are defined in terms of their low-pass filter responses as follows:[5]

$$\begin{aligned} H_0(z) &= \frac{1}{8} z \left(1 + z^{-1}\right)^2 \left(-z - z^{-1} + 4\right) \\ G_0(z) &= \frac{1}{2} z \left(1 + z^{-1}\right)^2 \end{aligned} \tag{6.35}$$

The filter coefficients are thus:

$$\begin{aligned} h_0[n] &= \{-1, 2, 6, 2, -1\}/8; \quad & g_0[n] &= \{1, 2, 1\}/2; \\ h_1[n] &= \{1, -2, 1\}/2; & g_1[n] &= \{1, 2, -6, 2, 1\}/8 \end{aligned}$$

Recalling from equation (6.29) that:

$$\begin{aligned} Y(z) &= \frac{1}{2} X(z) \left[G_0(z)H_0(z) + G_1(z)H_1(z)\right] \\ &\quad + \frac{1}{2} X(-z) \left[G_0(z)H_0(-z) + G_1(z)H_1(-z)\right] \\ &= F_0(z)X(z) + F_1(z)X(-z) \end{aligned} \tag{6.36}$$

for perfect reconstruction we require $F_0(z) = constant$ and $F_1(z) = 0$. So for the distortion term we have:

$$\begin{aligned} G_0(z)H_0(z) &= \left(-1 + 0z^{-1} + 9z^{-2} + 16z^{-3} + 9z^{-4} + 0z^{-5} - z^{-6}\right)\Big/16 \\ G_1(z)H_1(z) &= \left(+1 + 0z^{-1} - 9z^{-2} + 16z^{-3} - 9z^{-4} + 0z^{-5} + z^{-6}\right)\Big/16 \end{aligned}$$

Thus the distortion is constant for all frequencies:

$$F_0(z) = z^{-3}$$

[5] *Note:* The LeGall filters are sometimes defined where both are pre-multiplied by a factor 1/4 rather than 1/8 and 1/2.

In the case of the alias term we have:

$$G_0(z)H_0(-z) = \left(-1 - 4z + z^{-2} + 8z^{-3} + z^{-4} - 4z^{-5} - z^{-6}\right)\Big/16$$
$$G_1(z)H_1(-z) = \left(1 + 4z - z^{-2} - 8z^{-3} - z^{-4} + 4z^{-5} + z^{-6}\right)\Big/16$$

Thus the alias term is given by:

$$F_1(z) = 0$$

Hence the filter bank is also alias free and satisfies the conditions required for perfect reconstruction. The frequency responses of the four LeGall (5/3) biorthogonal filters are shown in Figure 6.18.

The LeGall wavelet has rational coefficients and is recommended for reversible or lossless compression in JPEG2000.

Daubechies 9/7 filters

The Daubechies 9/7 filters [26] are recommended for lossy compression in JPEG2000. They are defined as:

$$H_0(z) = \frac{2^{-5}}{\frac{64}{5\rho} - 6 + \rho} z^2 \left(1 + z^{-1}\right)^4 \left(z^2 + z^{-2} - \left(8 - \rho\right)\left(z + z^{-1}\right) + \frac{128}{5\rho} + 2\right)$$
$$G_0(z) = \frac{2^{-3}}{\rho - 2} z^2 \left(1 + z^{-1}\right)^4 \left(-z - z^{-1} + \rho\right) \tag{6.37}$$

where $\rho \approx 3.3695$. In comparisons of short length filters by Villasenor [23], the Daubechies 9/7 filters were ranked as the best performing for lossy image compression.

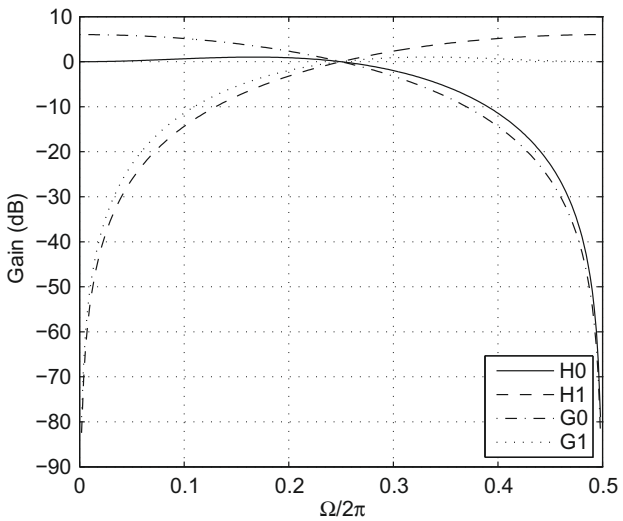


FIGURE 6.18

LeGall filter frequency responses.

6.4.3 Multistage (multiscale) decompositions

One important class of subband decomposition, commonly (although, as described previously, not always accurately) referred to as a discrete wavelet transform (DWT), is achieved by cascading a series of two-band systems, recursively applied to the low band. Using this method, the high frequency bands have good spatial resolution and poorer frequency resolution, while the low frequency bands have good frequency localization but with inferior spatial localization.

