CHAPTER 14

Wavelet Transforms

4.1 INTRODUCTION

Considerable interest has arisen in recent years regarding new transform techniques that specifically address the problems of image compression, edge and feature detection, and texture analysis. These techniques come under the headings of multiresolution analysis, time-frequency analysis, pyramid algorithms, and wavelet transforms [1].

In this chapter, we review some of the limitations of the traditional Fourier and similar transforms and define three types of wavelet transforms that promise improved performance for certain applications. We trace some of the developments that have led to the current state of wavelet analysis, noting the similarities that tend to unify these different approaches under the banner of *wavelet transforms.* Later in the chapter, we illustrate some of the applications of wavelet transforms.

We restrict ourselves to transforming real-valued, measurable, square-integrable functions of one and two dimensions, since these encompass the signals and images that are of interest to us. As before, we introduce each concept in one dimension for simplicity and then generalize it to two dimensions for application to images. We begin by introducing the three basic types of wavelet transforms. Then we illustrate some particular wavelets and some applications of wavelet transforms.

14.1.1 Waves and Wavelets

Recall that the Fourier transform uses, as its orthonormal basis functions, sinusoidal *waves,* so called because they resemble the waves of the ocean and propagating waves in other media. For the integral transform, these functions extend to infinity in both directions. The basis vectors of the discrete Fourier transform are also nonzero over their entire domain; that is, they do not have *compact support.*

By contrast, *transient* signal components are nonzero only during a short interval. Likewise, many important features in images (edges, for example) are highly localized in spatial position. Such components do not resemble any of the Fourier basis functions, and they are not represented compactly in the transform coefficients (i.e., the frequency spectrum), as discussed subsequently. This makes the Fourier and other wave transforms, such as those mentioned in the previous chapter, less than optimal representations for compressing and analyzing signals and images containing transient or localized components.

In fairness, we note that the Fourier transform can represent any analytic function—even a narrow transient signal—as a sum of sinusoids. It does this, however, by intricately arranging for the cancellation of sine waves (by destructive interference) to create a function that is zero over most of the interval. This is, of course, a valid way for an invertible transform to behave, but it leaves the spectrum a rather confusing picture of the function.

To combat such a deficiency, mathematicians and engineers have explored several approaches using transforms having basis functions of limited duration. These basis functions vary in position as well as frequency. They are waves of limited duration and are referred to as *wavelets.* Transforms based on them are called *wavelet transforms.* They are also called *ondelettes* in the considerable amount of French-language literature on the subject.

Figure 14–1 illustrates the difference between waves and wavelets. The top two curves are cosine waves that differ in frequency, but not in duration. The lower two are wavelets that differ in both frequency and position along the axis.

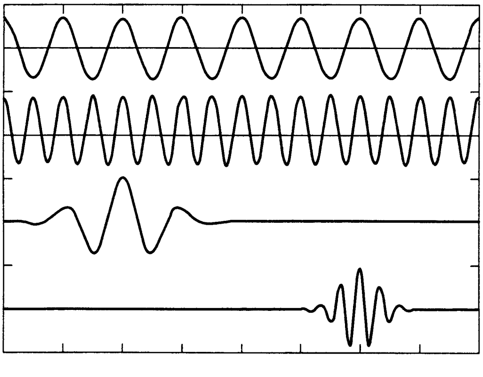

Figure 14–1 Waves and wavelets

The Haar transform (see Sec. 13.5.4) is the earliest example of what we now call a wavelet transform [2]. It differs from the other transforms in Chapter 13 in that its basis vectors are all generated by translations and scalings of a single function. The Haar function, which is an odd rectangular pulse pair, is the oldest and simplest wavelet.

14.1.2 Time-Frequency Analysis

The literature on signal processing includes considerable work regarding analyzing signals in terms of a two-dimensional *time-frequency* space. This approach actually preceded wavelet transforms, but it now fits into the same modern framework. According to it, each transient component of a signal maps to a position in the time-frequency plane that corresponds to that component's predominant frequency and time of occurrence (Figure 14–2).

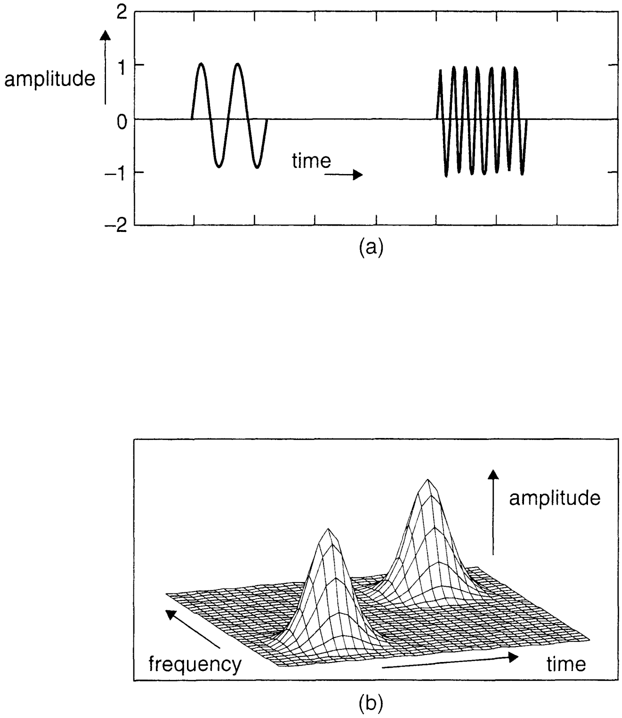

Figure 14–2 Time-frequency space: (a) signal; (b) representation

In image analysis, the space is three dimensional and can be viewed as an image stack. A localized component will appear primarily at the level in the stack that corresponds to the component's predominant frequency. Figure 14–3 shows an image containing two localized components being submitted to two bandpass filters. In this case the two filters almost completely isolate the two components.

This approach began with Gabor's [3] *windowed Fourier transform,* and led to the *short-time Fourier transform* (STFT) and then to *subband coding.*

14.1.2.1 Wavelets and Music

Consider the musical notation shown in Figure 14–4. It can be viewed as depicting a two-dimensional time-frequency space. Frequency (pitch) increases from the bottom of the scale to the top, while time (measured in beats) advances to the right. Each note on the sheet music corresponds to one wavelet component (tone burst) that would appear in the recording of a performance of the song. The duration of each wavelet is coded by the type of note (*e.g.*, quarter note, half note, etc.), rather than by its horizontal extent.

If we were to analyze a recorded musical performance and write out the corresponding score, we would have a type of wavelet transform. Similarly, a recording of a musician's performance of a song can be viewed as an inverse wavelet transform, since it reconstructs the signal from a time-frequency representation.

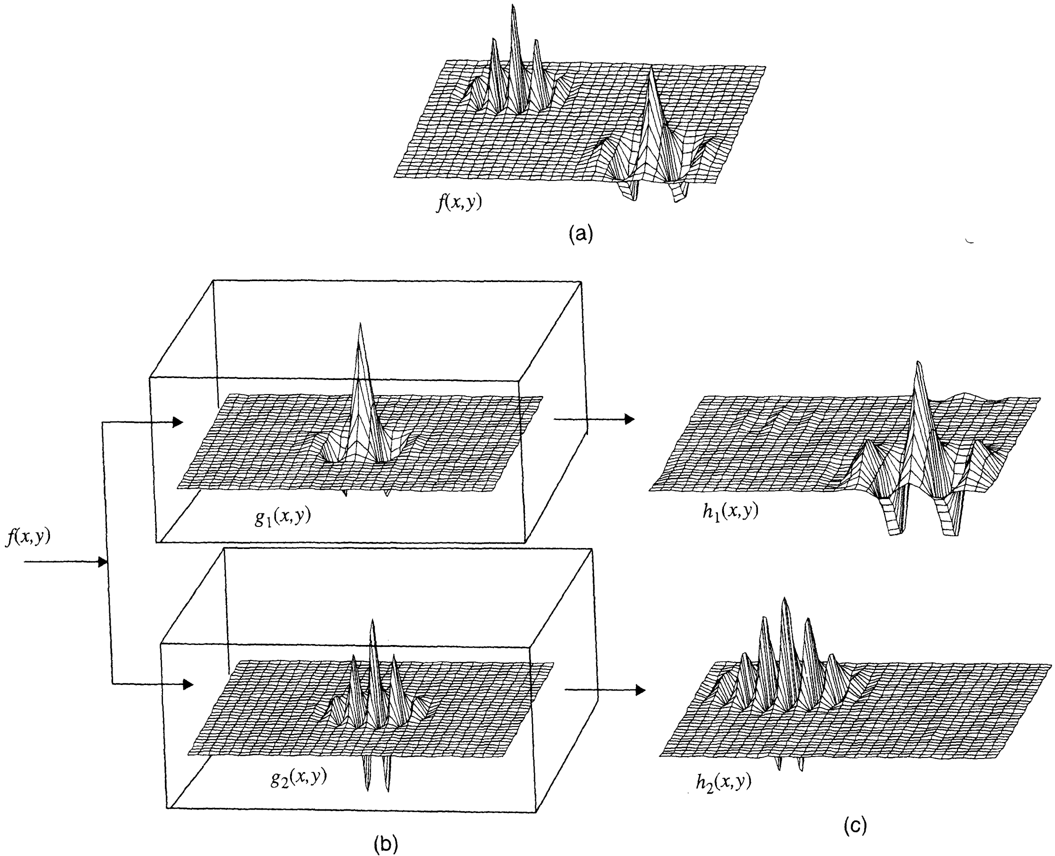

Figure 14–3 Space-frequency analysis of an image

Figure 14–4 Musical notation as a time-frequency plane

14.1.3 Transforms

Recall that each of the coefficients in a transform is determined by taking an inner product between the input function and one of the basis functions. This value represents, in some sense, the degree of similarity between the input function and that particular basis function. If the basis functions are orthogonal (or orthonormal), then an inner product taken between two basis functions is zero, indicating that these are all completely dissimilar. So if the signal or image is made up of components that are similar to one, or a few, of the basis functions, then all but one or a few of the coefficients will be small.

Similarly, the inverse transform can be viewed as reconstructing the original signal or image by summing basis functions that are weighted in amplitude by the transform coefficients. So if the signal or image is made up of components that are similar to one or a few of the basis functions, then this summation needs to have only a few terms of significant amplitude. Many of the terms can then be ignored, and the signal or image can be represented compactly by only a few transform coefficients.

Further, if the components of interest in the signal or image are similar to one or a few of the basis functions, then those components will manifest themselves in large coefficients for those (and only those) basis functions. They will thus be "easy to find" in the transform. Finally, if an undesirable (noise) component is similar to one or a few basis functions, then it, too, will be easy to find. It will be also easy to remove, simply by reducing (or setting to zero) the corresponding transform coefficients.

We conclude from all of this that there is potential value in using transforms with basis functions that are similar to the expected components of the signals or images to be transformed. We also note that transient components cannot be similar to the basis functions of the Fourier or other wave-type transforms.

14.1.3.1 Types of Transforms

Recall from Chapter 10 that there are three different, but related Fourier transformation techniques: the Fourier integral transform, the Fourier series expansion, and the DFT.

The Fourier integral transform associates two continuous functions (a signal and its spectrum). It and its inverse are given in one dimension by

$$F(s) = \int_{-\infty}^{\infty} f(x)e^{-j2\pi(xs)}dx \quad \text{and} \quad f(x) = \int_{-\infty}^{\infty} F(s)e^{j2\pi(xs)}ds \tag{1}$$

The Fourier series expansion represents a periodic function (or a transient function that can be considered to be one cycle of a periodic function) as a (finite or infinite) sequence of Fourier coefficients. It and its inverse are obtained by making $s = n\Delta s$ a discrete variable, so that

$$F_n = F(n\Delta s) = \int_0^L f(x)e^{-j2\pi(n\Delta sx)}dx \quad \text{and} \quad f(x) = \Delta s\sum_{n=0}^{\infty} F_n e^{j2\pi(n\Delta sx)} \tag{2}$$

where L is the period and $\Delta s = 1/L$.

The DFT represents a sampled function by a sampled spectrum, and the number of independent samples (degrees of freedom) is the same in both domains. It is obtained by making $x = i\Delta x$ a discrete variable as well. If $g(x)$ is bandlimited and sampled as required by the sampling theorem (Sec. 12.2.4), then $g_i = g(i\Delta x)$, and

$$G_k = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1} g_i e^{-j2\pi k\frac{i}{N}} \quad \text{and} \quad g_i = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} G_k e^{j2\pi i\frac{k}{N}} \tag{3}$$

In all three transformation techniques, sines and cosines of different frequencies form a set of orthonormal basis functions. Also, each transform coefficient is determined by an inner product of the function being transformed and one of the basis functions. A discrete inner product and discrete basis functions are used for the DFT, while an integral inner product and continuous basis functions serve for the other two transforms. In each case, the inverse transform consists of summing basis functions whose amplitudes are weighted by the transform coefficients. This summation becomes an integral for the continuous Fourier transform.

The discrete transforms discussed in the previous chapter also use discrete orthonormal basis functions. Thus, they behave in a manner generally similar to the way the DFT behaves. For most of them, the basis functions are real and the forward and inverse transforms are identical.

14.1.3.2 Types of Wavelet Transforms

As with the Fourier transform, the same three possibilities exist for wavelet transforms: a continuous wavelet transform (CWT), a wavelet series expansion, and a discrete wavelet transform (DWT). The situation is slightly more complex, however, since the wavelet basis functions may or may not be orthonormal.

A set of wavelet basis functions can support a transform even if the functions are not orthonormal. This means, for example, that a wavelet series expansion might represent a bandlimited function by infinitely many coefficients. If this sequence of coefficients is truncated to finite length, then we can reconstruct only an approximation of the original function. Likewise, a discrete wavelet transform might require more coefficients than the original function has sample points in order to reconstruct it exactly, or even to an acceptable approximation.

14.1.3.3 Notation and Definitions

Next, we introduce some definitions to clarify the concept of a wavelet transform. For the present, we restrict the discussion primarily to transforming functions of one dimension.

In order to conform with the bulk of the literature on wavelets we use j as an integer index in this chapter. As elsewhere in the book, we also use j to represent the imaginary unit $\sqrt{-1}$, taking care not to use it both ways in the same equation. The distinction should be clear from the context.

The class of functions we seek to represent by a wavelet transform is those that are square integrable on the real line (i.e., the set of all real numbers—the x-axis). This class is denoted as $L^2(R)$. Thus, the notation $f(x) \in L^2(R)$ means

$$\int_{-\infty}^{\infty} |f(x)|^2 dx < \infty \tag{4}$$

In wavelet analysis, we generate a set of basis functions by dilating and translating a single prototype function, $\psi(x)$, which we call a *basic wavelet.* This is some oscillatory function, usually centered upon the origin, that dies out rapidly as $|x| \to \infty$. Thus, $\psi(x) \in L^2(R)$.

14.2 THE CONTINUOUS WAVELET TRANSFORM

The continuous wavelet transform (also called the integral wavelet transform) was introduced by Grossman and Morlet [4].

14.2.1 Definition

If $\psi(x)$ is a real-valued function whose Fourier spectrum, $\Psi(s)$, satisfies the *admissibility criterion* [4,5]

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\Psi(s)|^2}{|s|} ds < \infty \tag{5}$$

then $\psi(x)$ is called a *basic wavelet.* Notice that, due to the s in the denominator of the integrand, it is necessary that

$$\Psi(0) = 0 \Rightarrow \int_{-\infty}^{\infty} \psi(x) dx = 0 \tag{6}$$

Furthermore, since $\Psi(\infty) = 0$ as well, we can see that the amplitude spectrum of an admissible wavelet is similar to the transfer function of a bandpass filter. In fact, any bandpass filter impulse response with zero mean [Eq. (6)] that decays to zero fast enough with increasing frequency [Eq. (5)] can serve as a basic wavelet for this transform.

A set of wavelet basis functions, $\{\psi_{a,b}(x)\}$, can be generated by translating and scaling the basic wavelet, $\psi(x)$, as

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}} \psi\left(\frac{x-b}{a}\right) \tag{7}$$

where $a > 0$ and b are real numbers. The variable a reflects the scale (width) of a particular basis function, while b specifies its translated position along the x-axis.

Normally the basic wavelet, $\psi(x)$, is centered at the origin, so that $\psi_{a,b}(x)$ is centered at $x = b$. Figure 14–5 shows an example of such a wavelet. This particular one is given by

$$\psi(x) = \frac{2}{\sqrt{3}\sqrt{\pi}}(1 - x^2)e^{-x^2/2} \tag{8}$$

The *continuous wavelet transform* of $f(x)$ with respect to the wavelet $\psi(x)$ is then [4,5]

$$W_f(a, b) = \langle f, \psi_{a,b} \rangle = \int_{-\infty}^{\infty} f(x) \psi_{a,b}(x) dx \tag{9}$$

The wavelet transform coefficients are once again given as inner products of the function being transformed with each of the basis functions.

Grossman and Morlet [4] showed that the inverse continuous wavelet transform is

$$f(x) = \frac{1}{C_\psi} \int_0^{\infty} \int_{-\infty}^{\infty} W_f(a, b) \psi_{a,b}(x) db \frac{da}{a^2} \tag{10}$$

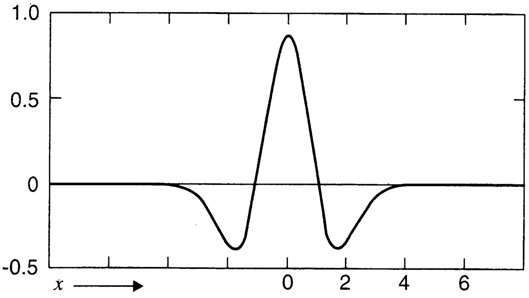

Figure 14–5 A wavelet

The scale factor in front of the right-hand side of Eq. (7) ensures that the norms of the wavelet basis functions are all equal, since

$$\left\|f\left(\frac{x-b}{a}\right)\right\| = \sqrt{\int_{-\infty}^{\infty}\left|f\left(\frac{x-b}{a}\right)\right|^2 dx} = \sqrt{a}\|f(x)\| \tag{11}$$

Since the basic wavelet has zero mean [Eq. (6)], all scalings and translations ŏf it [Eq. (7)] will likewise have zero mean, and the mean of $f(x)$ must be accounted for separately.

