



Chapter-6

Wavelets & Other Image Transforms

Wavelets and Multiresolution Processing

All this time, the guard was looking at her, first through a telescope, then through a microscope, and then through an opera glass.

Lewis Carroll, Through the Looking Glass

Introduction

Fourier Transform

- Decomposes a signal into a sum of sinusoids

Wavelet Transform

- Decomposes a signal (image) into small waves of varying frequency and limited duration.

Advantage: Know when (where) the frequencies appear

Major Applications

- Image compression
- Image transmission
- Image denoising

Introduction

- Statistics of images such as histograms can vary significantly from one part of the image to another
- Images have locally varying statistics due to combinations of edges, abrupt features and homogeneous regions
- Small objects are best viewed at high resolution
- Large objects require only a coarse resolution
- Need to analyze images at multiple resolutions

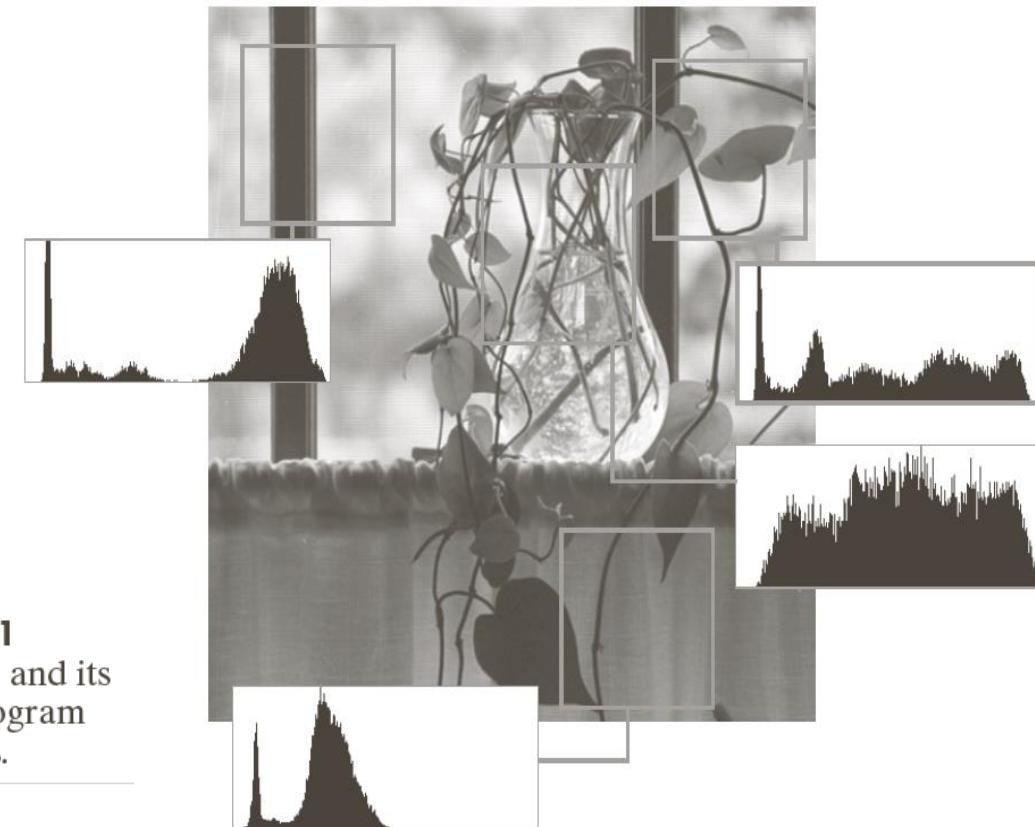


FIGURE 7.1
An image and its local histogram variations.

6.2 Matrix-Based Transforms

- An important class of 1-D linear transforms, denoted by $T(u)$

$$T(u) = \sum_{y=0}^{N-1} f(x)r(x,u) \quad (6-16)$$

$f(x)$: Input Signal

$r(x,u)$: Forward transformation kernel

- Inverse Transform

$$f(x) = \sum_{u=0}^{N-1} T(u)s(x,u) \quad (6-17)$$

where $s(x,u)$ is the *inverse transformation kernel*,
 $x = 0, 1, 2, \dots, N-1$.

- Assume $s(x,u)$ are Orthonormal Basis vectors in basis vector space, $T(u)$ is found using inner-product:

$$T(u) = \langle s(x,u), f(x) \rangle, \quad u = 0, 1, \dots, N-1 \quad (6-19)$$

Matrix-Based Transforms

- In Matrix form:

$$\mathbf{f} = \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(N-1) \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \end{bmatrix}, \quad (6-20)$$

$$\mathbf{t} = \begin{bmatrix} t(0) \\ t(1) \\ \vdots \\ t(N-1) \end{bmatrix} = \begin{bmatrix} t_0 \\ t_1 \\ \vdots \\ t_{N-1} \end{bmatrix}, \quad (6-21)$$

$$\mathbf{s}_u = \begin{bmatrix} s(0,u) \\ s(1,u) \\ \vdots \\ s(N-1,u) \end{bmatrix} = \begin{bmatrix} s_{u,0} \\ s_{u,1} \\ \vdots \\ s_{u,N-1} \end{bmatrix}; \quad u = 0, 1, \dots, N-1 \quad (6-22)$$

- Since \mathbf{s}_u 's represent a set of Orthonormal Basis vectors using matrix notation $T(u)$ is found as,

$$T(u) = \langle \mathbf{s}_u, \mathbf{f} \rangle, \quad u = 0, 1, \dots, N-1 \quad (6-23)$$

Matrix-Based Transforms

- Form Orthonormal Basis Matrix:

$$\mathbf{A} = \begin{bmatrix} \mathbf{s}_0^T \\ \mathbf{s}_1^T \\ \vdots \\ \mathbf{s}_{N-1}^T \end{bmatrix} = \mathbf{s}_0 \quad \mathbf{s}_1 \quad \cdots \quad \mathbf{s}_{N-1}^T \quad (6-24)$$

- Then **forward transform** is given by,

$$\mathbf{t} = \begin{bmatrix} \langle \mathbf{s}_0, \mathbf{f} \rangle \\ \langle \mathbf{s}_1, \mathbf{f} \rangle \\ \vdots \\ \langle \mathbf{s}_{N-1}, \mathbf{f} \rangle \end{bmatrix} = \mathbf{Af} \quad (6-28)$$

- And the **inverse transform**: $\mathbf{f} = \mathbf{A}^{-1}\mathbf{t}$ (6-29)

- \mathbf{s}_u 's : A set of Orthonormal Basis vectors,

$$\langle \mathbf{s}_k, \mathbf{s}_l \rangle = \mathbf{s}_k^T \mathbf{s}_l = \delta_{kl} = \begin{cases} 0, & k \neq l \\ 1, & k = l \end{cases} \quad (6-30)$$

- Furthermore,

$$\mathbf{A}\mathbf{A}^T = \mathbf{A}^T\mathbf{A} = \mathbf{I} \quad \text{and} \quad \mathbf{A}^{-1} = \mathbf{A}^T$$

- Therefore, the inverse transform is given by,

$$\mathbf{t} = \mathbf{Af} \iff \mathbf{f} = \mathbf{A}^{-1}\mathbf{t} = \mathbf{A}^T\mathbf{t}$$

(6-28)

(6-29)

- From Chapter 2, Section-5, 2-D linear transform is denoted by $T(u, v)$

Forward Transform:
$$T(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)r(x, y, u, v) \quad (6-31)$$

where,

- $f(x, y)$: input image,
- $r(x, y, u, v)$: *forward transformation kernel*,
- u and v : transform variables, $u = 0, 1, 2, \dots, M - 1$
and $v = 0, 1, \dots, N - 1$.

2-D Image Transform

- Given $T(u, v)$, the original image $f(x, y)$ can be recovered using the inverse transformation of $T(u, v)$.

Inverse Transform:
$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} T(u, v) s(x, y, u, v) \quad (6-32)$$

where,

- $s(x, y, u, v)$: *Inverse transform kernel*, $x = 0, 1, 2, \dots, M - 1$ and $y = 0, 1, \dots, N - 1$.
- $r(x, y, u, v)$ is *separable* if $r(x, y, u, v) = r_1(x, u)r_2(y, v)$
- $r(x, y, u, v)$ is *symmetric* if $r(x, y, u, v) = r_1(x, u)r_1(y, u)$

2-D Image Transform

- If the transformation kernels are real and orthonormal, and both r and s are separable and symmetric, then in matrix form,

$$\text{Forward Transform: } \mathbf{T} = \mathbf{A}\mathbf{F}\mathbf{A}^T \quad (6-35)$$

$$\text{Inverse Transform: } \mathbf{F} = \mathbf{A}^T\mathbf{T}\mathbf{A} \quad (6-36)$$

- The pre- and post multiplication of \mathbf{F} by \mathbf{A} and \mathbf{A}^T compute the columns and transforms of $\mathbf{F} \rightarrow$ breaks the 2D transform into two 1-D transforms.

Complex Orthonormal Basis Vectors

- Orthonormality condition for Complex-valued Basis vectors,

$$\langle \mathbf{s}_k, \mathbf{s}_l \rangle = \langle \mathbf{s}_k, \mathbf{s}_l \rangle^* = \mathbf{s}_k^{*T} \mathbf{s}_l = \delta_{kl} = \begin{cases} 0, & k \neq l \\ 1, & k = l \end{cases} \quad (6-40)$$

- Furthermore,
- Therefore, the 2D transform pair is given by,

$$\text{Forward Transform: } \mathbf{T} = \mathbf{A} \mathbf{F} \mathbf{A}^T \quad (6-43)$$

$$\text{Inverse Transform: } \mathbf{F} = \mathbf{A}^{*T} \mathbf{T} \mathbf{A}^* \quad (6-44)$$

Biorthonormal Basis Vectors

- Expansion functions, s_0, s_1, \dots, s_{N-1} are biorthonormal if there exists dual expansion functions, $\tilde{s}_0, \tilde{s}_1, \dots, \tilde{s}_{N-1}$

$$\langle \tilde{s}_k, s_l \rangle = \tilde{s}_k^T s_l = \delta_{kl} = \begin{cases} 0, & k \neq l \\ 1, & k = l \end{cases} \quad (6-46)$$

- Neither the expansion functions nor their duals are orthonormal by themselves.
- In this case, the 2D transform pair is given by,

$$\text{Forward Transform: } \mathbf{T} = \tilde{\mathbf{A}} \mathbf{F} \tilde{\mathbf{A}}^T \quad (6-47)$$

$$\text{Inverse Transform: } \mathbf{F} = \mathbf{A}^T \mathbf{T} \mathbf{A} \quad (6-48)$$

- Where, $\tilde{\mathbf{A}} = \tilde{s}_0 \quad \tilde{s}_1 \quad \cdots \quad \tilde{s}_{N-1}^T$

Discrete Fourier Transform (cont.)

- See Section 4.4 for a slightly different but equivalent DFT-pair
- DFT pair $f(x) \Leftrightarrow T(u)$ of length N :

$$s(x, u) = \frac{1}{\sqrt{N}} e^{j \frac{2\pi ux}{N}}, \quad 0 \leq x, u \leq N-1 \quad (6-56)$$

$$f(x) = \frac{1}{\sqrt{N}} \sum_{u=0}^{N-1} T(u) e^{j \frac{2\pi ux}{N}}, \quad 0 \leq x \leq N-1 \quad (6-57)$$

$$T(u) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} f(x) e^{-j \frac{2\pi ux}{N}}, \quad 0 \leq u \leq N-1 \quad (6-58)$$

Discrete Fourier Transform (DFT)

- Use $e^{j\frac{2\pi ux}{N}} = \cos \frac{2\pi ux}{N} + j \sin \frac{2\pi ux}{N}$ as its basis functions
- Fast Fourier Transform (FFT) - $O(N \log N)$
- Not so popular in image compression because
 - Performance is not good enough
 - Computational load for complex number is heavy

6.3 Correlation

- **Definition**

$$f * g(\Delta x) = \int_{-\infty}^{\infty} f^*(x)g(x + \Delta x)dx = \langle f(x), g(x + \Delta x) \rangle$$

- If $\Delta x = 0$, $f * g(0) = \langle f(x), g(x) \rangle$

- **Discrete Equivalent:**

$$\mathbf{f} * \mathbf{g}(m) = \sum_{x=-\infty}^{\infty} f_n^* g_{n+m}$$

$$\mathbf{f} * \mathbf{g}(0) = \sum_{x=-\infty}^{\infty} f_n^* g_n = \langle \mathbf{f}, \mathbf{g} \rangle : \text{Measures similarity between } \mathbf{f} \text{ and } \mathbf{g}$$

$$T(u) = \langle \mathbf{s}_u, \mathbf{f} \rangle = \mathbf{s}_u * \mathbf{f}(0) : \text{Measures similarity between } \mathbf{f} \text{ and } \mathbf{s}_u$$

basis

6.4 Basis Functions in Time-Frequency Domain

- Fig. 6.3 shows basis vectors of some common transforms

$h(t)$: Basis vectors

$g(t)$: Function being transformed

$\langle \mathbf{g}, \mathbf{h} \rangle = \mathbf{g} * \mathbf{h}(0)$: Measures similarity between h and g



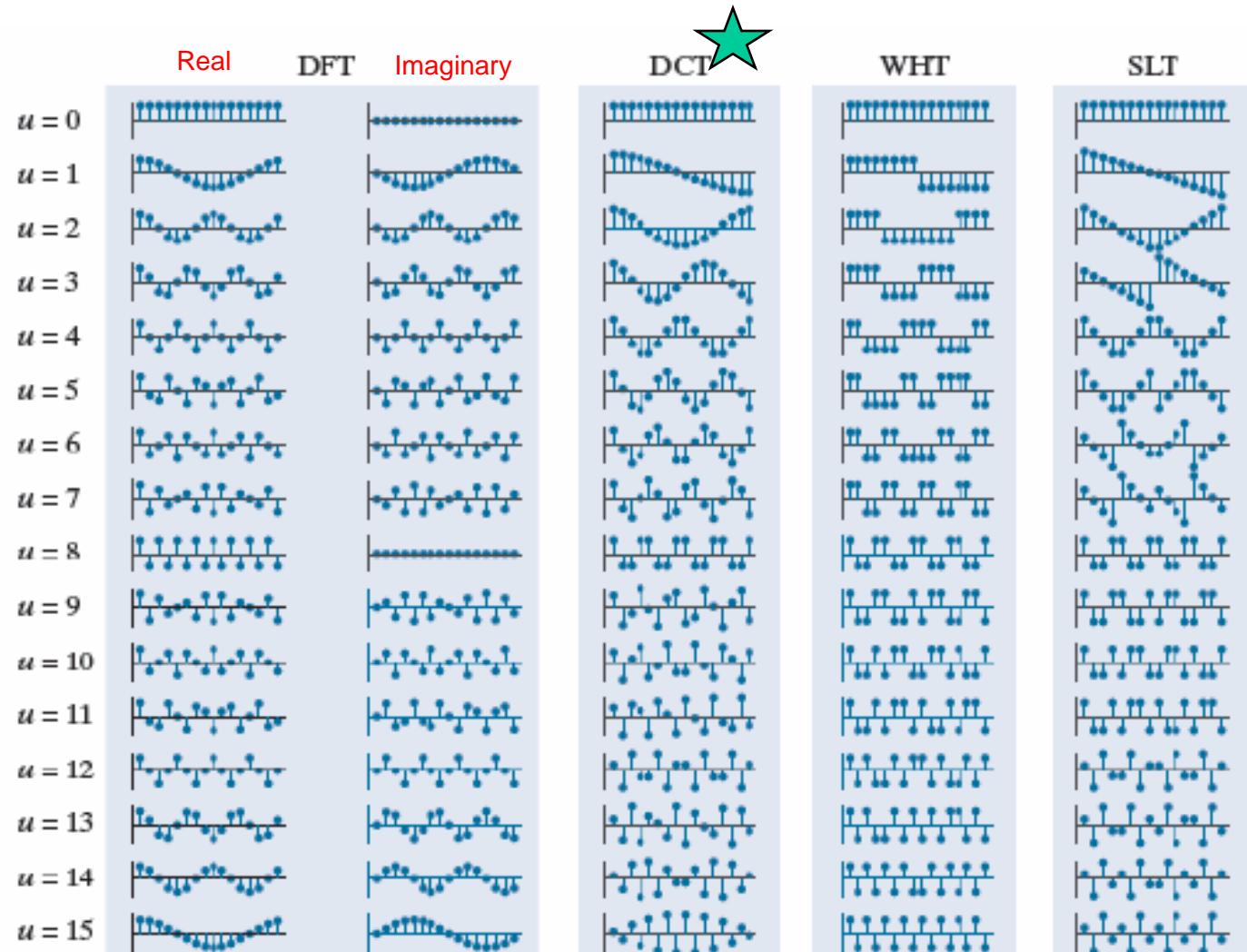
6.4 Basis Functions in Time-Frequency Domain (Orthonormal)

- For a given u , locations in time-frequency plane are similar

a b c d

FIGURE 6.3

Basis vectors (for $N = 16$) of some commonly encountered transforms:
(a) Fourier basis (real and imaginary parts),
(b) discrete Cosine basis,
(c) Walsh-Hadamard basis,
(d) Slant basis,





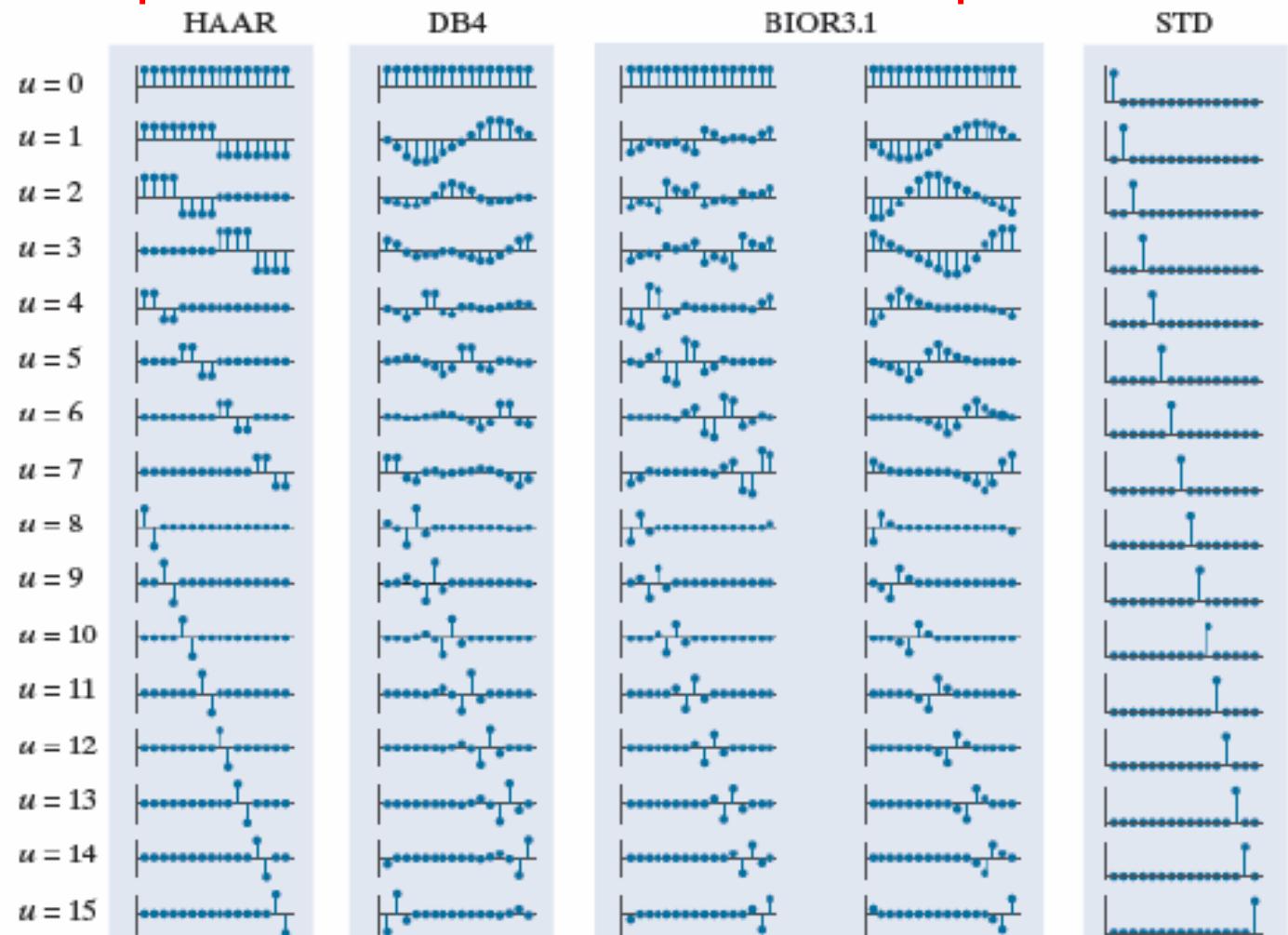
6.4 Basis Functions in Time-Frequency Domain

- Wavelets Basis Functions: Scaled & Shifted small waves

e f g h

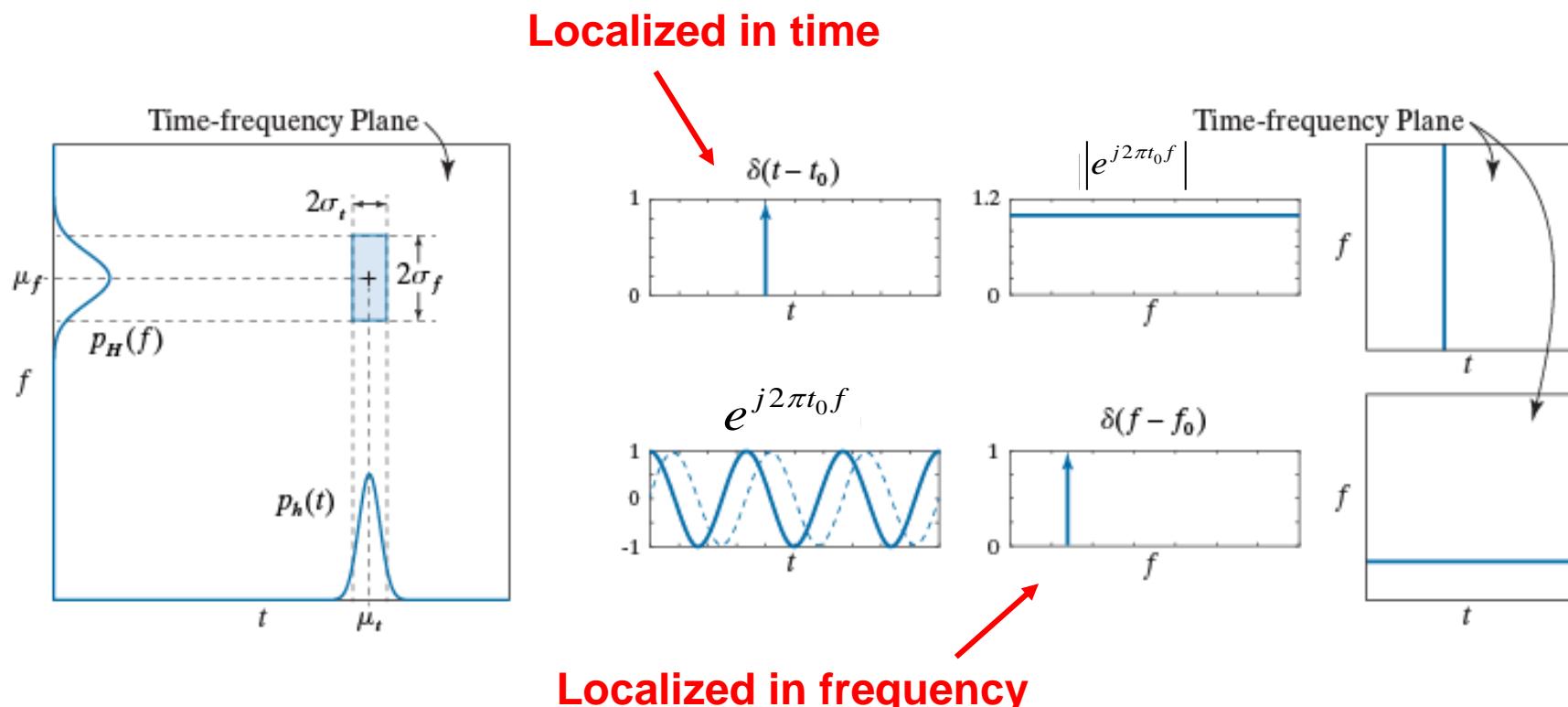
FIGURE 6.3

Basis vectors (for $N = 16$) of some commonly encountered transforms:
(e) Haar basis,
(f) Daubechies basis,
(g) Biorthogonal B-spline basis and its dual, and
(h) the standard basis, which is included for reference only (i.e., not used as the basis of a transform).





Basis Function Localization in Time-Frequency Plane



a | b
c

FIGURE 6.4 (a) Basis function localization in the time-frequency plane. (b) A standard basis function, its spectrum, and location in the time-frequency plane. (c) A complex sinusoidal basis function (with its real and imaginary parts shown as solid and dashed lines, respectively), its spectrum, and location in the time-frequency plane.

- **Wavelets Basis Functions:** Scaled & Shifted small waves

$$\psi_{s,\tau}(t) = 2^{s/2} \psi(2^s t - \tau), \quad s, \tau \text{ are integers}$$

- $\psi(t)$: **Mother Wavelet**
 - A real, square integrable function **with bandpass-like spectrum**
- τ : Determines the position of $\psi_{s,\tau}(t)$ on the t -axis
- s : Determines width (how broad or narrow on the t -axis)
- $2^{s/2}$: Controls Amplitude
- $\Psi(f)$: Fourier Transform of $\psi(t)$ with properties:

$$\mathcal{F}\{\psi(2^s t)\} = \frac{1}{|2^s|} \Psi\left(\frac{f}{2^s}\right) \quad \text{and} \quad \mathcal{F}\{\psi(t - \tau)\} = e^{-j2\pi\tau f} \Psi(f)$$

Heisenberg cells:

- The width of each rectangle in (a) represents one time instant.
- The height of each rectangle in (b) represents a single frequency.

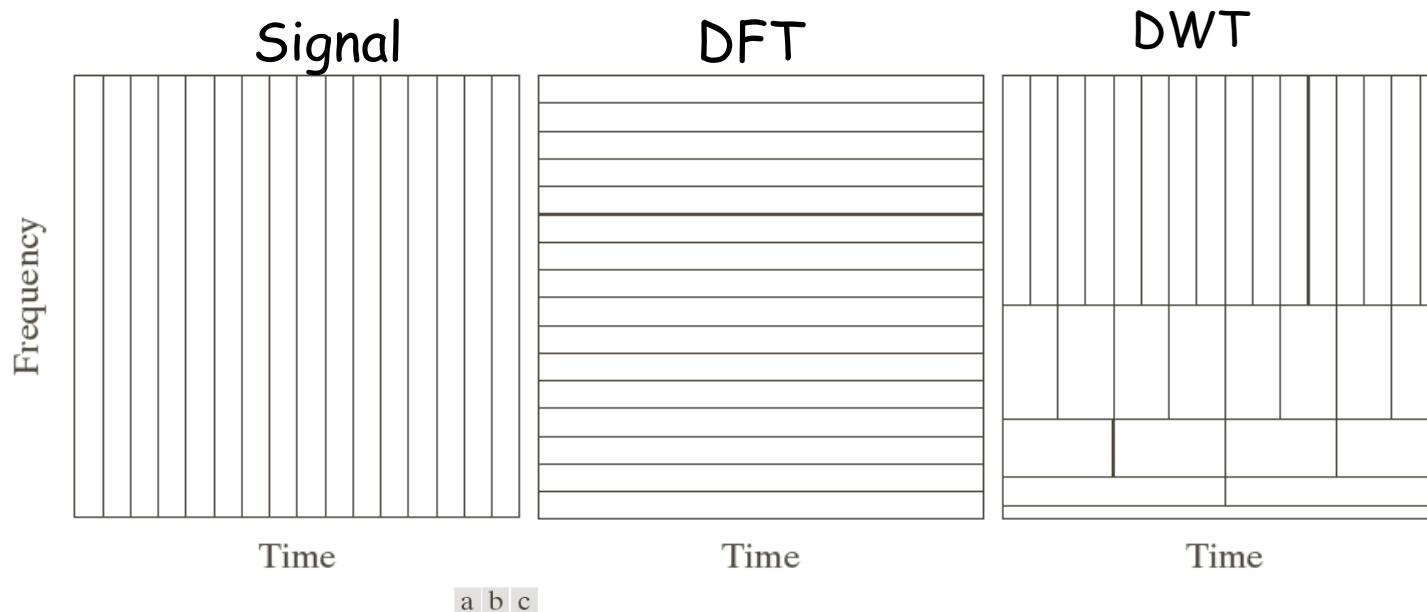


FIGURE 7.23 Time-frequency tilings for the basis functions associated with (a) sampled data, (b) the FFT, and (c) the FWT. Note that the horizontal strips of equal height rectangles in (c) represent FWT scales.