A 1-D three-stage decomposition is shown in Figure 6.19 and the associated time–frequency tilings at all stages of decomposition are given in Figure 6.20. This figure

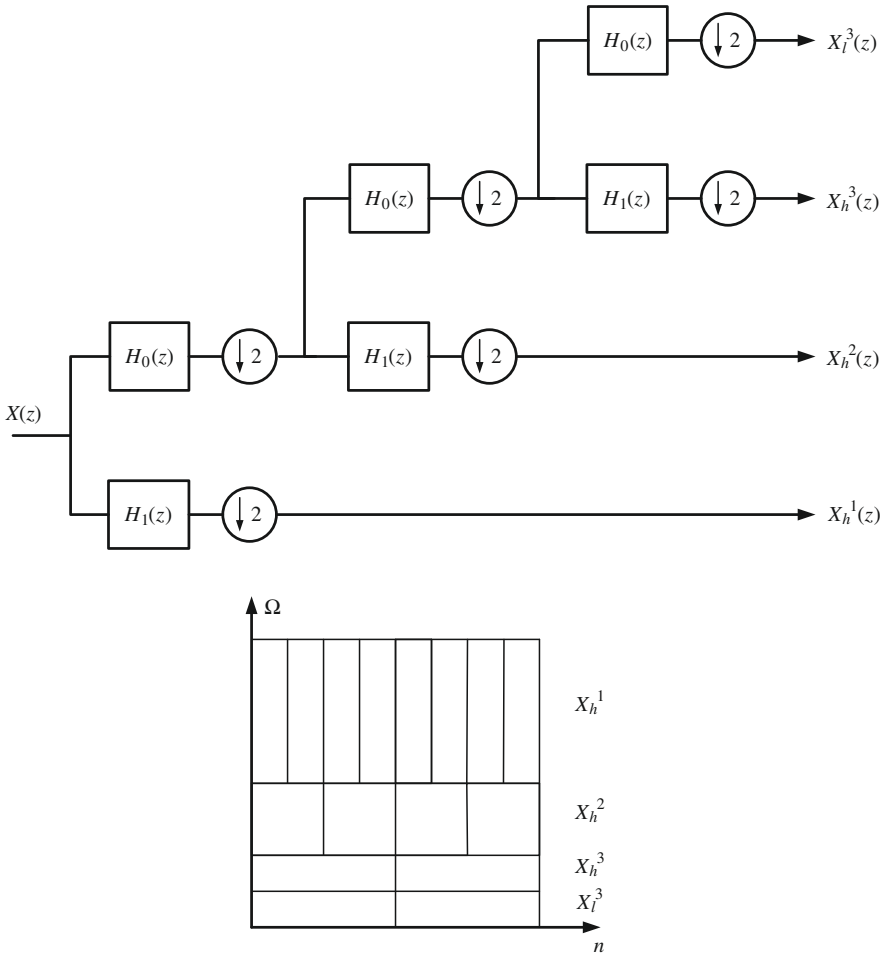


FIGURE 6.19

1-D three stage wavelet decomposition. Top: block diagram. Bottom: associated time–frequency tiling.

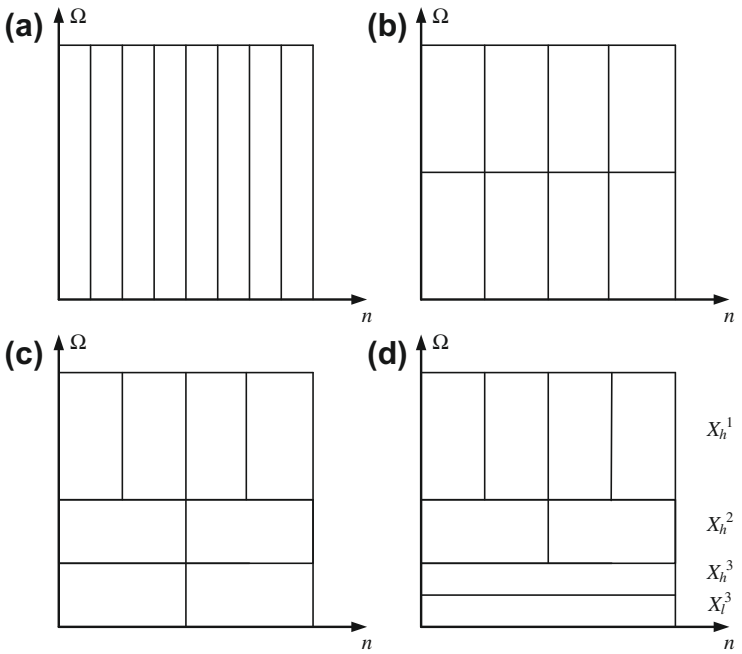


FIGURE 6.20

Time–frequency tiling for progressive stages of decomposition in a 1-D wavelet filter bank: (a) input of eight samples; (b) after the first stage; (c) after the second stage; (d) after the third stage.

shows an input of eight consecutive time domain samples, subsequently filtered and downsampled to yield four samples in each of two bands (b). The lower frequency band is then further filtered and downsampled to yield two more bands, each with two samples (c). Finally in (d), the low-pass band is once again filtered and downsampled to provide the tiling shown. This clearly shows the property of critical sampling as the number of samples (eight in this case) is identical at each decomposition stage.

In general, for an input sequence of length N, the relationship between the bandwidth and the number of samples in each subband is given in Table 6.1. This is shown for a five-stage decomposition, although the pattern for more or fewer stages should be obvious.

An alternative view of multistage decomposition

We can see from Figure 6.19 that each output from the analysis section is a series of filtering and downsampling operations. This can be restructured as shown in Figure 6.21. This is straightforward to show, based on the two alternative configurations at the bottom of Figure 6.21. For example, in the case of the downsampler followed by a filter $H(z)$, we have:

$$Y(z) = \frac{1}{2}H(z)\left(X\left(z^{\frac{1}{2}}\right) + X\left(-z^{\frac{1}{2}}\right)\right)$$

Table 6.1 Relationship between subband frequency range and number of samples for each subband in a five-stage decomposition.