14.2.2 The Two-Dimensional CWT

The continuous wavelet transform $W(a,b)$ of a one-dimensional function $f(x)$ is a function of two variables, one more than $f(x)$. The CWT is said to be *overcomplete,* as it represents a considerable increase in information content and in the volume required for data storage. For functions of more than one variable, this transform also increases the dimensionality by one.

If $f(x, y)$ is a function of two dimensions, its continuous wavelet transform is

$$W_f(a, b_x, b_y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)\,\psi_{a, b_x, b_y}(x, y)dx\,dy \tag{12}$$

where b_x and b_y specify the translation in two dimensions. The inverse two-dimensional continuous wavelet transform is

$$f(x, y) = \frac{1}{C_\psi}\int_0^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} W_f(a, b_x, b_y)\,\psi_{a, b_x, b_y}(x, y)\,db_x db_y\frac{da}{a^3} \tag{13}$$

where

$$\psi_{a, b_x, b_y}(x, y) = \frac{1}{|a|}\psi\left(\frac{x-b_x}{a}, \frac{y-b_y}{a}\right) \tag{14}$$

and $\psi(x, y)$ is a two-dimensional basic wavelet. The same generalization extends to cover functions of more than two variables.

14.2.3 The Filter Bank Interpretation

The following exercise illustrates one way of viewing the continuous wavelet transform. We first define the general wavelet basis function at scale a as

$$\psi_a(x) = \frac{1}{\sqrt{a}}\psi\left(\frac{x}{a}\right) \tag{15}$$

This is the basic wavelet scaled by a and normalized by $a^{-1/2}$. It defines a set of functions that become broader with increasing a. We also define

$$\breve{\psi}_a(x) = \psi_a^*(-x) = \frac{1}{\sqrt{a}}\psi^*\left(-\frac{x}{a}\right) \tag{16}$$

which is the reflected complex conjugate of the scaled wavelet. If $\psi(x)$ is real and even, as is often the case, the reflection and conjugation have no effect.

Now we can write the continuous wavelet transform [Eq. (9)] as

$$W_f(a, b) = \int_{-\infty}^{\infty} f(x)\breve{\psi}_a(b-x)dx = f*\breve{\psi}_a \tag{17}$$

For fixed a, then, $W_f(a, b)$ is the convolution of $f(x)$ with the reflected conjugate wavelet at scale a.

Figure 14–6 shows the integral wavelet transform as a bank of linear (convolution) filters acting upon $f(x)$. Each value of a defines a different bandpass filter, and the outputs of all the filters, taken together, comprise the wavelet transform. Further, Eq. (10) becomes

$$f(x) = \frac{1}{C_\psi}\int_0^{\infty}\int_{-\infty}^{\infty}[f*\breve{\psi}_a](b)\,\psi_a(b-x)\,db\frac{da}{a^2} = \frac{1}{C_\psi}\int_0^{\infty}[f*\breve{\psi}_a*\psi_a](x)\frac{da}{a^2} \tag{18}$$

which implies that the filter outputs, each filtered again by $\psi_a(x)$ and properly scaled, combine to reconstruct $f(x)$. This is a statement of Calderon's identity [6,7], which predates Grossman and Morlet by 20 years.

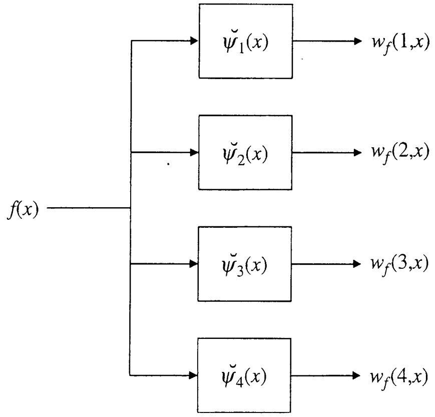

Figure 14–6 Filter bank analogy for the integral wavelet transform of a signal

Recall from the similarity theorem (Sec. 10.2.5) that

$$\mathfrak{F}\{f(ax)\} = \frac{1}{|a|}F\left(\frac{s}{a}\right) \tag{19}$$

This means that

$$\Psi_a(s) = \mathfrak{F}\{\psi_a(x)\} = \sqrt{a}\,\Psi(as) \tag{20}$$

and the center frequencies of the bandpass filters decrease as the transfer functions become more narrow with increasing a.

14.2.4 Two Dimensional Filter Banks

Figure 14–7 illustrates the filter bank approach in two dimensions. Here, each filter $\psi_a(x,y)$ is a two-dimensional impulse response, and its output is a bandpass-filtered version of the image. (Recall Figure 14–3). The stack of filtered images comprises the wavelet transform.

Again, the redundancy is considerable. In fact, if $\Psi(u,v)$, the transfer function of $\psi(x, y)$, is nonzero everywhere except at the origin, one could, theoretically, recover the

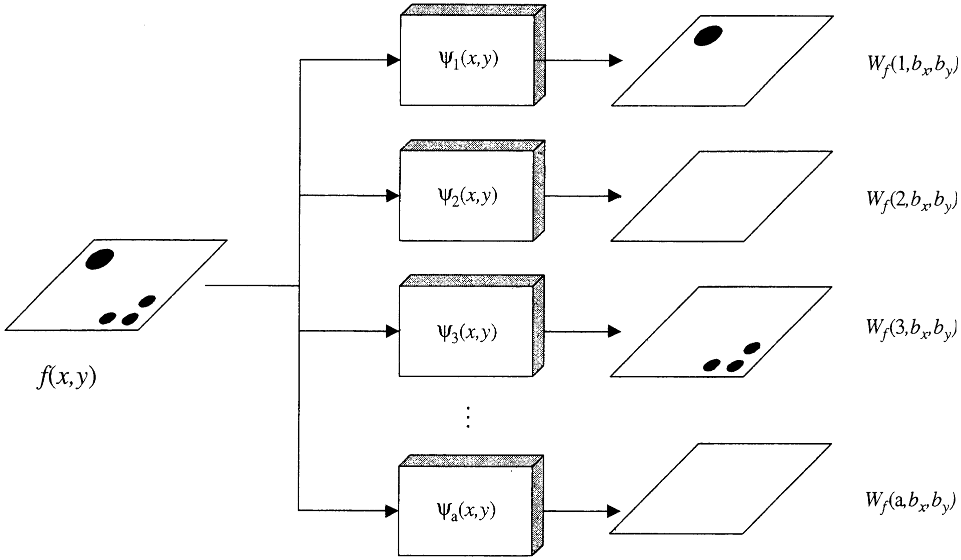

Figure 14–7 Filter bank analogy for the integral wavelet transform of an image

original image from *any one* of the filter outputs by inverse filtering (e.g., by deconvolution). Alternatively, if the image is bandlimited to an interval over which at least one $\Psi_a(u, v)$ is nonzero, then $f(x, y)$ could be recovered from that filter output alone. The conclusion, then, is that the potential value of the integral wavelet transform lies not in a compact representation, but in decomposition and analysis of signals and images.

To illustrate this, suppose that the image in Figure 14–7 contained, for example, circular objects of different sizes and that the basic wavelet were selected to respond only (or primarily) to circular objects of unit radius. Then an examination of the output image stack would reveal the locations of the objects. Further, each object would appear only (or primarily) in the specific output image that corresponded to its particular size.

14.3 THE WAVELET SERIES EXPANSION

14.3.1 Dyadic Wavelets

The second type of wavelet transform is somewhat more restrictive than the first. Again, a basic wavelet is scaled and translated to form a set of basis functions. Here, however, the scaling and translation are specified by integers rather than real numbers.

In this second definition, we restrict ourselves to forming the basis functions by *binary scalings* (shrinking by factors of two) and *dyadic translations* of the basic wavelet, $\psi(x)$. A dyadic translation is a shift by the amount $k/2^j$, which is an integer multiple of the binary scale factor and thus of the width of the wavelet as well. Binary scalings and dyadic translations are illustrated in Figure 14–8.

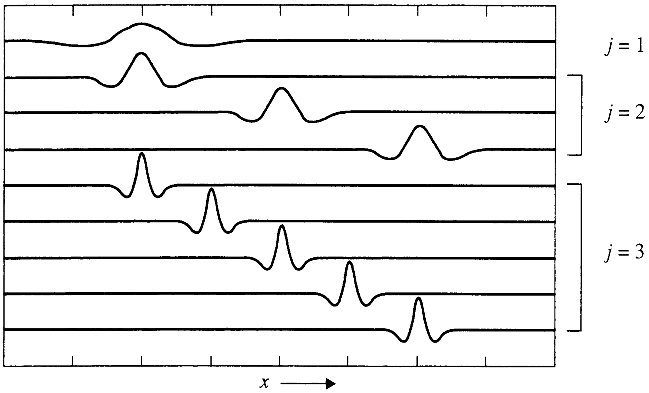

Figure 14–8 Binary scalings and dyadic translations of a wavelet

14.3.2 Definition

A function $\psi(x)$ is an *orthogonal wavelet* if the set $\{\psi_{j,k}(x)\}$ of functions defined by

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k) \tag{21}$$

where $-\infty < j,k < \infty$ are integers, forms an orthonormal basis of $L^2(R)$ [5]. The integer j determines the dilation, while k specifies the translation.

The preceding wavelet set forms an orthonormal basis if, first,

$$\langle \psi_{j,k}, \psi_{l,m}\rangle = \delta_{j,l}\delta_{k,m} \tag{22}$$

where l and m are integers, $\delta_{j,k}$ is the Kronecker delta function, and $\langle \cdot, \cdot\rangle$ indicates the inner product; and second, if any function $f(x) \in L^2(R)$ can be written as

$$f(x) = \sum_{j=-\infty}^{\infty}\sum_{k=-\infty}^{\infty} c_{j,k}\psi_{j,k}(x) \tag{23}$$

where the transform coefficients are again given by inner products; that is,

$$c_{j,k} = \langle f(x), \psi_{j,k}(x)\rangle = 2^{j/2}\int_{-\infty}^{\infty} f(x)\psi(2^j x - k)dx \tag{24}$$

Eqs. (23) and (24) specify a *wavelet series expansion* of $f(x)$ relative to the wavelet $\psi(x)$ [5].

Notice that here a continuous function is represented by a doubly infinite sequence, and, in general, the transform is again overcomplete. Since the basis functions commonly extend to infinity in both directions, a complete reconstruction must include all terms.

If $\psi(x)$ is properly chosen, however, one might be able to truncate the series without serious approximation error. If $f(x)$ is of finite duration, and the basic wavelet is well localized (i.e., it approaches zero rapidly away from the origin), then many of the coefficients with large $|k|$ will be negligible. Likewise, coefficients with large $|j|$ will usually be small as well, since the wavelet basis function then becomes extremely broad or narrow.

14.3.3 Compact Dyadic Wavelets

If we further restrict $f(x)$ and the basic wavelet to functions that are zero outside the interval [0,1], then the family of orthonormal basis functions can be specified by a single index, n; that is,

$$\psi_n(x) = 2^{j/2}\psi(2^j x - k) \tag{25}$$

where j and k are actually functions of n, as follows:

$$n = 2^j + k \quad \text{for} \quad j = 0, 1, \ldots \quad k = 0, 1, \ldots, 2^j - 1 \tag{26}$$

For any n, j is the largest integer such that $2^j \le n$, and $k = n - 2^j$.

Now the inverse transform is

$$f(x) = \sum_{n=0}^{\infty} c_n \psi_n(x) \tag{27}$$

where it is assumed that $\psi_0(x) = 1$. The transform coefficients are given by the inner product

$$c_n = \langle f(x), \psi_n(x)\rangle = 2^{j/2}\int_{-\infty}^{\infty} f(x)\psi(2^j x - k)dx \tag{28}$$

Here, a continuous function is being represented by a single infinite sequence, as with a Fourier series representation. The tremendous redundancy of the integral wavelet transform is absent. In fact, if one or a few of the $\psi_n(x)$ are similar to $f(x)$ (or its major components), then one might be able to truncate the series to a relatively few terms without appreciable approximation error.

We have here, as well, the basis of the discrete wavelet transform. If $f(i\Delta t)$ is a discrete function sampled with N points, where N is a power of two, and if $\psi(x)$ is a compact dyadic wavelet, then we can compute a discrete wavelet transform using discrete versions of Eqs. (27) and (28). Both equations become summations of N terms. The Haar transform offers an example of this.

14.3.3.1 Example: The Haar Transform

The Haar transform [2,8,9] is one of the earliest examples of what we now call a compact, dyadic, orthonormal wavelet transform. It differs from the other transforms mentioned in Chapter 13 in that its basis functions are all generated by translations and dilations of a basic wavelet. The Haar function, which is an odd rectangular pulse pair, is the simplest and oldest orthonormal wavelet with compact support.

The basic wavelet is progressively narrowed (reduced in scale) by powers of two. Each smaller wavelet is then translated by increments equal to its width, so that the complete set of wavelets at any scale completely covers the interval. As the basic wavelet is scaled down by powers of two, its amplitude is scaled up by powers of $\sqrt{2}$, to maintain orthonormality. The result of all this is a set of orthonormal basis functions (Figure 14–9). The basis function index, as defined in Eq. (26), differs slightly from that used in Sec. 13.5.4.

14.4 THE DISCRETE WAVELET TRANSFORM

The DWT most closely resembles the unitary transforms discussed in the previous chapter. It promises to be the most useful for image compression, processing, and analysis. Given a set of orthonormal basis functions, one can compute the discrete wavelet transform just as one does any other unitary transform, such as the Haar transform. Obtaining a suitable basic wavelet, however, requires further background material.

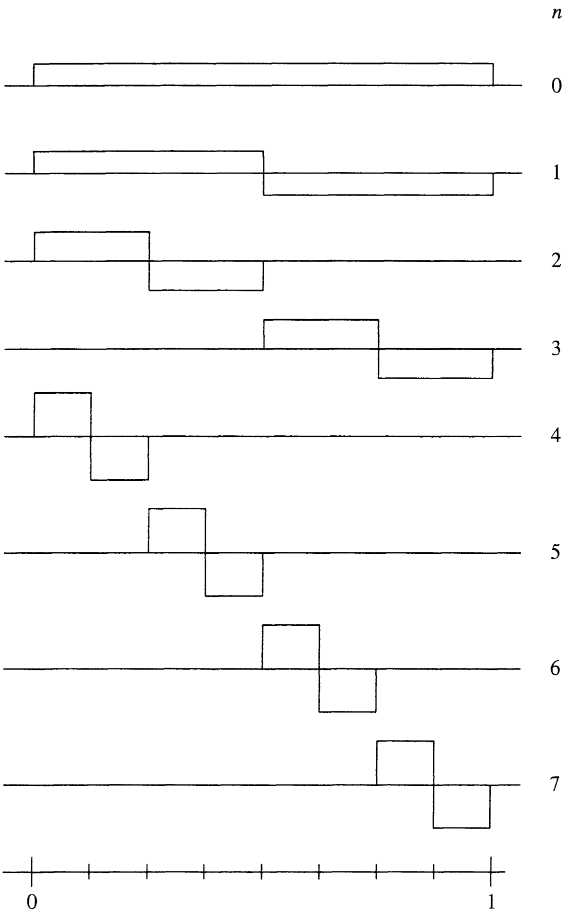

Figure 14–9 The Haar transform basis functions

In this section, we first review three techniques that have led to the development of the discrete wavelet transform: (1) filter bank theory, (2) multiresolution or time-scale analysis, particularly using pyramid representations, and (3) subband coding. This discussion is followed by an introduction to the discrete wavelet transform.

14.4.1 Filter Bank Theory

Workers in the area of speech analysis and acoustic signal processing have long used the concept of a bank of bandpass filters for decomposing a signal into components at different frequencies. Indeed, the method is a precursor to *time-frequency analysis,* in which the signal's components are displayed in a two-dimensional space whose dimensions are time of occurrence and frequency of oscillation. Here, we review the basics of that approach as a step leading toward a discussion of the discrete wavelet transform.

Suppose we have a signal composed of two *tone bursts* (sinusoids of short duration) embedded in random noise, as illustrated in Figure 14–10a. Suppose further that we wish to analyze this signal to detect the number, frequency, and position of the tone bursts.

The Fourier transform will, of course, reflect the entire content of the signal, but often not in a way that is easily interpreted. Position information, for example, is encoded in the

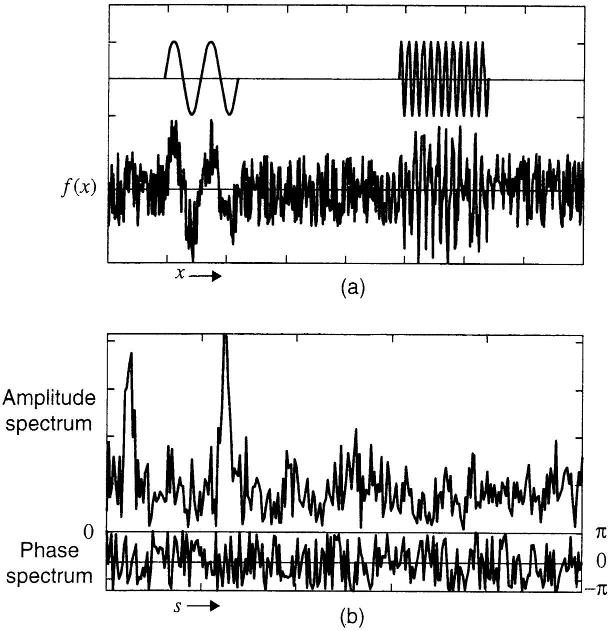

Figure 14–10 Composite signal containing two tone bursts and random noise: (a) the three components; (b) amplitude and phase spectra

phase spectrum in a complicated way. (Recall Figure 10–10.) While the amplitude spectrum may show distinct peaks due to each of the transient signal components, this is reliable only for transient detection when those components are large enough in amplitude and duration to dominate the spectrum. Figure 14–10b, for example, does manifest distinct peaks at the frequencies of the two tone bursts. The phase spectrum, however, gives little insight into the location of these components in time. Often, the uninteresting components of the signal (e.g., noise) complicate the spectrum to the point that a simple frequency analysis is insufficient to resolve the signal's components.