3rd Edition

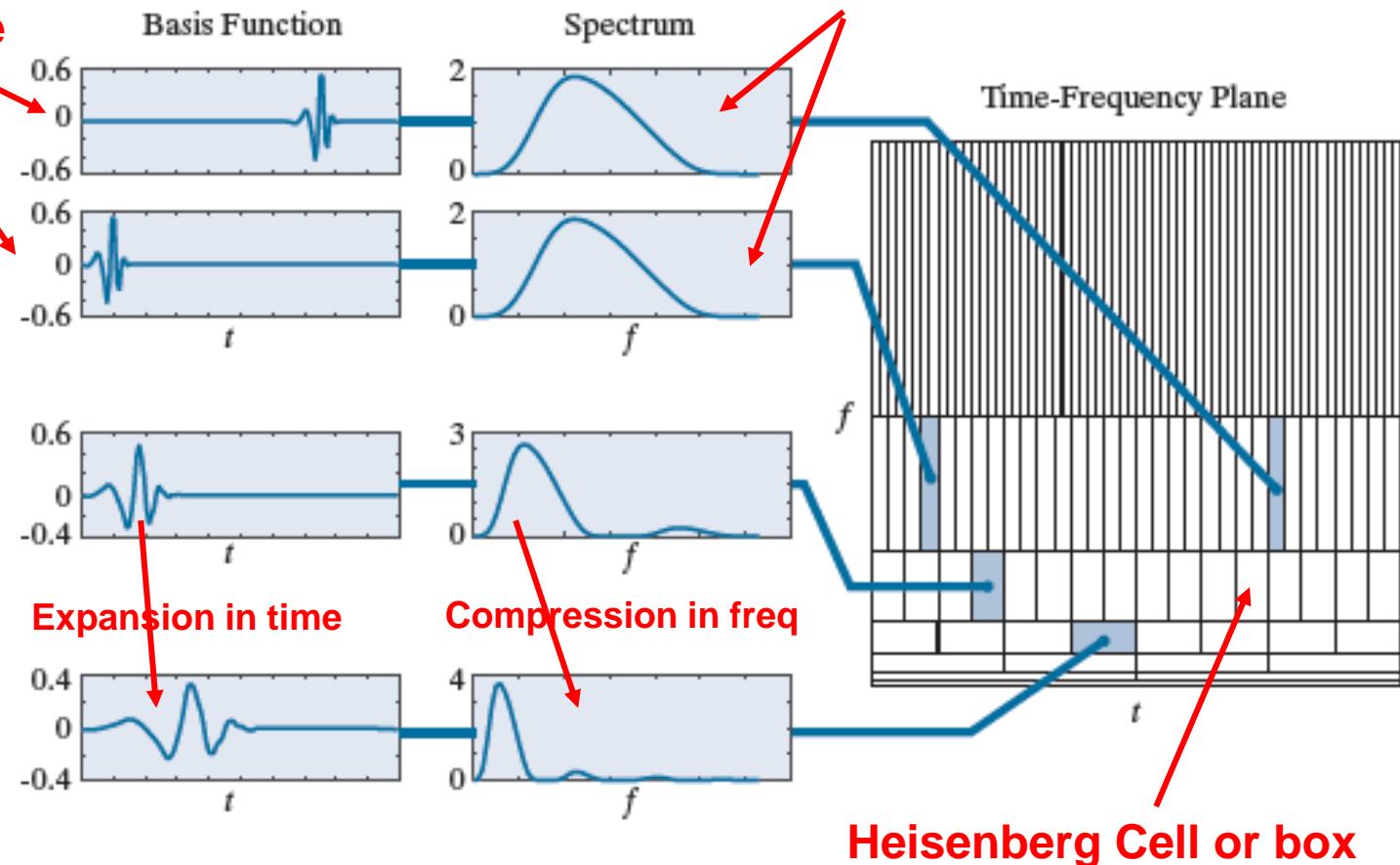
Time & Frequency Localization of Daubechies Basis Functions

Shift in time

a
b
c
d

FIGURE 6.5

Time and frequency localization of 128-point Daubechies basis functions.



- Width (support) in time is inversely proportional to width in frequency, and vice versa

6.5 Basis Images

- Recall $f(x, y)$ from (6-32): $f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} T(u, v) s(x, y, u, v)$
- Matrix version of $f(x, y)$: $\mathbf{F} = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} T(u, v) \mathbf{S}_{u,v}$
- N^2 Basis Images for $u, v = 0, 1, 2, \dots, N - 1$

$$\mathbf{S}_{u,v} = \begin{bmatrix} s(0,0,u,v) & s(0,1,u,v) & \cdots & s(0,N-1,u,v) \\ s(0,0,u,v) & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ s(N-1,0,u,v) & s(N-1,1,u,v) & \cdots & s(N-1,N-1,u,v) \end{bmatrix}$$

- For real, separable and symmetric $s(x, y, u, v) \rightarrow \mathbf{S}_{u,v} = \mathbf{S}_u \mathbf{S}_v^T$

Basis Images for 2-D DFT

- DFT Basis: $s(x, u) = \frac{1}{\sqrt{N}} e^{j \frac{2\pi ux}{N}}$, $0 \leq x, u \leq N-1$ for $N=8$
 from (6-56)

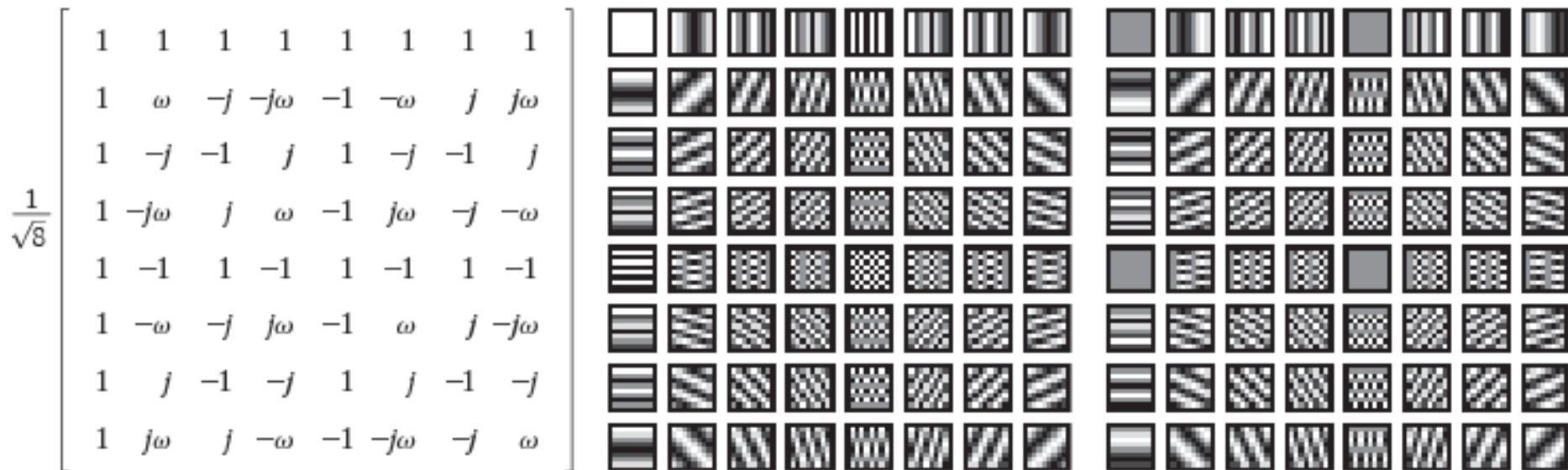


FIGURE 6.7 (a) Transformation matrix \mathbf{A}_F of the discrete Fourier transform for $N=8$, where $\omega = e^{-j2\pi/8}$ or $(1-j)/\sqrt{2}$. (b) and (c) The real and imaginary parts of the DFT basis images of size 8×8 . For clarity, a black border has been added around each basis image. For 1-D transforms, matrix \mathbf{A}_F is used in conjunction with Eqs. (6-43) and (6-44); for 2-D transforms, it is used with Eqs. (6-41) and (6-42).

Basis Images for Standard Basis

- Standard Basis $\mathbf{e}_0, \mathbf{e}_1, \dots, \mathbf{e}_{N-1}$ for $N=8$

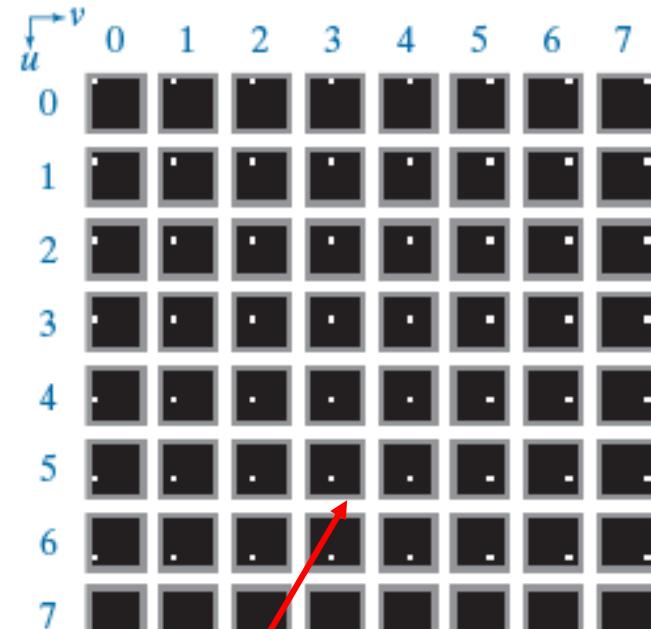
$$\mathbf{S}_{u,v} = \mathbf{e}_u \mathbf{e}_v^T = \mathbf{E}_{u,v} \quad (6-77)$$

a b

FIGURE 6.6

(a) Basis image organization and (b) a standard basis of size 8×8 . For clarity, a gray border has been added around each basis image. The origin of each basis image (i.e., $x = y = 0$) is at its top left.

$S_{0,0}$	$S_{0,1}$	$S_{0,N-1}$
$S_{1,0}$	\ddots			\vdots
\vdots				
$S_{N-1,0}$	\ddots			\vdots
				$S_{N-1,N-1}$



Only one 1 in u -th row and v -th column

6.6 Fourier-Related Transforms

Discrete Cosine Transform (DCT)

- Uses **cosine function** as its basis function
- Real valued (avoids complex number processing)
- **Fast algorithm** exists
- **Most popular** in image compression application
- The periodicity implied by DCT implies that it causes **less blocking** effect than DFT
- Can be implemented by **fast $2n$ points FFT** (by making data symmetric)
- Used in most popular standards:
 - **JPEG, H.26x, MPEG**

- 1-D DCT kernel Definition:

$$s(x, u) = \alpha(u) \cos \left[\frac{(2x+1)u\pi}{2N} \right] \quad \text{where, } \alpha(u \text{ or } v) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u, v = 0 \\ \sqrt{\frac{2}{N}} & \text{for } u, v = 1, 2, \dots, N-1 \end{cases} \quad (6-83)$$

- Form Basis vector

$$\mathbf{s}_u = \begin{bmatrix} s(0, u) \\ s(1, u) \\ \vdots \\ s(N-1, u) \end{bmatrix}; \quad u = 0, 1, \dots, N-1 \quad (6-22)$$

- Form Transformation matrix: $\mathbf{A}_C = \mathbf{s}_0 \quad \mathbf{s}_1 \quad \cdots \quad \mathbf{s}_{N-1}^T \quad (6-24)$

- Use in pair $\mathbf{t} = \mathbf{A}_C \mathbf{f} \iff \mathbf{f} = \mathbf{A}_C^{-1} \mathbf{t}$

(6-28) (6-29)

- 2-D DCT kernel Definition:

$$s(x, y, u, v) = \alpha(u)\alpha(v) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right]; \quad \alpha(u \text{ or } v) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u, v = 0 \\ \sqrt{\frac{2}{N}} & \text{for } u, v = 1, 2, \dots, N-1 \end{cases} \quad (6-83)$$

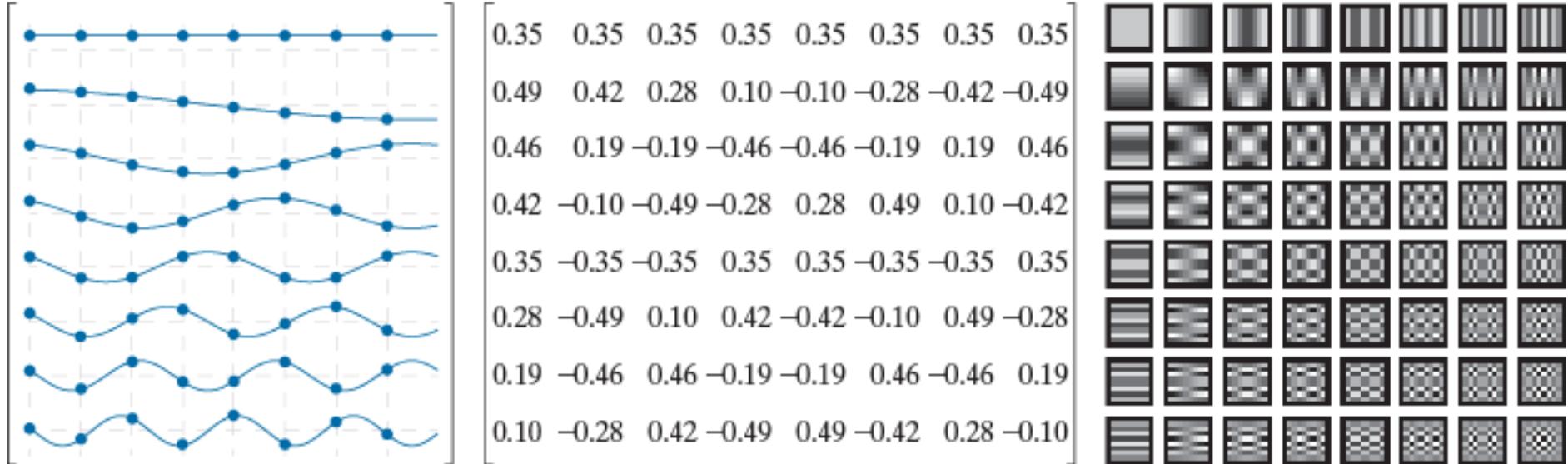
(6-85)

- Form Basis images and Use in pair

$$\text{Forward Transform: } \mathbf{T} = \mathbf{A} \mathbf{F} \mathbf{A}^T \quad (6-35)$$

$$\text{Inverse Transform: } \mathbf{F} = \mathbf{A}^T \mathbf{T} \mathbf{A} \quad (6-36)$$

DCT Transformation Matrix and Basis Images (N=8)



a | b | c

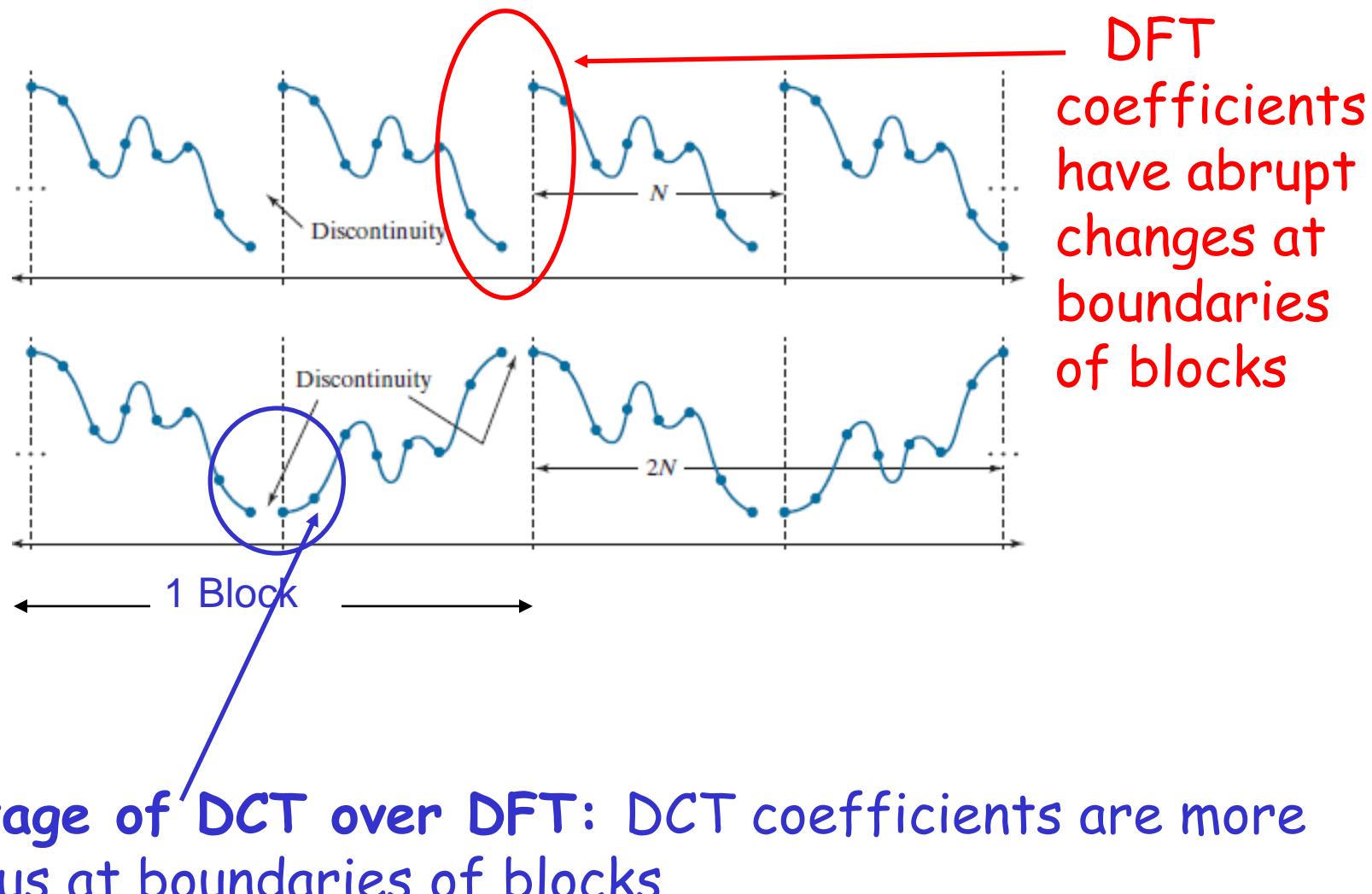
FIGURE 6.10 The transformation matrix and basis images of the discrete cosine transform for $N = 8$. (a) Graphical representation of orthogonal transformation matrix A_C (b) A_C rounded to two decimal places, and (c) basis images. For 1-D transforms, matrix A_C is used in conjunction with Eqs. (6-28) and (6-29); for 2-D transforms, it is used with Eqs. (6-35) and (6-36).

Periodicity implicit in DFT & DCT

a
b

FIGURE 6.11

The periodicity
implicit in the 1-D
(a) DFT and
(b) DCT.

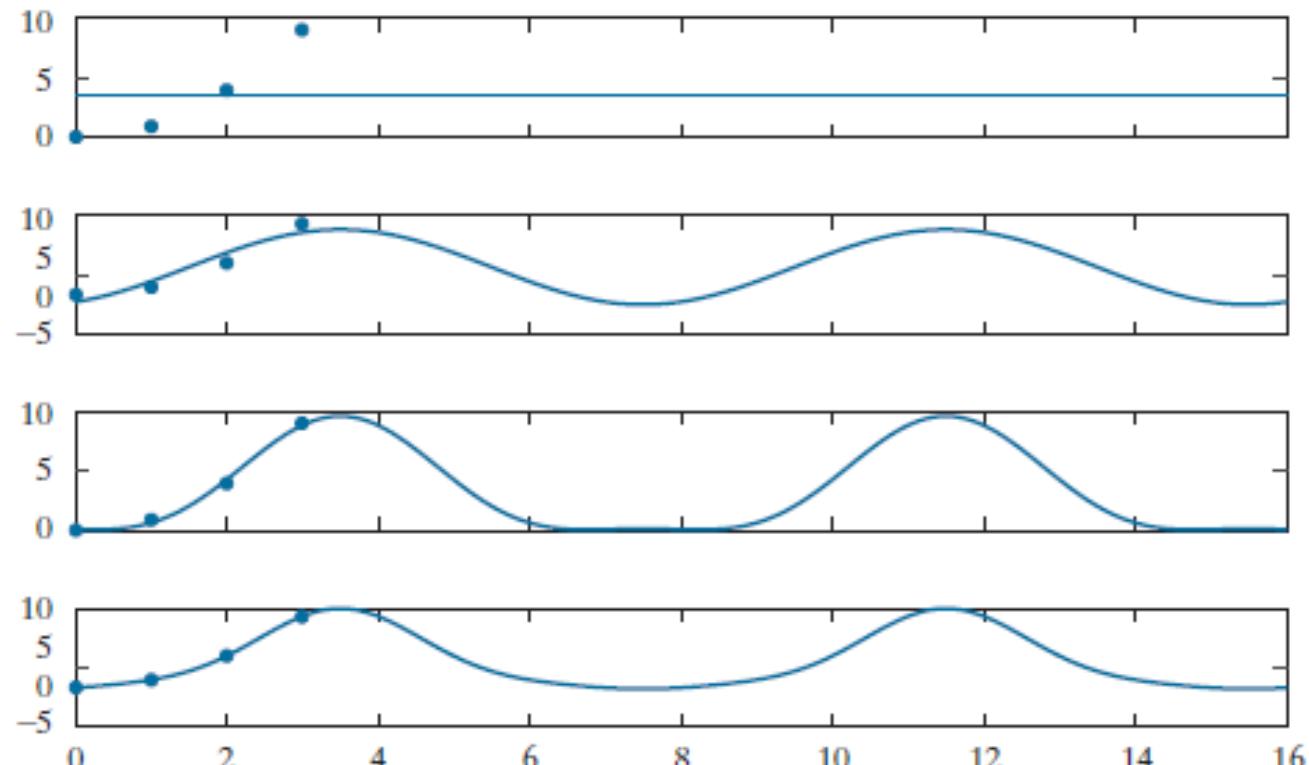




DCT Reconstruction

a
b
c
d

FIGURE 6.12
DCT reconstruction of a discrete function by the addition of progressively higher frequency components. Note the $2N$ -point periodicity and even symmetry imposed by the DCT.





Ideal Lowpass Filtering with Fourier-related transforms

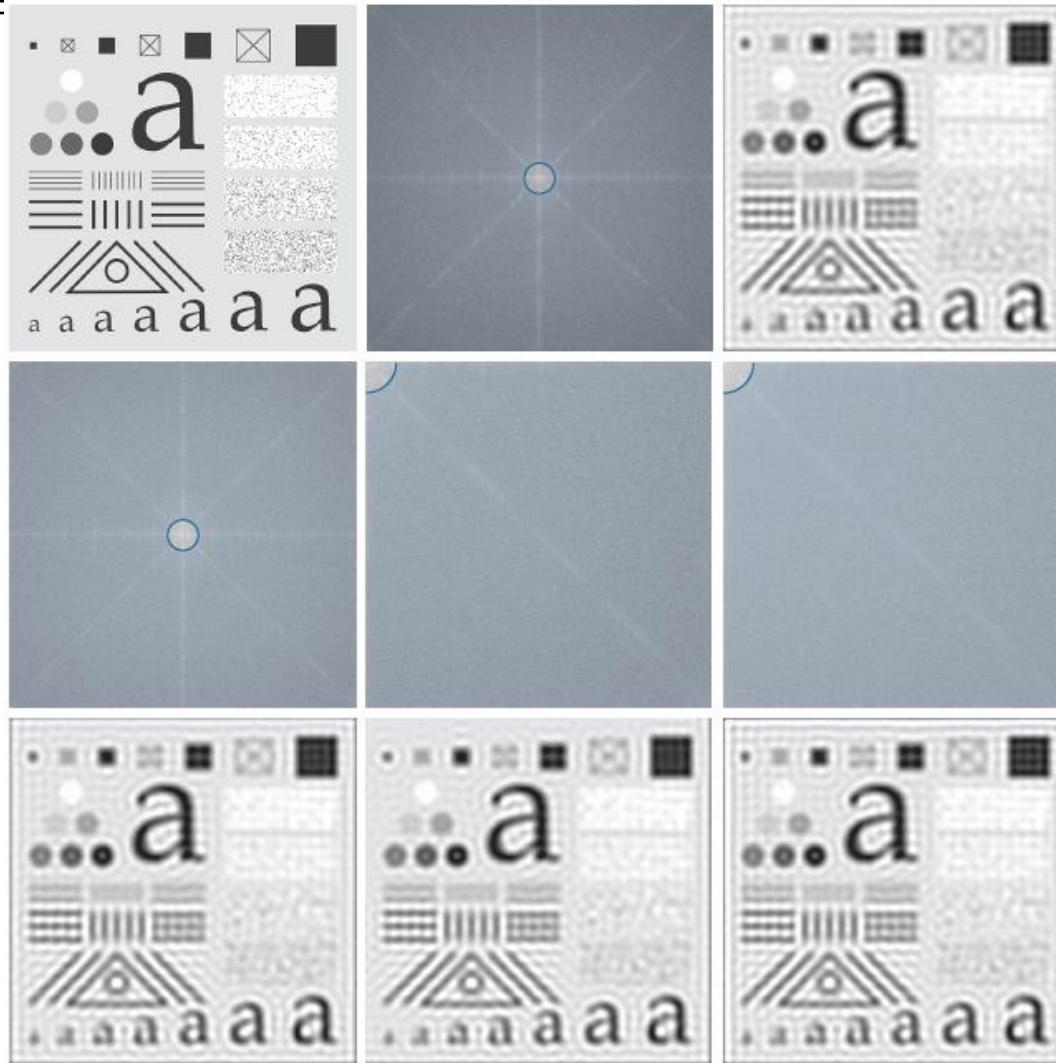


FIGURE 6.15 (a) Original image of the 688×688 test pattern from Fig. 4.41(a). (b) Discrete Fourier transform (DFT) of the test pattern in (a) after padding to size 1376×1376 . The blue overlay is an ideal lowpass filter (ILPF) with a radius of 60. (c) Result of Fourier filtering. (d)–(f) Discrete Hartley transform, discrete cosine transform (DCT), and discrete sine transform (DST) of the test pattern in (a) after padding. The blue overlay is the same ILPF in (b), but appears bigger in (e) and (f) because of the higher frequency resolution of the DCT and DST. (g)–(i) Results of filtering for the Hartley, cosine, and sine transforms, respectively.

w

6.7 Walsh-Hadamard Transform (WHT)

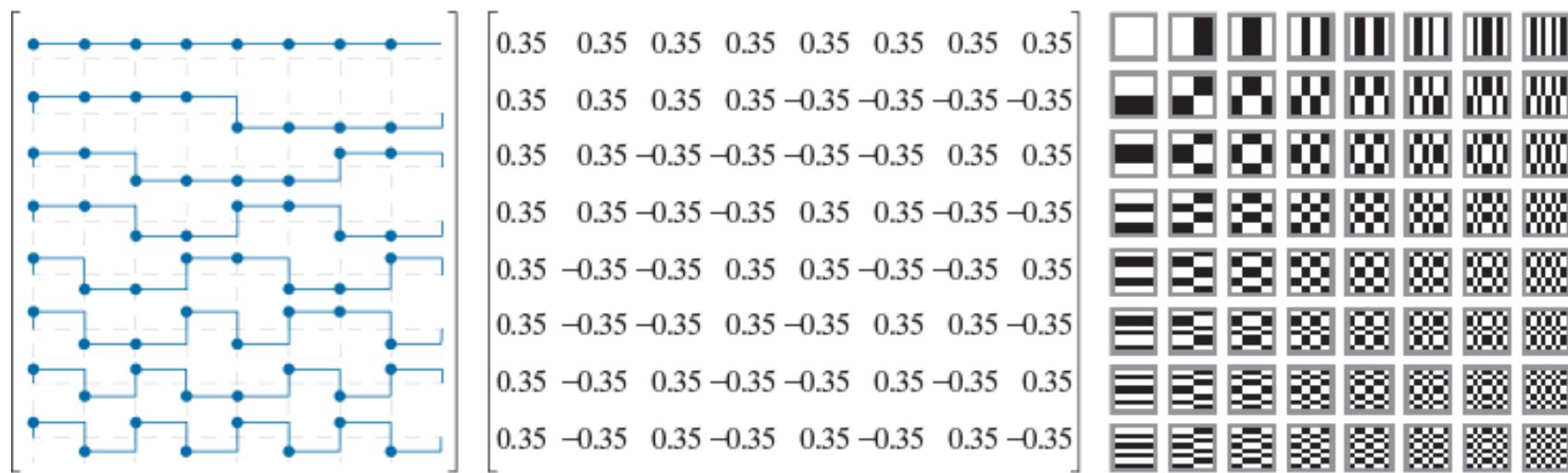
$$\mathbf{H}_{2N} = \begin{bmatrix} \mathbf{H}_N & \mathbf{H}_N \\ \mathbf{H}_N & -\mathbf{H}_N \end{bmatrix} \quad \text{Generate Recursively} \quad (6-98)$$

$$\mathbf{H}_1 = 1 \quad \mathbf{H}_2 = \begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_1 \\ \mathbf{H}_1 & -\mathbf{H}_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (6-99)$$

$$\mathbf{H}_4 = \begin{bmatrix} \mathbf{H}_2 & \mathbf{H}_2 \\ \mathbf{H}_2 & -\mathbf{H}_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \quad \mathbf{H}_8 = \begin{bmatrix} \mathbf{H}_4 & \mathbf{H}_4 \\ \mathbf{H}_4 & -\mathbf{H}_4 \end{bmatrix} \quad (6-101)$$

(6-100)

6.7 Walsh-Hadamard Transform (WHT)



a b c

FIGURE 6.16 The transformation matrix and basis images of the sequency-ordered Walsh-Hadamard transform for $N = 8$. (a) Graphical representation of orthogonal transformation matrix \mathbf{A}_{w^r} , (b) \mathbf{A}_{w^r} rounded to two decimal places, and (c) basis images. For 1-D transforms, matrix \mathbf{A}_{w^r} is used in conjunction with Eqs. (6-28) and (6-29); for 2-D transforms, it is used with Eqs. (6-35) and (6-36).