Subband	Bandwidth	No. samples
X_h^1	$\pi/2 \cdots \pi$	$N/2$
X_h^2	$\pi/4 \cdots \pi/2$	$N/4$
X_h^3	$\pi/8 \cdots \pi/4$	$N/8$
X_h^4	$\pi/16 \cdots \pi/8$	$N/16$
X_h^5	$\pi/32 \cdots \pi/16$	$N/32$
X_l^5	$0 \cdots \pi/32$	$N/32$

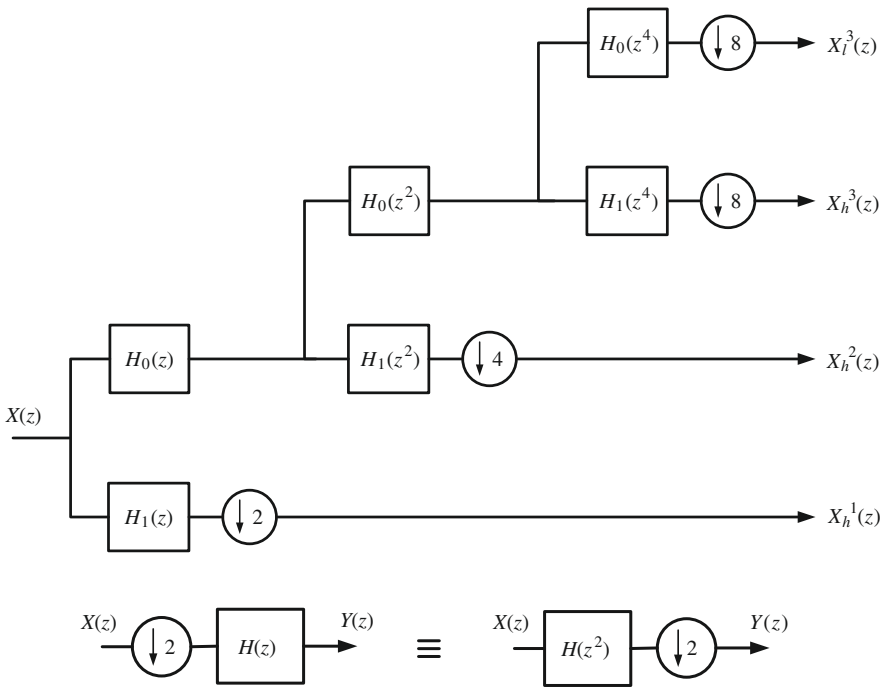


FIGURE 6.21

1-D three-stage wavelet decomposition—modified downsampling structure.

and, in the case of the modified filter followed by the downsampler, we have:

$$Y(z) = \frac{1}{2}\left(X\left(z^{\frac{1}{2}}\right)H(z) + X\left(-z^{\frac{1}{2}}\right)H(z)\right)$$

A similar relationship exists for an upsampler–filter combination.

6.4.4 Separability and extension to 2-D

A two-level subband decomposition applied in two dimensions (separably) is shown in Figure 6.22 and the corresponding tiling of the spatial frequency plane is shown in Figure 6.23. The upper left subband in Figure 6.23 contains the lowest frequency

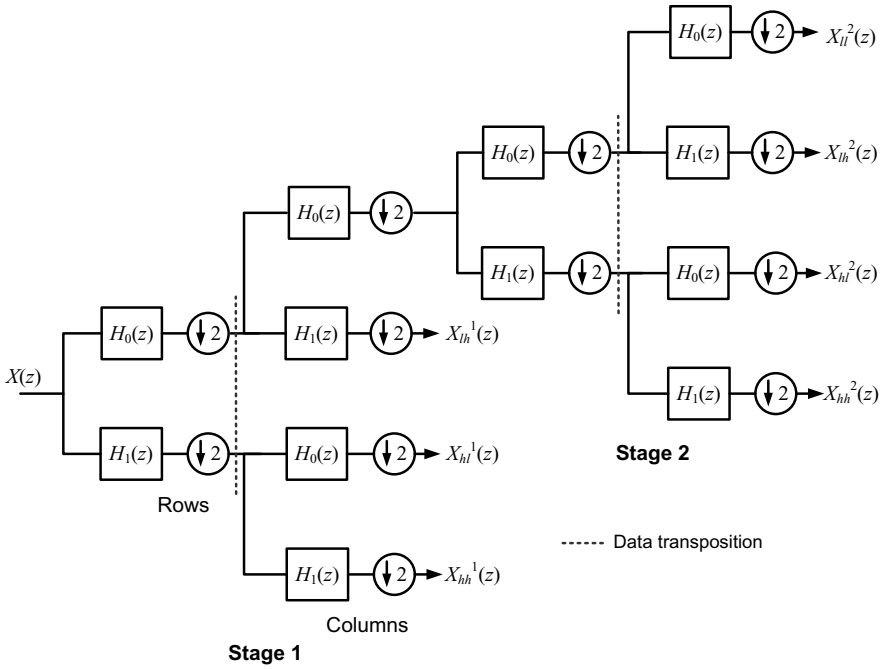


FIGURE 6.22

2D 2 channel 2 stage wavelet decomposition. Data transposition (as shown) is required between row and column processing.