14.4.1.1 Ideal Bandpass Filters

Suppose we partition the frequency axis into a set of disjoint (adjacent, nonoverlapping) intervals and use this partitioning to define a set of ideal bandpass transfer functions, as shown in Figure 14–11b. The corresponding impulse responses appear in Figure 14–11a. Figure 14–12 shows the implementation of a bandpass filter bank. The input signal is fed into each of the bandpass filters in parallel. The corresponding outputs are $g_i(x)$. The $H_i(s)$ are constructed so that they sum to 1 for all frequencies, and thus, the $g_i(x)$ will sum to form $f(x)$. That is,

$$\sum_{i=1}^{\infty} H_i(s) = 1 \Rightarrow \sum_{i=1}^{\infty} g_i(x) = f(x) \tag{29}$$

Figure 14–13 shows the output of three of the bandpass filters shown in Figure 14–12. Notice that the two tone burst signals (recall Figure 14–10a) emerge from separate filters. Further, their locations along the time axis are evident in those outputs. Thus, we have an approach to decomposing the composite signal and identifying the components of interest.

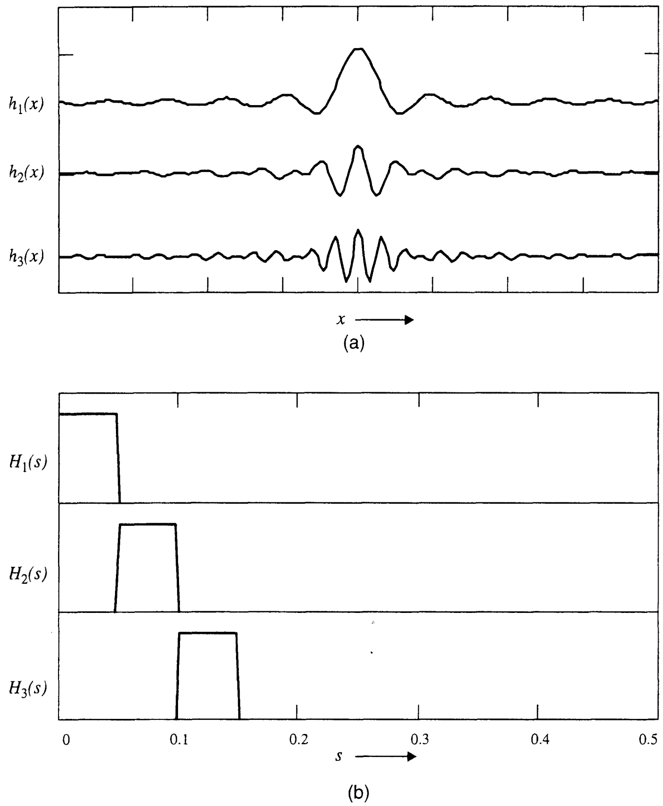

Figure 14–11 Generating a series of bandpass filters by partitioning the frequency axis: (a) impulse responses; (b) transfer functions

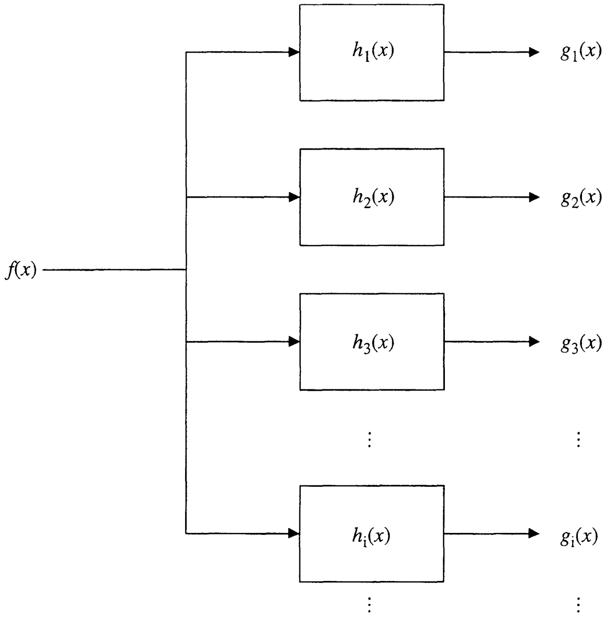

Figure 14–12 Implementation of a bandpass filter bank

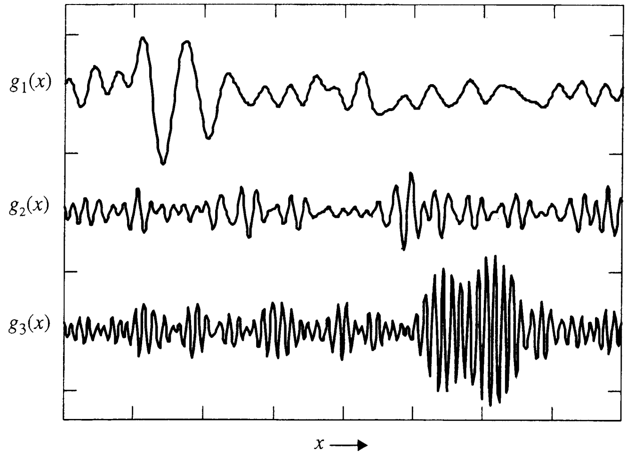

Figure 14–13 Bandpass filter outputs

Each of the bandpass filter outputs is formed by the convolution

$$g_i(x) = \int_{-\infty}^{\infty} f(t)h_i(x-t)dt \tag{30}$$

Since $H_i(s)$ is real and even, $h_i(x)$ will be as well. Then the reflection in the convolution integral has no effect, and the filter outputs can be written as

$$g_i(x) = \int_{-\infty}^{\infty} f(t)h_i(t-x)dt = \langle f(t), h_i(t-x)\rangle \tag{31}$$

Hence, each point on $g_i(x)$ is the inner product of $f(t)$ with a version of $h_i(t)$ that has been shifted to location x. We can also view $\{g_i(x)\}$ as a (two-dimensional) set of wavelet transform coefficients, where $\{h_i(x)\}$ is the set of wavelets. Further, $\{g_i(x)\}$ is sufficient to reconstruct $f(x)$ exactly, in view of Eq. (29).

The message borne by Eq. (31) is a significant one. The similarity between convolution, on the one hand, and taking inner products with shifted basis functions, on the other, is what brings the disparate pieces of the wavelet transform together into a unified whole.

14.4.1.2 Smooth Bandpass Filters

The functions $h_i(x)$ in Figure 14–11a lack one of the characteristics that good wavelet basis functions should have: They are not well localized. That is, they do not die out quickly away from their central region. This means that $h_i(x - x_0)$ will respond to strong components that are located distant from x_0. It is the sharp edges of $H_i(s)$ that give rise to the undesirable width of $h_i(x)$.

Designing the $H_i(s)$ functions to have smoother edges will reduce the width of the $h_i(x)$. Since the $H_i(s)$ must still sum to unity everywhere, the resulting bandpass transfer functions will overlap at their edges. One such construction is shown in Figure 14–14. Here, the passband edges are each a raised half-cycle of the cosine. The resulting narrowing of the impulse responses is evident.

Figure 14–15 shows the filter bank outputs with the signal of Figure 14–10a as input and smooth bandpass filters. Notice the improvement in localization. We have thus taken a

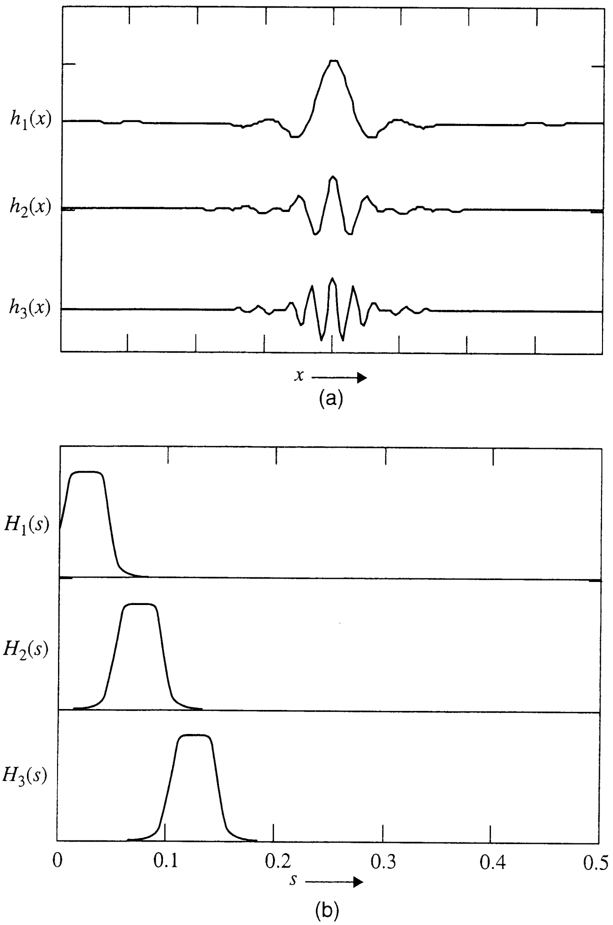

Figure 14–14 Smooth bandpass filters: (a) impulse responses; (b) transfer functions

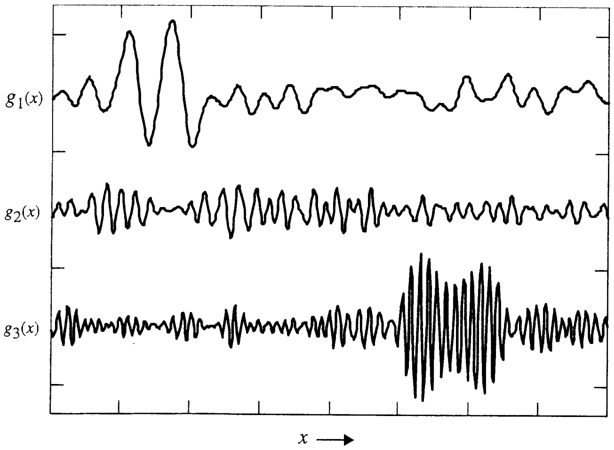

Figure 14–15 Smooth bandpass filter bank output

step toward a *time-frequency* analysis of the composite signal. That is, we have means of localizing the transient components of the signal in both time (or position) and frequency.

14.4.2 Multiresolution Analysis

Many of the developments preceding wavelet analysis came in a field generally called *multiresolution analysis*. These developments were intended to combat the limitations of the Fourier transform mentioned at the beginning of the chapter. We now summarize this approach as groundwork leading to modern wavelet analysis.

Filter bank theory offers a convenient means of representing signals composed of oscillatory components, such as musical notes and tone bursts. These components include several (or many) cycles of the oscillation within their duration. In image analysis, however, the localized components of interest often are not truly oscillatory, in that they include only one cycle or even just part of a cycle. Examples include lines, edges, and spots.

The objects in an image are observed to occur at different size scales. An edge, for example, can be either a sharp transition from black to white or one that occurs gradually over a considerable distance. In general, a multiresolution approach to image representation or analysis seeks to exploit this idea.

Cartography illustrates the approach. Maps are commonly drawn at different scales. The *scale* of a map is the ratio of the size of an actual territory to that of its representation on the map. At large scales, as on a globe, major features such as continents and oceans are visible, while details such as individual city streets fall below the resolution of the map. At smaller scales, the details become visible and the larger features are lost. Thus, to be able to navigate to a point at a distant location, one needs a set of maps drawn at different scales.

Wavelet transforms have developed along these multiresolution lines. As with time-frequency analysis, a signal is represented in a two-dimensional space, but here the vertical axis is scale rather than frequency. Scaling is achieved by dilating and contracting the basic wavelet to form a set of basis functions.

The basic wavelet, $\psi(x)$, is scaled as $\psi(x/a)$ (which is broadened if $a > 1$ and narrowed if $a < 1$) to form a set of basis functions. At large scale a, the dilated basis functions search for large features, while for small a, they seek out fine detail.

14.4.2.1 Pyramid Algorithms

Suppose we generated, from one 1,024-by-1,024–pixel digital image, 10 additional images by successively averaging 2-by-2–pixel blocks, each time discarding every second row and column of pixels. We would be left with images of 512 by 512, 256 by 256, etc., down to 1 by 1. If we then performed edge detection, for example, on each image, using one of the 3-by-3 edge detection operators mentioned in Chapter 18, we would find small edges in the original image, somewhat larger edges in the 512-by-512 and 256-by-256–pixel images, and only the very large edges in the 16-by-16–pixel and smaller images.

The Haar transform represents the dawn of this approach from almost a century ago. In its basis images (Figure 13–6), we see the concept of searching the image with edge detectors of different scales. The principle of binary dilation is evident there as well.

One might be tempted to observe that all edges, large and small, appear in the original 1,024-by-1,024–pixel image and that no change of resolution is required to locate them. The problem is that large edges—those manifesting a transition in gray level that spans a considerable distance—are difficult to detect with conventional (small) neighborhood operators such as the ones discussed in Chapter 18. One could scale the operators up to detect the larger edges, but it is more efficient to scale the image down. Using a large operator to search a high-resolution image for large edges is computational overkill.

Several forms of multiresolution analysis have been studied under different names over the years. It is only in recent years, however, that the basic similarity between multiresolution and filter bank approaches has been recognized, and these have been unified under the heading of wavelet transforms.

14.4.2.2 Laplacian Pyramid Coding

Burt and Adelson [10] introduced a pyramid coding scheme based on the Gaussian function. The image is lowpass filtered with a Gaussian impulse response, and the result is subtracted from the original image. The high-frequency detail in the image is retained in this difference image. The lowpass filtered image can then be subsampled without loss of detail. The process is illustrated as follows.

Let $f_0(i, j)$ be the original image, and let $g(i, j)$ be a Gaussian-shaped lowpass filter impulse response. Then, at each step of the encoding process, the image is decomposed into half-resolution low-frequency and full-resolution high-frequency components, $f_1(i, j)$ and $h_1(i, j)$, respectively for the first step, by

$$f_1(i, j) = [f_0 * g](2i, 2j) \quad \text{and} \quad h_1(i, j) = f_0(i, j) - [f_0 * g](i, j) \tag{32}$$

This process is iterated each time on the subsampled image. After n iterations of an N-by-N image, where $N = 2^n$, $f_n(i, j)$ is a single point. The encoded image pyramid consists of the $h_k(i, j)$'s and the final low-frequency image $f_n(i, j)$. This is shown in Figure 14–16.

Image decoding is done in the reverse order. *Upsampling* is the process of inserting zeros between sample points. Each subsampled image, $f_k(i, j)$, beginning with the last one, $f_n(i, j)$, is upsampled and interpolated by convolution with $g(i, j)$. Then the result is added to the next (previous) image $f_{k-1}(i, j)$, and the process is repeated on the resulting image. This reconstructs the original image without error [10].

Each $h_k(i, j)$ is the difference of two images obtained by convolving a single image with Gaussians of single and double width. This is equivalent to convolving the image with

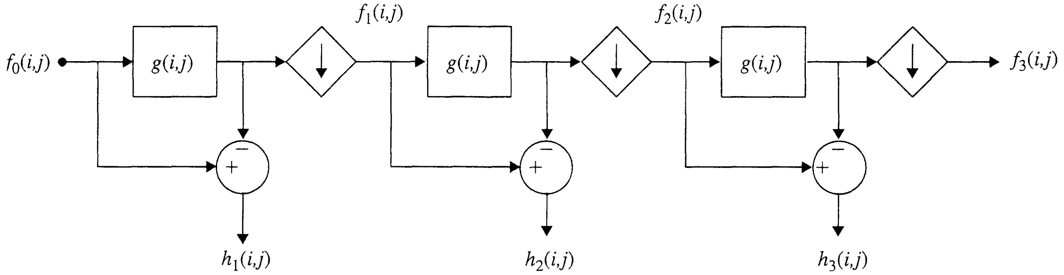

Figure 14–16 The Laplacian pyramid coding scheme

the difference of two Gaussians, which, in turn, approximates the "Laplacian of a Gaussian" highpass filter; hence the name chosen for this pyramid coding algorithm.

Although Laplacian pyramid coding increases the number of pixels required to represent the image by 33 percent, it can nevertheless accomplish a significant degree of image compression [10]. This occurs because the $h_k(i, j)$ images have significantly reduced correlation and dynamic range and are thus amenable to coarse quantization and even to setting some of the pixel values to zero. Further, the design of the Laplacian pyramid provided the inspiration that later led to the discrete wavelet transform.

14.4.3 Subband Coding

As further background leading to the discrete wavelet transform, we now describe a time-frequency technique called *subband coding.* Originally developed for compact coding of digitized audio signals, subband coding seeks to decompose a signal (or an image) into narrow-band (bandpass-filtered) components and represent these, without redundancy, in such a way that it is possible to reconstruct the original signal without error [11–13].

Given a bandlimited signal $f(t)$, that is,

$$\mathfrak{F}\{f(t)\} = F(s) = 0 \quad \text{for} \quad |s| \geq s_{\max} \tag{33}$$

we can sample the signal with uniform sample spacing Δt to form

$$f(i\Delta t) \quad i = 0, 1, \ldots, N-1 \qquad s_{\max} \leq s_N = \frac{1}{2\Delta t} \tag{34}$$

(Figure 14–17a), where s_N is the Nyquist (folding) frequency. (Recall Chapter 12, Eq. 22.)

We begin the analysis by partitioning the frequency axis into disjoint subintervals. While any subinterval length could be used, we now choose $s_N/2$, as shown in Figure 14–17b, for reasons that will become clear later. Here, the spectrum $F(s)$ is periodic with period $2s_N$.