7.1.3 The Haar Transform

- Due to Alfred Haar [1910].
- Basis Functions: **Oldest and simplest known orthonormal wavelets.**
- The Haar transform is both separable and symmetric:
$$\mathbf{T} = \mathbf{A}_H \mathbf{F} \mathbf{A}_H$$
- \mathbf{F} : $N \times N$ - Image
- \mathbf{A}_H : $N \times N$ - Transformation matrix
- \mathbf{T} : $N \times N$ - Transformed image
- Matrix \mathbf{A}_H contains Haar basis functions

The Haar Transform (cont...)

- The Haar basis functions $h_u(x)$ are defined for $0 \leq x \leq 1$, for $u = 0, 1, \dots, N-1$, where $N=2^n$
- To generate \mathbf{H} :
 - Define integer $u = 2^p + q$
 - p : Largest power of 2 in u
 - $q = 2^p - u$

(6-114)

For the above pairs of p and q , a value for u is determined and the Haar basis functions are computed.

The Haar Transform (cont...)

Haar basis functions:

$$h_u(x) = \begin{cases} 1 & u = 0 \text{ and } 0 \leq x < 1 \\ 2^{p/2} & u > 0 \text{ and } q/2^p \leq x < (q+0.5)/2^p \\ -2^{p/2} & u > 0 \text{ and } (q+0.5)/2^p \leq x < (q+1)/2^p \\ 0 & \text{otherwise} \end{cases} \quad (6-115)$$

First row is always 1

The u -th row of a $N \times N$ Haar transform matrix contains the elements of $h_u(x)$ for $x = 0/N, 1/N, 2/N, \dots, (N-1)/N$.

The Haar Transform (cont...)

- $N \times N$ Haar matrix is formed as:

$$\mathbf{H}_N = \begin{bmatrix} h_0(0/N) & h_0(1/N) & \cdots & h_0(N-1/N) \\ h_1(0/N) & h_1(1/N) & \cdots & \vdots \\ \vdots & \cdots & \ddots & \vdots \\ h_{N-1}(0/N) & \cdots & \cdots & h_{N-1}(N-1/N) \end{bmatrix} \quad (6-117)$$

- The resulting transformation matrix is

$$\mathbf{A}_H = \frac{1}{\sqrt{N}} \mathbf{H}_N \quad (6-118)$$

- For $N=2$, the 2×2 transformation matrix is:

$$\mathbf{A}_H = \frac{1}{\sqrt{2}} \begin{bmatrix} h_0(0) & h_0(1/2) \\ h_1(0) & h_1(1/2) \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (6-119)$$

The Haar Transform (cont...)

For $N=4$, p , q and u have the following values:

u	p	q
1	0	0
2	1	0
3	1	1

For $u=0$, the first row
is always 1

and the 4×4 transformation matrix is:

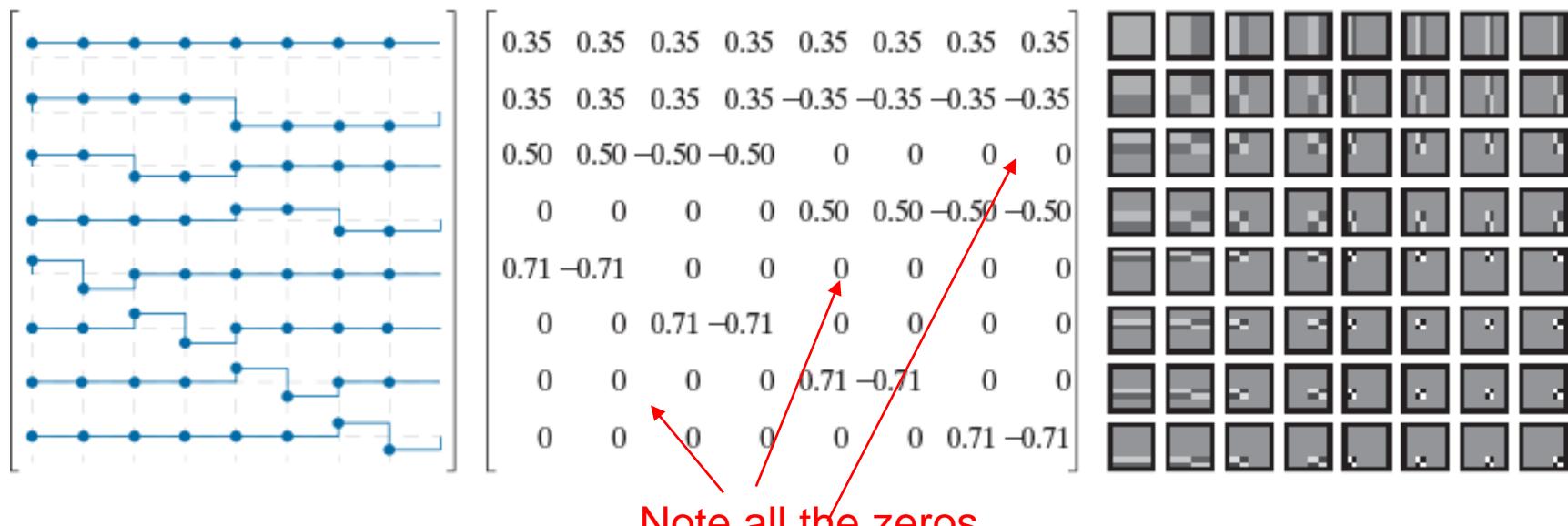
$$\mathbf{A}_H = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix} \quad (6-120)$$

The Haar Transform (cont...)

The number of positive-going zero crossings

- A is real, orthogonal, and sequency ordered.
- It can be decomposed into products of matrices with fewer non-zero entries (even fewer than other transforms studied so far)
- Reduced complexity algorithm on the order $O(N)$
- Can be used as Scaling and Wavelet vectors (defined later)
- One of the simplest and oldest Wavelet transform

The Haar Transform (cont...)



a b c

FIGURE 6.18 The transformation matrix and basis images of the discrete Haar transform for $N = 8$. (a) Graphical representation of orthogonal transformation matrix \mathbf{A}_H , (b) \mathbf{A}_H rounded to two decimal places, and (c) basis images. For 1-D transforms, matrix \mathbf{A}_H is used in conjunction with Eqs. (6-28) and (6-29); for 2-D transforms, it is used with Eqs. (6-35) and (6-36).

6.2 Wavelet Transforms

- Image pyramids, subband coding and the Haar transform play an important role in a mathematical framework called *Multiresolution Analysis (MRA)*
- **Scaling Function:** Used to create a series of approximations of a signal, each differing by a factor of 2 in resolution from its nearest neighbouring approximation
- **Wavelets:** Used to encode the difference (i.e., residues) between adjacent approximations

Scaling Functions

- Consider the set of **expansion functions** composed of **integer translations** and **binary scalings** of a real, square-integrable function $\varphi(x)$:
- $$\varphi_{j,k}(x) = 2^{j/2} \varphi(2^j x - k), \quad j, k \in \mathbb{Z}, \quad \varphi(x) \in L^2(\mathbb{R}). \quad (6-121)$$
- $\varphi(x)$: To be defined (Haar basis, for example)
 - Parameter k (shift)**: Determines the position of $\varphi_{j,k}(x)$ along the horizontal axis
 - Parameter j (width-scaling)**: Determines how broad or narrow it is along the horizontal axis
 - The multiplication term $2^{j/2}$ controls the amplitude
 - The shape of $\varphi_{j,k}(x)$ changes for each j , hence $\varphi(x)$ is called **scaling function**

Haar Scaling Function

- Consider the unit-height, unit-width Haar scaling function:

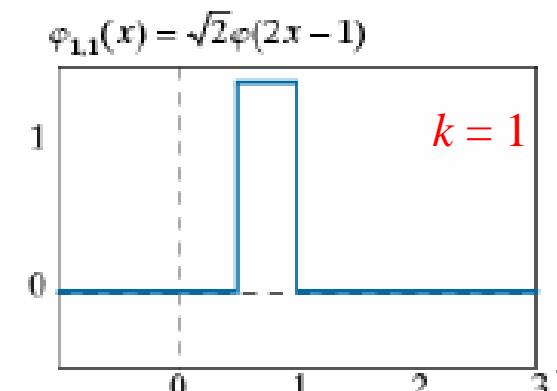
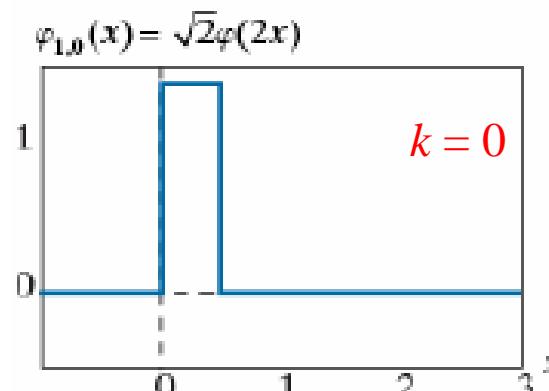
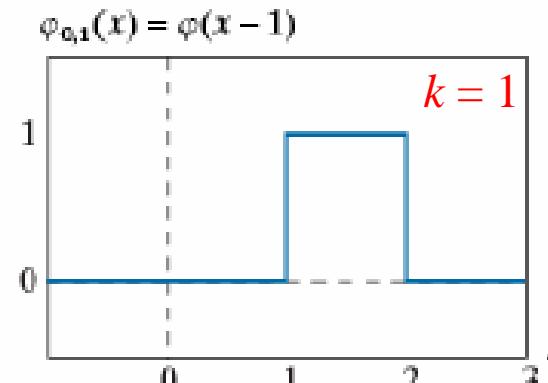
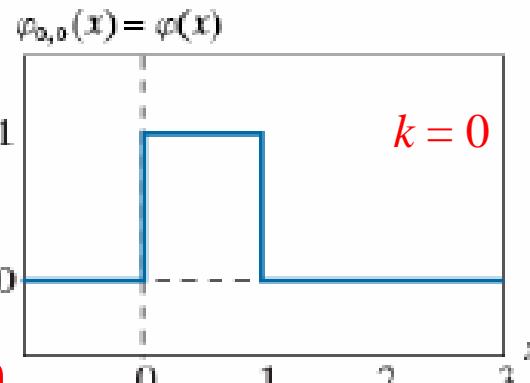
$$\varphi(x) = \begin{cases} 1, & 0 \leq x \leq 1 \\ 0, & \text{otherwise} \end{cases} \quad \begin{matrix} \text{Father Scaling} \\ \text{Function} \end{matrix} \quad (6-122)$$

- Then observe some of the expansion functions $\varphi_{j,k}(x)$ generated by scaling and translations of the original function.

Multiresolution Analysis Scaling Functions (cont...)

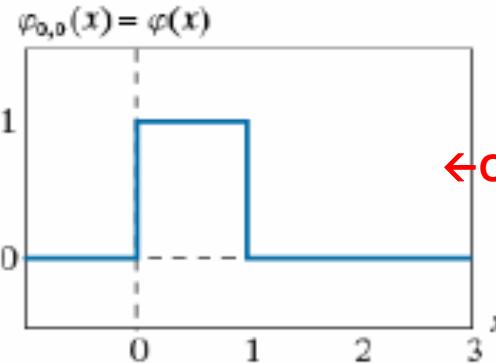
Example: Haar Scaling Function

$$\varphi_{j,k}(x) = 2^{j/2} \varphi(2^j x - k)$$

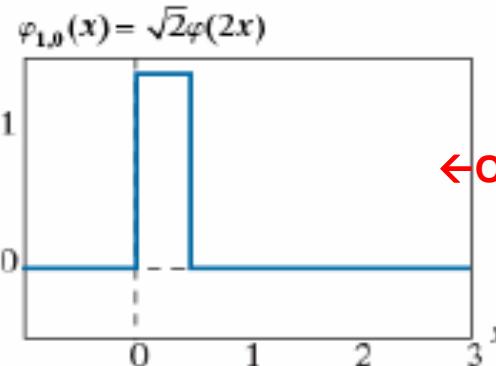
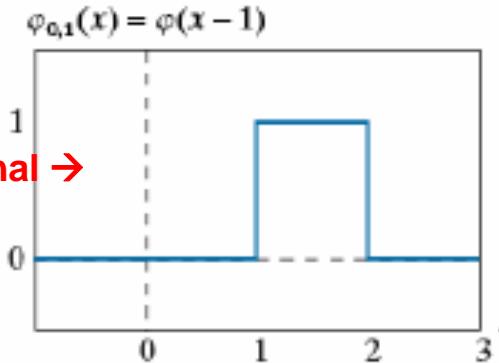


- As j increases, the functions become narrower → scaling

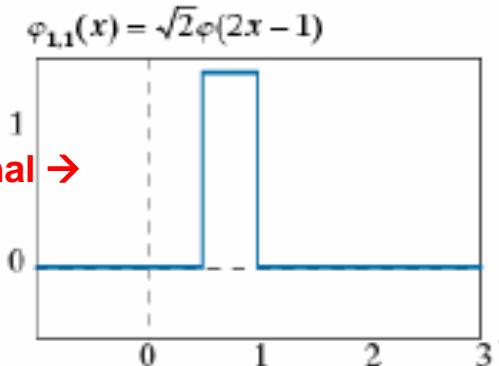
Scaling Functions (cont...)



←Orthogonal →

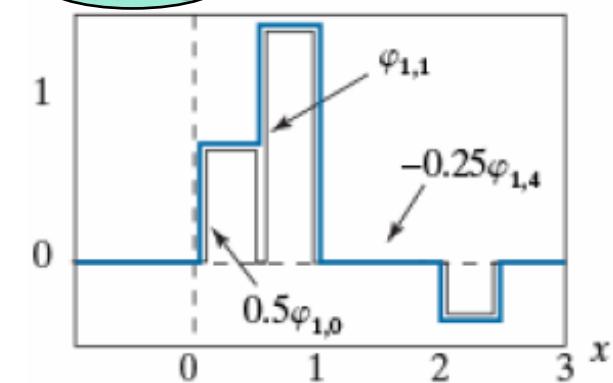


←Orthogonal →



Notice flat-top $f(x)$
→ Practical

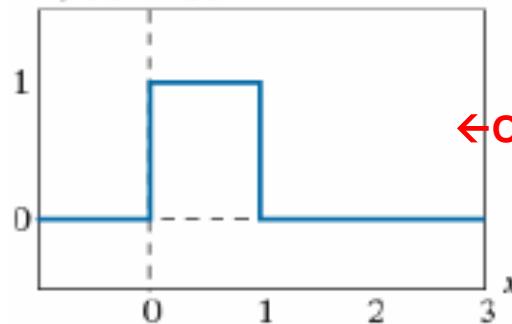
$$f(x) \in V_1$$



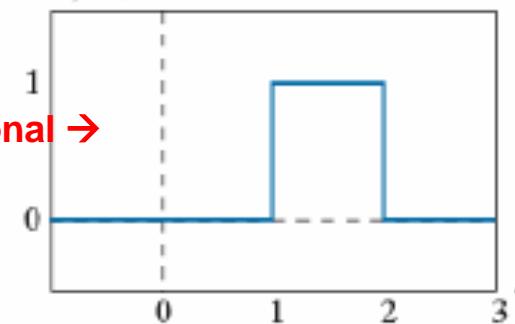
Function $f(x)$ at right does not belong to V_0 because the V_0 expansion functions are too coarse to represent it.
Higher resolution functions are required for $f(x)$.

Scaling Functions (cont...)

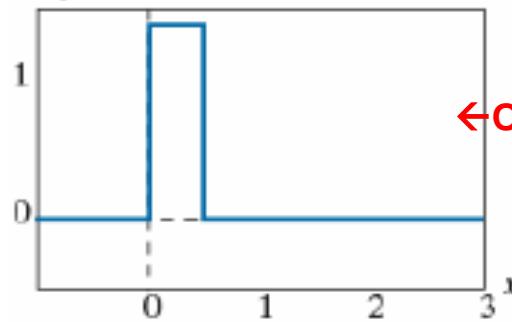
$$\varphi_{0,0}(x) = \varphi(x)$$



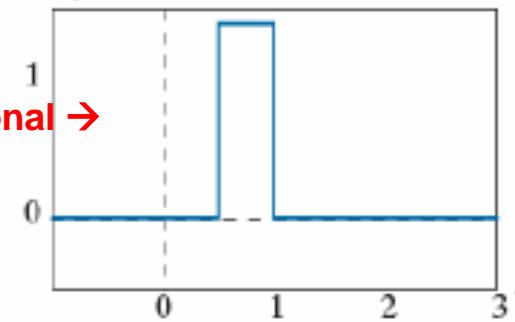
$$\varphi_{0,1}(x) = \varphi(x - 1)$$



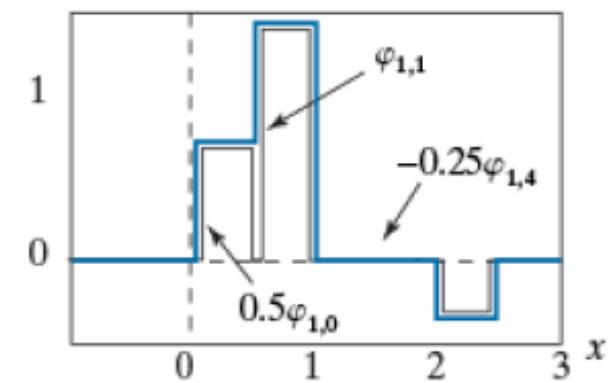
$$\varphi_{1,0}(x) = \sqrt{2}\varphi(2x)$$



$$\varphi_{1,1}(x) = \sqrt{2}\varphi(2x - 1)$$



$$f(x) \in V_1$$



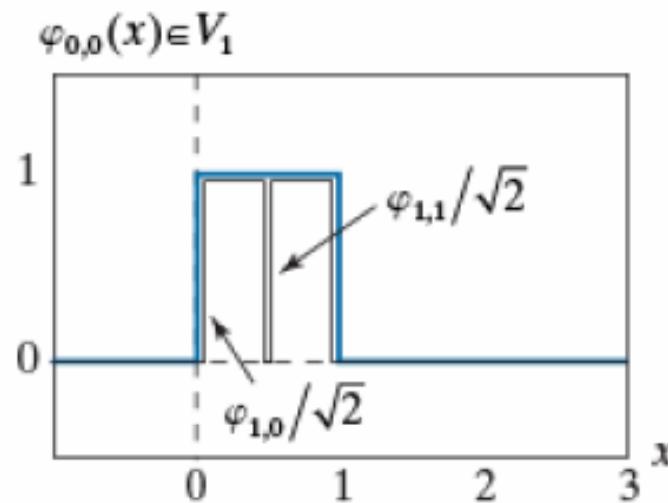
Indeed, $f(x) \in V_1$

$$f(x) = 0.5\varphi_{1,0}(x) + \varphi_{1,1}(x) - 0.25\varphi_{1,4}(x)$$



Multiresolution Analysis Scaling Functions (cont...)

Note also that $\varphi_{0,0}(x)$ may be decomposed as a sum of V_1 expansion functions.



$$\varphi_{0,0}(x) = \frac{1}{\sqrt{2}} \varphi_{1,0}(x) + \frac{1}{\sqrt{2}} \varphi_{1,1}(x) = \varphi(2x) + \varphi(2x - 1)$$



- In a similar manner, any V_0 expansion function may be decomposed as a sum of V_1 expansion functions:

$$\varphi_{0,k}(x) = \frac{1}{\sqrt{2}} \varphi_{1,2k}(x) + \frac{1}{\sqrt{2}} \varphi_{1,2k+1}(x)$$

- Therefore, if $f(x) \in V_0$, then $f(x) \in V_1$.
- This is because all V_0 expansion functions are contained in V_1 . Mathematically, we say that V_0 is a subspace of V_1 :

$$V_0 \subset V_1$$

- The simple scaling function in the preceding example obeys the four fundamental requirements of Multiresolution analysis [Mallat 1989].
- MRA Requirement 1: *The scaling function is orthogonal to its integer translates.*
 - Easy to see for the Haar function.
 - Hard to satisfy for functions with support different than $[0, 1]$.

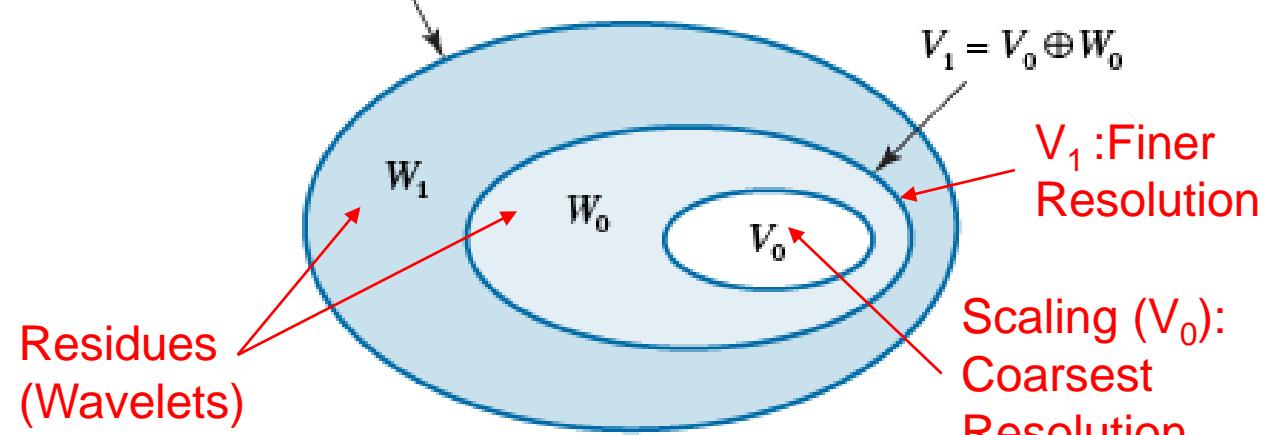
- **MRA Requirement 2:** The subspaces spanned by the scaling function at low scales are nested within those spanned at higher scales.
- This implies that:

$$V_{-\infty} \subset \cdots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots \subset V_\infty \quad (6-123)$$

$$V_2 = V_1 \oplus W_1 = V_0 \oplus W_0 \oplus W_1$$

\oplus Denotes Union of Sets

FIGURE 6.20
The relationship between scaling and wavelet function spaces.



- Moreover, if $f(x) \in V_j$, then $f(2x) \in V_{j+1}$. Compressed:
Need Finer
Resolution
- The fact that the Haar scaling function satisfies this requirement is not an indication that any function with support of width 1 satisfies the condition.
- For instance, the simple function:

$$\varphi(x) = \begin{cases} 1, & 0.25 \leq x \leq 0.75 \\ 0, & \text{otherwise} \end{cases}$$

is not a valid scaling function for MRA (violates MRA-2 - See problem 6.34).

- **MRA Requirement 3:** The only common function to all subspaces V_j is $f(x) = 0$.
- In the coarsest possible expansion, the only representable function is the **function with no information** $f(x) = 0$. That is: $j \rightarrow -\infty, V_{-\infty} = \{0\}$
- **MRA Requirement 4:** Any function may be represented with arbitrary precision.
- This means that in the limit: $j \rightarrow \infty, V_\infty = \{L^2(\mathbb{R})\}$
 $\{L^2(\mathbb{R})\}$: Set of all measurable square-integrable functions (6–124)

Scaling Functions (cont...)

- Father Scaling Function: $\varphi_{0,0}(x) = \varphi(x)$
 - Under MRA conditions, $\varphi(x)$ can be expressed as a linear combination of double resolution copies of itself:
- $$\varphi(x) = \sum_{k \in \mathbb{Z}} h_\varphi(k) \sqrt{2} \varphi(2x - k) \quad \text{Analogous to Fourier Series Expansion} \quad (6-125)$$
- The coefficients $h_\varphi(n)$, are called the **scaling function coefficients**.
 - This fundamental equation is called the **Refinement Equation**, or the **Dilation Equation**.