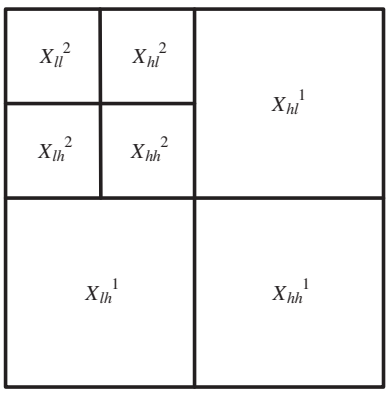


FIGURE 6.23

2-D wavelet—frequency plane tiling.

FIGURE 6.24

Frame from Akiyo sequence (left) wavelet transformed (right) using a three-stage decomposition with LeGall 5/3 filters.

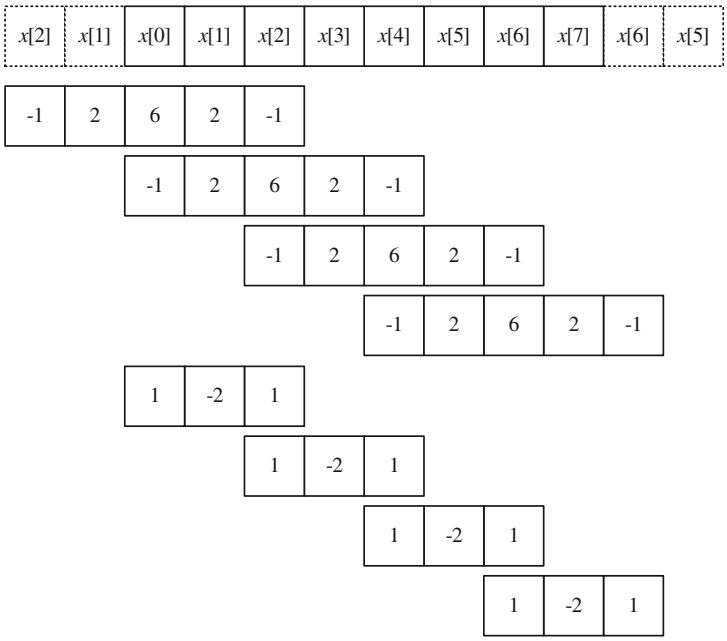


FIGURE 6.25

Illustration of boundary extension.

information and this would be recognizable as a small sub-sampled version of the original image. Intuitively therefore, this subband is important from an HVS point of view and must be coded accurately (usually using DPCM). The other subbands correspond to mid- and high-range spatial frequencies and usually contain relatively little energy. Thus the encoder can allocate fewer (or no) bits to coding this information. We will look at bit allocation strategies and entropy coding of these bands a little later.

An example showing a three-level 2-D wavelet decomposition of one frame in the Akiyo sequence is shown in Figure 6.24. This decomposition employs the LeGall 5/3 filters described earlier.

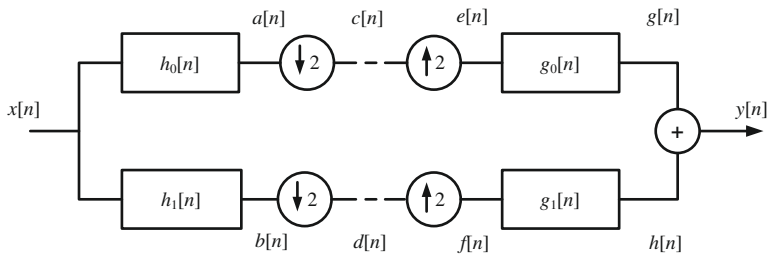
6.4.5 Finite length sequences, edge artifacts, and boundary extension

When we discussed critical sampling previously, we have assumed the context of infinitely long sequences. It is clear in the case of finite length sequences, such as those typical in image coding, that the number of subband samples will be greater than the number of input samples. If we maintain strict critical sampling, then distortion would be introduced by the filter bank, due to discontinuities at the signal boundaries, and perfect reconstruction would not be achieved. This can be reduced by extending the boundary samples either periodically or symmetrically or by zero padding. Symmetrical extension is normally preferred as this does not create discontinuities and has the effect of introducing lower boundary distortions.

We can easily maintain critical sampling with boundary extension and an example for an eight-sample input with 5/3 filters is shown in Figure 6.25. This illustrates the effective convolution shifts for each filter taking account of sub-sampling.

Example 6.4 (Boundary effects and critical sampling)
Consider a 1-D two-channel single stage filter bank based on the LeGall 5/3 filters. Compute the signal values at all intermediate nodes for the case of an input signal: $x[n] = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

Solution. We can compute the convolution of the filters with the input sequence at all intermediate nodes, as defined in the following figure:



Let us consider first the case where critical sampling is not strictly observed. The signal values at each intermediate node are given in the following table:

$x[n]$	1 2 3 4 5 6 7 8
$a[n]$	−1 0 7 16 24 32 40 48 65 56 9 −8
$b[n]$	1 0 0 0 0 0 0 0 −9 8
$c[n]$	−1 7 24 40 65 9
$d[n]$	1 0 0 0 −9

$e[n]$	$-1\ 0\ 7\ 0\ 24\ 0\ 40\ 0\ 65\ 0\ 9$
$f[n]$	$1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ -9$
$g[n]$	$-1\ -2\ 6\ 14\ 31\ 48\ 64\ 80\ 105\ 130\ 74\ 18\ 9\ 0\ 0$
$h[n]$	$1\ 2\ -6\ 2\ 1\ 0\ 0\ 0\ -9\ -18\ 54\ -18\ -9$
$y[n]$	**0 0 0 16 32 48 64 80 96 112 128 0 0**

Note that the filter gain terms have been ignored here for convenience. However, it is clear that the answer is correct when we divide the output by 16. The problem with this is that we have 11 samples at the output of the analysis stage (c and d) rather than 8. If we maintain strict critical sampling, then distortion would be introduced by the filter bank due to discontinuities at the signal boundaries and perfect reconstruction would not be achieved.