14.4.3.1 The Lower Halfband

Figure 14–17b shows an ideal halfband lowpass filter, $h_0(i\Delta t)$, so called because it passes only the frequency band $[-s_N/2, s_N/2]$, which is the low-frequency half of the total frequency band $[-s_N, s_N]$. The impulse response and transfer function of h_0 are

$$h_0(t) = \text{sinc}\left(\pi\frac{t}{2\Delta t}\right) \quad \text{and} \quad H_0(s) = \Pi\left(\frac{s}{s_N}\right) \tag{35}$$

where the rectangular pulse is

$$\Pi(x) = \begin{cases} 1 & |x| < \frac{1}{2} \\ \frac{1}{2} & |x| = \frac{1}{2} \\ 0 & |x| > \frac{1}{2} \end{cases} \tag{36}$$

and

$$\text{sinc}(x) = \frac{\sin(x)}{x} \tag{37}$$

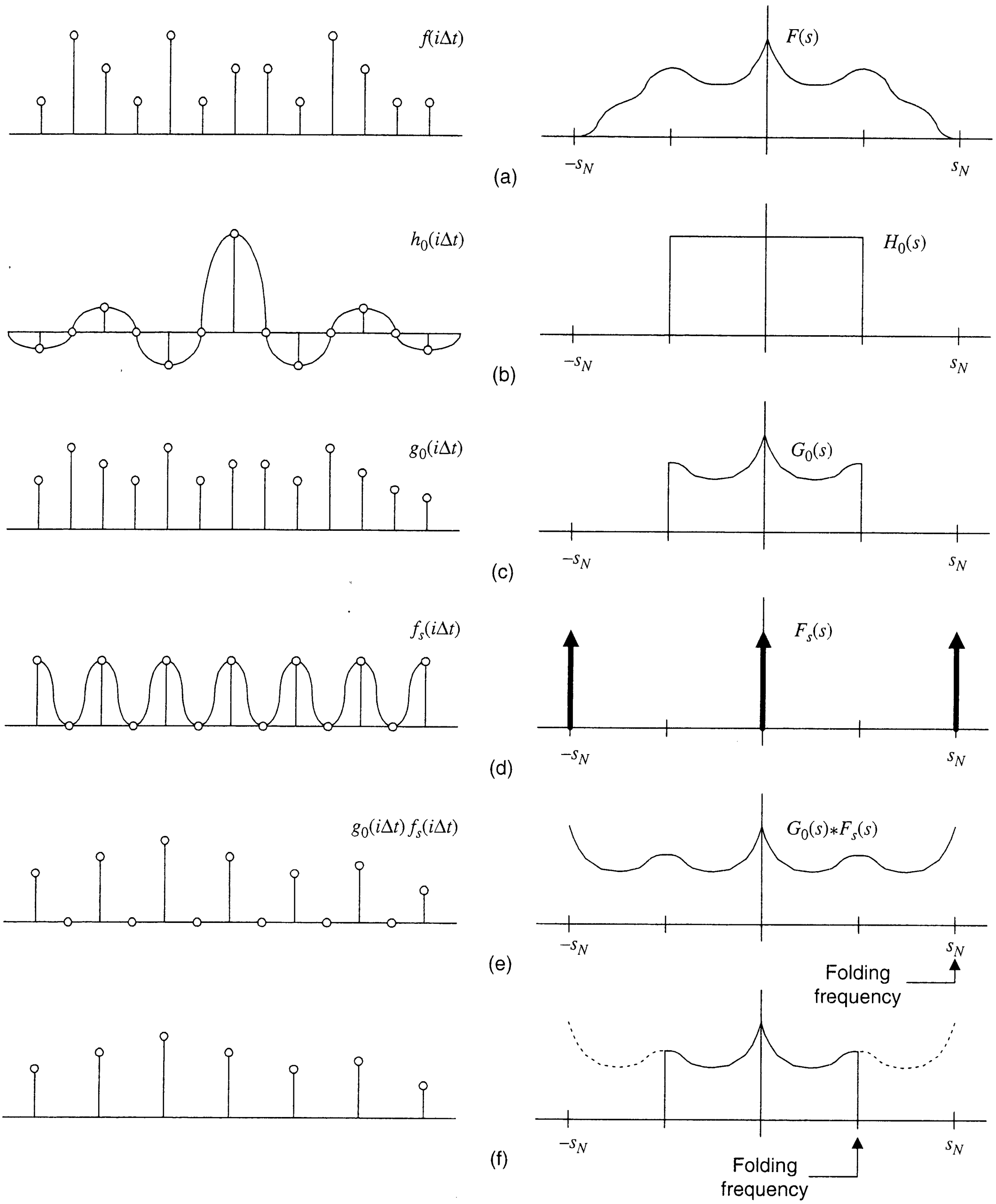

Figure 14–17 Subband coding, the lower halfband: (a) a sampled signal and its bandlimited spectrum; (b) the ideal halfband lowpass filter; (c) the lowpass filtered signal; (d) the subsampling function; (e) odd sample points replaced with zeros; (f) odd sample points discarded

Applying this filter to $f(i\Delta t)$ (Figure 14–17a) yields the signal $g_0(i\Delta t)$ (Figure 14–17c), which is bandlimited at $s = s_N/2$. This is a low-resolution (blurred) version of $f(i\Delta t)$. It retains the basic shape of $f(i\Delta t)$, but has lost the details.

Since $g_0(i\Delta t)$ has no energy above $s_N/2$, it could be sampled with sample spacing as large as $2\Delta t$ without introducing aliasing. In fact, we can discard every second sample and represent g_0 with only the remaining $N/2$ samples (Figure 14–17f). This process is called *subsampling* or *decimation.*

We can model subsampling as first multiplying the signal by a subsampling function that drives the odd-numbered samples to zero and then discarding the odd-numbered samples. Such a subsampling function

$$f_s(i\Delta t) = \frac{1}{2}[1 + \cos(2\pi s_N i\Delta t)] \tag{38}$$

and its spectrum

$$F_s(s) = \frac{1}{2}[\delta(s) + \delta(s - s_N) + \delta(s + s_N)] \tag{39}$$

are shown in Figure 14–17d.

When we multiply the signal $g_0(i\Delta t)$ by $f_s(i\Delta t)$, we convolve its spectrum with $F_s(s)$. The result is to make the spectrum symmetric in such a way that its period is reduced from $2s_N$ to s_N, as shown in Figure 14–17e. Its amplitude is also cut in half; we write

$$F_s(s) * G_0(s) = \frac{1}{2}G_0(s) + \frac{1}{2}G_0(s + s_N) + \frac{1}{2}G_0(s - s_N) \tag{40}$$

Clearly, we can now discard the odd-numbered sample points without loss of information (Figure 14–17f). This reduces the folding frequency to $s_N/2$ and leaves us with a signal that is properly sampled with sample spacing $2\Delta t$.

No information has been lost in the process of subsampling $g_0(i\Delta t)$. To see this, notice that we could recover $g_0(i\Delta t)$ from the subsampled signal in Figure 14–17f simply by (1) computing its ($N/2$-point) discrete spectrum, (2) padding it with zeros from $s_N/2$ to s_N to reconstruct $G_0(s)$ (Figure 14–17c), and (3) taking the inverse (N-point) DFT of $G_0(s)$ to reconstruct $g_0(i\Delta t)$, the signal shown in Figure 14–17c. While this is not the preferred method, it argues that subsampling $g_0(i\Delta t)$ produces no loss of information.

A simpler way to recover $g_0(i\Delta t)$ can also be seen in Figure 14–17. We first upsample the encoded lowband signal (Figure 14–17f) by inserting the zero-valued odd-numbered samples (to form Figure 14–17e). Then we filter that signal with $2h_0(i\Delta t)$, the ideal halfband lowpass filter (Figure14–17b). This will reconstruct the spectrum, and hence the signal, in Figure 14–17c, thereby recovering $g_0(i\Delta t)$.

In the frequency domain, we write

$$F_s(s) * G_0(s) \times H_0(s) =$$

$$\left[\frac{1}{2}G_0(s) + \frac{1}{2}G_0(s + s_N) + \frac{1}{2}G_0(s - s_N)\right] \times \Pi\left(\frac{s}{s_N}\right) = \frac{1}{2}G_0(s) \tag{41}$$

Notice that the lowpass filter impulse response, $h_0(i\Delta t)$, is $\operatorname{sinc}(\pi x/2\Delta t)$, which has zero-crossings at even multiples of the sample spacing, except at zero (Figure14–17b). Thus, it interpolates the intermediate (odd-numbered) values of $g_0(i\Delta t)$, where the zeros are located, and leaves the even-numbered samples alone.

14.4.3.2 The Upper Halfband

Turning now to the upper halfband of $f(i\Delta t)$ (Figure 14–18a), we can isolate the energy there with an ideal halfband bandpass filter (Figure 14–18b). This filter's impulse response and transfer function are, respectively,

$$h_1(t) = \delta(t) - \mathrm{sinc}\left(\pi\frac{t}{2\Delta t}\right) \quad \text{and} \quad H_1(s) = 1 - \Pi\left(\frac{s}{s_N}\right) \tag{42}$$

where $\Pi(x)$ is as in Eq. (36).

The filter produces a signal, $g_1(i\Delta t)$, whose spectrum is nonzero only in the upper halfband (Figure 14–18c). This signal contains exactly the high-frequency information that was eliminated from $f(i\Delta t)$ by the lowpass filter in Figure 14–17b. Thus, $g_0(i\Delta t)$ and $g_1(i\Delta t)$, taken together, contain all the information that was present in the original signal, $f(i\Delta t)$. In fact,

$$f(i\Delta t) = g_0(i\Delta t) + g_1(i\Delta t) = f(i\Delta t) * h_0(i\Delta t) + f(i\Delta t) * h_1(i\Delta t) \tag{43}$$

since

$$H_0(s) + H_1(s) = 1 \tag{44}$$

Figure 14–18d shows the subsampling function $f_s(i\Delta t)$ that was used in the analysis in Sec. 14.4.3.1. When $g_1(i\Delta t)$ (Figure14–18c) is subsampled by $f_s(i\Delta t)$, its spectrum is convolved with $F_s(s)$. This fills the interval $[-s_N/2, s_N/2]$ with a replicate of its spectrum and produces the spectrum shown in Figure 14–18e. We write

$$F_s(s) * G_1(s) = \frac{1}{2}G_1(s) + \frac{1}{2}G_1(s+s_N) + \frac{1}{2}G_1(s-s_N) \tag{45}$$

This spectrum is now periodic with period $s_N/2$ and could be sampled at spacing $2\Delta t$ without aliasing. Thus, we now have another signal that is confined to the lower halfband, and it can be subsampled as before (Figure 14–18f).

This leaves the N-point signal $f(i\Delta t)$ encoded into two $N/2$-point signals. We have seen that $g_0(i\Delta t)$ can be recovered from the encoded lowband signal. It remains only to show that $g_1(i\Delta t)$ can be recovered from the encoded highband signal to see that $f(i\Delta t)$ [and hence $f(t)$] can be reconstructed without error.

Figure 14–18e shows the upsampled highband signal. Its spectrum is identical to that in Figure 14–18f, except that, after upsampling, the folding frequency is once again s_N. We can reconstruct $G_1(s)$, and thus $g_1(i\Delta t)$ (Figure 14–18c), simply by filtering this upsampled signal with $2h_1(i\Delta t)$ (Figure 14–18b) to eliminate the low-frequency energy. We write

$$F_s(s) * G_1(s) \times H_1(s) =$$
$$\left[\frac{1}{2}G_1(s) + \frac{1}{2}G_1(s+s_N) + \frac{1}{2}G_1(s-s_N)\right] \times \left[1 - \Pi\left(\frac{s}{s_N}\right)\right] = \frac{1}{2}G_1(s) \tag{46}$$

Thus, we have, in two-channel subband coding, an invertible representation of the signal in terms of two subsampled discrete filter outputs, and it is without redundancy (i.e., not overcomplete).

14.4.3.3 Aliasing the Upper Halfband

Clearly, subsampling $g_1(i\Delta t)$ by discarding every other sample point will result in aliasing. The energy at frequencies between $s_N/2$ and s_N will be aliased down to the interval $[0, s_N)$, as indicated in Figure 14–19a. This process, however, is nondestructive, since that interval is

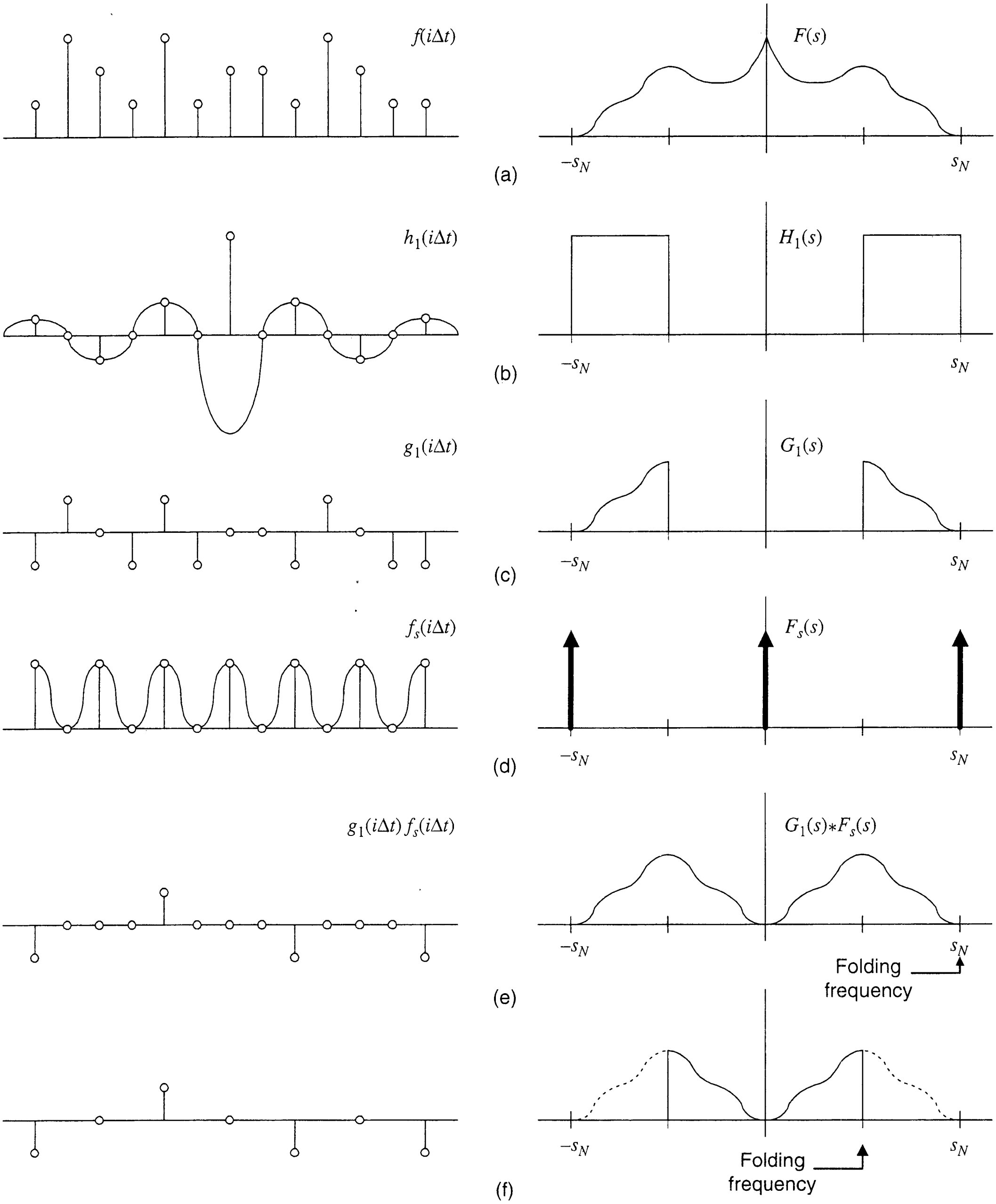

Figure 14–18 Subband coding, the upper halfband: (a) a sampled signal and its bandlimited spectrum; (b) the ideal halfband highpass filter; (c) the highpass filtered signal; (d) the subsampling function; (e) odd sample points replaced with zeros; (f) odd sample points discarded

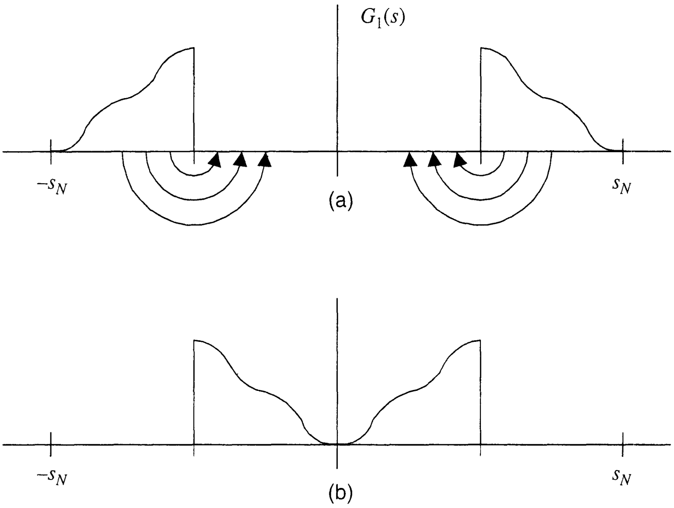

Figure 14–19 Aliasing the upper halfband: (a) aliasing the spectrum in Figure 14–18(c); (b) the result of aliasing

already vacant. It produces the spectrum shown in Figure 14–19b, which is bandlimited at $s_N/2$ and contains all the energy of $g_1(i\Delta t)$.

Ironically, aliasing, the bugaboo that usually threatens our ability to process continuous signals and images digitally, now comes to our aid. It is only necessary to subsample $g_1(i\Delta t)$ to obtain the upper subband coded signal. Furthermore, upsampling followed by filtering with $2h_1(i\Delta t)$, recovers $g_1(i\Delta t)$.

14.4.3.4 Subband Coding and Decoding

Two-channel subband coding, then, requires only filtering $f(i\Delta t)$ with $h_0(i\Delta t)$ and $h_1(i\Delta t)$, followed by subsampling each output. This yields the two half-length subband signals

$$g_0(k\Delta t) = \sum_i f(i\Delta t)h_0((-i+2k)\Delta t) \tag{47}$$

and

$$g_1(k\Delta t) = \sum_i f(i\Delta t)h_1((-i+2k)\Delta t) \tag{48}$$

Reconstruction is effected by upsampling the lower and upper subband signals, interpolating them with $2h_0(i\Delta t)$ and $2h_1(i\Delta t)$, respectively, and adding them together. This is given by

$$f(i\Delta t) = 2\sum_k [g_0(k\Delta t)h_0((-i+2k)\Delta t) + g_1(k\Delta t)h_1((-i+2k)\Delta t)] \tag{49}$$

and is illustrated in Figure 14–20.