- How to find the scaling functions?
 - For Orthonormal Scaling Functions,

$$h_\varphi(k) = \langle \varphi(x), \sqrt{2} \varphi(2x - k) \rangle \quad \text{Dot-Product} \quad (6-126)$$

Note:

- Choice of reference V_0 is arbitrary
- We can start at any resolution level.

Haar Scaling Functions (cont...)

Example (Haar): The scaling function coefficients, $h_\varphi(n)$ for the Haar function are the elements of the first row of matrix \mathbf{A}_H for $N = 2$:

$$h_\varphi(0) = h_\varphi(1) = \frac{1}{\sqrt{2}}$$

Thus, the **Refinement Equation**:

$$\begin{aligned}\varphi(x) &= \frac{1}{\sqrt{2}} [\sqrt{2} \varphi(2x)] + \frac{1}{\sqrt{2}} [\sqrt{2} \varphi(2x-1)] \\ &= \varphi(2x) + \varphi(2x-1)\end{aligned}$$

Wavelet Functions

- Given a Father Scaling Function $\varphi(n)$, that meets the MRA criteria there exists a Wavelet Function $\psi(x)$ whose integer translates and binary scalings,

$$\psi_{j,k}(x) = 2^{j/2} \psi(2^j x - k), \quad j, k \in \mathbb{Z} \quad (6-127)$$

span the difference between any two adjacent scaling subspaces V_j and V_{j+1} .

- Let W_{j_0} denote the function space spanned by $\{\psi_{j_0,k}(x) | k \in \mathbb{Z}\}$ then

$$V_{j_0+1} = V_{j_0} \oplus W_{j_0} \quad (6-128)$$

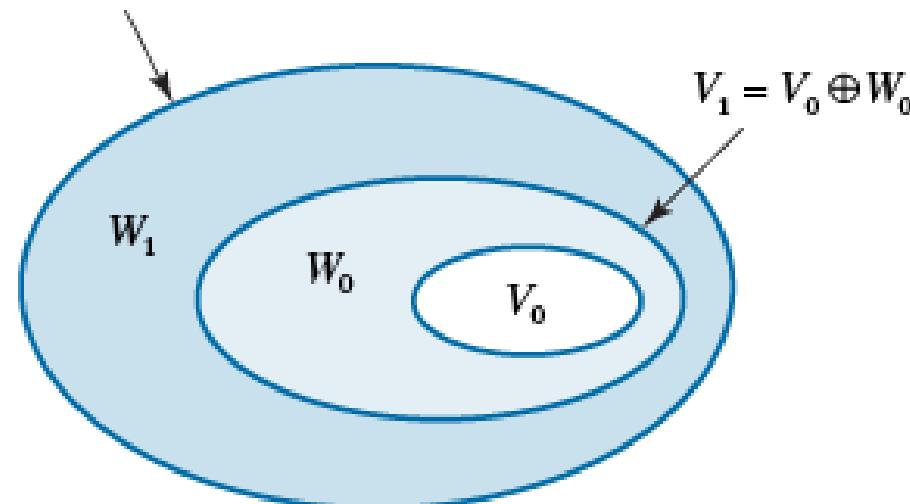
\oplus Denotes Union of Sets

- The **orthogonal complement** of V_{j_0} in V_{j_0+1} is W_{j_0}
- Scaling functions that are the basis of V_{j_0} are orthogonal to wavelet functions that are the basis of W_{j_0}

$$\langle \varphi_{j_0,k}(x), \psi_{j_0,l}(x) \rangle = 0, \text{ for } k \neq l \quad (6-129)$$

FIGURE 6.20
The relationship between scaling and wavelet function spaces.

$$V_2 = V_1 \oplus W_1 = V_0 \oplus W_0 \oplus W_1$$

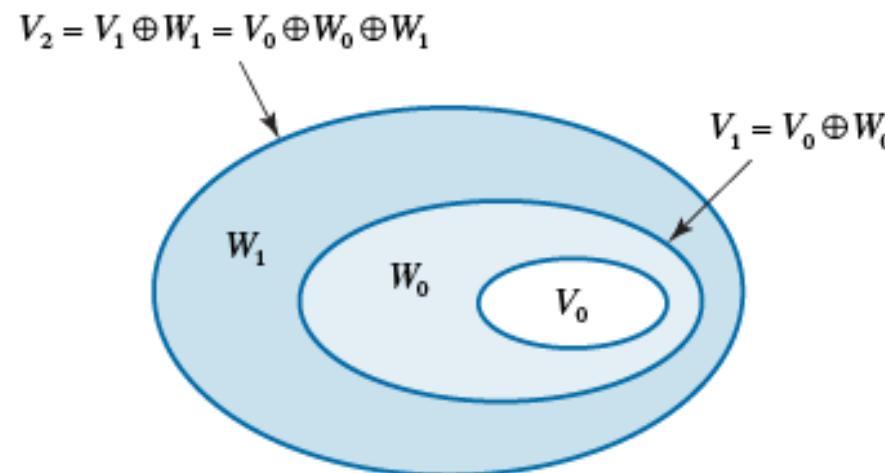


- If $f(x) \in V_1$ but $f(x) \notin V_0$, its expansion using

$$L^2(\mathbb{R}) = V_0 \oplus W_0$$

contains an approximation using scaling functions V_0 ; and wavelets from W_0 would encode the difference between this approximation and the actual function.

FIGURE 6.20
The relationship between scaling and wavelet function spaces.



Multiresolution Analysis Wavelet Functions (cont...)

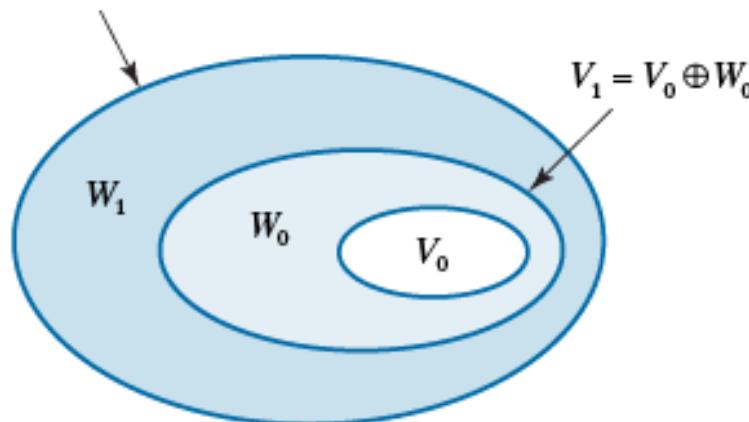
The representation may be generalized to yield

$$L^2(\mathbb{R}) = V_{j_0} \oplus W_{j_0} \oplus W_{j_0+1} \oplus \dots$$

starting from an arbitrary scale j_0 and adding the appropriate wavelet functions that capture the difference between the coarse scale representation and the actual function.

$$V_2 = V_1 \oplus W_1 = V_0 \oplus W_0 \oplus W_1$$

FIGURE 6.20
The relationship between scaling and wavelet function spaces.



- Any wavelet function, like its scaling function counterpart, resides in the space spanned by the next higher resolution level
- Therefore, it can be expressed as a weighted sum of shifted, **double-resolution scaling functions**:

$$\psi(x) = \sum_k h_\psi(k) \sqrt{2} \varphi(2x - k) \quad (6-130)$$

- $h_\psi(n)$: **Wavelet function coefficients**
- Scaling function coefficients and wavelet function coefficients are related [Burrus et al, 1998] :

$$h_\psi(k) = (-1)^k h_\varphi(1-k) \quad (6-131)$$

Example: The Haar scaling function coefficients were defined as

$$h_\varphi(0) = h_\varphi(1) = \frac{1}{\sqrt{2}}$$

- The corresponding wavelet coefficients:

$$h_\psi(0) = (-1)^0 h_\varphi(1-0) = \frac{1}{\sqrt{2}}$$

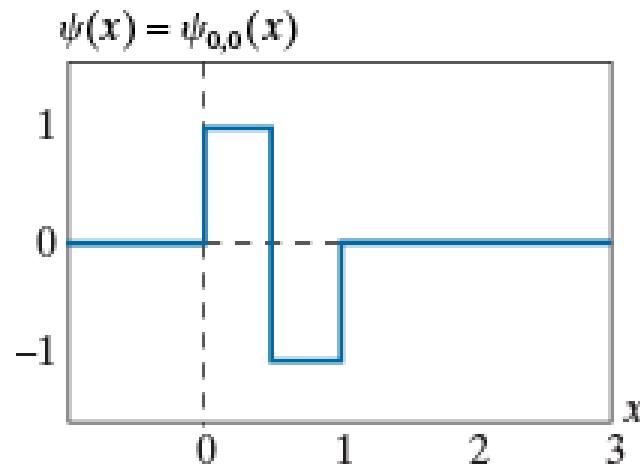
$$h_\psi(1) = (-1)^1 h_\varphi(1-1) = -\frac{1}{\sqrt{2}}$$

- These coefficients are the elements of second row of Haar transformation matrix \mathbf{A}_H for $N=2$

Multiresolution Analysis Wavelet Functions (cont...)

Substituting this result into: $\psi(x) = \sum_n h_\psi(n) \sqrt{2} \varphi(2x - n)$

we get, $\psi(x) = \varphi(2x) - \varphi(2x - 1) = \begin{cases} 1 & 0 \leq x \leq 0.5 \\ -1 & 0.5 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases}$

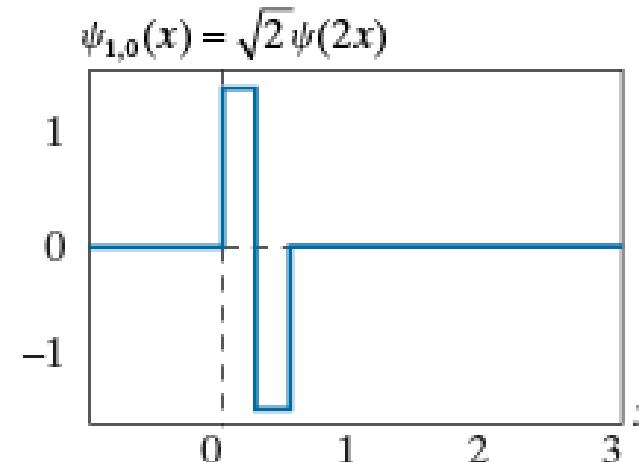
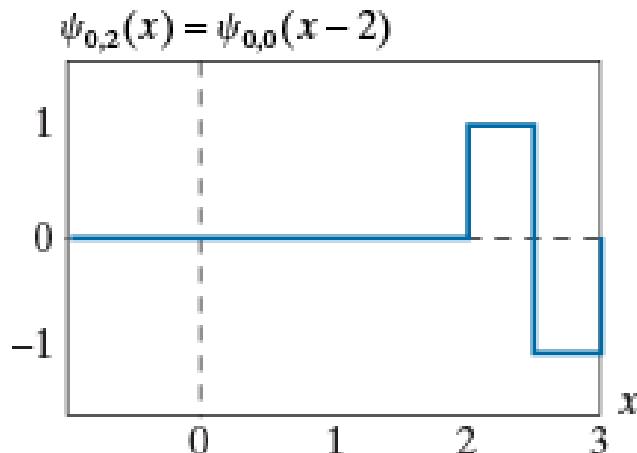




Multiresolution Analysis Wavelet Functions (cont...)

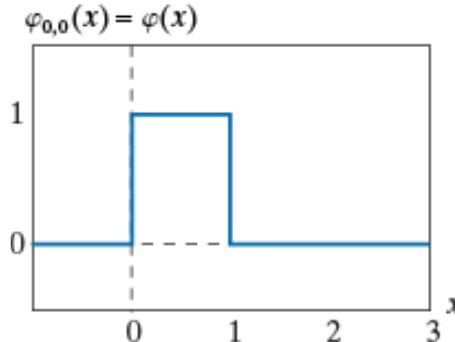
Using $\psi_{j,k}(x) = 2^{j/2} \psi(2^j x - k)$, $j, k \in \mathbb{Z}$

The universe of translated and scaled Haar wavelets can now be generated.

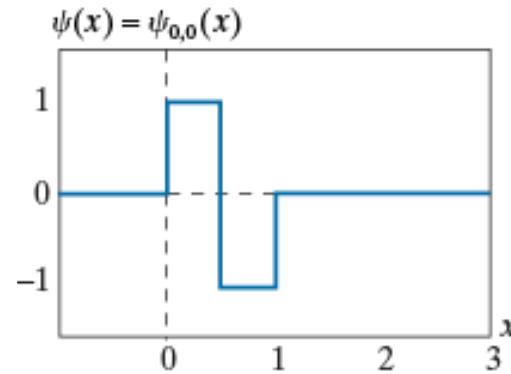


Multiresolution Analysis Wavelet Functions (cont...)

Any function $f(x) \in V_1$ may be expressed by the scaling function $\varphi(x)$ describing the coarse form



and the wavelet function $\psi(x)$ describing the details that cannot be represented in V_0 by $\varphi(x)$.

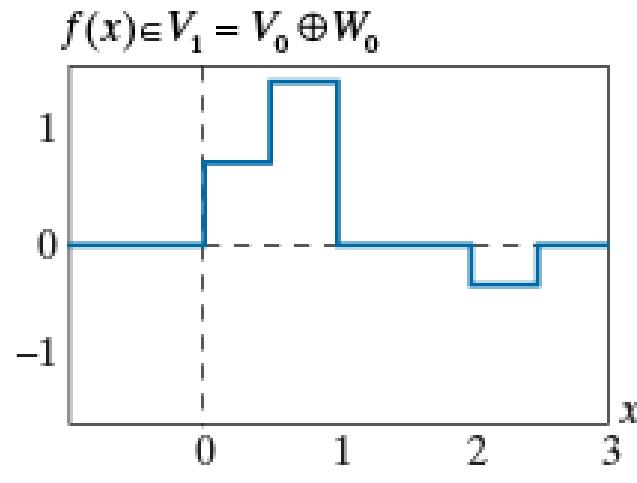


Note: $\varphi(x)$ is orthogonal to $\psi(x)$

Multiresolution Analysis Wavelet Functions (cont...)

Recall:

- The function of an earlier example $f(x) \in V_1$ but $f(x) \notin V_0$.
- This indicates that it could be expanded using V_0 to capture the coarse characteristics of the function and W_0 to encode the details that cannot be represented by V_0 .





Multiresolution Analysis Wavelet Functions (cont...)

approximation + detail

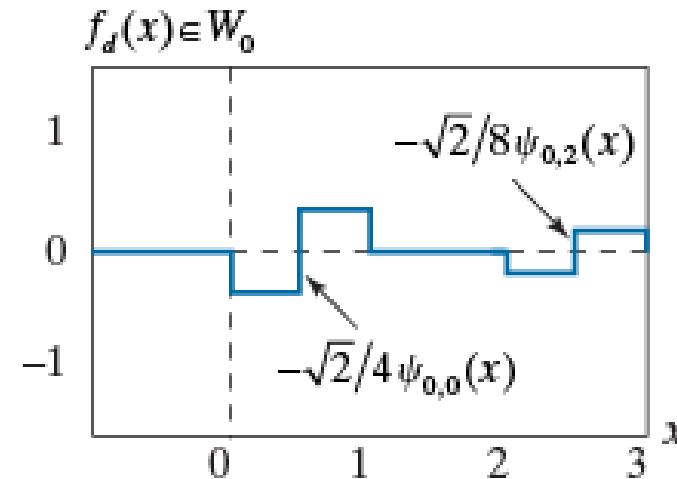
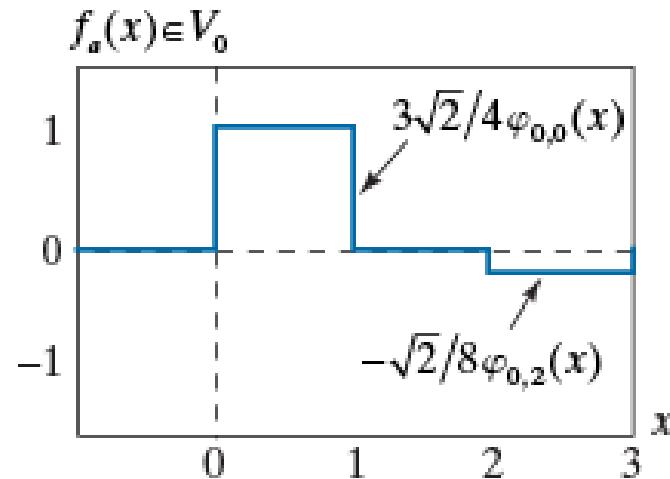
Approximation

$$f(x) = f_a(x) + f_d(x)$$

Detail

$$f_a(x) = \frac{3\sqrt{2}}{4} \varphi_{0,0}(x) - \frac{\sqrt{2}}{8} \varphi_{0,2}(x)$$

$$f_d(x) = -\frac{\sqrt{2}}{4} \psi_{0,0}(x) - \frac{\sqrt{2}}{8} \psi_{0,2}(x)$$



- Notice the equivalence to low pass and high pass filtering.
- Also notice the analogy with fundamental and harmonics

Wavelet Series Expansion

- A **continuous signal** $f(x)$ may be represented by a scaling function in a subspace V_{j_0} and a number of wavelet functions in subspaces $W_{j_0}, W_{j_0+1}, W_{j_0+2}, \dots$

$$f(x) = \sum_k c_{j_0}(k) \varphi_{j_0,k}(x) + \sum_{j=j_0}^{\infty} \sum_k d_j(k) \psi_{j,k}(x)$$

j_0 : Level (6-133)
 k : Translations

**Scaling
coefficients:**

$$c_{j_0}(k) = \langle f(x), \varphi_{j_0,k}(x) \rangle = \int f(x) \varphi_{j_0,k}(x) dx$$

(6-134)

**Detail (Wavelet)
Coefficients:**

$$d_j(k) = \langle f(x), \psi_{j,k}(x) \rangle = \int f(x) \psi_{j,k}(x) dx$$

(6-135)

1-D Wavelet Transforms

The Wavelet Series (cont...)

Example: Use Haar wavelets starting from $j_0=0$ to compute the Wavelet Series of

$$y = f(x) = \begin{cases} x^2, & 0 \leq x \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

**Scaling
coefficients:**

$$c_0(0) = \int_0^1 x^2 \varphi_{0,0}(x) dx = \int_0^1 x^2 dx = \frac{1}{3}$$

- There is only one scaling coefficient for $k=0$.
- Integer translations of the scaling function do not overlap with the signal ($0 \leq x \leq 1$).

1-D Wavelet Transforms

The Wavelet Series (cont...)

Example (continued): Detail (Wavelet) coefficients

$$d_0(0) = \int_0^1 x^2 \psi_{0,0}(x) dx = \int_0^{0.5} x^2 dx - \int_{0.5}^1 x^2 dx = -\frac{1}{4}$$

$j = j_0 = 0, k = 0$

$$d_1(0) = \int_0^1 x^2 \psi_{1,0}(x) dx = \int_0^{0.25} x^2 \sqrt{2} dx - \int_{0.25}^{0.5} x^2 \sqrt{2} dx = -\frac{\sqrt{2}}{32}$$

$j = 1, k = 0$

$$d_1(1) = \int_0^1 x^2 \psi_{1,1}(x) dx = \int_{0.5}^{0.75} x^2 \sqrt{2} dx - \int_{0.75}^1 x^2 \sqrt{2} dx = -\frac{3\sqrt{2}}{32}$$

$j = 1, k = 1$

1-D Wavelet Transforms

The Wavelet Series (cont...)

Example (continued): Substituting these values:

$$f(x) = \underbrace{\frac{1}{3} \varphi_{0,0}(x)}_{V_0} + \underbrace{\left[-\frac{1}{4} \psi_{0,0}(x) \right]}_{W_0} + \underbrace{\left[-\frac{\sqrt{2}}{32} \psi_{1,0}(x) - \frac{\sqrt{2}}{32} \psi_{1,1}(x) \right]}_{W_1} + \dots$$

$V_1 = V_0 \oplus W_0$

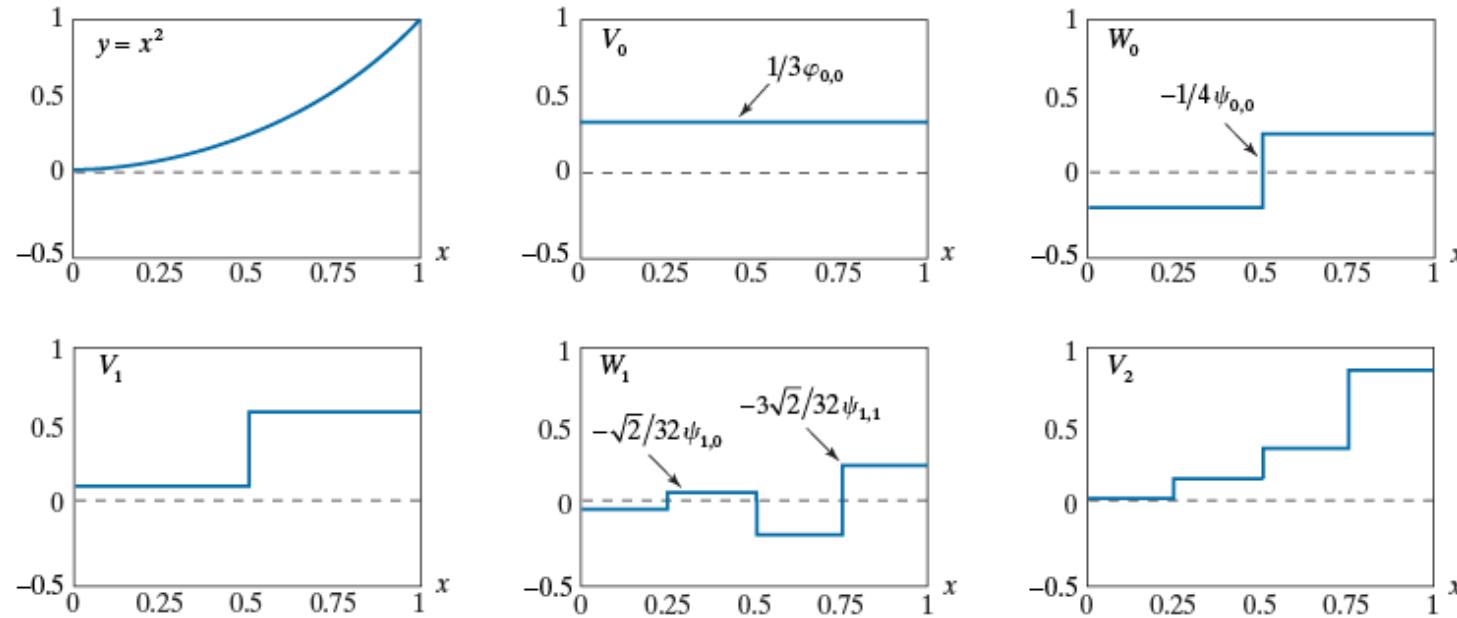
$V_2 = V_1 \oplus W_1 = V_0 \oplus W_0 \oplus W_1$

1-D Wavelet Transforms

The Wavelet Series (cont...)

Example (continued):

- The **scaling function** approximates the signal by its **average value**.
- Each **wavelet subspace** adds a level of **detail** in the wavelet series representation of the signal.



a
b
c
d
e
f

FIGURE 6.22 A wavelet series expansion of $y = x^2$ using Haar wavelets.

Thus far, we have done
Continuous case. Discrete →

- If the signal $f(x)$ is discrete of length N (= power of 2), its DWT is computed as:

$$f(x) = \frac{1}{\sqrt{N}} \left[T_\varphi(0,0)\varphi(x) + \sum_{j=0}^{J-1} \sum_{k=0}^{2^j-1} T_\psi(j,k)\psi_{j,k}(x) \right] \quad (6-136)$$

- Approximation Coefficients:**

$$T_\varphi(0,0) = \langle f(x), \varphi_{0,0}(x) \rangle = \langle f(x), \varphi(x) \rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} f(x) \varphi^*(x) \quad (6-137)$$

- Detail Coefficients (Wavelets):** Discrete Dot-Product

$$T_\psi(j,k) = \langle f(x), \psi_{j,k}(x) \rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} f(x) \psi_{j,k}^*(x) \quad (6-138)$$

for $j = 0, 1, 2, \dots, J-1$, $k = 0, 1, 2, \dots, 2^j - 1$

Note: Summation,
not integration

$$f(x) = \frac{1}{\sqrt{N}} \left[T_{\varphi}(0,0)\varphi(x) + \sum_{j=0}^{J-1} \sum_{k=0}^{2^j-1} T_{\psi}(j,k)\psi_{j,k}(x) \right] \quad (6-136)$$

- We take N equally spaced samples over the support of the scaling and wavelet functions.
 → Normally, we let $j_0=0$ and $N=2^j$.
- Therefore, the summations are performed over
 $x = 0, 1, 2, \dots, N-1, \quad j = 0, 1, 2, \dots, J-1, \quad k = 0, 1, 2, \dots, 2^j - 1$
- For Haar wavelets, the discretized scaling and wavelet functions correspond to the rows of the $N \times N$ Haar matrix.

1-D Wavelet Transforms
The Discrete Wavelet Transform (cont...)

Example: $f(n) = \{1, 4, -3, 0\}$.

- We will compute the DWT of the signal using the Haar scaling function and the corresponding wavelet (detail) functions.
- Here, $N = 4 = 2^J$, $J = 2$ and with $j_0 = 0$.
- The summations are performed over $n = 0, 1, 2, 3$, $j = 0, 1$ and $k = 0$ for $j = 0$ or $k = 0, 1$ for $j = 1$.
- Values of sampled scaling and wavelet functions are the elements of the rows of \mathbf{A}_H for $N = 4$ (slide-41)

$$\mathbf{A}_H = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

1-D Wavelet Transforms

The Discrete Wavelet Transform (cont...)

Example (continued): $f(x) = \{1, 4, -3, 0\}$.

$$\mathbf{A}_H = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

$$T_\varphi(0,0) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} f(x) \varphi^*(x) = \frac{1}{2} \sum_{x=0}^3 f(x) \varphi(x) \Leftrightarrow$$

$$T_\varphi(0,0) = \frac{1}{2} [(1)(1) + (4)(1) - (3)(1) + (0)(1)] = 1$$

Approximation
coefficient

$$T_\psi(0,0) = \frac{1}{2} [(1)(1) + (4)(1) - (3)(-1) + (0)(-1)] = 4$$

$$T_\psi(1,0) = \frac{1}{2} [(1)(\sqrt{2}) + (4)(-\sqrt{2}) - (3)(0) + (0)(0)] = -1.5\sqrt{2}$$

$$T_\psi(1,1) = \frac{1}{2} [(1)(0) + (4)(0) - (3)(\sqrt{2}) + (0)(-\sqrt{2})] = -1.5\sqrt{2}$$

Detail
coefficients

Example (continued): $f(n)=\{1, 4, -3, 0\}$.

The DWT relative to the Haar scaling and wavelet functions is

$$\{1, 4, -1.5\sqrt{2}, -1.5\sqrt{2}\}$$

To reconstruct the signal, we compute

$$f(x) = \frac{1}{\sqrt{N}} \left[T_\varphi(0,0)\varphi(x) + \sum_{j=0}^{J-1} \sum_{k=0}^{2^j-1} T_\psi(j,k)\psi_{j,k}(x) \right]$$

$$= \frac{1}{2} \left[T_\varphi(0,0)\varphi(x) + T_\psi(0,0)\psi_{0,0}(x) + T_\psi(1,0)\psi_{1,0}(x) + T_\psi(1,1)\psi_{1,1}(x) \right]$$

for $x = 0, 1, 2, 3$. Notice that we could have started from a different approximation level $j_0 \neq 0$.