Let us now consider what happens if we keep only eight samples for processing by the synthesis section. Let us discard the first two samples of the a and b signals as these are associated with the filter transients, thus:

$c[n]$	$7\ 24\ 40\ 65$
$d[n]$	$0\ 0\ 0\ -9$
$e[n]$	$7\ 0\ 24\ 0\ 40\ 0\ 65$
$f[n]$	$0\ 0\ 0\ 0\ 0\ 0\ -9$
$g[n]$	$7\ 14\ 31\ 48\ 64\ 80\ 105\ 130\ 65\ 0\ 0$
$h[n]$	$0\ 0\ 0\ 0\ 0\ 0\ -9\ -18\ 54\ -18\ -9$
$y[n]$	**7 14 31 48 64 80 96 112 119 −18 −9**

We can now see that the filter bank output is corrupted at its boundaries. We can improve this by boundary extension using, for example, symmetric extension.

6.4.6 Wavelet compression performance

An excellent comparison between the performances of various types of wavelet filter is provided by Villasenor [23]. Unser and Blu [27] also provide a rigorous comparison between the LeGall 5/3 and the Daubechies 9/7 filters in the context of JPEG2000. Both papers show that the 9/7 filters provide excellent performance with compact support for lossy coding. Considering their size, the 5/3 filters also perform exceptionally well in an image coding context, with the added advantage that lossless reconstruction can always be guaranteed because of their use of integer coefficients.

A comparison between wavelet and DCT compression methods (based on PSNR) is provided for the case of the 256 × 256 Lena image in Figure 6.26. Although this shows clear benefits of wavelet methods over their DCT counterparts, we must be careful of reading too much into this graph because of the significant differences in the coding methods used and in the artifacts produced by them. In subjective testing, however, it has been widely reported that wavelet coding (for example in JPEG2000) significantly outperforms DCT coding (as in JPEG). Subjective performance comparisons are provided in Section 6.6 (see Figure 6.27).

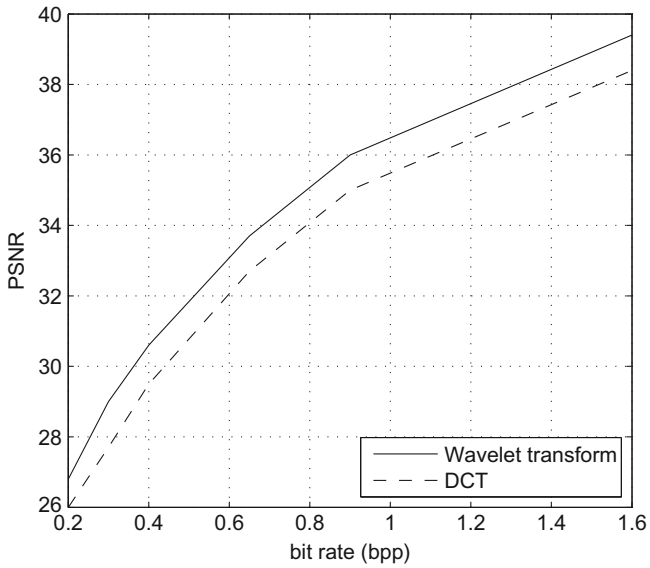


FIGURE 6.26

Compression performance, DCT vs wavelet, for the 256 × 256 luma Lena image.

FIGURE 6.27

Comparisons of wavelet-coded images, 256 × 256 Lena. Left: LeGall 5/3 wavelets at 0.3 bpp. Right: LeGall 5/3 wavelets at 0.5 bpp.

6.5 Coefficient quantization and bit allocation

6.5.1 Bit allocation and zonal coding

In order to quantize subband coefficients we can, in a similar manner to our approach with the DCT, apply different quantizer weights to different subbands according to their perceptual importance. This is shown in Figure 6.28, where different weights, W_0 to W_6, are used to differentiate both subband orientation and scale.

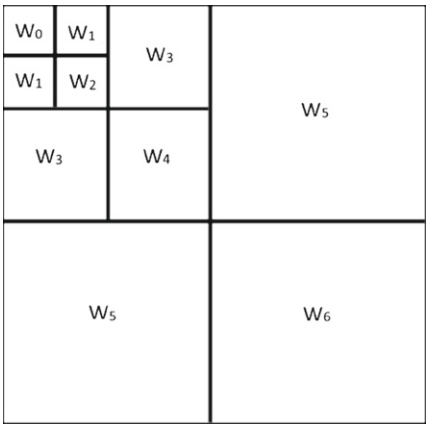


FIGURE 6.28

Subband weighting.