We have a slight problem at the midfrequency point $s = s_N/2$, since encoding and decoding entails filtering $f(i\Delta t)$ twice, once with $h_0(i\Delta t)$ and once with $h_1(i\Delta t)$, and since $H_0(s_N/2) = 1/2$ and $H_1(s_N/2) = 1/2$. This problem could be avoided by using $\Pi(\pm\frac{1}{2}) = \sqrt{\frac{1}{2}}$ in Eq. (36). In the next section, where we use more general bandpass filters, we handle the situation explicitly.

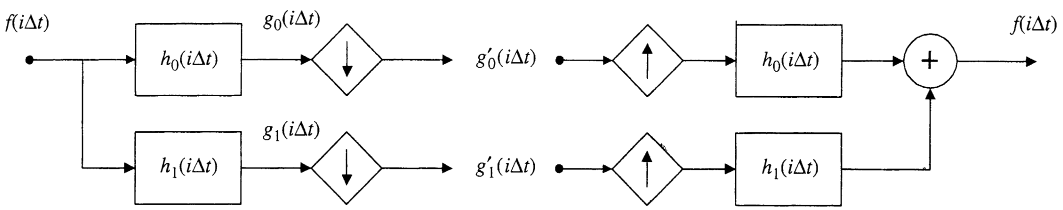

Figure 14–20 Two-band subband coding and reconstruction

We could have chosen to partition the frequency axis into M shorter intervals of length $2s_N/M$, producing M subband signals of N/M points each, as is commonly done in subband coding. Different frequency components then show up in separate subband channels. Since we are moving toward the DWT, however, we stick with the choice of two subbands ($M = 2$).

14.4.4 The Fast Wavelet Transform Algorithm

Mallat [14] defined a discrete wavelet transform algorithm that is more efficient than computing a full set of inner products. It applies two-band subband coding in an iterative fashion and builds the wavelet transform from the bottom up, that is, computing small-scale coefficients first.

After the first step of subband coding, as outlined in Sec. 14.4.3, the lower subband signal, $g_0(i\Delta t)$, is once again subjected to halfband subband coding. This leaves us with the $N/2$-point upper halfband signal and two $N/4$-point subband signals corresponding to the first and second quarters of the interval $[0, s_N]$.

The process is continued, at each step retaining the upper halfband signal and further encoding the lower halfband signal, until a one-point lowband signal is obtained. The transform coefficients are then the lowband point and the collection of subband-coded upper halfband signals. This is shown in Figure 14–21. The first $N/2$ coefficients come from the upper halfband of $F(s)$, the next $N/4$ points from the second quarterband, etc.

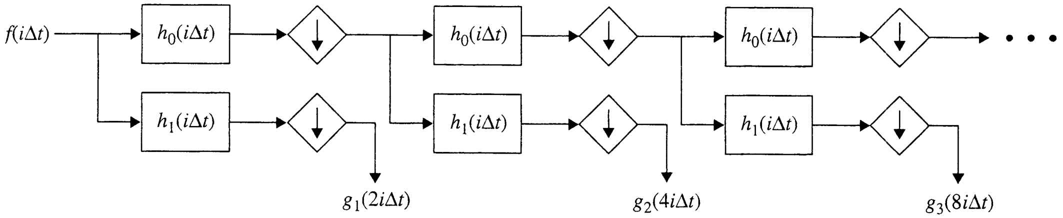

Figure 14–21 The discrete wavelet transform algorithm

The impulse response, h_j, doubles in scale at each iteration. Thus, we have an orthonormal wavelet transform. The basic wavelet is $h(t) = \delta(t) - \text{sinc}(at)$, and the basis functions are $\{2^{-j/2}h(2^jt - n)\}$ [15]. Thus, subband coding, which is basically a time-frequency transform technique, has been employed to define a time-scale wavelet transform.

The foregoing algorithm is sometimes referred to as the *fast wavelet transform* (FWT), or Mallat's *herringbone algorithm,* due to the appearance of the diagram in Figure 14–21. The inverse transform is obtained by reversing the process, as shown in Figure 14–22.

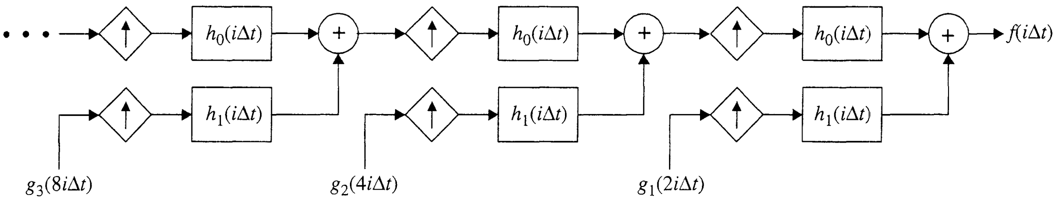

Figure 14–22 The inverse discrete wavelet transform

14.4.4.1 Basis Functions

We see in Figure 14–21 that each set of transform coefficients is obtained by convolving $f(i\Delta t)$ repeatedly with $h_0(i\Delta t)$ and then once with $h_1(i\Delta t)$. Thus, the basis functions of this wavelet transform are $h_1(i\Delta t)$ and other functions derived by convolving $h_1(i\Delta t)$ repeatedly with $h_0(i\Delta t)$. This is explored in more detail next.

14.4.5 Discrete Wavelet Transform Design

We are now prepared to approach the design of a basic wavelet for use in a discrete wavelet transform. As we saw in Sec. 14.4.1, it is not necessary that the filters in a filter bank implementation be ideal lowpass and bandpass filters. Similarly, for the DWT, we can use any pair of subband coding filters that allows Eq. (49) to hold.

Writing Eq. (49) in the frequency-domain, we have

$$\begin{aligned} F(s) &= 2\left[\frac{1}{2}G_0(s)H_0(s) + \frac{1}{2}G_1(s)H_1(s)\right] \\ &= 2\left[\frac{1}{2}F(s)H_0(s)H_0(s) + \frac{1}{2}F(s)H_1(s)H_1(s)\right] \end{aligned} \tag{50}$$

which means that

$$F(s) = F(s)\,[H_0^2(s) + H_1^2(s)] \tag{51}$$

and the two filter transfer functions must satisfy the condition

$$H_0^2(s) + H_1^2(s) = 1 \quad \text{for} \quad 0 \le |s| \le s_N \tag{52}$$

The transfer functions are squared here because $f(t)$ is convolved twice with each filter, once during coding and once during decoding. This resolves the problem that was noted in Sec. 14.4.3.4.

Suppose $H_0(s)$ is a smooth-edged lowpass transfer function that we wish to use in a wavelet transform. Clearly, the corresponding $H_1(s)$ is given by

$$H_1^2(s) = 1 - H_0^2(s) \tag{53}$$

Thus, a well-selected lowpass filter is all that is required to design a discrete wavelet transform.

14.4.5.1 Mirror Filters

Comparing Figures 14–17b and 14–18b, we see that, for the case of the ideal bandpass filter, $h_1(i\Delta t)$ can be viewed as $h_0(i\Delta t)$, shifted by an amount s_N along the frequency axis. According to the shift theorem (Sec. 10.2.3)

$$\mathfrak{F}^{-1}\{H(s-a)\} = e^{j2\pi a t}h(t) \Rightarrow \mathfrak{F}^{-1}\{H(s-s_N)\}$$
$$= e^{j2\pi\left(\frac{1}{2\Delta t}\right)i\Delta t}h(i\Delta t) = (-1)^i h(i\Delta t) \tag{54}$$

and such a half-period shift of the spectrum can be effected simply by changing the sign of the odd-numbered samples of the signal. Note the use of the imaginary unit in Eq. (54).

We can use this approach in the design of more general subband filters. Selecting $h_1(i\Delta t)$ so that

$$h_1((N-1-i)\Delta t) = (-1)^i h_0(i\Delta t) \tag{55}$$

where N is the length of $h_0(i\Delta t)$, we obtain the corresponding highpass filter. The filter $h_1(i\Delta t)$ is called the *mirror filter* of $h_0(i\Delta t)$. If $h_0(i\Delta t)$ is of short duration, we can be assured that $h_1(i\Delta t)$ will be short as well.

The symmetry property that $H_0(s)$ must have in order for Eq. (53) to hold, and for this entire approach to work, is

$$H_0^2\left(\frac{s_N}{2}+s\right) = 1 - H_0^2\left(\frac{s_N}{2}-s\right) \tag{56}$$

14.4.5.2 The Scaling Vector

To develop a discrete wavelet transform, then, we need only a discrete lowpass filter impulse response $h_0(k)$ that meets certain conditions [16]. This impulse response is sometimes called a *scaling vector.*

From $h_0(k)$ we can generate a related function $\phi(t)$, called the *scaling function.* We can also generate $h_1(k)$ and, from it and $\phi(t)$, the basic wavelet, $\psi(t)$. If the scaling vector has only a finite number of nonzero entries, then $\phi(t)$, $\psi(t)$, and the resulting wavelets will all have compact support [16]. That is, they will be zero outside a relatively short interval on the t-axis.

Actually, if we have either $h_0(k)$ or $\phi(t)$, we can use it to generate the other. It is usually easier to start with $h_0(k)$, which must satisfy Eq. (56). Let the scaling vector be a sequence such that

$$\sum_k h_0(k) = \sqrt{2} \quad\text{and}\quad \sum_k h_0(k)h_0(k+2l) = \delta(l) \tag{57}$$

Then there exists a scaling function

$$\phi(t) = \sum_k h_0(k)\phi(2t-k) \tag{58}$$

that can be built as a weighted sum of half-scale copies of itself, using $h_0(k)$ as the weights. From the observation in Sec. 14.4.4.1, $\phi(t)$ can also be computed numerically [16] by repeated convolution of $h_0(k)$ with scaled versions of the rectangular pulse function (Figure 14–23); that is,

$$\phi(x) = \lim_{i\to\infty}\eta_i(x) \tag{59}$$

where

$$\eta_i(x) = \sqrt{2}\sum_n h_0(n)\eta_{i-1}(2x-n) \tag{60}$$

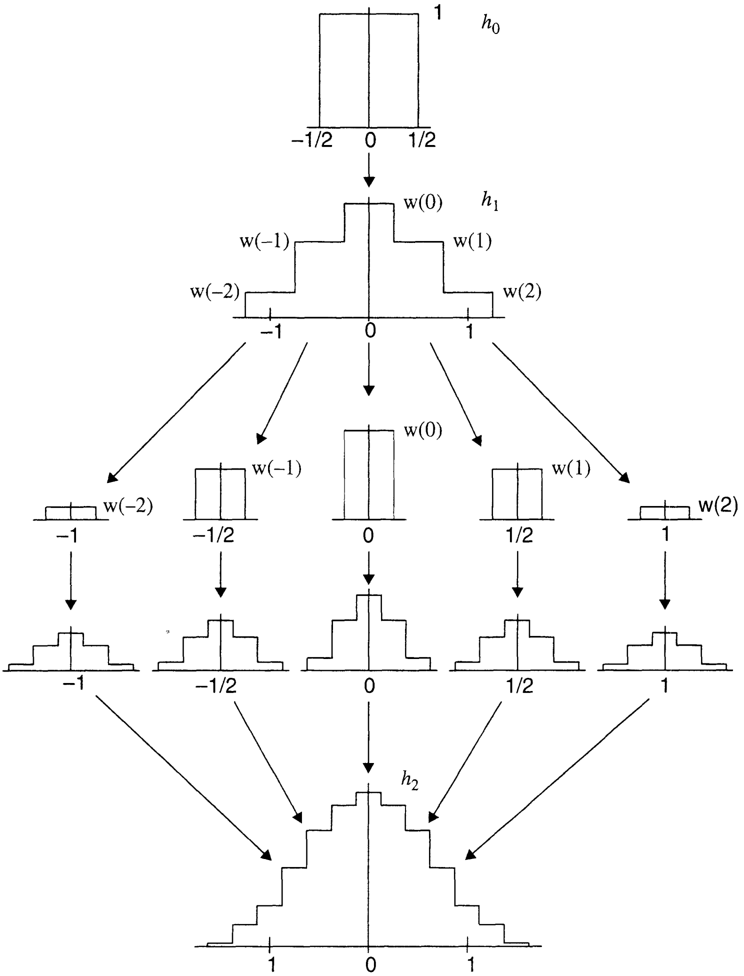

Figure 14–23 Constructing the scaling function (after Daubechies [16])

is a piecewise constant approximation to $\phi(t)$ and

$$\eta_0(x) = \Pi(x) = \begin{cases} 1 & |x| < \frac{1}{2} \\ \frac{1}{2} & |x| = \frac{1}{2} \\ 0 & |x| > \frac{1}{2} \end{cases} \tag{61}$$

Notice that the first iteration creates a piecewise constant function having the values of $h_0(k)$. Further, the resolution of the approximation doubles, and the approximation becomes smoother, with each iteration. Nine iterations, for example, will take a four-point sequence into a 1,024-point sampled function, and this is adequate for most digital implementations. The scaling function $\phi(t)$ is, then, a continuous function that has the same general shape as the discrete lowpass filter impulse response $h_0(k)$.

If, on the other hand, we start with a scaling function $\phi(t)$, it must be orthonormal under unit shifts; that is,

$$\langle \phi(t-m), \phi(t-n) \rangle = \delta_{m,n} \tag{62}$$

Then $h_0(k)$ can be computed from

$$h_0(k) = \langle \phi_{1,0}(t), \phi_{0,k}(t) \rangle \tag{63}$$

where

$$\phi_{j,k}(t) = 2^{j/2}\phi(2^j t - k) \qquad j = 0, 1, \ldots \qquad k = 0, 1, \ldots, 2^j - 1 \tag{64}$$

If the scaling vector has only a finite number of nonzero entries, then the resulting wavelets will have compact support [16].

If a desired scaling function $\hat{\phi}(t)$ is not orthonormal, it can be used to generate one that is orthonormal by proper normalization of its spectrum, $\hat{\Phi}(s)$ [7]. That is,

$$\Phi(s) = \frac{C\hat{\Phi}(s)}{\sqrt{\sum_{n=-\infty}^{\infty} \hat{\Phi}(s - 2\pi n)}} \tag{65}$$

where C is a constant.

14.4.5.3 The Wavelet Vector

Once we have both $\phi(t)$ and $h_0(k)$ in hand, we continue the development by defining a discrete highpass impulse response called the *wavelet vector* as

$$h_1(k) = (-1)^k h_0(-k+1) \tag{66}$$

and, from that, a basic wavelet

$$\psi(t) = \sum_k h_1(k)\phi(2t - k) \tag{67}$$

from which an orthonormal wavelet set

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k) \tag{68}$$

follows.

14.4.5.4 Computing the Wavelet Transform

Given the set of orthonormal wavelets, the wavelet series expansion of the bandlimited continuous function $f(t)$ is

$$c_{j,k} = \int_{-\infty}^{\infty} f(t)\psi_{j,k}(t)dt \quad \text{and} \quad f(t) = \sum_{j,k} c_{j,k}\psi_{j,k}(t) \tag{69}$$

and the discrete wavelet transform of the sampled function is

$$c_{j,k} = \sum_i f(i\Delta t)\psi_{j,k}(i\Delta t) \quad \text{and} \quad f(i\Delta t) = \sum_{j,k} c_{j,k}\psi_{j,k}(i\Delta t) \tag{70}$$

The coefficients and summations can also be indexed by the single integer $n = 0, 1, \ldots, N-1$, where

$$n = 2^j + k \quad \text{for} \quad j = 0, 1, \ldots, \log_2(N) - 1 \qquad k = 0, 1, \ldots, 2^j - 1 \tag{71}$$

We refer to this as the *top-down* algorithm, since it computes large-scale coefficients first. By contrast, Mallat's herringbone algorithm computes small-scale coefficients first.

The design task, then, involves first finding a sequence $h_0(k)$ that satisfies Eq. (56) and then constructing the corresponding scaling function, or choosing an orthonormal scaling function and determining $h_0(k)$ from Eq. (63). A scaling function can be made orthonormal by Eq. (65). Then the wavelet vector, $h_1(k)$, is obtained from Eq. (66) and the basic wavelet from Eq. (67).

The discrete wavelet transform can be implemented either directly, by Eq. (70), or with the FWT herringbone algorithm. The latter does not require explicit construction of the scaling function and wavelet, and it is more computationally efficient.

To be mathematically precise, the conditions in Eq. (57) establish that the wavelets $\{\psi_{j,k}(t)\}$ constitute a *tight frame* and thus will support an invertible transform. They are not, however, adequate to guarantee that these basis functions will always be orthonormal. Lawton [17,18] and Cohen [19] give strict orthonormality conditions on $h_0(k)$, but the differences between a tight frame and an orthonormal transform are so slight that digital implementations are not affected. Thus, we can be satisfied using Eq. (57).

14.4.5.5 Examples

We illustrate the construction of a wavelet transform with three examples.

Example 1. Using ideal lowpass and bandpass filters, we have [15]

$$h_0(k) = \frac{1}{\sqrt{2}}\operatorname{sinc}\left(\pi\frac{k}{2}\right) \quad \text{and} \quad h_1(k) = \sqrt{2}\delta(k) - h_0(k) \tag{72}$$

and

$$\phi(t) = \operatorname{sinc}(\pi t) \quad \text{and} \quad \psi(t) = 2\phi(2t) - \phi(t) \tag{73}$$

This gives a discrete wavelet transform based on *sinc wavelets* (Figure 14–11). Notice that these wavelets do not have compact support.

Example 2. If we let

$$h_0(k) = \begin{cases} \frac{1}{\sqrt{2}} & k = 0, 1 \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad h_1(k) = \begin{cases} \frac{1}{\sqrt{2}} & k = 0 \\ \frac{-1}{\sqrt{2}} & k = 1 \\ 0 & \text{otherwise} \end{cases} \tag{74}$$

then $\psi(t)$ is the Haar function, and we are led to the Haar transform. This scaling vector has two nonzero entries, and, as expected, the Haar transform does have compact support.