The Fast Wavelet Transform

- FWT: A computationally efficient implementation of the DWT [Mallat 1989]
- Resembles **subband coding** (discussed next)
- Recall the *multiresolution refinement equation*

$$\varphi(x) = \sum_{k \in \mathbb{Z}} h_\varphi(k) \sqrt{2} \varphi(2x - k) \quad (\text{slide-56}) \quad (6-125)$$

- **Scale x by 2^j , translate by k**

$$\varphi(2^j x - k) = \sum_n h_\varphi(n) \sqrt{2} \varphi\left(2(2^j x - k) - n\right)$$

$$= \sum_n h_\varphi(n) \sqrt{2} \varphi(2^{j+1} x - 2k - n)$$

$$= \sum_m h_\varphi(m - 2k) \sqrt{2} \varphi(2^{j+1} x - m)$$

- **Let $m = 2k + n$**

1-D Wavelet Transforms

The Fast Wavelet Transform (cont...)

- Substituting (6.121:slide-45) into (6-134:slide-70)

$$\begin{aligned}
 c_j(k) &= \left\langle f(x), \varphi_{j,k}(x) \right\rangle = \int f(x) \varphi_{j,k}(x) dx \quad \text{Continuous case} \\
 &= \int f(x) 2^{j/2} \varphi(2^j x - k) dx \quad \text{Using (6-121)} \\
 &= \int f(x) 2^{j/2} \sum_m h_\varphi(m - 2k) \sqrt{2} \varphi(2^{j+1} x - m) dx \\
 &= \sum_m h_\varphi(m - 2k) \left[\int f(x) 2^{(j+1)/2} \varphi(2^{j+1} x - m) dx \right] \\
 \therefore c_j(k) &= \sum_m h_\varphi(m - 2k) c_{j+1}(m) \quad \text{Recursive!!} \quad (6-139)
 \end{aligned}$$

- A similar sequence of operations leads to the recursive computation of the **wavelet functions**

$$d_j(k) = \sum_m h_\psi(m - 2k) c_{j+1}(m) \quad (6-140)$$

1-D Wavelet Transforms

The Fast Wavelet Transform (cont...)

Further analysis leads to the following important equations relating the DWT coefficients of adjacent scales:

Scaling coefficients

$$T_\varphi(j, k) = \sum_n h_\varphi(n - 2k) T_\varphi(j+1, n) \quad (6-141)$$

Wavelet
coefficients

$$T_\psi(j, k) = \sum_n h_\psi(n - 2k) T_\varphi(j+1, n) \quad (6-142)$$

- Both the scaling and the wavelet coefficients of a certain scale j may be obtained by
 - Convolution of the scaling coefficients of the next scale $j+1$ (the **finer scale**), with the order-reversed scaling and wavelet vectors $h_\varphi(-n)$ and $h_\psi(-n)$.
 - Subsample the result (due to $2k$)
- The complexity is $O(N)$ ← Less than FFT!

1-D Wavelet Transforms

The Fast Wavelet Transform (cont...)

$$T_\varphi(j, k) = T_\varphi(j+1, n) * h_\varphi(-n) \quad (6-143)$$

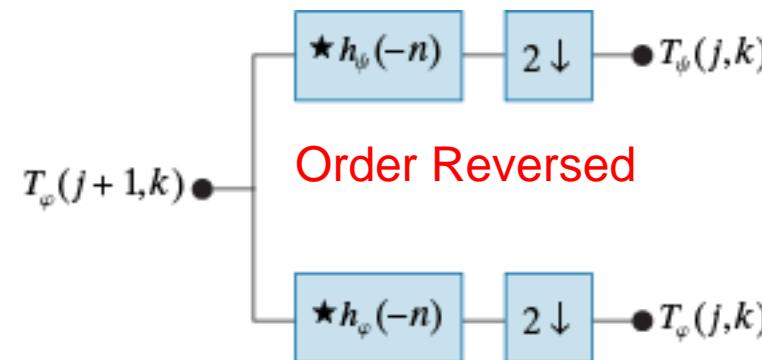
$$T_\psi(j, k) = T_\varphi(j+1, n) * h_\psi(-n) \quad (6-144)$$

- The **convolutions** are evaluated at non-negative even indices.
- This is equivalent to filtering and downsampling by 2.

$$y_{2\downarrow}(n) = y(2n) \quad \text{for } n = 0, 1, \dots \quad (6-145)$$

FIGURE 6.23

A FWT analysis filter bank for orthonormal filters. The \star and $2 \downarrow$ denote convolution and downsampling by 2, respectively.



1-D Wavelet Transforms

The Fast Wavelet Transform (cont...)

- The procedure may be iterated to create multistage structures of more scales.
- The highest scale coefficients are assumed to be the values of the signal itself $T_\varphi(J,k) = f(x)$

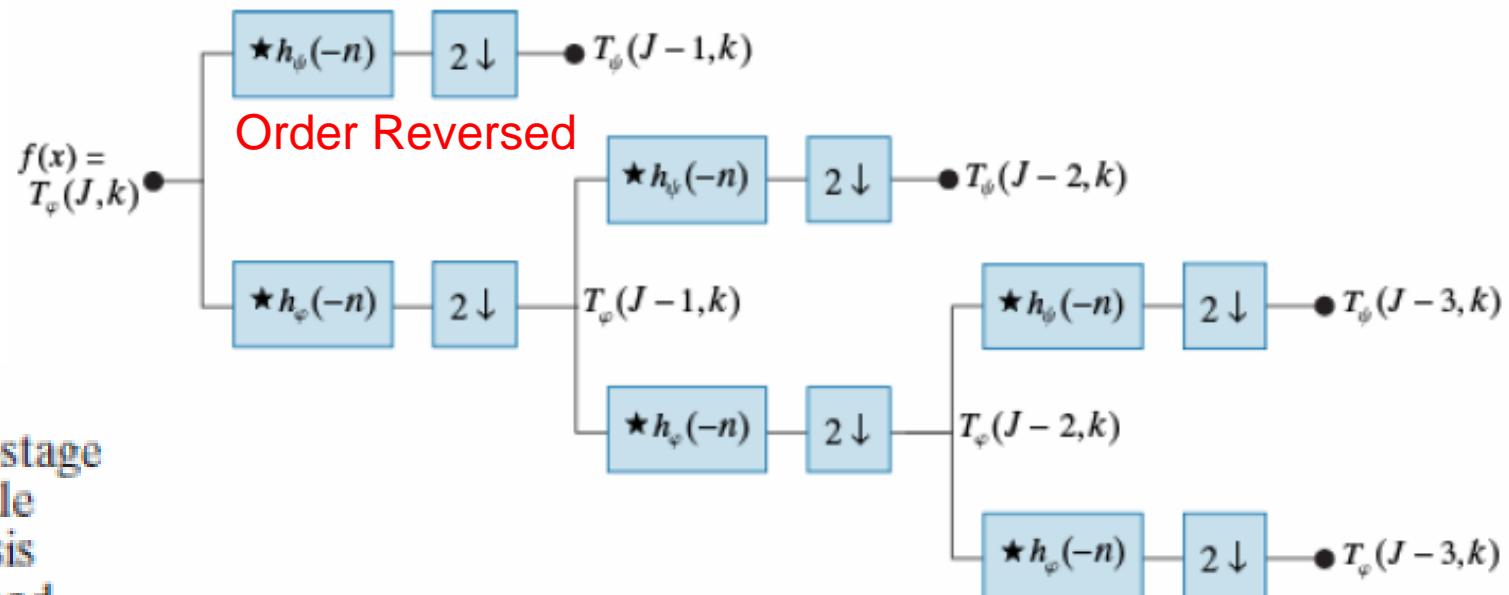


FIGURE 6.24

(a) A three-stage or three-scale FWT analysis filter bank and



1-D Wavelet Transforms

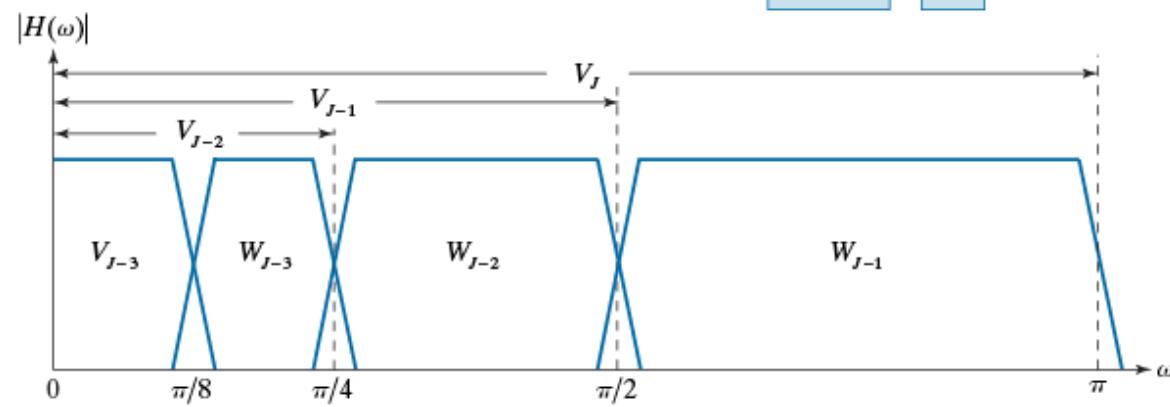
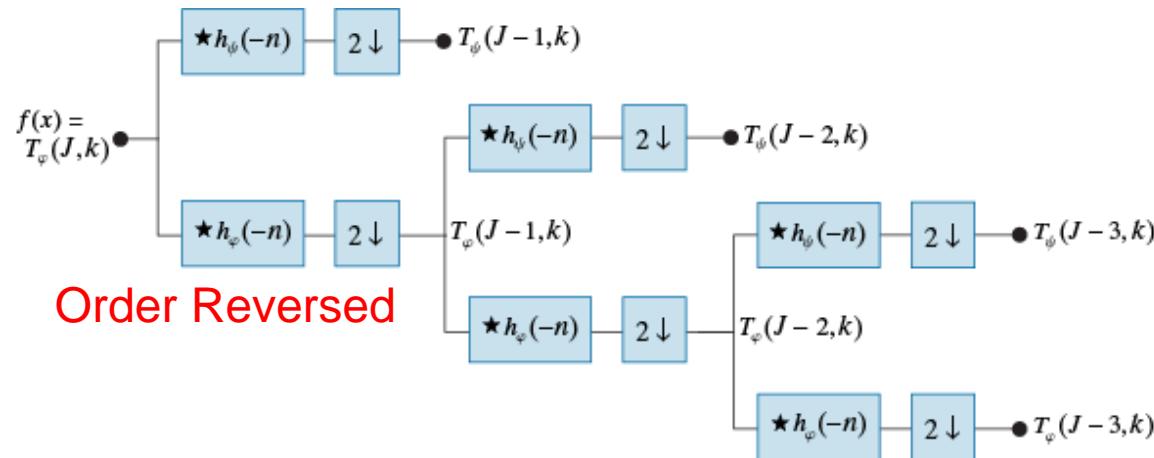
The Fast Wavelet Transform (cont...)

- The corresponding **frequency splitting** characteristics for the three-stage procedure:

a
b

FIGURE 6.24

(a) A three-stage or three-scale FWT analysis filter bank and (b) its frequency-splitting characteristics. Because of symmetry in the DFT of the filter's impulse response, it is common to display only the $[0, \pi]$ region.



1-D Wavelet Transforms

The Fast Wavelet Transform (cont...)

- Example 6.2 using FWT and 2×2 Haar (slide-40)
- Same results as in slide-78 using DWT, where 4×4 Haar was used

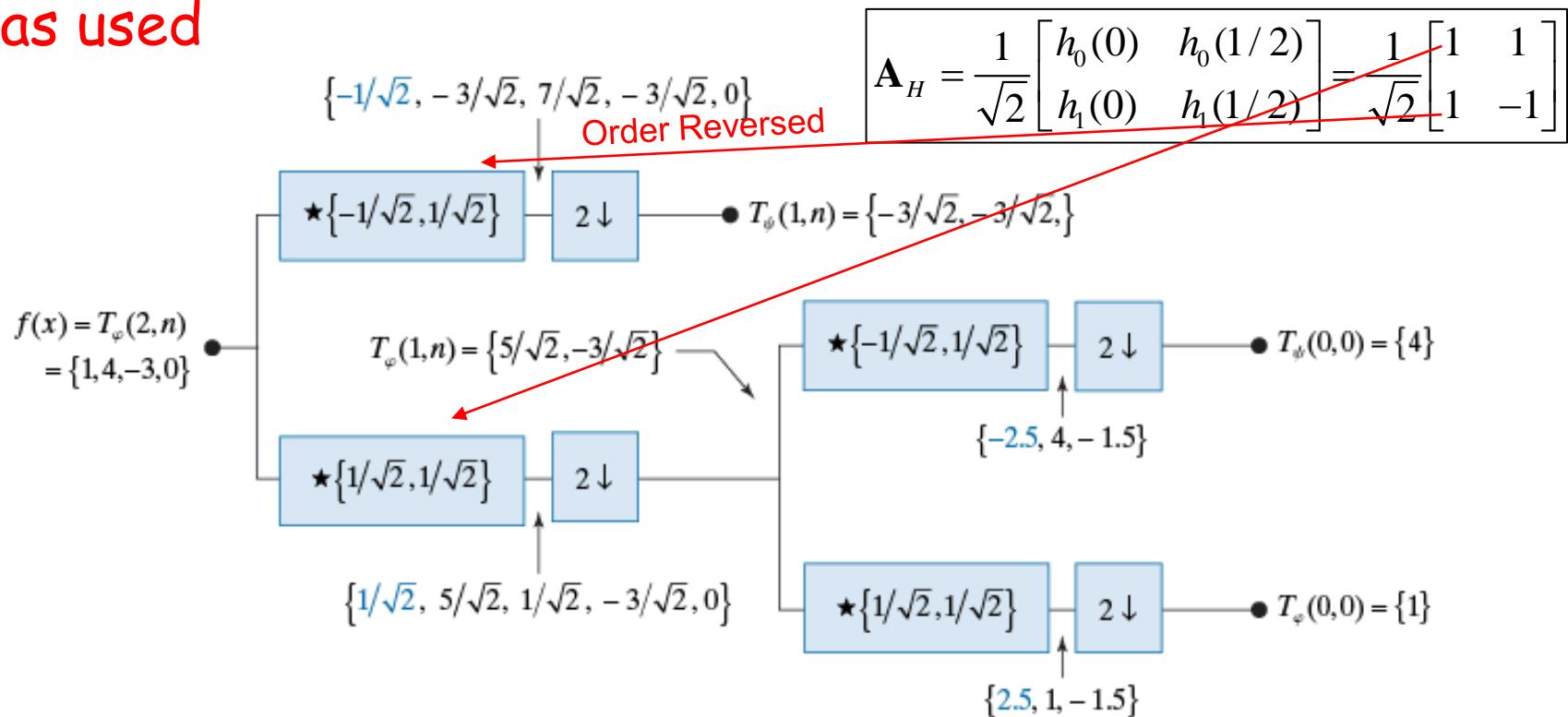


FIGURE 6.25 Computing a two-scale fast wavelet transform of sequence $\{1, 4, -3, 0\}$ using Haar scaling and wavelet coefficients.

Subband Coding

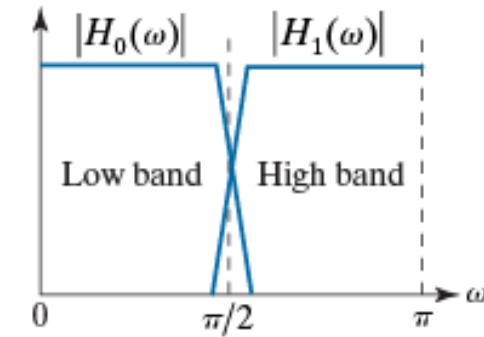
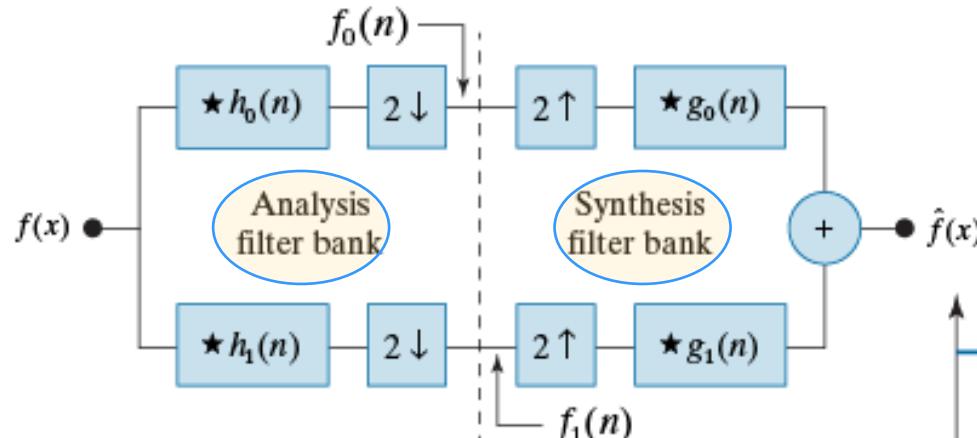
- The filters in Fig 6.23 through Fig 6.25 are FIR filters.
- These filters decompose images into a set of bandlimited components (subbands).
- The decomposition is carried out by digital filtering and downsampling.
- If the filters are properly selected the image may be reconstructed without error by filtering and upsampling.

Two-Band Subband Coding-Decoding

- **Analysis filters:** $h_0(n)$ and $h_1(n)$
 - Decompose $f(x)$ into half length sequences $f_0(n)$ and $f_1(n)$
- **Synthesis Filters:** $g_0(n)$ and $g_1(n)$
 - Reconstructs $\hat{f}(x)$ from upsampled versions of $f_0(n)$ and $f_1(n)$

a b

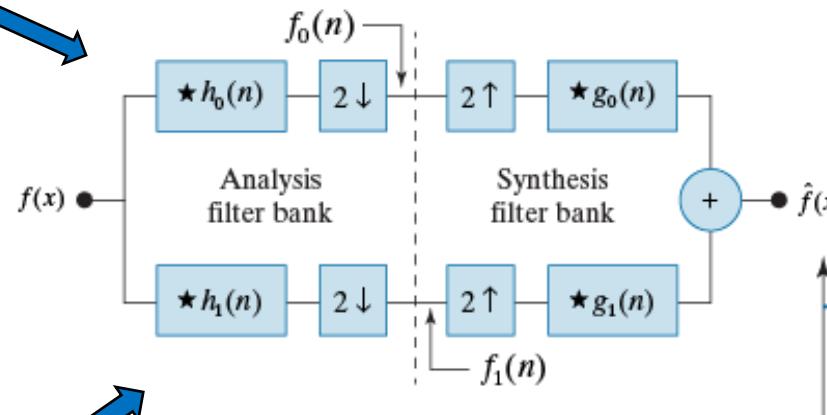
FIGURE 6.26
 (a) A two-band digital filtering system for subband coding and decoding and
 (b) its spectrum-splitting properties.



Two-Band Subband Coding-Decoding

A two-band subband coding

Approximation
filter (low pass)



a b

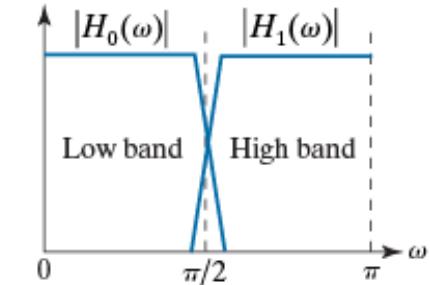
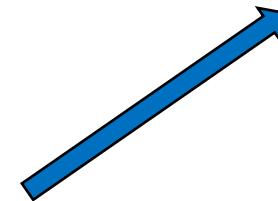


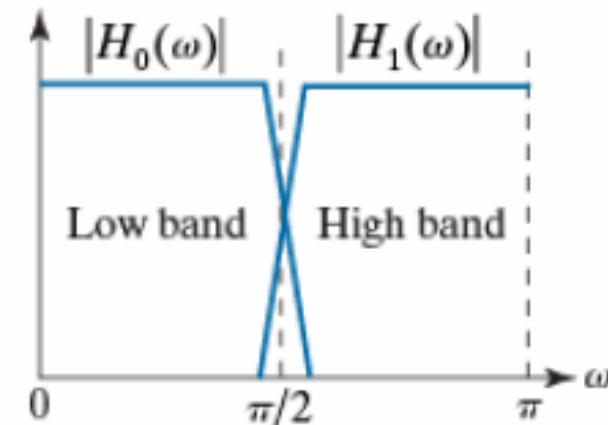
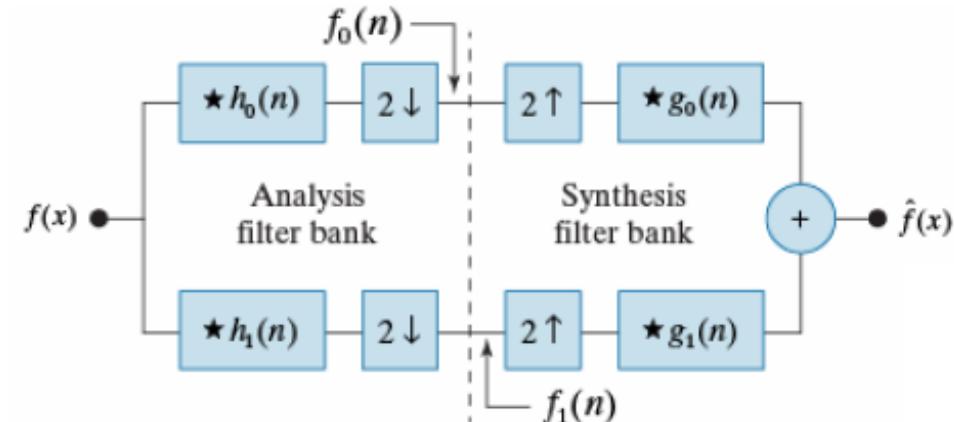
FIGURE 6.26
(a) A two-band digital filtering system for subband coding and decoding and
(b) its spectrum-splitting properties.

Detail filter (high pass)
- Difference between



Subband Coding (cont...)

- The goal of subband coding is to select the analysis and synthesis filters in order to have perfect reconstruction of the signal.
- It may be shown that the synthesis filters should be
 - Modulated versions of the analysis filters with one (and only one) synthesis filter being sign reversed of an analysis filter.



7.1.2 Subband Coding

- With Perfect Reconstruction filters:

$$\hat{f}(x) = f(x)$$

– Biorthogonal filters

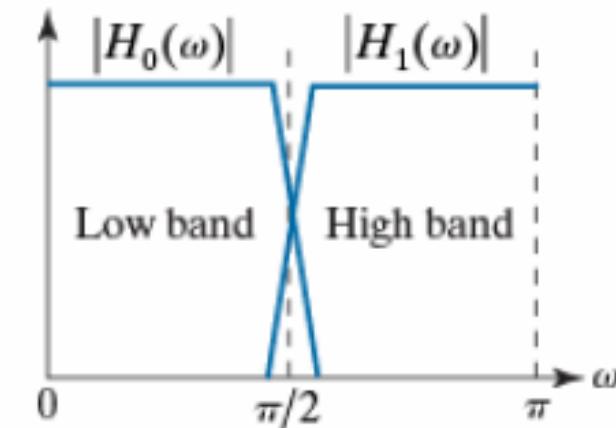
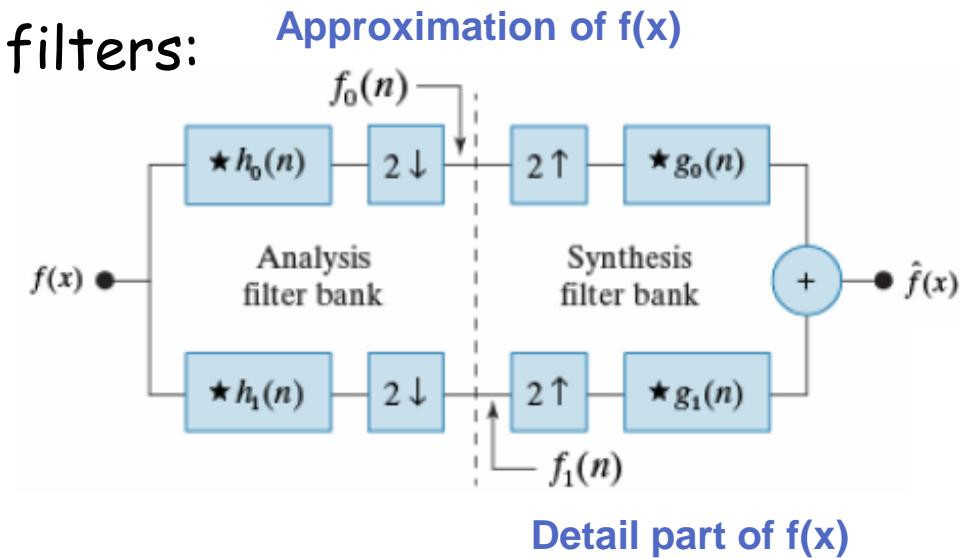
- Two prototypes required

$$g_0(n) = (-1)^n h_1(n)$$

$$g_1(n) = (-1)^{n+1} h_0(n)$$

– Orthonormal filters

- Given a single prototype filter, remaining three can be computed to satisfy the orthonormality constraint

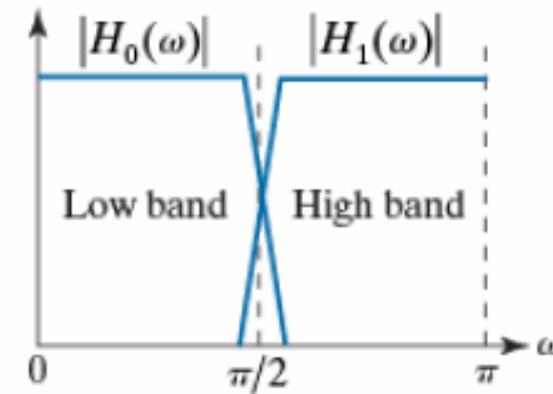
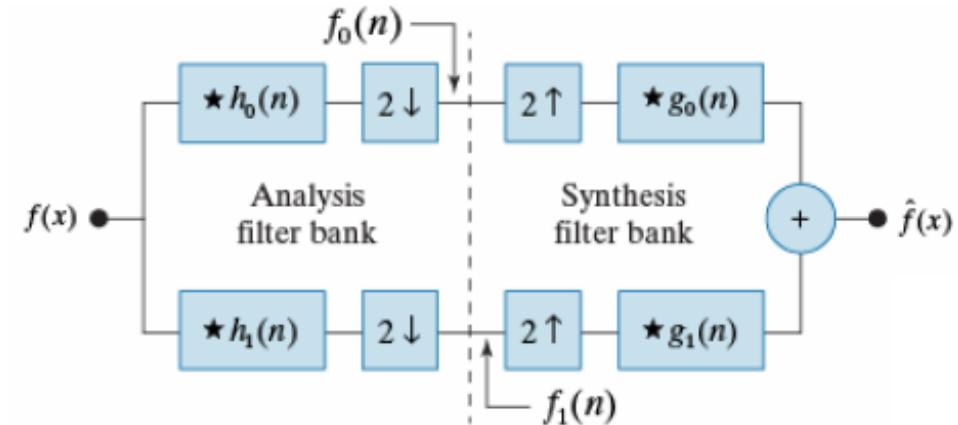


Subband Coding (cont...)