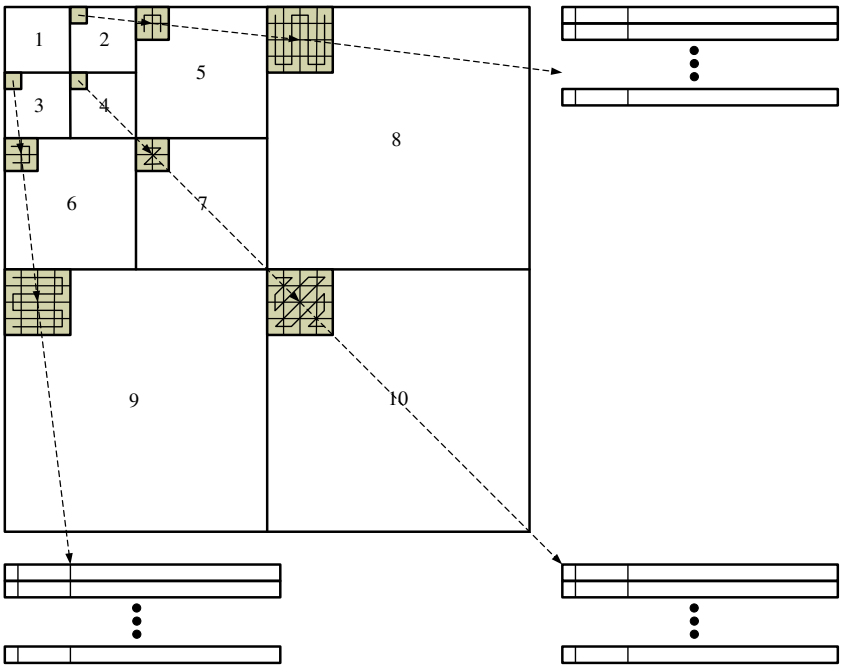


FIGURE 6.29

Scanning of wavelet subbands.

Similar to the way that we applied zig-zag scanning to a DCT block in Chapter 5, we can also scan subband coefficients from high to low subbands and in a manner that is matched to the orientation of energy in the subband. This is shown in Figure 6.29

where, for each of the three orientations, a data structure is used that supports efficient quantization. For example, subbands 2, 5, and 8 are likely to be highly correlated since subband 2 is a coarse approximation of subband 5 and subband 5 is a coarse approximation of subband 8. Similarly for subbands 4, 7, and 10 and subbands 3, 6, and 9. We can observe that the scanning patterns have been devised to exploit these relationships.

In the case of subband 1 (DC), coefficients will typically be encoded differentially and then {size, value} symbols entropy coded. For the AC coefficients, after scanning, a {run length/size, value} format, similar to that used with JPEG, would normally be employed to yield symbols for entropy coding.

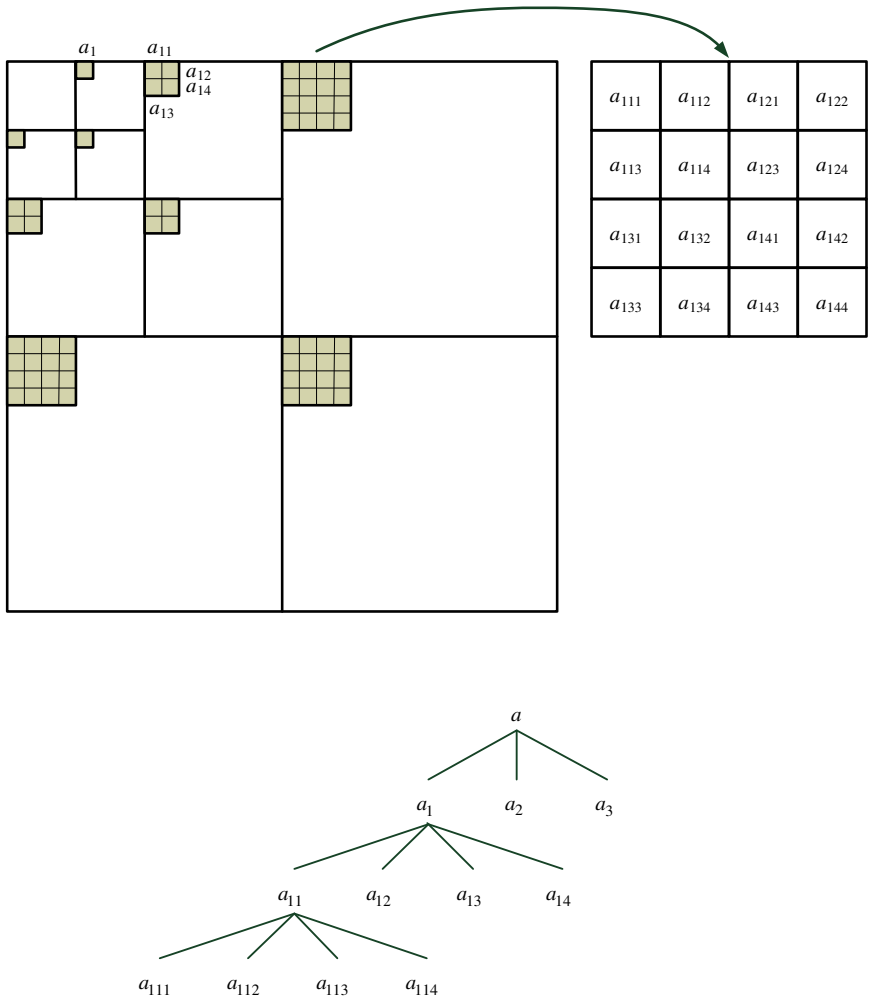


FIGURE 6.30

Tree structures for embedded wavelet coding.

6.5.2 Hierarchical coding

Alternative scanning methods have been proposed in the context of wavelet compression, that provide the basis for scalable encoding using embedded bitstreams. In particular the reader is referred to the work of Shapiro [28], who introduced the concept of scalable coding using embedded zero-tree coding (EZW), and that of Said and Pearlman [16], who extended this idea into an approach called set partitioning in hierarchical trees (SPIHT). We will not consider these methods in detail, but provide a brief introduction below.