Example 3. The sequence $h_0(k)$, having four nonzero elements and given by

$$4\sqrt{2}h_0(k) = \begin{cases} (1+\sqrt{3}) & k=0 \\ (3+\sqrt{3}) & k=1 \\ (3-\sqrt{3}) & k=2 \\ (1-\sqrt{3}) & k=3 \\ 0 & \text{otherwise} \end{cases} \tag{75}$$

satisfies Eq. (56) and is thus a scaling vector [16]. Its scaling function and wavelet, constructed by the procedure outlined in Figure 14–23, are shown in Figure 14–24. This is one of a family of finite-length sequences that give rise to orthonormal wavelets having compact support. The family is discussed further in the following section.

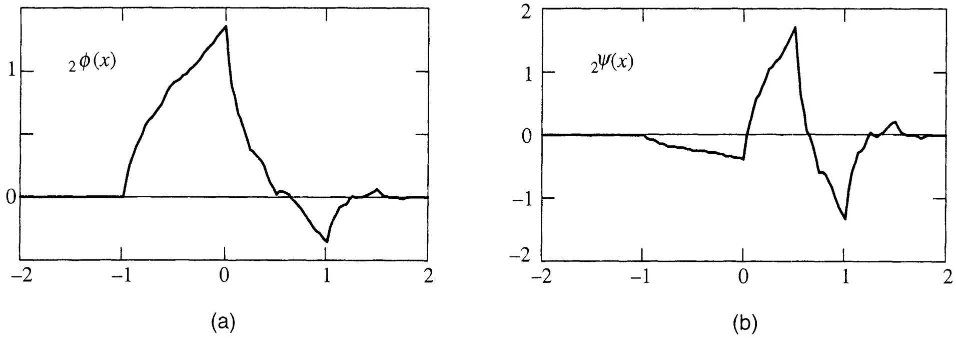

Figure 14–24 Daubechies' (a) scaling function and (b) wavelet for $r = 2$ (after [16])

14.4.5.6 Orthonormal Wavelets with Compact Support

Daubechies [16] has constructed a family, $\{{}_r\psi(x)\}$, of orthonormal wavelets having compact support. For each integer value of the index r, the set of wavelets

$$\{{}_r\psi_{j,k}(x)\} = \{2^{j/2}{}_r\psi(2^j x - k)\} \tag{76}$$

where j and k are integers, forms an orthonormal basis. Further, ${}_r\psi(x)$ is zero outside the interval $[0,2r-1]$, its first r moments vanish, that is,

$$\int_{-\infty}^{\infty} x^n {}_r\psi(x)dx = 0 \qquad n = 0, 1, \ldots, r \tag{77}$$

and its number of continuous derivatives is approximately r/5. This describes a rather well-behaved, or *regular,* group of functions. Interestingly, ${}_1\psi(x)$ is the basic wavelet of the Haar transform.

Table 14–1 shows the sequences $h_0(k)$ that generate the orthonormal wavelets for $r = 3,5,7$, and 9. The construction technique was outlined in Sec. 14.4.5. Figure 14–25 shows plots of the corresponding wavelets. Notice that these functions become both broader and more regular with increasing r.

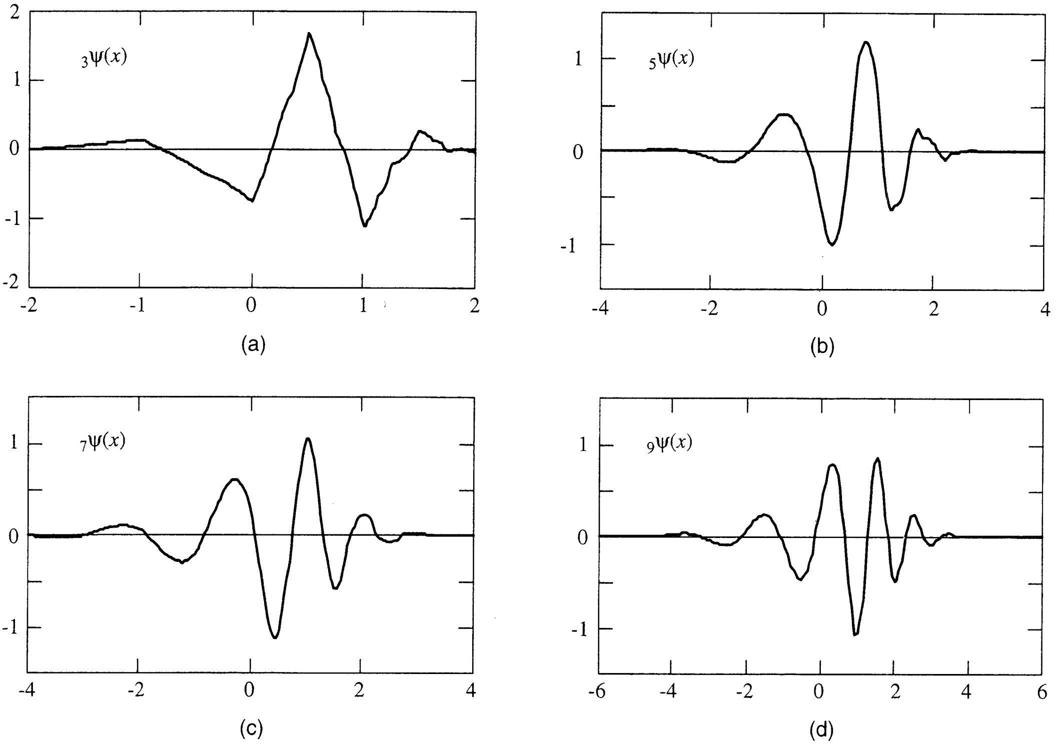

Figure 14–25 Orthonormal wavelets for (a) $r = 3$, (b) $r = 5$, (c) $r = 7$, and (d) $r = 9$ (after [16])

TABLE 14–1 DISCRETE FILTER SEQUENCES FOR THE ORTHONORMAL WAVELETS IN FIGURE 14–25 ($r = 3$, 5, 7 AND 9; FROM [16])

.3327	.8069	.4599	−.1350	−.0854	.0352				
.1601	.6083	.7243	.1384	−.2423	−.0322	.0776	−.0062	−.0126	.0033
.0779	.3965	.7291	.4698	−.1439	−.2240	.0713			
.0806	−.0380	−.0166	.0126	.0004	−.0018	.0004			
.0381	.2438	.6048	.6573	.1332	−.2933	−.0968	.1485	.0307	
−.0676	.0003	.0224	−.0047	−.0043	.0018	.0002	−.0003	.0000	

14.4.6 The Two-Dimensional Discrete Wavelet Transform

The concepts developed for the representation of one-dimensional signals generalize easily to two dimensions [5,7,14,16]. As with unitary image transforms, we consider the case where the two-dimensional scaling function is *separable;* that is,

$$\phi(x, y) = \phi(x)\phi(y) \tag{78}$$

where $\phi(x)$ is a one-dimensional scaling function. If $\psi(x)$ is its companion wavelet (Eq. 67), then the three two-dimensional basic wavelets

$$\psi^1(x, y) = \phi(x)\psi(y) \qquad \psi^2(x, y) = \psi(x)\phi(y) \qquad \psi^3(x, y) = \psi(x)\psi(y) \tag{79}$$

establish the foundation for a two-dimensional wavelet transform. Note that the superscript is used here as an index rather than an exponent. In particular, the set of functions

$$\{\psi^l_{j,m,n}(x,y)\} = \{2^j\psi^l(x-2^jm, y-2^jn)\} \qquad j \geq 0 \qquad l = 1,2,3 \tag{80}$$

where j, l, m, and n are integers, is an orthonormal basis for $L^2(R^2)$.

14.4.6.1 The Forward Transform

We begin with an N-by-N image, $f_1(x, y)$, where the subscript indicates scale and N is a power of two. For $j = 0$, the scale is $2^j = 2^0 = 1$, and this is the scale of the original image. Each larger integer value of j doubles the scale and halves the resolution. Some of the literature uses j to index the resolution rather than the scale. In that case $j \leq 0$, and its sign in the equations that follow is reversed.

The image can be expanded in terms of the two-dimensional wavelets as follows. At each stage of the transform, the image is decomposed into four quarter-size images, as shown in Figure 14–26. Each of the four images is formed by inner products with one of the wavelet basis images, followed by subsampling in x and y by a factor of two. For the first stage ($j = 1$), we write

$$\begin{aligned} f_2^0(m,n) &= \langle f_1(x,y), \phi(x-2m, y-2n)\rangle \\ f_2^1(m,n) &= \langle f_1(x,y), \psi^1(x-2m, y-2n)\rangle \\ f_2^2(m,n) &= \langle f_1(x,y), \psi^2(x-2m, y-2n)\rangle \\ f_2^3(m,n) &= \langle f_1(x,y), \psi^3(x-2m, y-2n)\rangle \end{aligned} \tag{81}$$

For subsequent stages ($j > 1$), $f^0_{2^j}(x, y)$ is decomposed in exactly the same way to form four smaller images at scale 2^{j+1} [Figure 14–26(c)]. The final result is an arrangement like that of the Haar transform, as shown in Figure 14–26(d).

Writing the inner products as convolutions, we have

$$\begin{aligned} f^0_{2^{j+1}}(m,n) &= \left\{\left[f^0_{2^j}(x,y) * \phi(-x,-y)\right](2m,2n)\right\} \\ f^1_{2^{j+1}}(m,n) &= \left\{\left[f^0_{2^j}(x,y) * \psi^1(-x,-y)\right](2m,2n)\right\} \\ f^2_{2^{j+1}}(m,n) &= \left\{\left[f^0_{2^j}(x,y) * \psi^2(-x,-y)\right](2m,2n)\right\} \\ f^3_{2^{j+1}}(m,n) &= \left\{\left[f^0_{2^j}(x,y) * \psi^3(-x,-y)\right](2m,2n)\right\} \end{aligned} \tag{82}$$

and the same four subsampled filtering operations are required at each stage.

Since the scaling and wavelet functions are separable, each convolution breaks down into one-dimensional convolutions on the rows and columns of $f^0_{2^j}(x, y)$. Figure 14–27 shows this in diagram form.

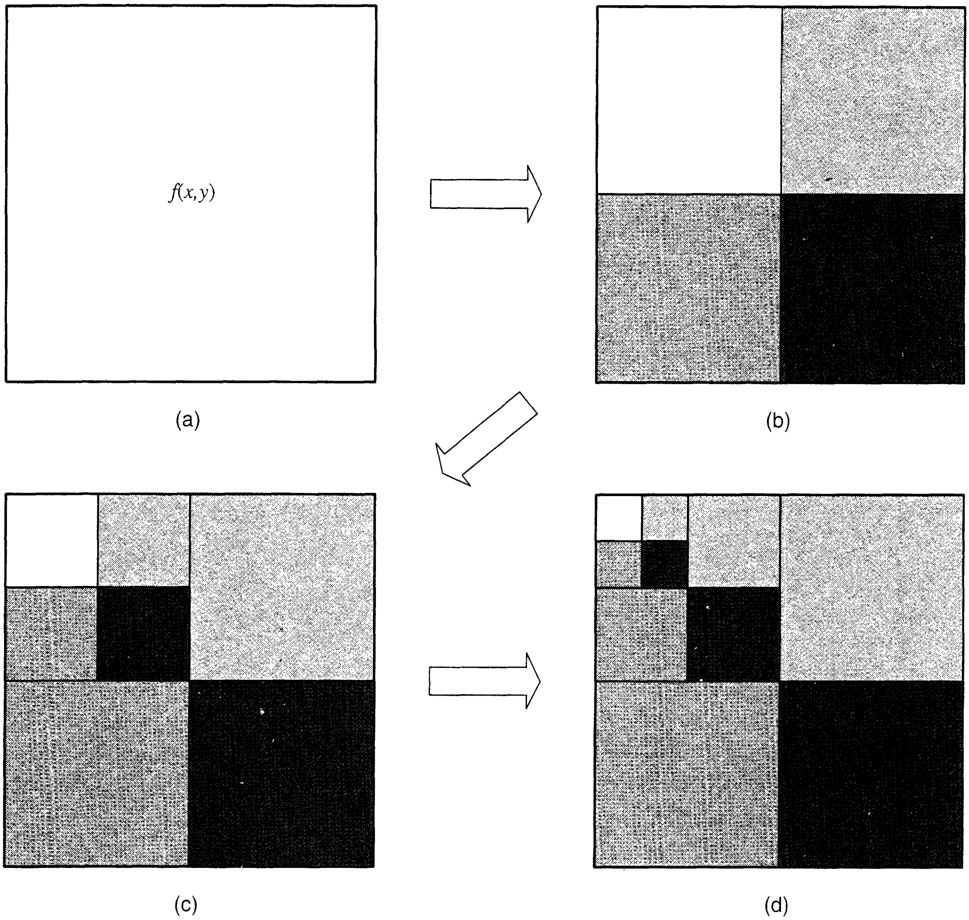

Figure 14–26 The two dimensional discrete wavelet transform: (a) original; (b) first, (c) second, (d) third step

At stage 1, for example, we first convolve the rows of the image $f_1(x, y)$ with $h_0(-x)$ and with $h_1(-x)$ and discard the odd-numbered columns (counting the leftmost as zero) of the two resulting arrays. The columns of each of the $N/2$-by-N arrays is then convolved with $h_0(-x)$ and with $h_1(-x)$, and the odd-numbered rows are discarded (counting the top row as zero). The result is the four $N/2$-by-$N/2$ arrays required for that stage of the transform.

The two-dimensional separable wavelet transform thus can be computed quickly. The transform process can be carried to J stages, where the integer $J \leq \log_2(N)$ for an N-by-N pixel image. If the transform coefficients are computed with floating-point accuracy, the inverse transform can reconstruct the original image with little degradation.

Figure 14–28 shows from where in the frequency plane each of the four next-higher scale images come, if we were to use sinc wavelets (that is, ideal halfband lowpass and bandpass filters). At each scale, $f^0_{2^j}(x, y)$ contains the low-frequency information from the previous stage, while $f^1_{2^j}(x, y)$, $f^2_{2^j}(x, y)$, and $f^3_{2^j}(x, y)$ contain the horizontal, vertical, and diagonal edge information, respectively.

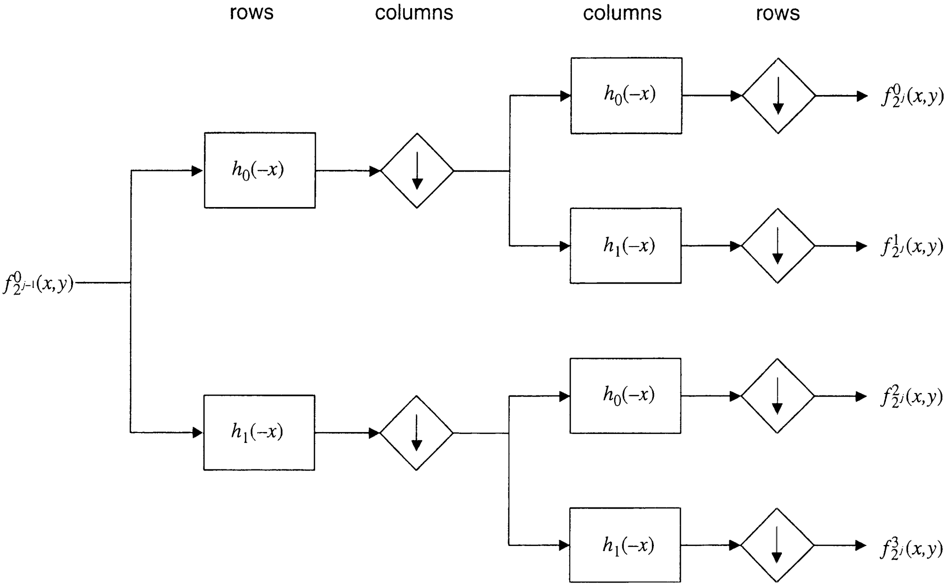

Figure 14–27 The DWT image decomposition step

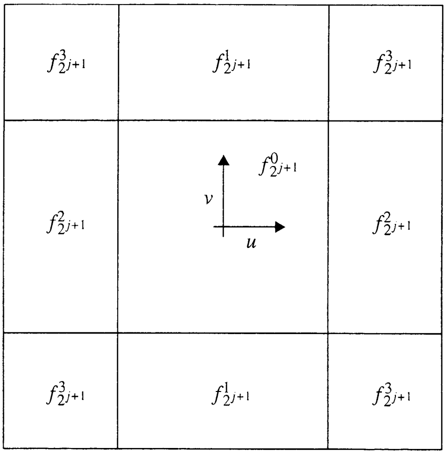

Figure 14–28 DWT decomposition in the frequency domain

14.4.6.2 The Inverse Transform

Inversion of the transform is done by a process similar to that just outlined [5,7,14,16]. This process is diagrammed in Figure 14–29.

At each stage (e.g., the last), we upsample each of the four previous stage arrays by inserting a column of zeros to the left of each column. Then we convolve the rows either with $h_0(x)$ or with $h_1(x)$, as shown in the figure, and add the $N/2$-by-N arrays together in pairs. The two resulting arrays are then upsampled to size N by N by adding a row of zeros above each row. The columns of these two arrays are then convolved with $h_0(x)$ and with $h_1(x)$, as shown. The sum of the two resulting arrays is the result for that stage of the reconstruction.