The analysis and synthesis filters should be related in one of the two ways:

$$g_0(n) = (-1)^n h_1(n)$$

$$g_1(n) = (-1)^{n+1} h_0(n)$$



These filters are called *cross-modulated*. (because related by modulation with diagonal opposite in block-diagram)

- Of special interest in subband coding are filters that go beyond biorthogonality and require to be *orthonormal* also (for FWT, and perfect Reconstruction; See 7.4) :

$$\langle g_i(n), g_j(n+2m) \rangle = \delta(i-j)\delta(m), \quad i, j = \{0,1\}$$

- Orthonormal filters satisfy the following conditions:

$$g_1(n) = (-1)^n g_0(K-1-n)$$

$$h_0(n) = g_0(K-1-n)$$

$$h_1(n) = g_1(K-1-n)$$

- K must be divisible by 2, i.e., the size of the filter should be even

Summary Notes:

- **Synthesis filters (g_0 and g_1):** Related by order reversal and modulation.
- **Analysis filters (h_0 and h_1):** Both order reversed versions of the synthesis filters.
- An **orthonormal filter bank** may be constructed by starting with the impulse response of g_0 which is called the **prototype**.
- Biorthogonal filter banks require **two prototypes**, h_0 and h_1 . Other two found using eqs. in slides 91-92
- 1-D orthonormal filters may be used as 2-D separable filters for subband image coding.

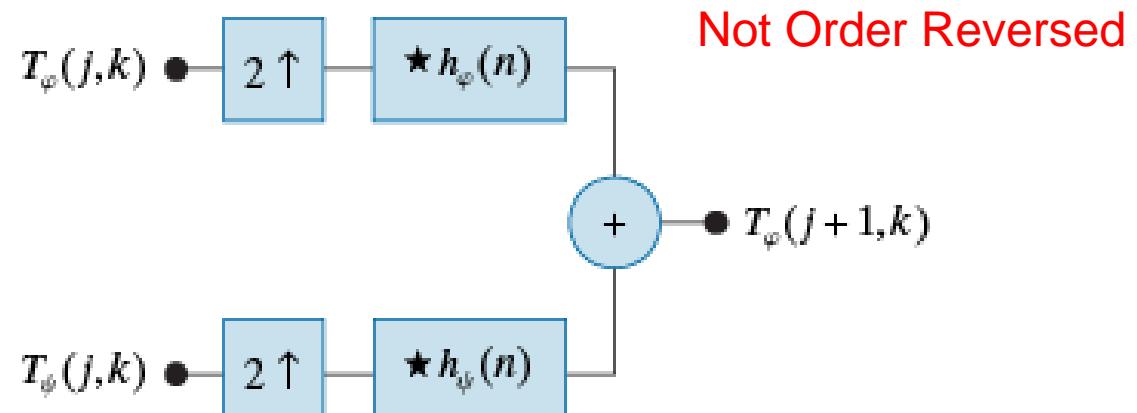
Reconstruction may be obtained by the IFWT:

- Upsample by 2 (by inserting zeros)
- Convolution of the scaling coefficients by $h_\varphi(n)$ and the wavelet coefficients by $h_\psi(n)$ and Summation

$$T_\varphi(j+1, k) = h_\varphi(k) * T_\varphi^{2\uparrow}(j, k) + h_\psi(k) * T_\psi^{2\uparrow}(j, k) \Big|_{k \geq 0}$$

- See Fig. 6.23 for the corresponding FWT Analysis Filter bank (slides 82-83)

FIGURE 6.27
An inverse FWT synthesis filter bank for orthonormal filters.



1-D Wavelet Transforms

The Fast Wavelet Transform (cont...)

- Reconstruction - See Fig 6.25 for the output of Analysis Filter bank

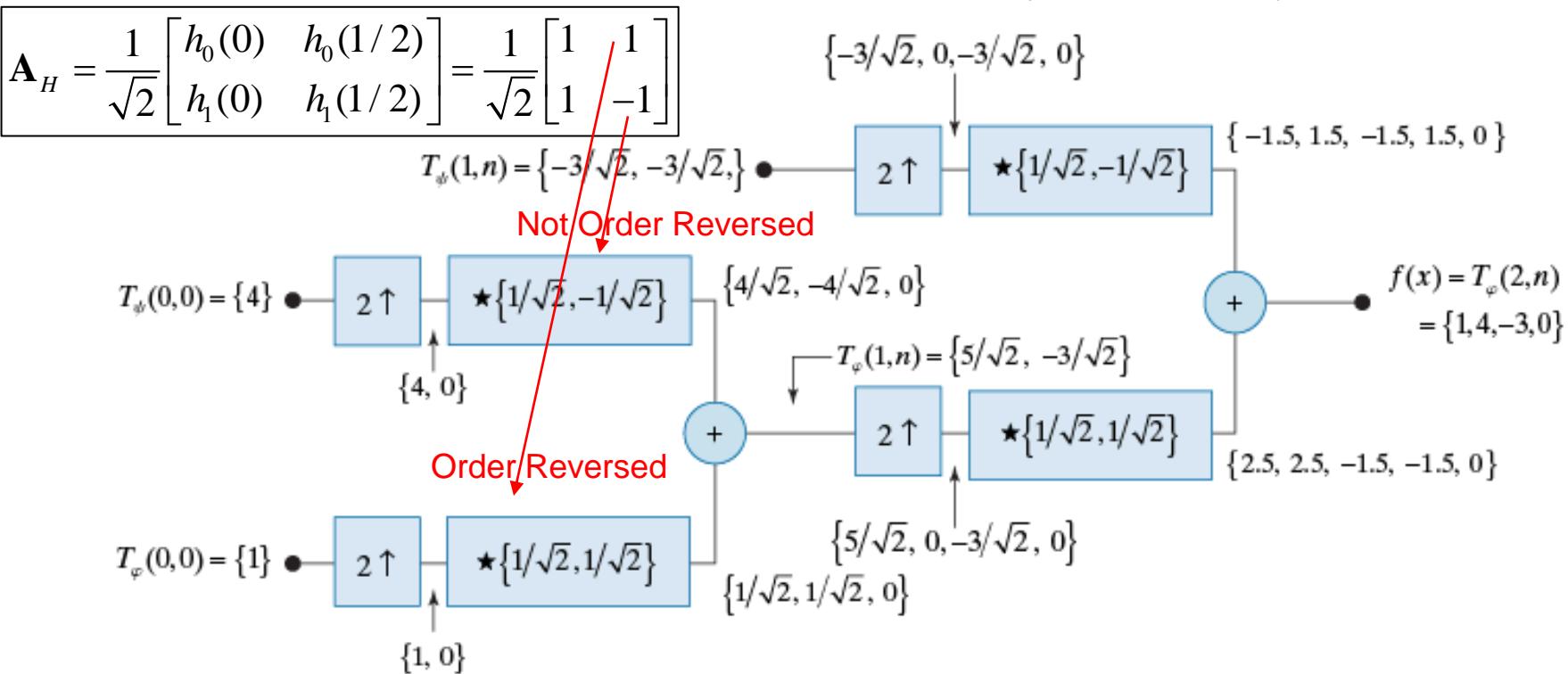


FIGURE 6.28 Computing a two-scale inverse fast wavelet transform of sequence $\{1, 4, -1.5\sqrt{2}, -1.5\sqrt{2}\}$ with Haar scaling and wavelet functions.

1-D Wavelet Transforms: Relation to the Fourier Transform

- The Fourier basis functions guarantee the existence of the transform for energy signals.
- The wavelet transform **depends upon the availability** of scaling functions for a given wavelet function.
- The wavelet transform depends on the orthonormality (or biorthogonality) of the scaling and wavelet functions.
- The F.T. informs us about the frequency content of a signal. F.T. does not inform us on the specific time instant when a certain frequency occurs.
- The W.T. provides information on “**when the frequency occurs**”.

Heisenberg cells:

- The width of each rectangle in (a) represents one time instant.
- The height of each rectangle in (b) represents a single frequency.

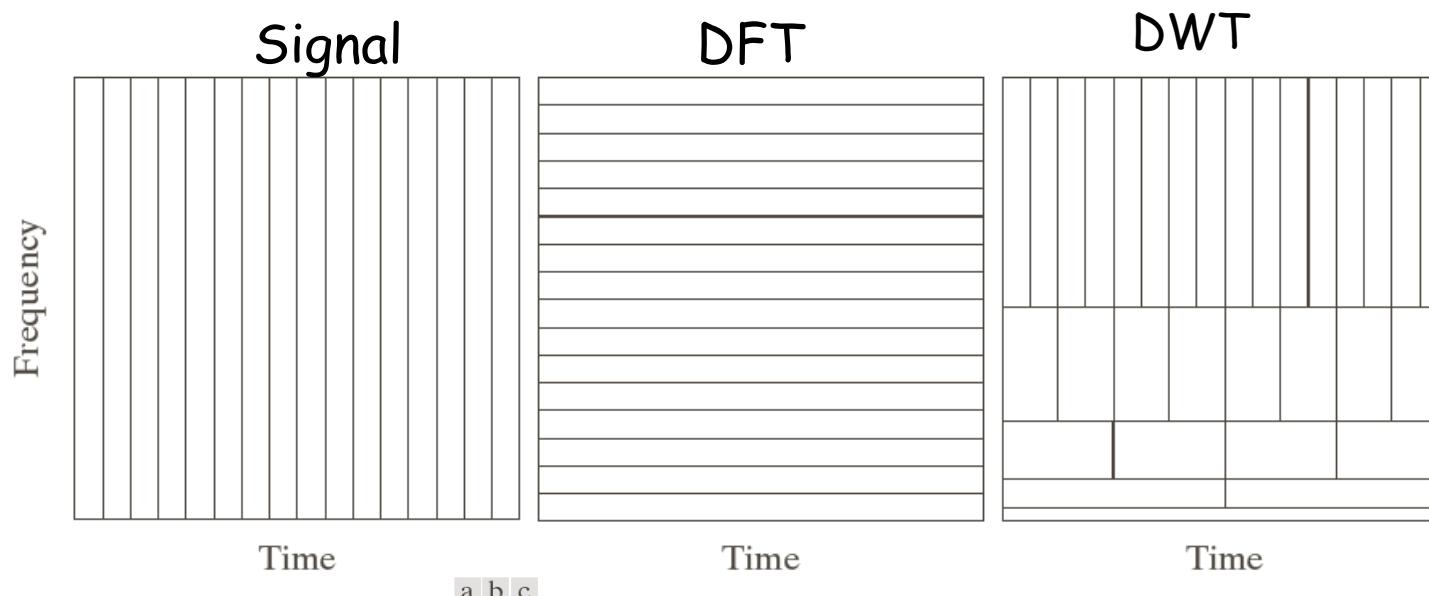


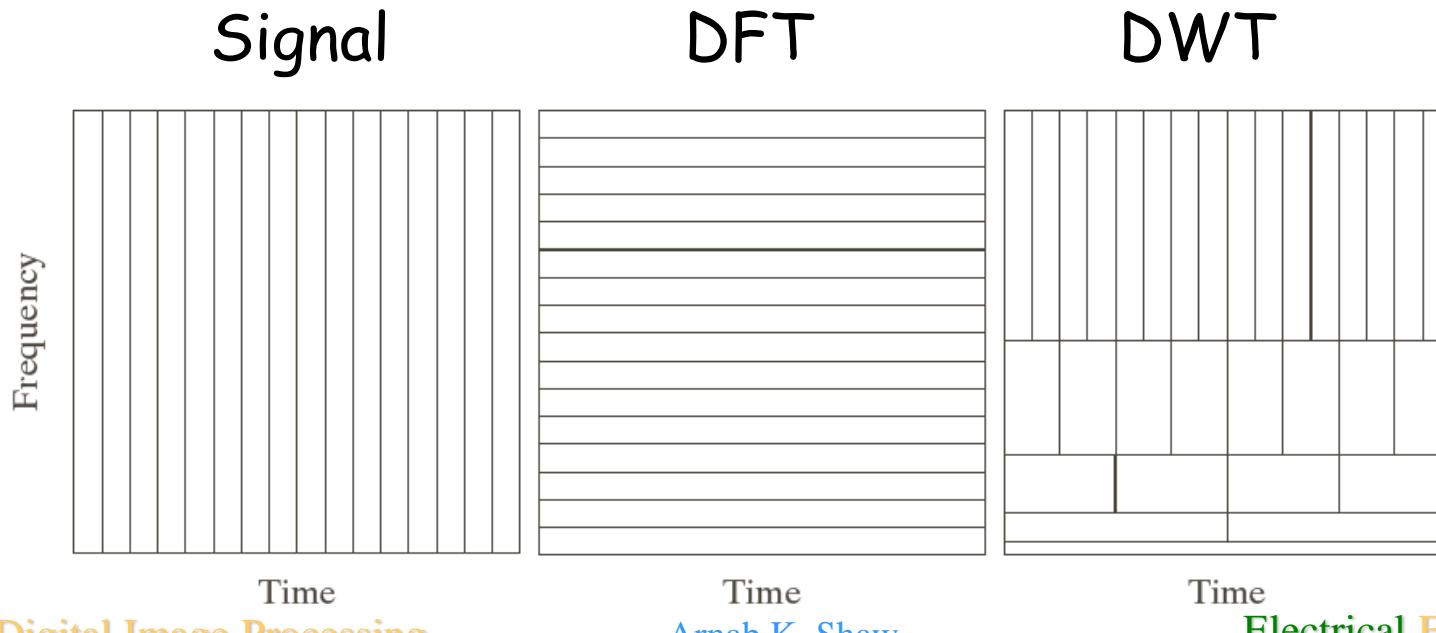
FIGURE 7.23 Time-frequency tilings for the basis functions associated with (a) sampled data, (b) the FFT, and (c) the FWT. Note that the horizontal strips of equal height rectangles in (c) represent FWT scales.

3rd
Edition

1-D Wavelet Transforms

Relation to the Fourier Transform (cont...)

- FT: Time domain pinpoints the instance an event occurs but has no frequency information.
- DFT domain pinpoints the frequencies that are present in the events but provides no time resolution (when a certain frequency appears).

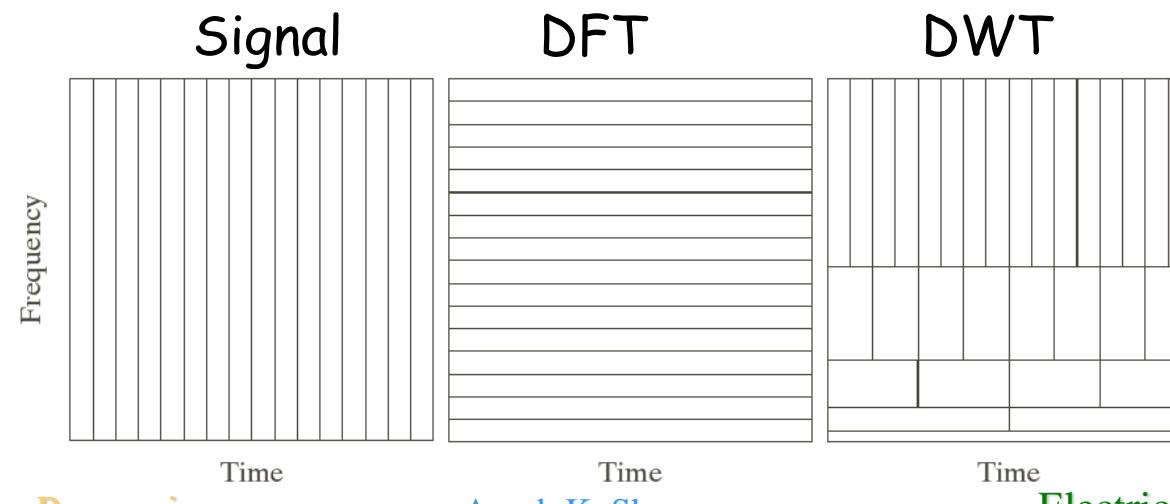


3rd
Edition

1-D Wavelet Transforms

Relation to the Fourier Transform (cont...)

- In DWT the time-frequency resolution varies but the area of each tile is the same. There is a compromise between time and frequency resolutions.
- At low frequencies (slowly varying) the tiles are shorter (better frequency resolution, less ambiguity regarding frequency) but wider (more ambiguity regarding time).
- At high frequencies the opposite happens (time resolution is improved).



**3rd
Edition**

7.5 2-D Discrete Wavelet Transform

Extension from 1-D wavelet transforms to 2-D:

- A 2-D scaling function and three 2-D wavelet functions are required
- Separable
- Product of two 1-D functions

Approximation (low pass): $\varphi(x, y) = \varphi(x)\varphi(y)$

Variations along columns: $\psi^H(x, y) = \psi(x)\varphi(y)$

Variations along rows: $\psi^V(x, y) = \varphi(x)\psi(y)$

Variations along diagonals: $\psi^D(x, y) = \psi(x)\psi(y)$

- Given **separable** scaling and wavelet functions the scaled and translated 2-D basis are:

$$\varphi_{j,m,n}(x, y) = 2^{j/2} \varphi(2^j x - m, 2^j y - n)$$

$$\psi_{j,m,n}^i(x, y) = 2^{j/2} \psi^i(2^j x - m, 2^j y - n), i = \{H, V, D\}$$

where, i, j : Indices

- There are two translations: m and n .

The 2D DWT is given by (Forward Transform)

$$W_\varphi(j_0, m, n) = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) \varphi_{j_0, m, n}(m, n)$$

$$W_\psi^i(j, m, n) = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) \psi_{j_0, m, n}^i(m, n), \quad i = \{H, V, D\}$$

and the Inverse Transformation is

$$\begin{aligned} f(m, n) = & \frac{1}{\sqrt{MN}} \sum_m \sum_n W_\varphi(j_0, m, n) \varphi_{j_0, m, n}(m, n) \\ & + \frac{1}{\sqrt{MN}} \sum_{i=H, V, D} \sum_{j=j_0}^{\infty} \sum_m \sum_n W_\psi^i(j, m, n, k) \psi_{j, m, n}^i(m, n) \end{aligned}$$

**3rd
Edition**

2-D Fast Wavelet Transform - Implementation

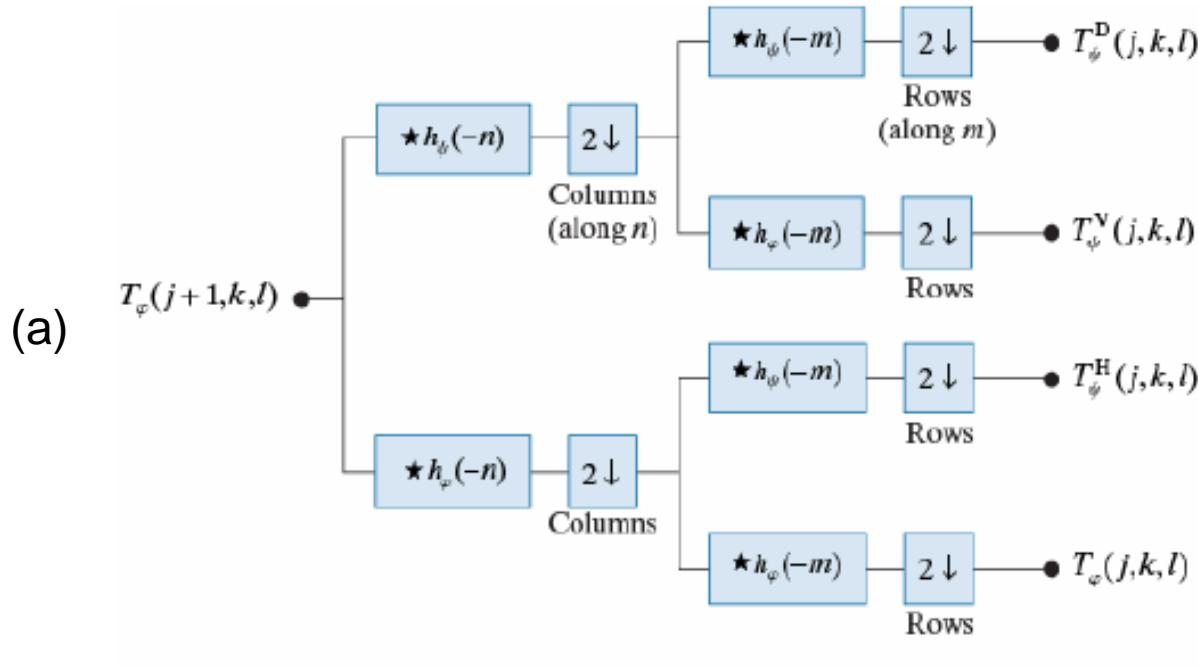
- Take M equally spaced samples over the support of the scaling and wavelet functions.
- Normally, let $j_0 = 0$ and
- Select $N=M=2^J$ so that,

$$m=n=0, 1, 2, \dots, M-1, \quad j=0, 1, 2, \dots, J-1, \quad k=0, 1, 2, \dots, 2^j-1$$

- Like the 1-D transform it can be implemented using filtering and down-sampling
- Take the 1-D FWT of the rows followed by the 1-D FWT of the resulting columns



2-D Fast Wavelet Transform (cont...)

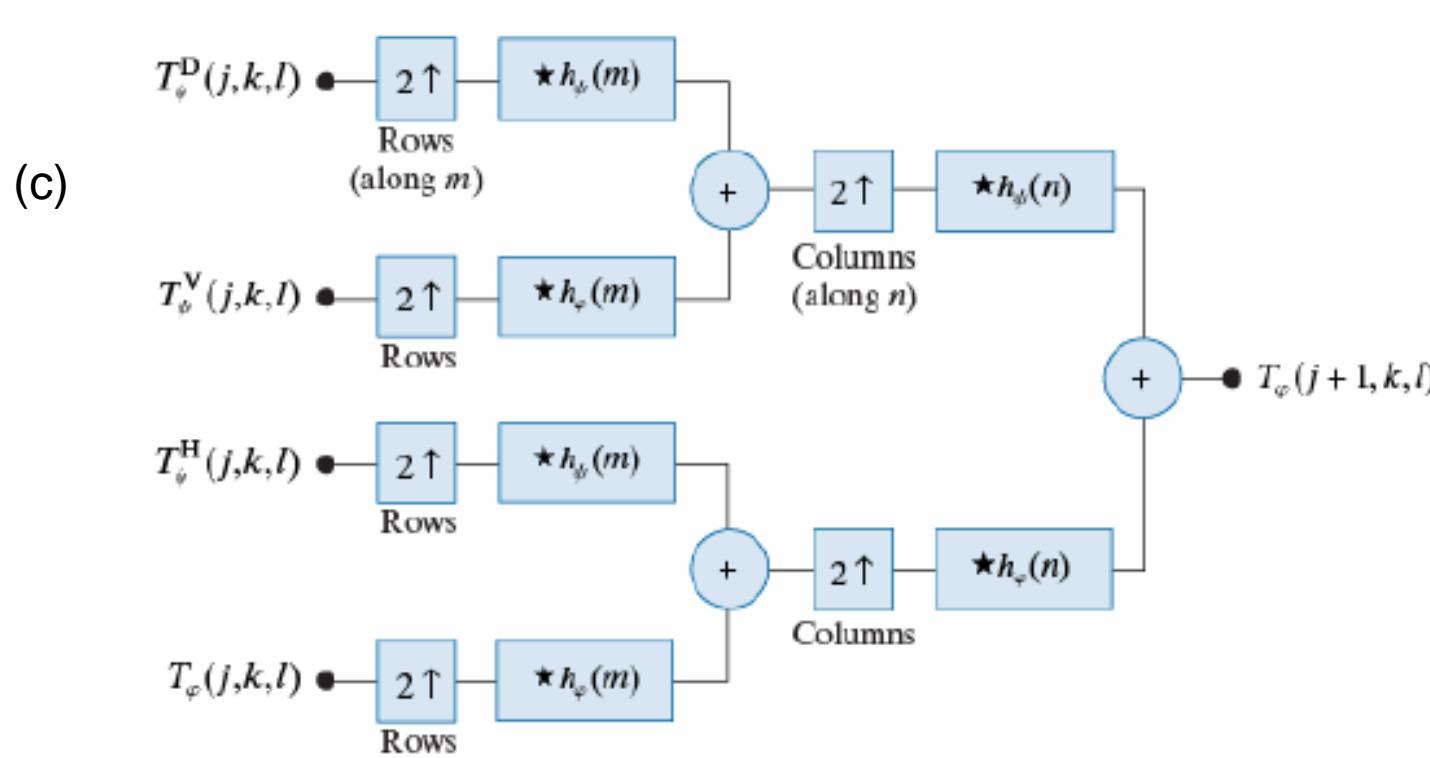


a
b
c

FIGURE 6.29
The 2-D fast wavelet transform: (a) the analysis filter bank; (b) the resulting decomposition; and (c) the synthesis filter bank.

Note m and n are dummy variables of convolution, while j , like in the 1-D case, is scale, and k and l are translations.

Synthesis/Reconstruction (Inverse 2-D FWT)

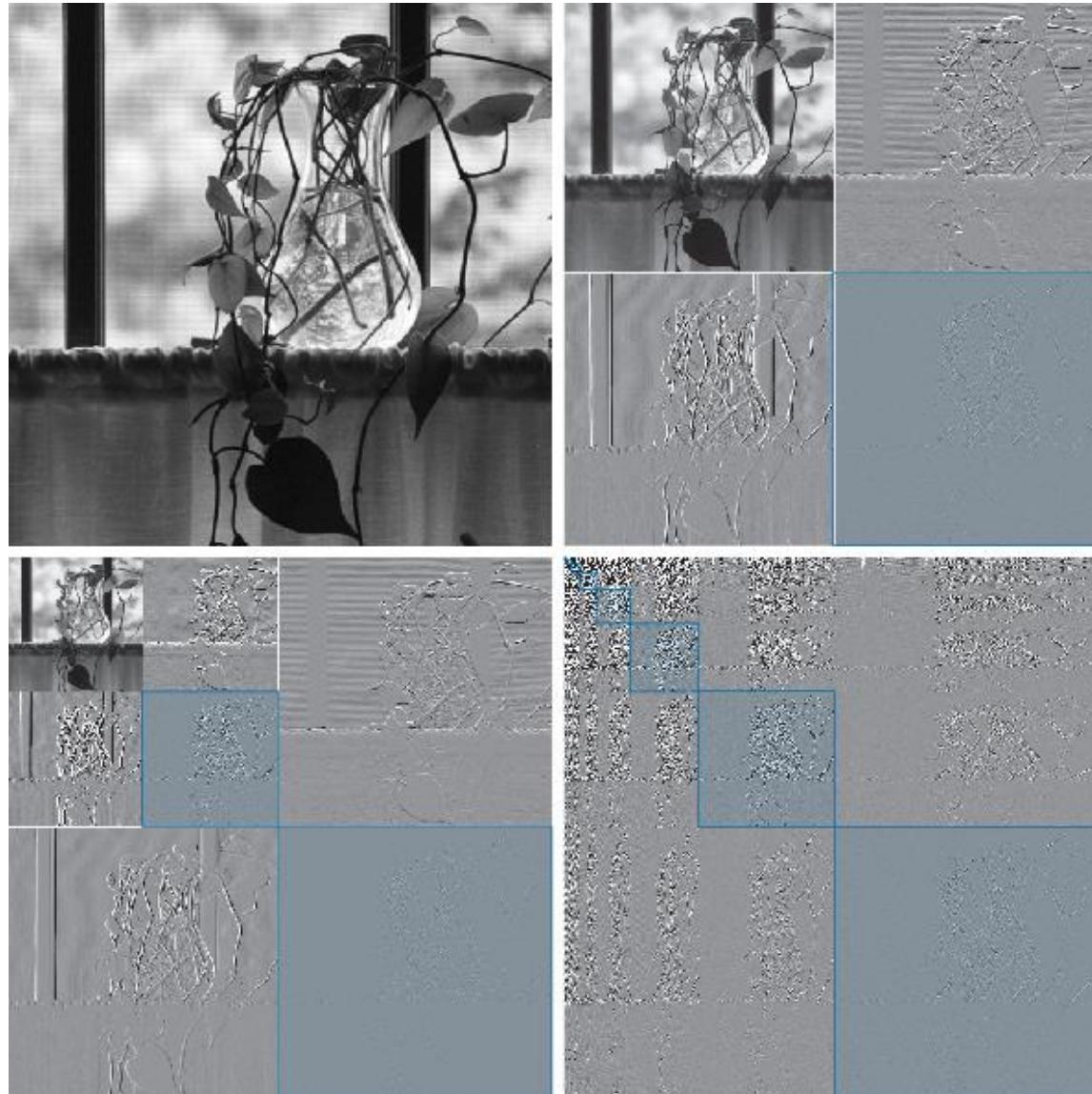


a
b
c

FIGURE 6.29
The 2-D fast wavelet transform: (a) the analysis filter bank; (b) the resulting decomposition; and (c) the synthesis filter bank.