Embedded coding methods generally follow the approach illustrated in Figure 6.30, where multiscale dependencies (self-similarities) are exploited across the wavelet subbands when coding the coefficients. The tree structure in the lower subfigure captures these dependencies. In EZW, Shapiro showed how these dependencies could be exploited using a succession of coding passes through the tree, in each pass coding significant coefficients that exceed a threshold. As the threshold decreases, more bits are included and a natural bit-plane data structure results. In EZW, arithmetic coding is essential in order to compress the data resulting from the significance passes. Said and Pearlman extended EZW with SPIHT which provides more efficient subset partitioning and which can perform exceptionally well, even without arithmetic coding.

6.6 JPEG2000

6.6.1 Overview

JPEG has been a big success with around 80% of all images still stored in this format. However, in the late 1990s, JPEG's limited coding efficiency, the presence of annoying visual blocking artifacts at high compression ratios, limited color gamut, and limited resolution were all reasons why work on an improved standard began. Furthermore, many applications were emerging that required functionalities not supported by JPEG, for example: spatial scalability, SNR scalability, and region of interest (ROI) coding.

The aim of JPEG2000 was to achieve a 30% bit rate saving for the same quality compared to JPEG, supporting 2–16 million colors and both lossless and lossy compression within the same architecture at bit depths greater than 8 bits. Additional features to support region of interest coding, error resilience, and data security were also included and a high emphasis was placed on scalability and progressive transmission. JPEG2000 Part 1 became an International Standard (ISO/IEC 15444-1) in December 2000 [21,29,30].

6.6.2 Architecture—bit planes and scalable coding

In JPEG2000, each subband is split into a number of code blocks. The encoder uses Embedded Block Coding with Optimal Truncation (EBCOT) [17,21] to code the quantized coefficients. Each bit plane of each code block is encoded during three passes (Significance Propagation, Magnitude Refinement and Cleanup), the first encoding bits (and signs) of insignificant coefficients with significant neighbors, the

Table 6.2 Technical overview of JPEG2000.

Property	Attribute
Wavelet transform	Dyadic decomposition
Filters	5/3 Integer filter (enables strictly reversible compression) [25] 9/7 Floating point filter [26]
Block structure	Each subband is divided into codeblocks
Quantization	Scalar quantization
Entropy coding	Block-based (EBCOT) and context-based arithmetic coding
Regions of Interest	Maxshift with multiple regions supported—ROI shape not encoded
Error resilience	JPWL for wireless transmission
Security tools	JPSEC
Streaming tools	JPIP

second refining bits of significant coefficients and finally coding coefficients without significant neighbors. In the JPEG2000 lossless mode, all bit planes must be encoded, whereas in lossy mode some of the bit planes can be dropped. The bits selected during EBCOT are passed to a binary MQ encoder (a context-driven binary arithmetic coder), where the context of each coefficient is related to the state of its nine neighbors in the code block.

An overview of the main features of JPEG2000 is given in Table 6.2.

6.6.3 Coding performance

Examples of JPEG2000 coding are provided in Figure 6.31. This shows the Puppet test image (512×512), encoded and decoded at compression ratios of 64:1 and 128:1, compared to JPEG. It can be seen that the JPEG2000 codec significantly outperforms JPEG at high compression ratios.

In terms of compression performance, JPEG2000 achieves similar results to the intra mode of H.264/AVC and is generally inferior to HEVC. The latter two methods benefit from variable block sizes and sophisticated intra-prediction techniques. This perhaps also provides an explanation of why wavelet methods have not been widely adopted for video coding, where advanced prediction modes and adaptive coding structures, used in combination with block-based transforms, can often provide equivalent or superior results.

6.6.4 Region of interest coding

ROI processing in JPEG2000 enables the coding of important portions of an image at higher qualities than the surrounding content (or background). Before entropy coding, the bit planes of the coefficients belonging to the ROI mask are shifted up. This scaling corresponds to a local increase of dynamic range, which is then prioritized in the rate allocation process. At the decoder, the quantized wavelet coefficients within each ROI are scaled back to their original values. This process is illustrated in Figure 6.32.

FIGURE 6.31

JPEG2000 encoding of *Puppet* test image (512 × 512). Top left to bottom right: JPEG at 64:1; JPEG2000 at 64:1; JPEG at 128:1; JPEG2000 at 128:1.

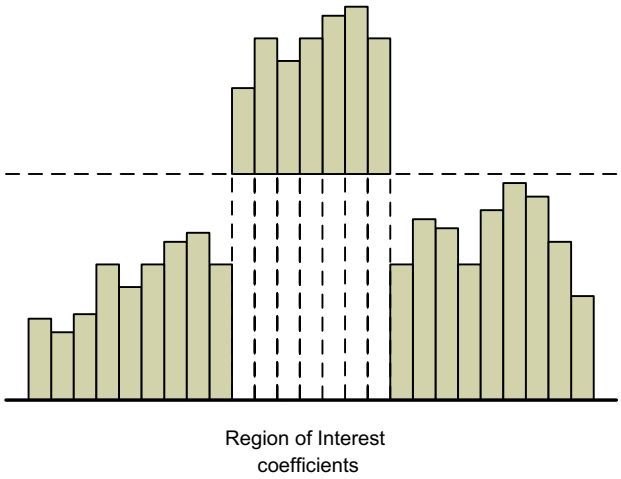


FIGURE 6.32

JPEG2000 ROI coefficient coding using Maxshift.