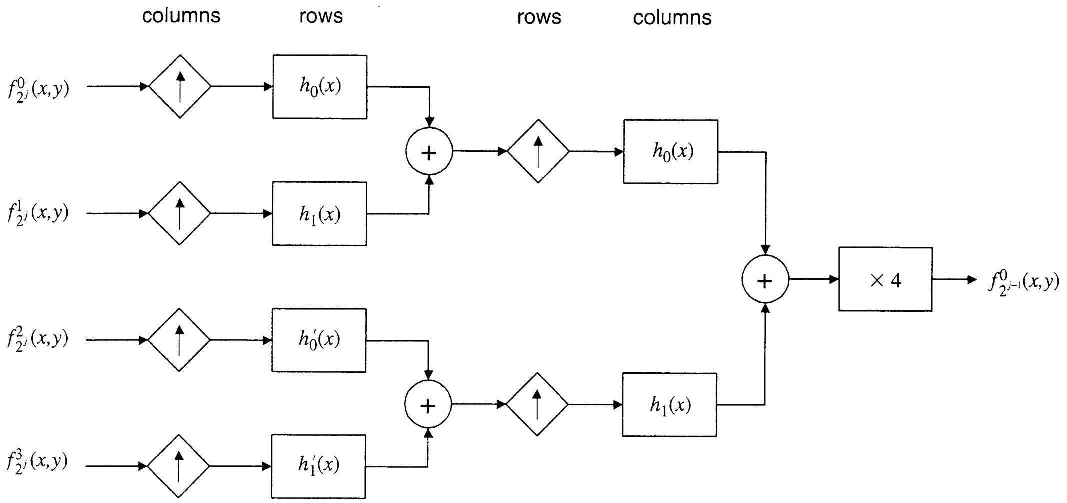

Figure 14–29 The DWT image reconstruction step

14.4.6.3 Examples

Figure 14–30 shows a numerical example of computing the first stage of the two-dimensional discrete wavelet transform. The figure depicts an eight-by-eight pixel image of a

$$\mathbf{f}=\begin{bmatrix}0&0&0&0&0&0&0&0\\0&0&0&1&1&0&0&0\\0&0&2&4&4&2&0&0\\0&1&4&8&8&4&1&0\\0&1&4&8&8&4&1&0\\0&0&2&4&4&2&0&0\\0&0&0&1&1&0&0&0\\0&0&0&0&0&0&0&0\end{bmatrix}\quad \mathbf{f}_{1,0}=\begin{bmatrix}0&0&0&0&0&0&0&0\\0&0&1&1&1&0&0&0\\0&2&4&5&4&2&0&0\\1&3&8&11&8&3&1&0\\1&3&8&11&8&3&1&0\\0&2&4&5&4&2&0&0\\0&0&1&1&1&0&0&0\\0&0&0&0&0&0&0&0\end{bmatrix}\quad \mathbf{f}_{1,1}=\begin{bmatrix}0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&1&1&0&-1&-1&0&0\\1&2&3&0&-3&-2&-1&0\\1&2&3&0&-3&-2&-1&0\\0&1&1&0&-1&-1&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\end{bmatrix}$$

$$\mathbf{f}_{2,0}=\begin{bmatrix}0&0&0&0\\0&1&1&0\\0&4&4&0\\1&8&8&1\\1&8&8&1\\0&4&4&0\\0&1&1&0\\0&0&0&0\end{bmatrix}\quad \mathbf{f}_{2,1}=\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&1&-1&0\\1&3&-3&-1\\1&3&-3&-1\\0&1&-1&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\quad \mathbf{f}_{3,0}=\begin{bmatrix}0&1&1&0\\0&4&4&0\\1&9&9&1\\1&11&11&1\\1&9&9&1\\0&4&4&0\\0&1&1&0\\0&0&0&0\end{bmatrix}\quad \mathbf{f}_{3,1}=\begin{bmatrix}0&1&1&0\\0&2&2&0\\0&3&3&0\\0&0&0&0\\0&-3&-3&0\\0&-2&-2&0\\0&-1&-1&0\\0&0&0&0\end{bmatrix}\quad \mathbf{f}_{3,2}=\begin{bmatrix}0&0&0&0\\0&1&-1&0\\1&3&-3&1\\1&4&-4&-1\\1&3&-3&-1\\0&1&-1&0\\0&0&0&0\\0&0&0&0\end{bmatrix}\quad \mathbf{f}_{3,3}=\begin{bmatrix}0&0&0&0\\0&1&-1&0\\0&1&-1&0\\0&0&0&0\\0&-1&1&0\\0&-1&1&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$$

$$\mathbf{f}_{4,0}=\begin{bmatrix}0&1&1&0\\1&9&9&1\\1&9&9&1\\0&1&1&0\end{bmatrix}\quad \mathbf{f}_{4,1}=\begin{bmatrix}0&1&1&0\\0&3&3&0\\0&-3&-3&0\\0&-1&-1&0\end{bmatrix}\quad \mathbf{f}_{4,2}=\begin{bmatrix}0&0&0&0\\1&3&-3&-1\\1&3&-3&-1\\0&0&0&0\end{bmatrix}\quad \mathbf{f}_{4,3}=\begin{bmatrix}0&0&0&0\\0&1&-1&0\\0&-1&1&0\\0&0&0&0\end{bmatrix}$$

Figure 14–30 Example of computing the two-dimensional discrete wavelet transform

Gaussian-like pulse. Figure 14–31 shows the corresponding (last) stage of the inverse discrete wavelet transform of the same image. (*Note:* Optionally, one can reverse the order of processing rows and columns in both the forward and inverse transforms.)

$$g_{4,0} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 9 & 9 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 9 & 9 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \quad g_{4,1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -3 & -3 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 \end{bmatrix} \quad g_{4,2} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 3 & -3 & -1 \\ 0 & 0 & 0 & 0 \\ 1 & 3 & -3 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad g_{4,3} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$g_{3,0} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 6 & 6 & 1 \\ 1 & 6 & 6 & 1 \\ 1 & 6 & 6 & 1 \\ 1 & 6 & 6 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \quad g_{3,1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -2 & -2 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & -2 & -2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad g_{3,2} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & -2 & 0 \\ 0 & 2 & -2 & 0 \\ 0 & 2 & -2 & 0 \\ 0 & 2 & -2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad g_{3,3} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$g_{2,0} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 4 & 4 & 0 \\ 1 & 8 & 8 & 1 \\ 1 & 8 & 8 & 1 \\ 0 & 4 & 4 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad g_{2,1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 3 & -3 & -1 \\ 1 & 3 & -3 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad g_{1,0} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 4 & 0 & 0 \\ 0 & 1 & 0 & 8 & 0 & 8 & 0 & 1 \\ 0 & 1 & 0 & 8 & 0 & 8 & 0 & 1 \\ 0 & 0 & 0 & 4 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad g_{1,1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 3 & 0 & -3 & 0 & -1 \\ 0 & 1 & 0 & 3 & 0 & -3 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$g_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 3 & 3 & 3 & 3 & 0 & 0 \\ 1 & 1 & 6 & 6 & 6 & 6 & 1 & 1 \\ 1 & 1 & 6 & 6 & 6 & 6 & 1 & 1 \\ 0 & 0 & 3 & 3 & 3 & 3 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad g_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & -2 & 2 & 2 & -2 & 0 & 0 \\ 0 & 0 & -2 & 2 & 2 & -2 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad g = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 4 & 4 & 2 & 0 & 0 \\ 0 & 1 & 4 & 8 & 8 & 4 & 1 & 0 \\ 0 & 1 & 4 & 8 & 8 & 4 & 1 & 0 \\ 0 & 0 & 2 & 4 & 4 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 14–31 Example of computing the two-dimensional inverse discrete wavelet transform

Figure 14–32 shows an example of separable two-dimensional wavelets [20]. These were constructed from Daubechies' $r = 2$ wavelet and scaling function (Figure 14–24) by Eqs. (78) and (79).

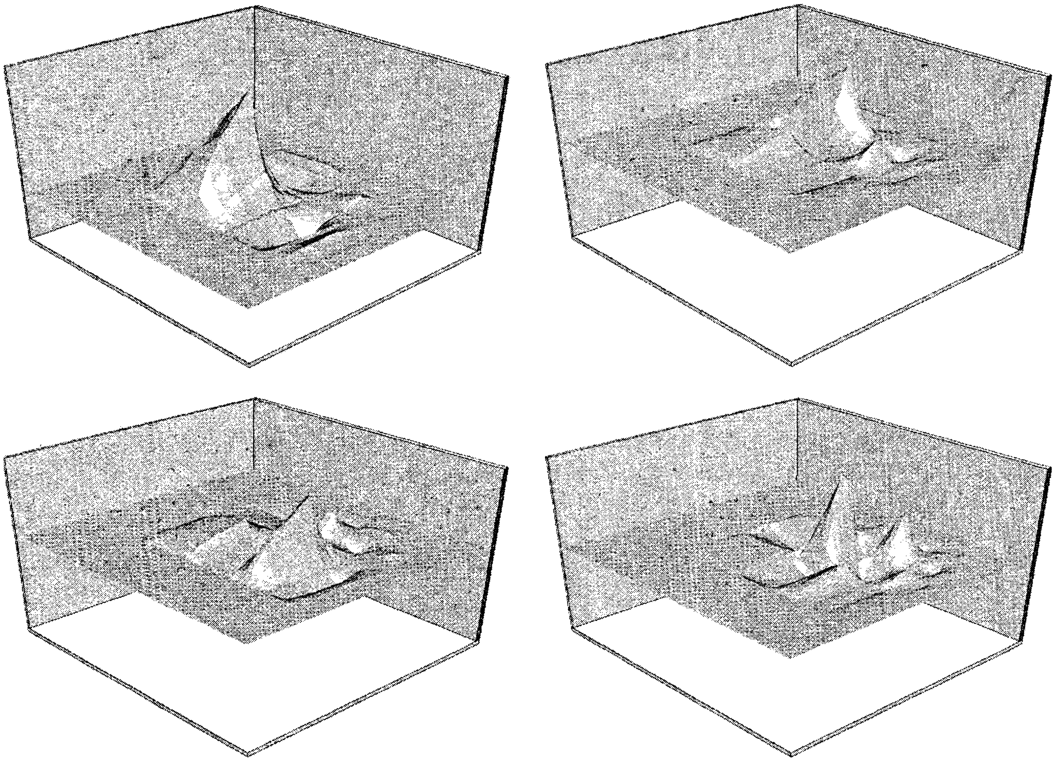

Figure 14–32 Separable two-dimensional wavelets constructed from Daubechies' $r = 2$ wavelet and scaling function (Courtesy Marcus Gross and Lars Lippert, reprinted by permission from [20])

14.4.7 Biorthogonal Wavelet Transforms

The functions that qualify as orthonormal wavelets with compact support lack desirable symmetry properties. It would be convenient, for example, if $\psi(t)$ could be an even or an odd function. By using two different wavelet bases, $\psi(x)$ and $\tilde{\psi}(x)$ —one for decomposition (analysis) and the other for reconstruction (synthesis)—we can have symmetrical wavelets with compact support [5,7,21,22,23]. The two wavelets are *duals* of each other, and the wavelet families $\{\psi_{jk}(x)\}$ and $\{\tilde{\psi}_{jk}(x)\}$ are *biorthogonal;* that is,

$$\langle \psi_{j,k}, \tilde{\psi}_{l,m} \rangle = \delta_{j,l}\delta_{k,m} \tag{83}$$

Then we have

$$c_{j,k} = \langle f(x), \tilde{\psi}_{j,k}(x) \rangle \quad \text{and} \quad d_{j,k} = \langle f(x), \psi_{j,k}(x) \rangle \tag{84}$$

for the decomposition, and

$$f(x) = \sum_{j,k} c_{j,k}\psi_{j,k}(x) = \sum_{j,k} d_{j,k}\tilde{\psi}_{j,k}(x) \tag{85}$$

for the reconstruction. Either wavelet can be used for the decomposition, provided that the other one is used for the reconstruction. The biorthogonal wavelet transform allows the use of symmetric (even or odd) wavelets having compact support.

14.4.7.1 Implementation

The one-dimensional biorthogonal wavelet transform requires four discrete filters (impulse response vectors). We must choose two lowpass filters (scaling vectors), $h_0(n)$ and $\tilde{h}_0(n)$, whose transfer functions satisfy

$$H_0(0) = \tilde{H}_0(0) = 1 \quad \text{and} \quad H_0(s_N) = \tilde{H}_0(s_N) = 0 \tag{86}$$

where $s_N = 1/2\Delta x$ is the folding frequency. From these, we generate two bandpass filters (wavelet vectors), as before, by half-period shifts of their transfer functions [recall Eq. (54)]:

$$h_1(n) = (-1)^n h_0(1-n) \qquad \tilde{h}_1(n) = (-1)^n \tilde{h}_0(1-n) \tag{87}$$

Now we can implement the FWT herringbone algorithm using these four filters, as shown in Figure 14–33.

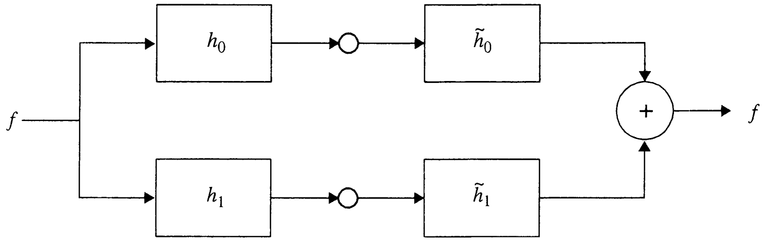

Figure 14–33 One decomposition step and one reconstruction step of the biorthogonal wavelet transform

14.4.7.2 Biorthogonal Wavelets

The conditions upon biorthogonal wavelet filters are

$$\sum_n h_0(n) = \sum_n \tilde{h}_0(n) = \sqrt{2} \quad \text{and} \quad \sum_n h_1(n) = \sum_n \tilde{h}_1(n) = 0 \tag{88}$$

and the perfect reconstruction property requires that

$$H_0(s)\tilde{H}_0(s) + H_1(s)\tilde{H}_1(s) = H_0(s)\tilde{H}_0(s) + H_0(s-s_N)\tilde{H}_0(s-s_N) = 1 \tag{89}$$

The two scaling functions are given, in the frequency domain, by

$$\Phi(2s) = \tilde{H}_0(s)\Phi(s) = \prod_{n=0}^{\infty} \tilde{H}_0(s/2^n) \quad \text{and} \quad \tilde{\Phi}(2s) = H_0(s)\tilde{\Phi}(s) = \prod_{n=0}^{\infty} H_0(s/2^n) \tag{90}$$

and the wavelets are then

$$\psi(x) = \sqrt{2}\sum_n h_1(n+1)\phi(2x-n) \quad \text{and} \quad \tilde{\psi}(x) = \sqrt{2}\sum_n \tilde{h}_1(n+1)\tilde{\phi}(2x-n) \tag{91}$$

14.4.7.3 Constructing Biorthogonal Wavelets

Biorthogonal wavelet design requires developing discrete impulse responses (scaling vectors) $h_0(n)$ and $\tilde{h}_0(n)$ whose transfer functions satisfy Eq. (86) and (89). This is an active area of research, and several authors have catalogued such filters and the corresponding biorthogonal wavelets.

Cohen, Daubechies, and Feauveau [21], for example, select $\phi(x)$ as a B-spline function (e.g., the triangle function) and develop $H_0(s)$ as a polynomial in $\cos(s)$. Vetterli and Herley [22] present approaches based on the theory of diophantine equations and on the theory of continued fractions. Generally, using longer impulse responses gives rise to more regular wavelets, that is, those having a larger number of derivatives and vanishing moments. Table 14–2 presents three pairs of scaling vectors, and Figure 14–34 shows the corresponding biorthogonal wavelets, constructed by the procedure outlined in Figure 14–23.

TABLE 14–2 DISCRETE FILTER SEQUENCES FOR THE BIORTHOGONAL WAVELETS IN FIGURE 14–33 (FROM[21] AND [22]).

Laplacian analysis filter:	$h_0 = \sqrt{2}[-.05 \quad .25 \quad .6 \quad .25 \quad -.05]^t$
Laplacian synthesis filter:	$h_0 = \sqrt{2}[-.0107 \quad -.0536 \quad .2607 \quad .6071 \quad .2607 \quad -.0536 \quad -.0107]^t$
Spline 2 filter:	$h_0 = \sqrt{2}[.25 \quad .5 \quad .25]^t$
Spline 4 filter:	$\tilde{h}_0 = \frac{\sqrt{2}}{128}[3 \quad -6 \quad -16 \quad 38 \quad 90 \quad 38 \quad -16 \quad -6 \quad 3]^t$
18-point analysis filter:	$h_0 = [.0012 \quad -.0007 \quad -.0118 \quad .0117 \quad .0713 \quad -.0310 \quad -.2263 \quad .0693 \quad .7318$ $.7318 \quad .0693 \quad -.2263 \quad -.0310 \quad .0713 \quad .0117 \quad -.0118 \quad -.0007 \quad .0012]^t$
18-point synthesis filter:	$h_0 = [.0012 \quad .0007 \quad -.0113 \quad -.0114 \quad .0235 \quad .0017 \quad -.0444 \quad .2044 \quad .6479$ $.6479 \quad .2044 \quad -.0444 \quad .0017 \quad .0235 \quad -.0114 \quad -.0113 \quad .0007 \quad .0012]^t$

14.4.7.4 Two-dimensional Biorthogonal Wavelets

The biorthogonal wavelets for the forward two-dimensional transform are given by Eq. (79), as before. For the inverse transform, they are

$$\tilde{\psi}^1(x, y) = \tilde{\phi}(x)\tilde{\psi}(y) \qquad \tilde{\psi}^2(x, y) = \tilde{\psi}(x)\tilde{\phi}(y) \qquad \tilde{\psi}^3(x, y) = \tilde{\psi}(x)\tilde{\psi}(y) \tag{92}$$

The implementation of the two-dimensional biorthogonal FWT is a straightforward extension of the orthonormal case.

14.5 WAVELET SELECTION

The ideal basic wavelet would be an oscillatory function of brief duration (i.e., having compact support or small amplitude outside a short interval) where all dyadic translations of binary scalings of the function are orthonormal. The Haar function illustrates this. Other available wavelet functions may fail to meet all these criteria.