Note m and n are dummy variables of convolution, while j , like in the 1-D case, is scale, and k and l are translations.

2-D Fast Wavelet Transform (cont...)



a
b
c
d

FIGURE 6.30

(a) A 512×512 image of a vase;
 (b) a one-scale FWT; (c) a two-scale FWT; and
 (d) the Haar transform of the original image.
 All transforms have been scaled to highlight their underlying structure. When corresponding areas of two transforms are shaded in blue, the correspondent pixels are identical.

See slides
37-43

a b

FIGURE 6.31

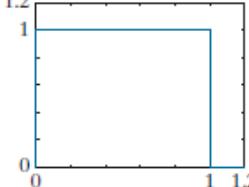
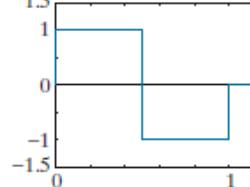
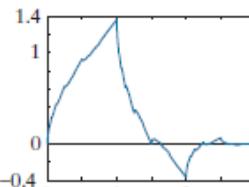
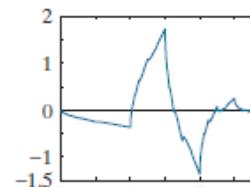
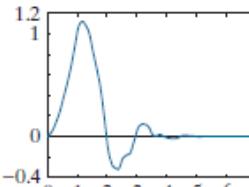
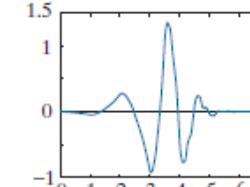
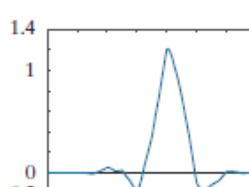
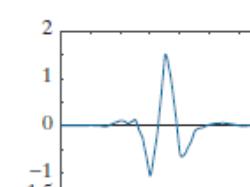
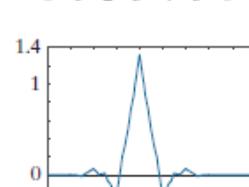
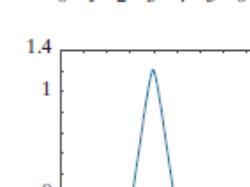
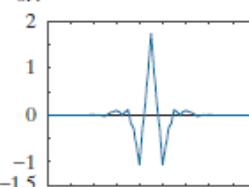
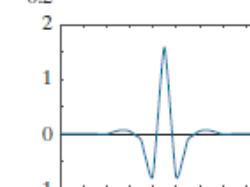
(a) Haar basis images of size 8×8 [from Fig. 6.18(c)] and (b) the basis images of a three-scale 8×8 discrete wavelet transform with respect to Haar basis functions.



See slide-43



TABLE 6.1
Some representative wavelets.

Wavelet Name or Family	Scaling Function	Wavelet Function	Filter Coefficients
Haar The oldest and simplest wavelets. Orthogonal and discontinuous.			$g_0(n) = \{1/\sqrt{2}, 1/\sqrt{2}\}$
Daubechies family Orthogonal with the most vanishing moments for a given support. Denoted db N , where N is the number of vanishing moments; db2 and db4 shown; db1 is the Haar of the previous row.			$g_0(n) = \{0.482963, 0.836516, 0.224144, -0.129410\}$
			$g_0(n) = \{0.230372, 0.714847, 0.630881, -0.027984, -0.187035, 0.030841, 0.032883, -0.010597\}$
Symlet family Orthogonal with the least asymmetry and most vanishing moments for a given support (sym4 or 4th order shown).			$g_0(n) = \{0.032231, -0.012604, -0.099220, 0.297858, 0.803739, 0.497619, -0.029636, -0.075766\}$
Cohen-Daubechies-Feauveau 9/7 Biorthogonal B-spline used in the irreversible JPEG2000 compression standard (see Chapter 8).			$h_0(n) = \{0.026749, -0.016864, -0.078223, 0.266864, 0.602949, 0.266864, -0.078223, -0.016864, 0.026749\}$
			$h_1(n) = \{-0.091271, -0.057544, 0.591272, -1.115087, 0.591272, 0.057544, -0.091271, 0\}$



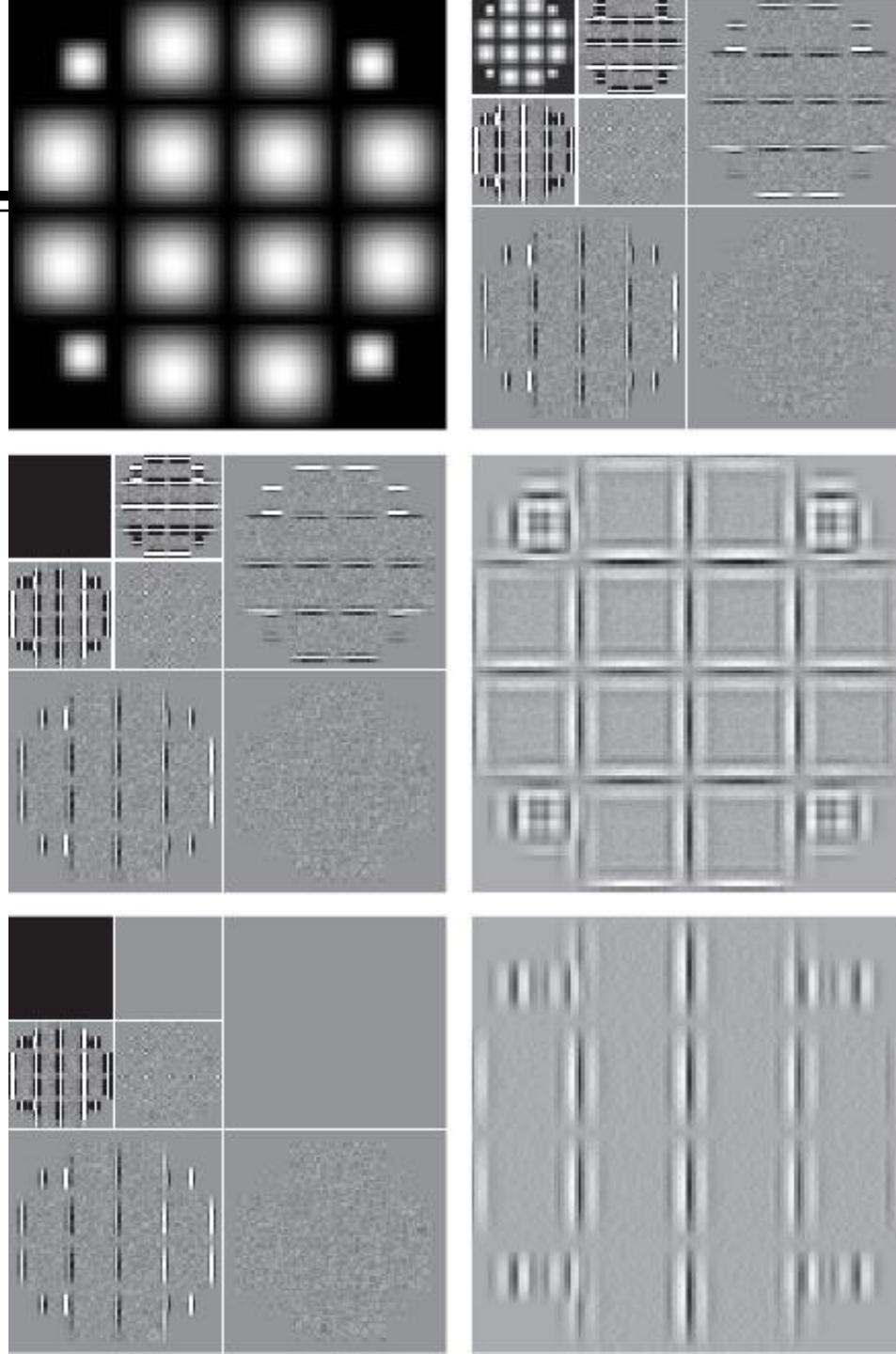
a b
c d
e f

FIGURE 6.32

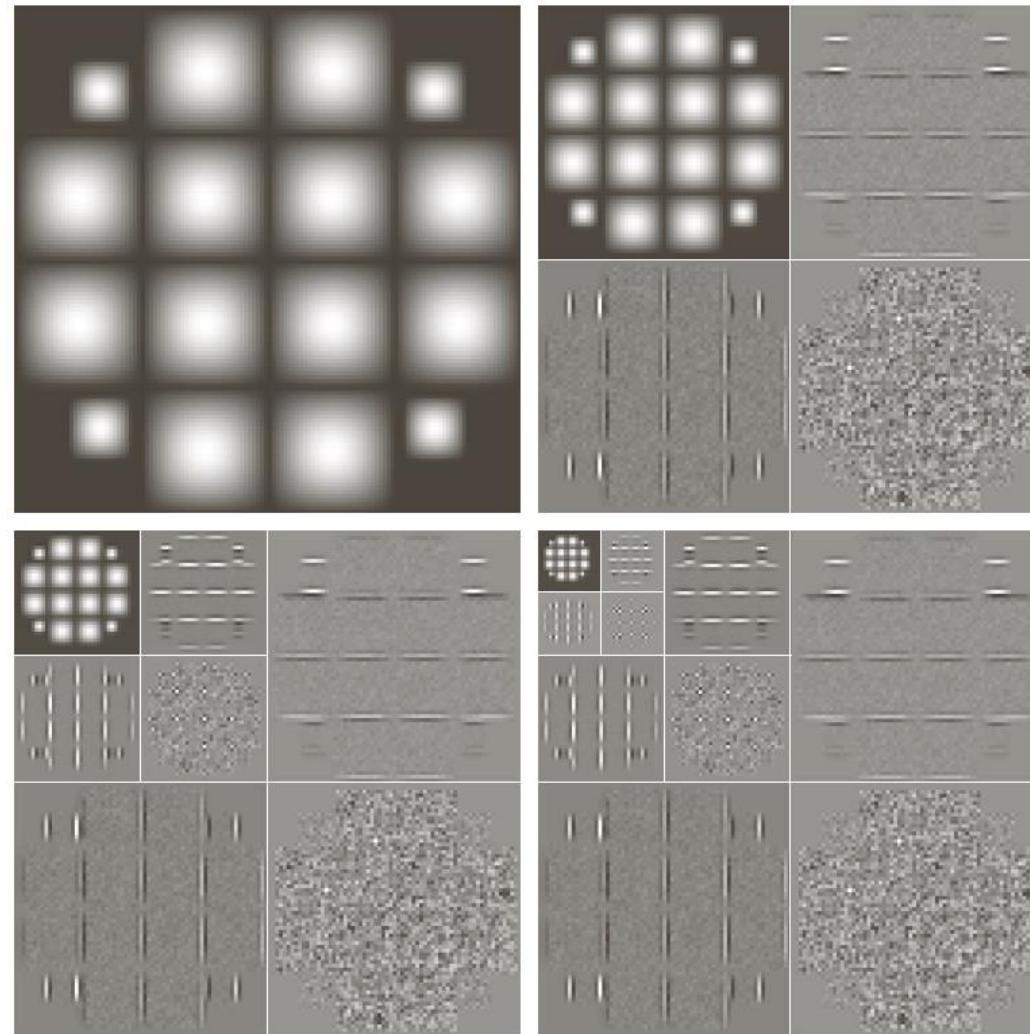
Modifying a DWT
for edge
detection:

(a) original image;
(b) two-scale
DWT with respect
to 4th-order sym-
lets; (c) modified
DWT with the
approximation set
to zero; (d) the
inverse DWT
of (c); (e) modi-
fied DWT with
the approximation
and horizontal
details set to zero;
and (f) the inverse
DWT of (e).

(Note when the
detail coefficients
are zero, they
are displayed as
middle gray; when
the approxima-
tion coefficients
are zeroed, they
display as black.)



Using Symlets



a
b
c
d

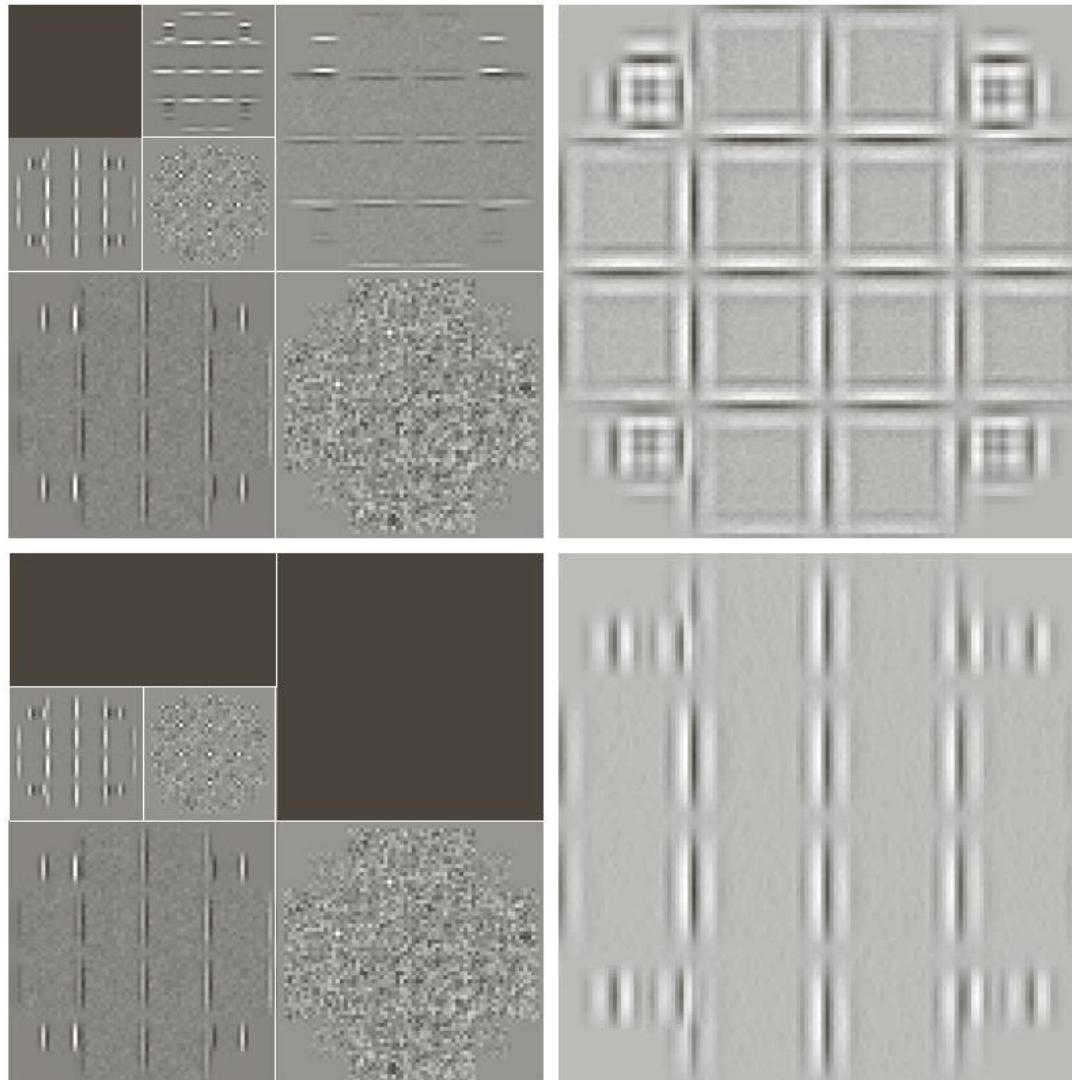
FIGURE 7.25
Computing a 2-D three-scale FWT:
(a) the original image; (b) a one-scale FWT; (c) a two-scale FWT; and (d) a three-scale FWT.

3rd
Edition



2-D Discrete Wavelet Transform: Edge Detection

- Start with Fig 7.25
- Delete selected Coefficients
- Reconstruct



a
b
c
d

FIGURE 7.27
Modifying a DWT for edge detection: (a) and (c) two-scale decompositions with selected coefficients deleted; (b) and (d) the corresponding reconstructions.

3rd
Edition



2-D Discrete Wavelet Transform: Noise Reduction

a
b
c
d
e
f

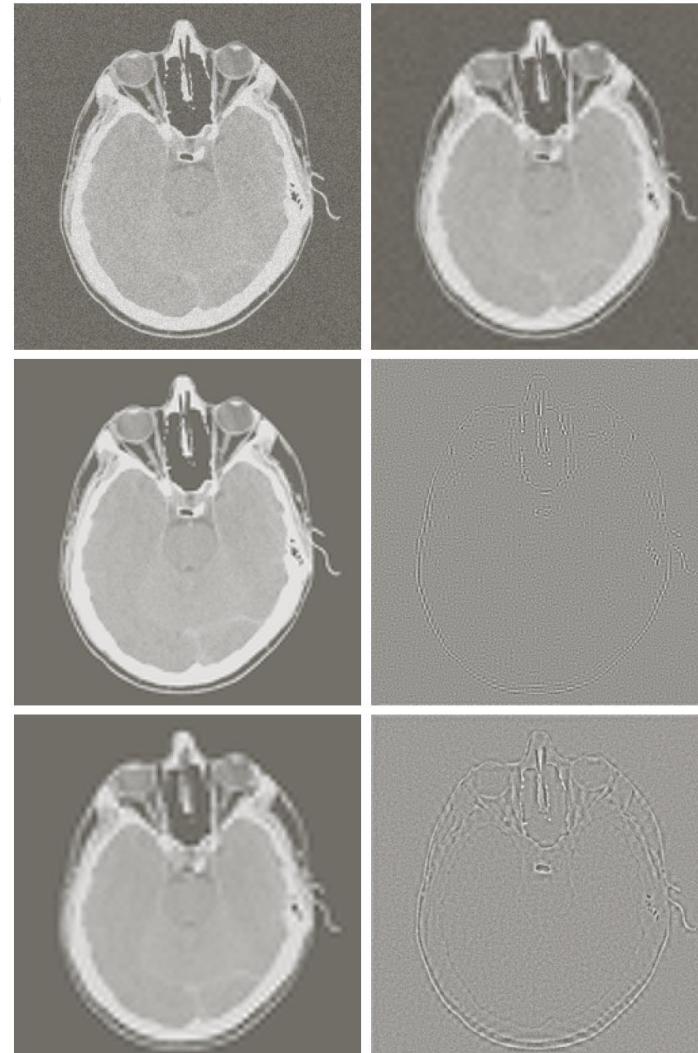
FIGURE 7.28

Modifying a DWT
for noise removal:

a) Noisy CT image.
Goal: Remove noise
by manipulating the
DWT (4th order
symlet)

c) Zeroing the
highest resolution
detail coefficients
(not thresholding
the lowest resolution
details).

e) Zeroing all of the
detail coefficients
of the two scale
DWT.



b) Thresholding all of the
detail coefficients of the
two scale DWT (P=2) →
Blurring. Significant edge
information is lost.

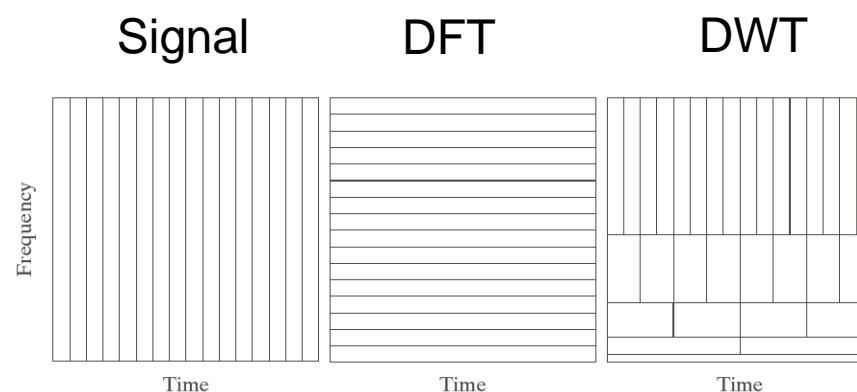
d) Information loss
in c) with respect to
the original image.
The edges are
slightly disturbed.

f) Information loss
in e) with respect to
the original image.
Significant edge
information is removed.

3rd
Edition

Wavelet Packets

- The DWT, as it was defined, decomposes a signal into a sum of scaling and wavelet functions whose bandwidths are logarithmically related.
- Low frequencies → Narrow bandwidths/Large Duration
- High frequencies → Wider bandwidths/Smaller Duration
- For greater flexibility of the partitioning of the time-frequency plane the DWT must be generalized.



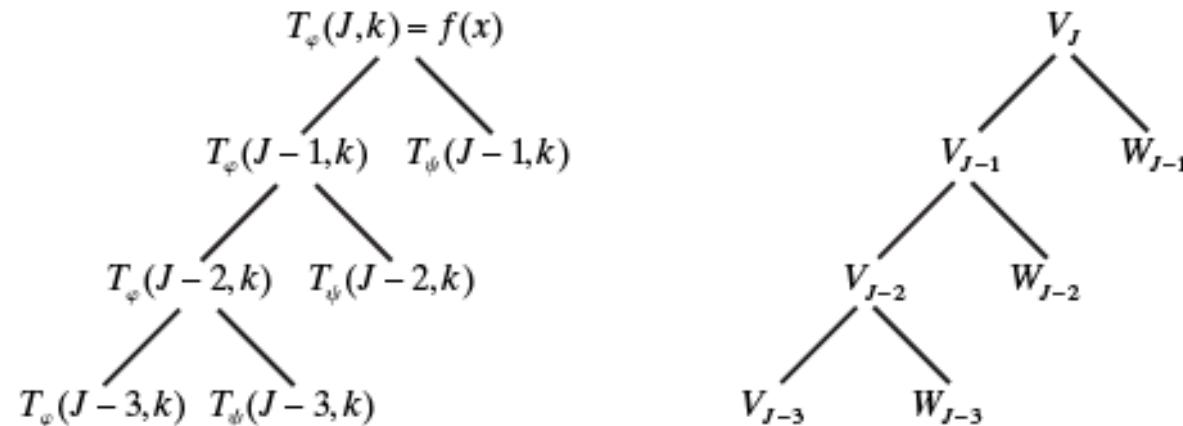
Wavelet Packets (cont...)

- Coefficient tree and subspace analysis tree for the two-scale (three levels) FWT filter bank.
- The subspace analysis tree provides a more compact representation of the decomposition in terms of subspaces.

a b

FIGURE 6.33

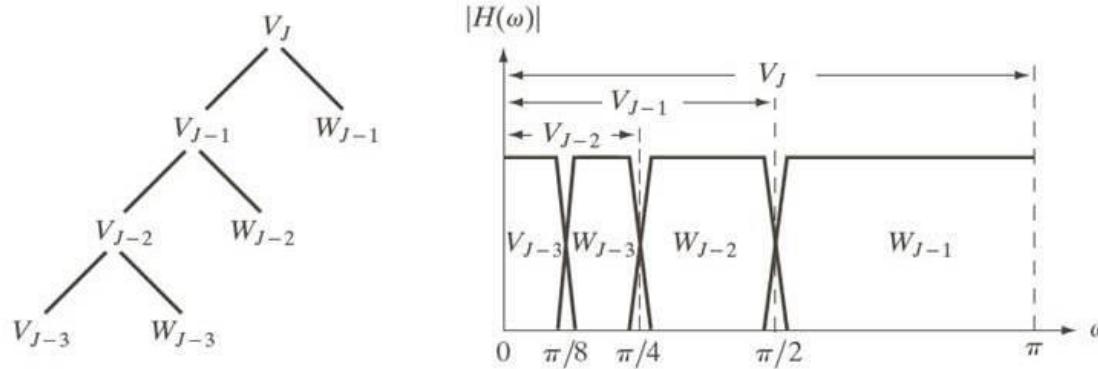
An (a) coefficient tree and (b) analysis tree for the two-scale FWT analysis bank of Fig. 6.24.



$$V_J = V_{J-1} \oplus W_{J-1}$$

Wavelet Packets (cont...)

- Subspace analysis tree and spectrum splitting for **three-scale** (four levels) FWT filter bank.
- Here, we have three options for the expansion:



$$V_J = V_{J-1} \oplus W_{J-1}$$

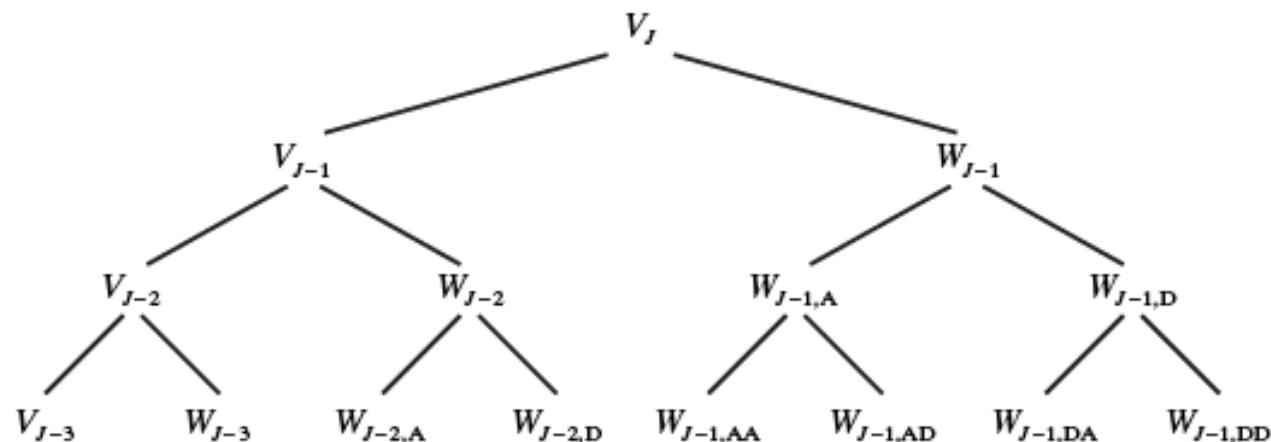
$$V_J = V_{J-2} \oplus W_{J-2} \oplus W_{J-1}$$

$$V_J = V_{J-3} \oplus W_{J-3} \oplus W_{J-2} \oplus W_{J-1}$$

Wavelet Packets (cont...)

- Wavelet packets are conventional wavelet transforms in which the *details are ALSO filtered iteratively*.
- For example, 3-scale wavelet packet tree:

FIGURE 6.34
A three-scale
wavelet packet
analysis tree.