6.6.5 Benefits and status

Although the core image coding part of JPEG2000 is intended by ITU-T to be royalty-free, it is not patent-free. It has been widely reported that the reason why the standard has not gained the success it perhaps deserves is because of fears of patent infringement. JPEG2000 has however gained favor in many military and surveillance applications, where its improved performance and its ability to offer scalable delivery have proved attractive.

6.7 Summary

This chapter has explored an approach to image decorrelation and source coding based on multiscale analysis and synthesis. Through the combination of appropriate filters with up- and downsampling, we have seen how critical constant sample rates can be maintained throughout the filter bank. We have also demonstrated how practical filters can be designed that, in the absence of quantization, achieve perfect reconstruction. We finally considered specific cases of wavelet filters, particularly in the context of the JPEG2000 still image coding standard.

When used in a lossy image compression system, wavelet-based coding generally produces less annoying visual artifacts than block-based transforms and it can also cope better with localized non-stationarities.

References

[1] A. Haar, Zur theorie der orthogonalen funktionensysteme, Mathematical Analysis 69 (1910) 331–371.

[2] A. Crosier, D. Esteban, C. Galand, Perfect channel splitting by use of interpolation/decimation techniques, in: Proceedings of the IEEE Conference on Information Science and Systems, 1976.

[3] M. Smith, T. Barnwell, A procedure for designing exact reconstruction filterbanks for tree-structured sub-band coders, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 1984.

[4] M. Smith, T. Barnwell, Exact reconstruction techniques for tree structured subband coders, IEEE Transactions on Acoustics, Speech and Signal Processing 34 (1986) 434–441.

[5] M. Vetterli, Filter banks allowing perfect reconstruction, Signal Processing 10 (3) (1986) 219–244.

[6] J. Woods, S. O'Neill, Sub-band coding of images, IEEE Transactions on Acoustics, Speech and Signal Processing 34 (1986) 1278–1288.

[7] P. Vaidyanathan, Theory and design of M-channel maximally decimated quadrature mirror filters with arbitrary M, having the perfect reconstruction property, IEEE Transaction on Acoustics, Speech and Signal Processing 35 (4) (1987) 476–492.

[8] A. Grossman, J. Morlet, Decomposition of hardy functions into square-integrable wavelets of constant shape, SIAM Journal on Mathematical Analysis 15 (4) (1984) 723–736.

[9] I. Daubechies, Orthonormal bases of compactly supported wavelets, Communications on Pure and Applied Mathematics 41 (1988) 909–996.

[10] S. Mallat, A theory for multiresolution signal decomposition: the wavelet representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (7) (1989) 674–693.
[11] S. Mallat, Multifrequency channel decomposition of images and wavelet models, IEEE Transactions on Acoustics, Speech and Signal Processing 37 (12) (1989) 2091–2110.
[12] M. Vetterli, J. Kovacevic, Wavelets and Subband Coding, Prentice Hall, 1995 (reissued 2007).
[13] R. Rao, A. Bopardikar, Wavelet Transforms – Introduction to Theory and Applications, Addison Wesley, 1998.
[14] G. Strang, T. Nguyen, Wavelets and Filter Banks, Wellesley-Cambridge Press, 1996.
[15] O. Rioul, M. Vetterli, Wavelets and signal processing, IEEE Signal Processing Magazine 8 (4) (1991) 14–38.
[16] A. Said, W. Pearlman, A new, fast, and efficient image codec based on set partitioning in hierarchical trees, IEEE Transactions on Circuits and Systems for Video Technology 6 (3) (1996) 243–250.
[17] D. Taubman, High performance scalable image compression with EBCOT, IEEE Transactions on Image Processing 9 (7) (2000) 1151–1170.
[18] D. Taubman, A. Secker, Highly scalable video compression with scalable motion coding, IEEE Transactions on Image Processing 13 (8) (2004) 1029–1041.
[19] S. Bokhari, A. Nix, D. Bull, Rate-distortion-optimized video transmission using pyramid vector quantization, IEEE Transactions on Image Processing 21 (8) (2012) 3560–3572.
[20] P. Schelkens, A. Skodras, T. Ebrahimi, The JPEG 2000 Suite, Wiley-IS&T Series in Imaging Science and Technology, Wiley, 2009.
[21] D. Taubman, M. Marcellin, JPEG 2000: Image Compression Fundamentals, Standards and Practice, Kluwer International Series in Engineering and Computer Science, 2001.
[22] J. Johnson, A filter family designed for use in quadrature mirror filter banks, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 1980, pp. 291–294.
[23] J. Villasenor, B. Belzer, J. Liao, Wavelet filter evaluation for image compression, IEEE Transactions on Image Processing 4 (8) (1995) 1053–1060.
[24] N. Kingsbury, Good Filters/ Wavelets, Connexions. <http://cnx.org/content/m11139/2.3/>, June 8, 2005.
[25] D. LeGall, A. Tabatabai, Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 1988, pp. 761–765.
[26] M. Antonini, M. Barlaud, P. Mathieu, I. Daubechies, Image coding using wavelet transform, IEEE Transactions on Image Processing 1 (2) (1992) 205–220.
[27] M. Unser, T. Blu, Mathematical properties of the JPEG2000 wavelet filters, IEEE Transactions on Image Processing 12 (9) (2003) 1080–1090.
[28] J. Shapiro, Embedded image coding using zerotrees of wavelet coefficients, IEEE Transactions on Signal Processing 41 (12) (1993) 3445–3462.
[29] JPEG2000 Core Coding System, ISO/IEC Standard ISO/IEC 15444-1, 2000.
[30] <http://www.itu.int/rec/T-REC-T.800/en>.