First, while the basic wavelet must go to zero as $|x| \rightarrow \infty$ at least as fast as $1/x$ in order to meet the admissibility criterion, many wavelets still have infinite, rather than compact, support. This means that they are nonzero over the entire real line, except for their zero-crossings. It may be that dyadic translations of the wavelet at each scale are orthogonal, but

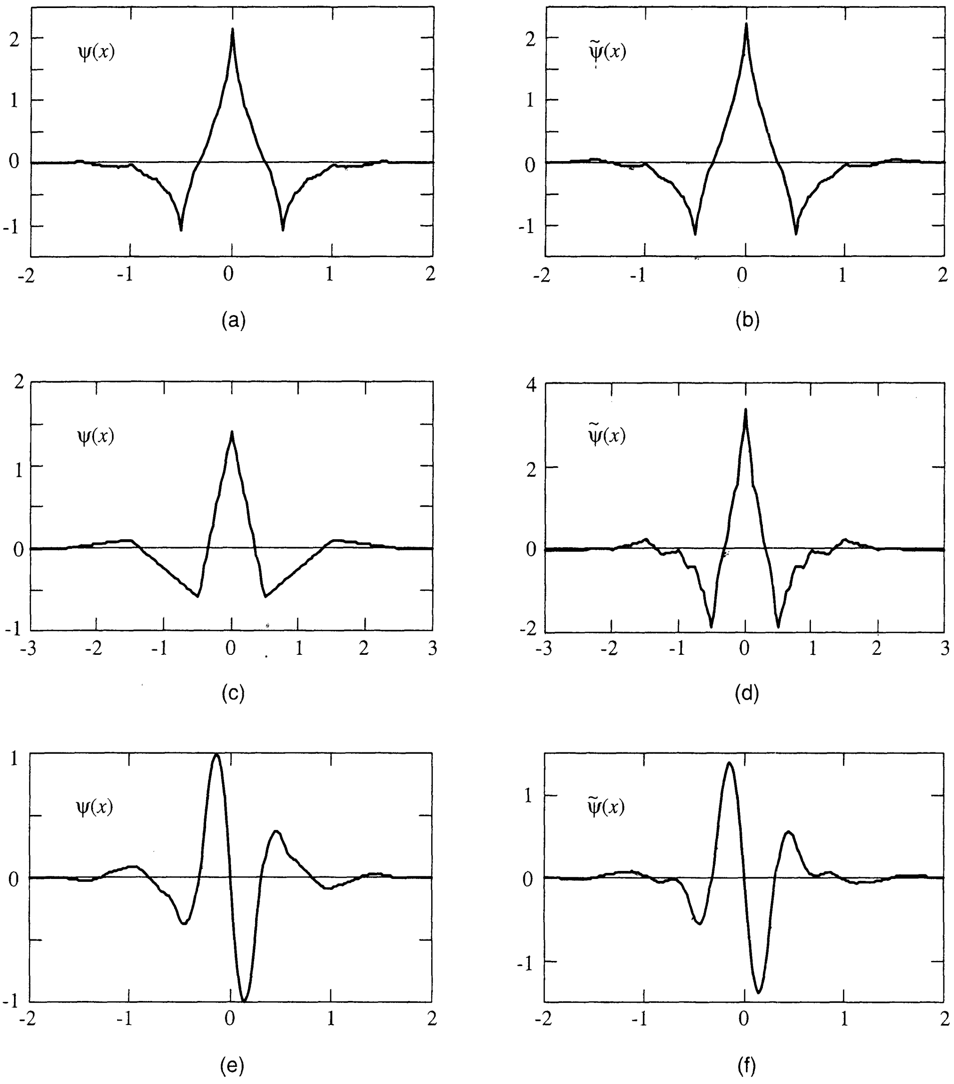

Figure 14–34 Examples of biorthogonal wavelets: (a) Laplacian pyramid wavelet [21]; (b) linear spline function wavelet [21]; (c) 18-point linear phase wavelet [22]

wavelets at different scales are not. Similarly, it may be that different scales of the wavelet are orthogonal, but some or all dyadic translations at the same scale are not.

Notice that some wavelet transforms (e.g., the CWT) are overcomplete, while others (e.g., the DWT) are not. For overcomplete transforms, the restrictions on the basis functions are relatively mild. For transforms involving little or no redundancy, such as the orthonormal discrete wavelet transform, the restrictions placed on basis functions are much more severe.

The biorthogonal DWT requires two scaling vectors and two wavelet vectors rather than one each, but this does not increase the computational burden of the process. The biorthogonal transform, however, affords a much wider choice of wavelet shape than the orthonormal transform, so it is preferable in many applications.

The choice of a basic wavelet is usually governed by the application. For lossless compression, for example, an orthonormal or biorthogonal basis is desirable or required, since the objective is to represent the function exactly and compactly. An overcomplete transform increases the amount of data required to represent the function exactly. If, on the other hand, the goal is lossy compression, the detection of specific components such as edges in an image, or noise removal, then it is more important to select a wavelet that is similar to the components of interest.

Wavelet transforms offer the promise of compact representation and efficient detection of image components that match the waveshape of the chosen wavelet. The orthonormal wavelet transform is inherently compact, but it does not behave well under slight shifts of the image components [24]. An image component that matches a wavelet will appear compactly in the transform if it happens to align with one of the dyadic positions of the wavelet, but not otherwise. For this reason, non-orthonormal transforms often perform better in detection tasks.

14.6 APPLICATIONS

Although wavelet transforms are relatively new on the image processing scene, they have already begun to see application in practice.

14.6.1 Image Compression

The discrete wavelet transform decomposes an image into a set of successively smaller orthonormal images. Further, while the gray-level histogram of the original image can be of any shape, those of the wavelet transform images are commonly unimodal and symmetrical about zero [14]. This simplifies an analysis of the statistical properties of the image.

Often, one can either coarsely quantize or eliminate entirely those coefficients having small value. Mallat and others have studied the possibility of reconstructing an image from only the zero-crossing locations of its wavelet transform [25]. While perfect reconstruction is generally impossible [7], many images can be adequately approximated by this highly compact coding.

14.6.2 Image Enhancement

The DWT decomposes an image into components of different size, position, and orientation. As with linear filtering in the Fourier frequency domain, one can alter the amplitude of coefficients in the wavelet transform domain prior to obtaining the inverse transform. This can selectively accentuate interesting components at the expense of undesirable ones. Figure 14–35 shows an example of edge-specific contrast enhancement [26,27]. Notice how the four peaks in the histogram are separated by the process.

14.6.3 Image Fusion

Image fusion combines two or more registered images of the same object into a single image that is more easily interpreted than any of the originals. This technique finds application in

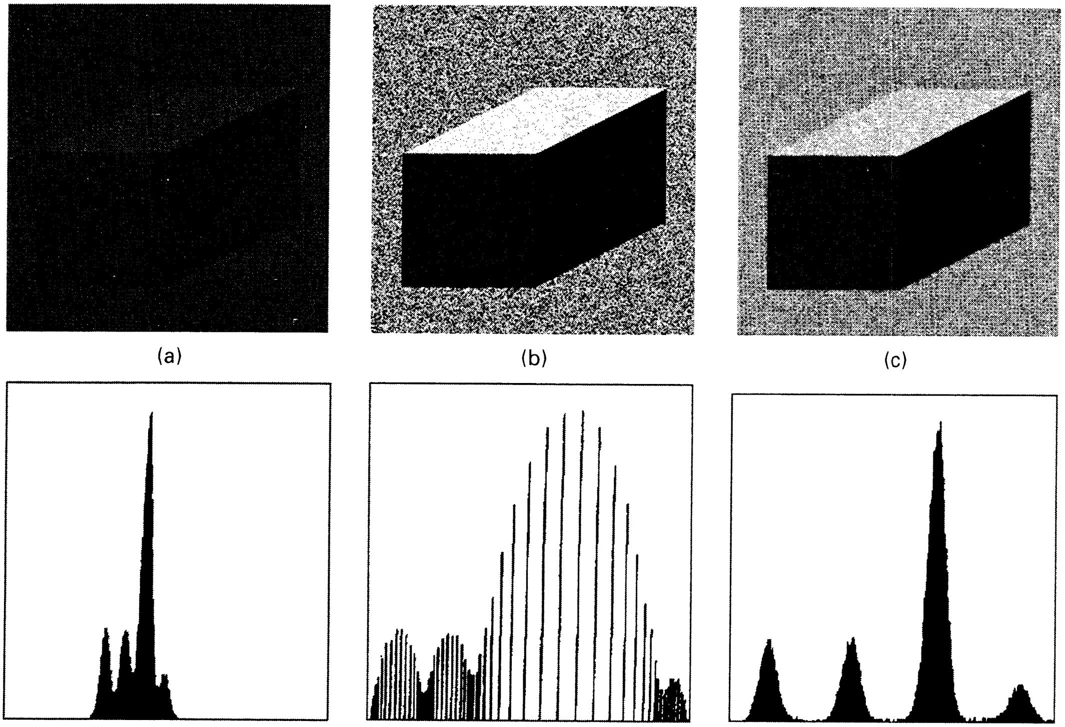

Figure 14–35 Image enhancement by multiscale gradient: (a) original; (b) enhanced by histogram equalization; (c) enhanced by scale-variable edge stretching. Gray-level histograms appear below each image (Courtesy Jian Lu, reprinted by permission from [26])

multispectral image interpretation, as well as medical imaging, where images of the same body part are obtained by several different imaging modalities.

Figure 14–36 shows two examples of image fusion using a wavelet transform [28]. In each case, the two images were combined in the transform domain by taking the maximum-amplitude coefficient at each coordinate. An inverse DWT of the resulting coefficients then reconstructed the fused image. In the first case, the process combined the in-focus information from the two input images. In the second case, the anatomic information of the MRI image was combined with the functional information of the PET scan to produce a convenient composite.

14.7 SUMMARY OF IMPORTANT POINTS

1. A basic wavelet is an oscillatory function that dies out as $|x| \rightarrow \infty$. Its spectrum resembles the transfer function of a bandpass filter.
2. A set of basis functions for a wavelet transform can be generated from dilations and translations of a basic wavelet.
3. The continuous wavelet transform represents a signal as a function of two variables: time and scale. It represents an image as a function of three variables: two for spatial position and one for scale.

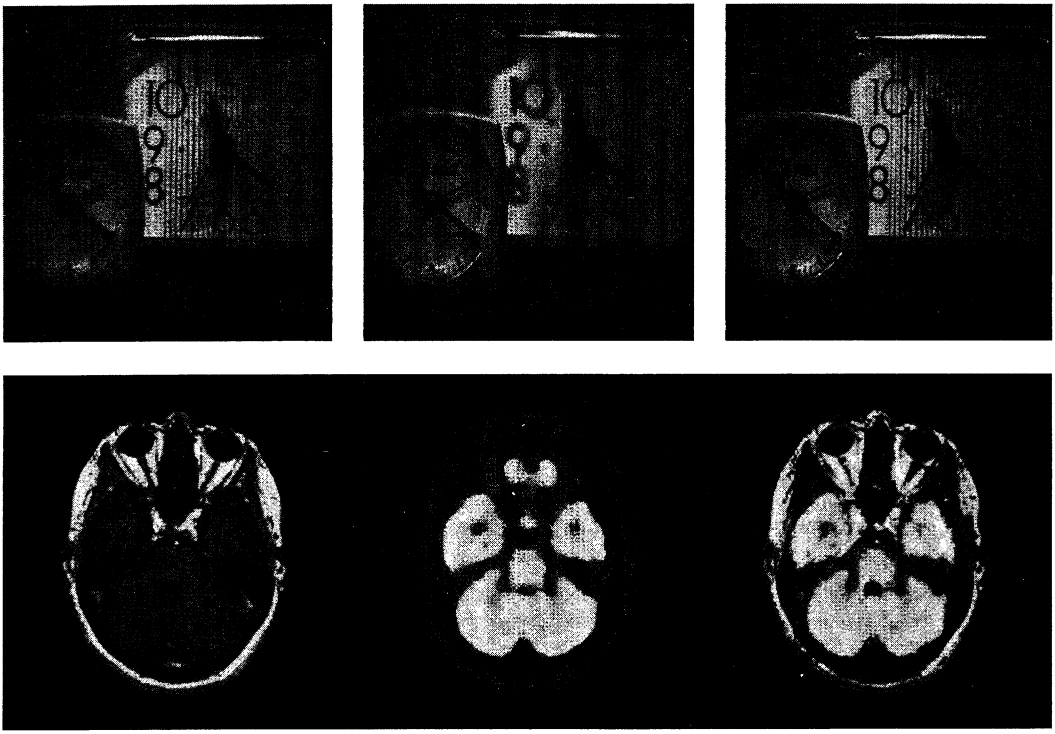

Figure 14–36 Wavelet transform image fusion: (a), (b) images taken at different focus settings; (c) fused image; (d) MRI image; (e) PET image; (f) fused image (Courtesy Henry Hui Li, reprinted by permission from [28])

4. The wavelet series expansion represents a periodic or finite-length signal with a series of coefficients.
5. The discrete wavelet transform represents an N-point signal with N coefficients. It represents an N-by-N image with N^2 coefficients.
6. The Haar transform is the simplest discrete wavelet transform.
7. The DWT can be implemented directly or, indirectly, by the fast wavelet transform (FWT, or herringbone) algorithm.
8. The separable two-dimensional DWT can also be implemented by the FWT algorithm.
9. Biorthogonal wavelet systems permit the DWT to use less restricted (e.g., symmetric) wavelets with compact support.

PROBLEMS

1. Which wavelet transform would you expect to perform best in detecting lines in an engineering drawing? Why?
2. Which wavelet transform would you expect to perform best in compressing fingerprint images? Why?

3. Which wavelet transform would you expect to perform best in image fusion? Why?
4. Which wavelet transform would you expect to perform best in detecting stars in a telescope image? Why?
5. Which wavelet transform would you expect to perform best in segmenting aerial photographs on the basis of texture? Why?

PROJECTS

1. Develop a program implementing the continuous wavelet transform, and use the program to locate the notes in a digitized recording of a simple song.
2. Develop a program for computing a wavelet series expansion of a signal, and use the program to compress a signal.
3. Develop a program for computing the discrete wavelet transform of a signal, and use the program to locate transient components in a signal.
4. Develop a program for computing a continuous wavelet transform of an image, and use the program to locate the spots in a simple image.
5. Develop a program for computing a wavelet series expansion of an image, and use the program to compress an image.
6. Develop a program for computing the discrete wavelet transform of an image, and use the program to locate edges in an image.

REFERENCES

1. S. Pittner, J. Schneid, and C. W. Ueberhuber, *Wavelet Literature Survey,* Technical University of Vienna, Vienna, Austria, 1993.
2. A. Haar, "Zur Theorie der Orthogonalen Funktionen-System," Inaugural Dissertation, *Math. Annalen,* **5:**17–31, 1955.
3. D. Gabor, "Theory of Communication," *IEEE Proc.,* **93:**429–441, 1946.
4. A. Grossman and J. Morlet, "Decomposition of Hardy Functions into Square Integrable Wavelets of Constant Shape," *SIAM J. Appl. Math.,* **15:**723–736, 1984.
5. C. K. Chui, *An Introduction to Wavelets,* Academic Press, San Diego, 1992.
6. A. P. Calderon, "Intermediate Spaces and Interpolation, the Complex Method," *Studia Math,* **24:**113–190, 1964.
7. Y. Meyer (transl. by R. D. Ryan), *Wavelets: Algorithms and Applications,* Society for Industrial and Applied Mathematics, Philadelphia, 1993.
8. H. C. Andrews, *Computer Techniques in Image Processing,* Academic Press, New York, 1970.
9. J. E. Shore, "On the Application of Haar Functions," *IEEE Trans. Comm.,* **COM-21,** 209–216, 1973.
10. P. J. Burt and E. H. Adelson, "The Laplacian Pyramid as a Compact Image Code," *IEEE Trans.,* **C-31:**532–540, 1983.
11. J. W. Woods and S. D. O'Neill, "Subband Coding of Images," *IEEE Trans.,* **ASSP-34:**1278–1288, 1986.
12. D. Esteban and C. Galand, "Application of Quadrature Mirror Filters to Split-Band Voice Coding Systems," *Int. Conf. on Acoustics, Speech and Signal Processing,* Washington, DC, pp. 191–195, 1977.

13. A. Croisier, D. Esteban, and C. Galand, "Perfect Channel Splitting by use of Interpolation, Decimation, Tree Decomposition Techniques," *Int. Conf. on Information Sciences and Systems,* pp. 443–446, 1976.

14. S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Trans.,* **PAMI-11:**674–693, 1989.

15. R. A. Gopinath and C. S. Burrus, "Wavelet Transforms and Filter Banks," *Wavelets—A Tutorial in Theory and Applications* (C. K. Chui, ed.), pp. 603–654, Academic Press, 1992.

16. I. Daubechies, "Orthonormal Bases of Compactly Supported Wavelets," *Commun. on Pure and Appl. Math.,* **41:**909–996, 1988.

17. W. M. Lawton, "Tight Frames of Compactly Supported Wavelets," *J. Math. Phys.,* **31:**1898–1900, 1990.

18. W. M. Lawton, "Necessary and Sufficient Conditions for Existence of ON Wavelet Bases," *J. Math. Phys.,* **32:**57–61, 1991.

19. L. Cohen, "Time-Frequency Distributions—A Review," *Proc. IEEE,* **77**(7):941–981, 1989.

20. M. H. Gross, R. Koch, L. Lippert, and A. Dreger, "Multiscale Image Texture Analysis in Wavelet Spaces," *Proc. ICIP '94,* **III:**412–416, IEEE Computer Society Press, Los Alamitos, CA, 1994.

21. A. I. Cohen, I. Daubechies, and J. C. Feauveau, "Bi-Orthogonal Bases of Compactly Supported Wavelets," *Comm. Pure and Applied Math.,* **45:**485–560, 1992.

22. M. Vetterli and C. Herley, "Wavelets and Filter Banks: Theory and Design," *IEEE Trans.,* **SP-40**(9):2207–2232, 1992.

23. J. C. Feauveau, P. Mathieu, M. Barlaud, and M Antonini, "Recursive Biorthogonal Wavelet Transform for Image Coding," *ICASSP '91: 1991 Int. Conf. on Acoustics, Speech and Signal Processing,* IEEE Press, New York, **4:**2649–2652, 1991.

24. E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Shiftable Multiscale Transforms," *IEEE Trans.,* **IT-38**(2):587–607, 1992.

25. S. Mallat, "Zero-Crossings of a Wavelet Transform," *IEEE Trans.,* **IT-37**(4):1019–1033, 1991.

26. J. Lu, D. M. Healy, and J. B. Weaver, "Contrast Enhancement of Medical Images Using Multiscale Edge Representation," *Optical Engineering,* **33**(7):2151–2161, 1994.

27. J. Lu and D. M. Healy, "Contrast Enhancement via Multiscale Gradient Transformation," *Proc. ICIP '94,* **II:**482–486, IEEE Computer Society Press, Los Alamitos, CA, 1994.

28. H. Hui Li, B. S. Manjunath, and S. K. Mitra, "Multi-Sensor Image Fusion Using the Wavelet Transform," *Proc. ICIP '94,* **I:**51–55, IEEE Computer Society Press, Los Alamitos, CA, 1994.