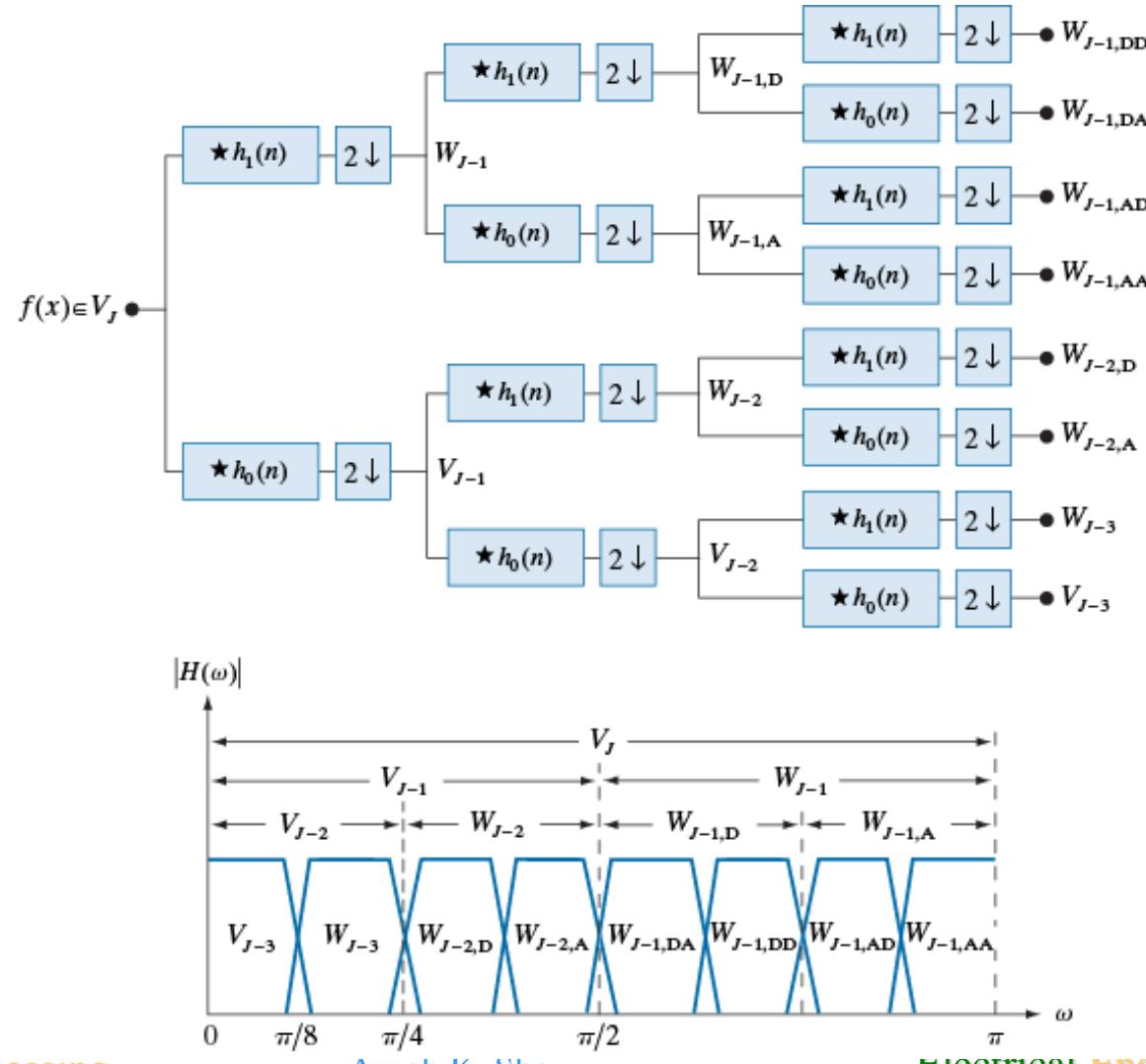
- Indices *A* (Approximation) and *D* (Detail) denote the path from the parent to the node.

Wavelet Packets (cont...)

Filter bank and spectrum (three scale wavelet packet).

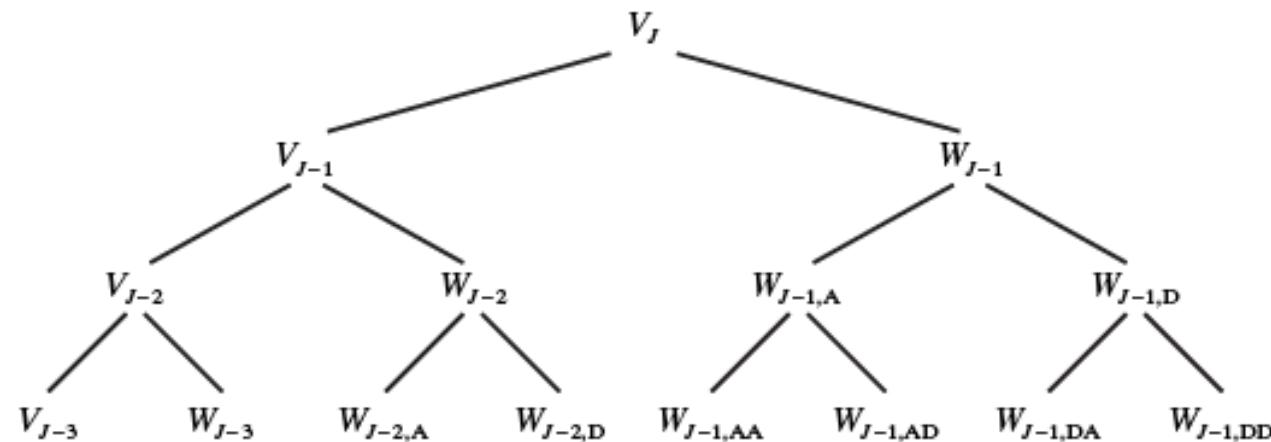
a
b

FIGURE 6.35
The (a) filter bank and (b) spectrum-splitting characteristics of a three-scale full wavelet packet analysis tree.

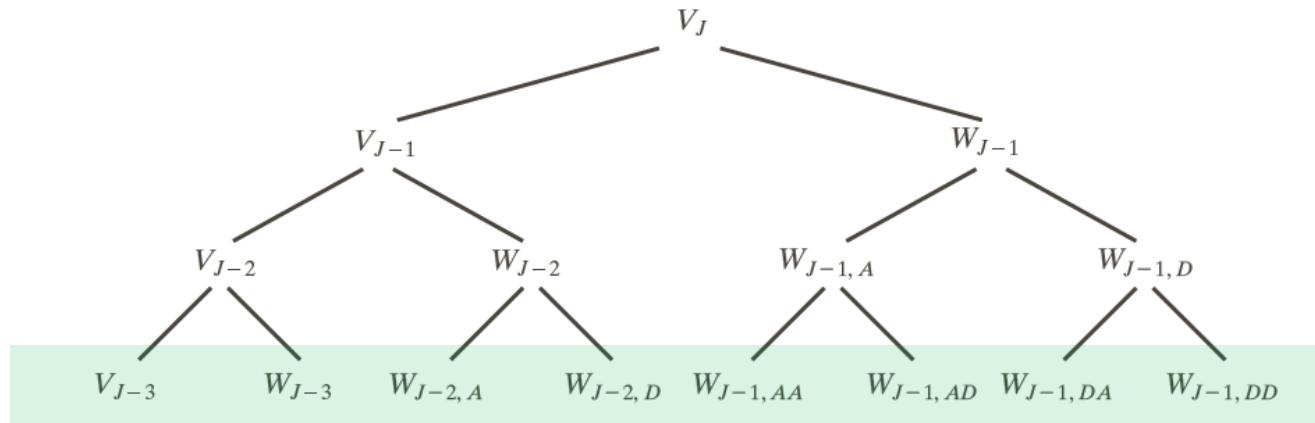


Wavelet Packets (cont...)

- The **cost** of this generalization **increases** the computational complexity of the transform from $O(M)$ to $O(M \log M)$.
- Also, the 3-scale packet almost triples the number of decompositions available from the three low pass bands.

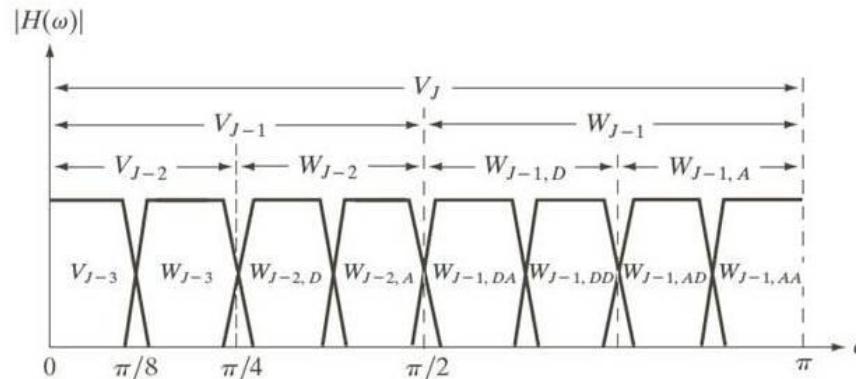


Wavelet Packets (cont...)



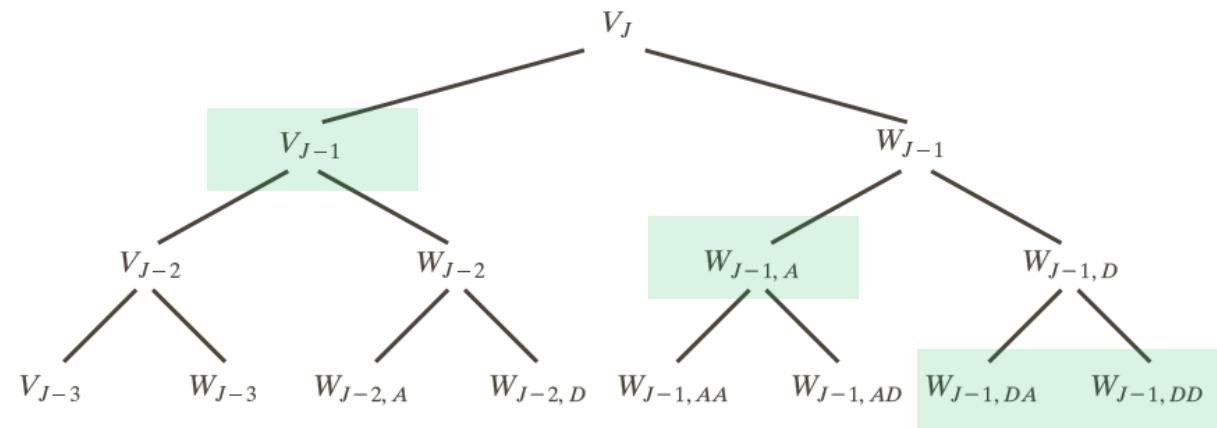
The subspace of the signal may be expanded as

$$V_J = V_{J-3} \oplus W_{J-3} \oplus W_{J-2,A} \oplus W_{J-2,D} \oplus W_{J-1,AA} \oplus W_{J-1,AD} \oplus W_{J-1,DA} \oplus W_{J-1,DD}$$



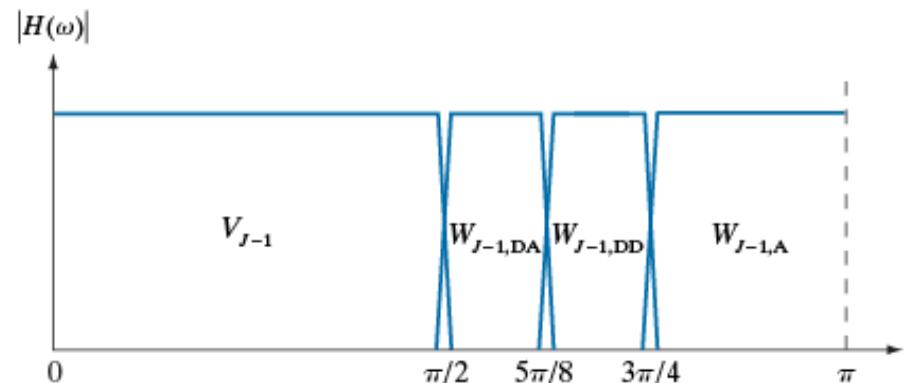
Spectrum:

Wavelet Packets (cont...)



It can also be expanded as

$$V_J = V_{J-1} \oplus W_{J-1,A} \oplus W_{J-1,DA} \oplus W_{J-1,DD}$$



Spectrum:

FIGURE 6.36
The spectrum of
the decomposition
in Eq. (6-160).

Wavelet Packets (cont...)

- In general, a *P-scale 1-D wavelet* packet (associated with $P+1$ level analysis trees) supports

$$D(P+1) = [D(P)]^2 + 1$$

unique decompositions, where $D(1)=1$ is the initial signal.

- For instance, $D(2)=2$, $D(3)=5$, $D(4)=26$ and $D(5)=677$.

Wavelet Packets (cont...)

- The problem increases dramatically for 2-D.
- A *P-scale 2-D wavelet* packet (associated with $P+1$ level analysis trees) supports

$$D(P+1) = [D(P)]^4 + 1$$

unique decompositions, where $D(1)=1$ is the initial image.

- For instance, $D(4)=83,522$ possible decompositions.
- Question: How do we select among them?
- *Impractical* to examine each one of them for a given application.

Wavelet Packets (cont...)

- A common application of wavelets is **image compression**.
- Consider a simplified example.
- We seek to compress the fingerprint image by selecting the “best” three-scale wavelet packet decomposition.

- Criterion for the decomposition: "Energy"

$$E(f) = \sum_m \sum_n |f(m, n)|$$

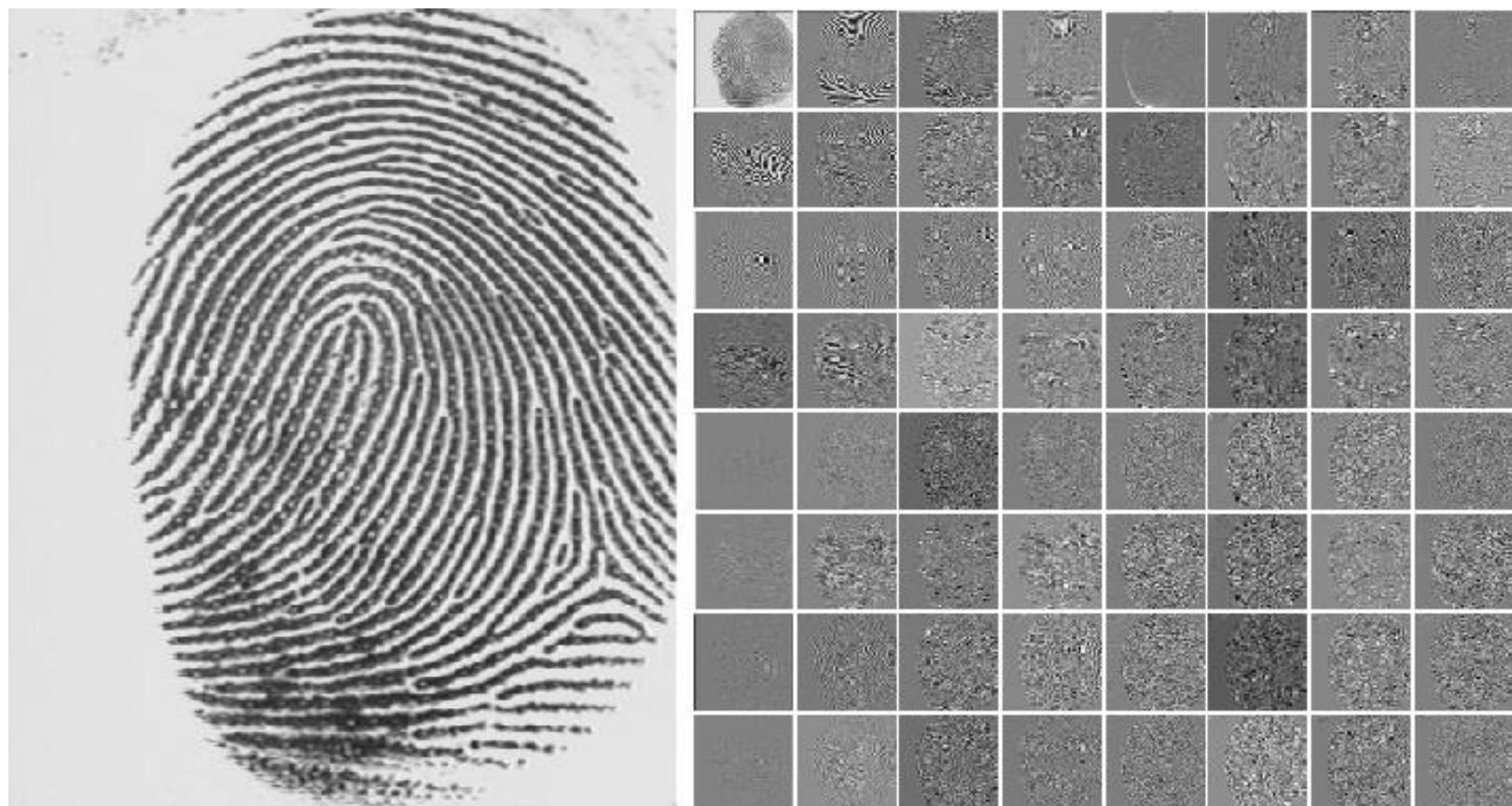
Additive Cost Function

Algorithm

- For each node of the analysis tree, from the root to the leaves:
- Step 1: Compute the energy of the parent-node (E_P) and the energy of its offspring (E_A, E_H, E_V, E_D).
- Step 2: If $E_A + E_H + E_V + E_D < E_P$ include the offspring in the analysis tree as they reduce the initial energy
- Else, keep only the parent which is a leaf of the tree
- Why? → Smaller energy implies more near-zero values
→ Easier to compress



Wavelet Packets (cont...)



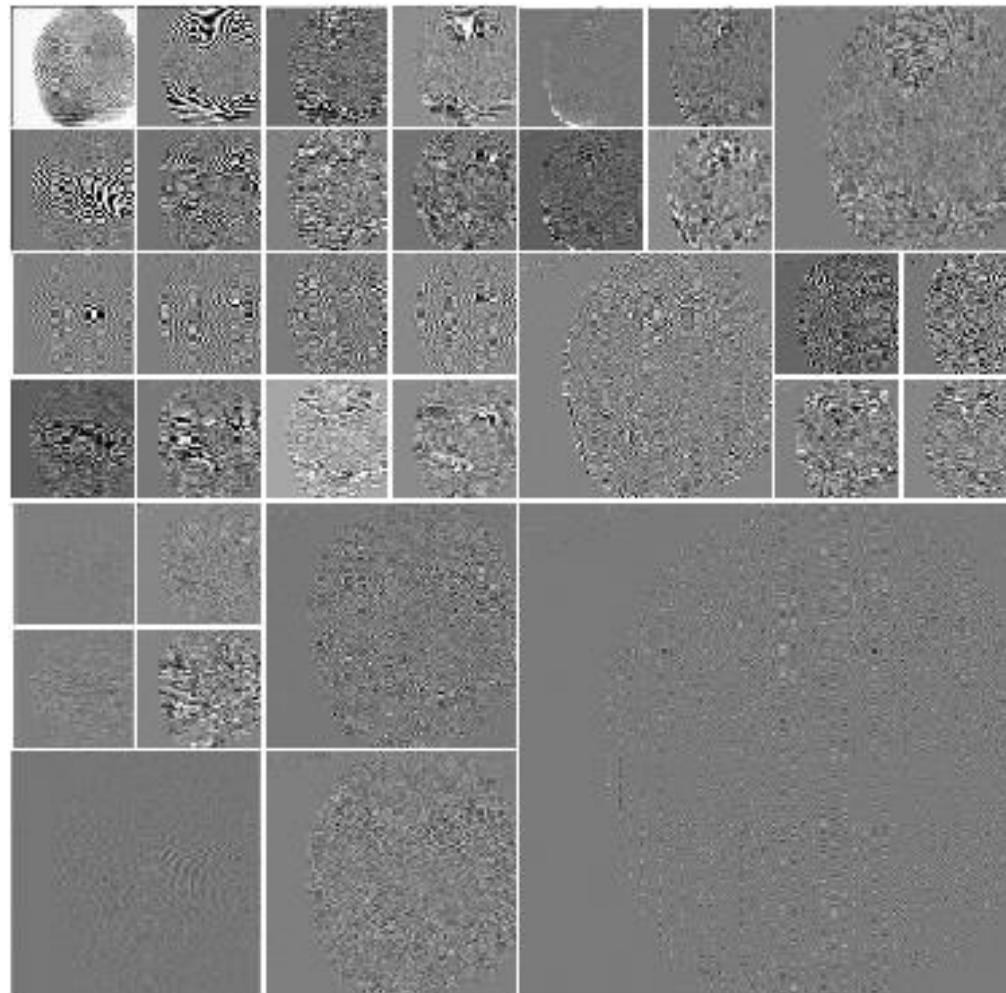
a b

FIGURE 6.39 (a) A scanned fingerprint and (b) its three-scale, full wavelet packet decomposition. Although the 64 subimages of the packet decomposition appear to be square (e.g., note the approximation subimage), this is merely an aberration of the program used to produce the result. (Original image courtesy of the National Institute of Standards and Technology.)

Wavelet Packets (cont...)

Many of the 64 initial subbands are eliminated.

FIGURE 6.40
An optimal
wavelet packet
decomposition for
the fingerprint of
Fig. 6.39(a).



Wavelet Packets (cont...)

- The preceding algorithm can be used to (1) **prune** wavelet packet trees or (2) **design** procedures for computing **optimal trees** from scratch.
- **Nonessential siblings-descendants** of nodes that would be **eliminated** in Step 2 of the algorithm → would not be computed.
- Many of the original full packet decomposition's 64 subbands have been eliminated.
- In addition, the subimages that are not split (or further decomposed) are relatively smooth and composed of pixels that are middle gray in value.
- These subimages contain little information. There would be no overall decrease in entropy realized by splitting them.

Acknowledgements

The slides are primarily based on the figures and images in the Digital Image Processing textbook by Gonzalez and Woods:

- http://www.imageprocessingplace.com/DIP-3E/dip3e_book_images_downloads.htm

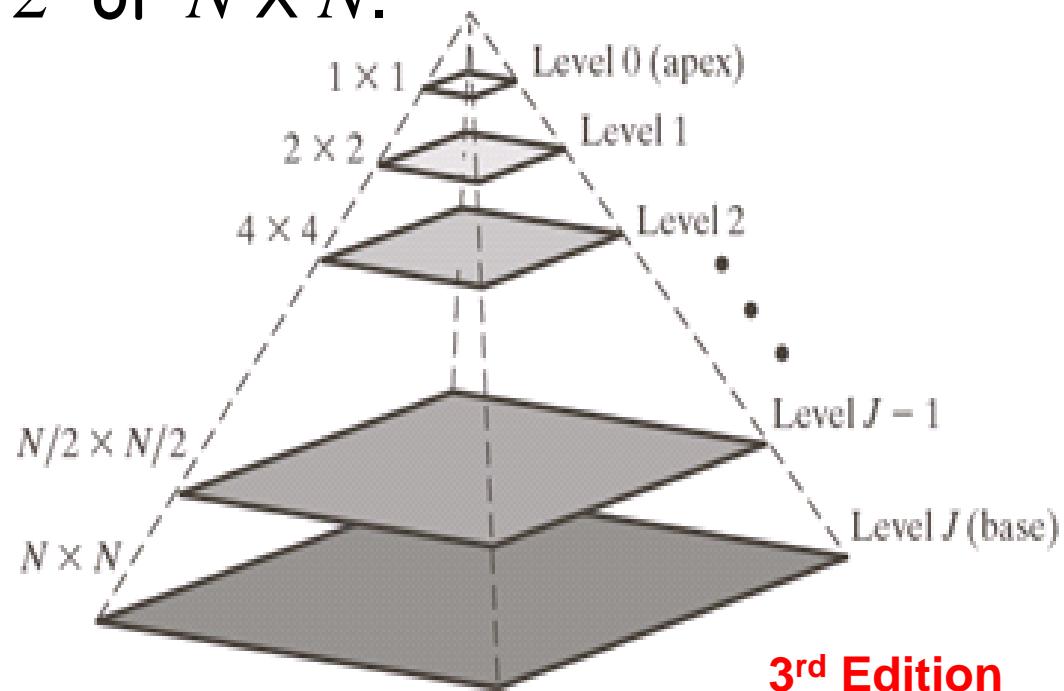
In addition, slides have been adopted and modified from the following sources:

- http://www.cs.uoi.gr/~cnikou/Courses/Digital_Image_Processing

7.1.1 Image Pyramids

- Originally devised for machine vision and image compression
- Collection of images at decreasing resolution levels
- Base level size; $2^J \times 2^J$ or $N \times N$.
- Level- j size: $2^j \times 2^j$.

FIGURE 7.2
(a) An image pyramid.



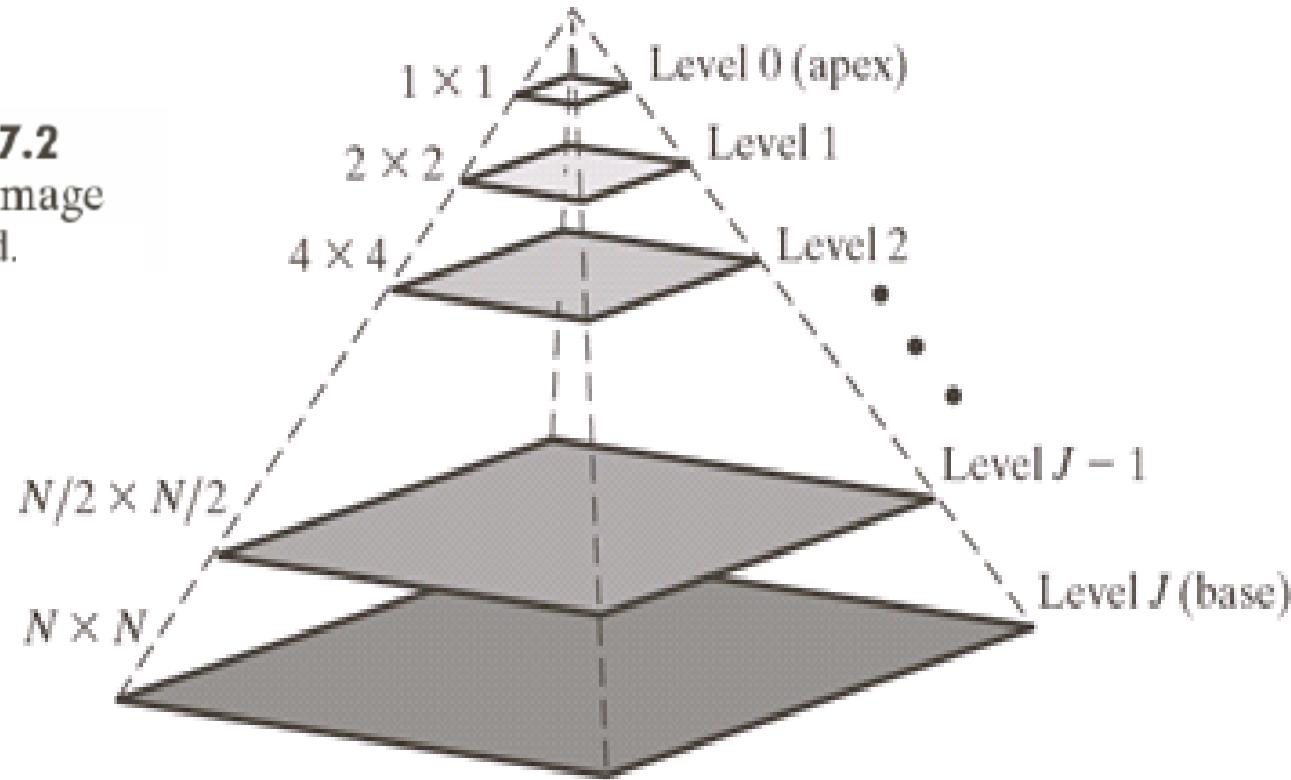
3rd Edition

Image Pyramids (cont...)

Approximation Pyramid:

- Each reduced resolution level is a **filtered** and **downsampled** image: $f_{\downarrow 2}(n) = f(2n)$

FIGURE 7.2
(a) An image pyramid.

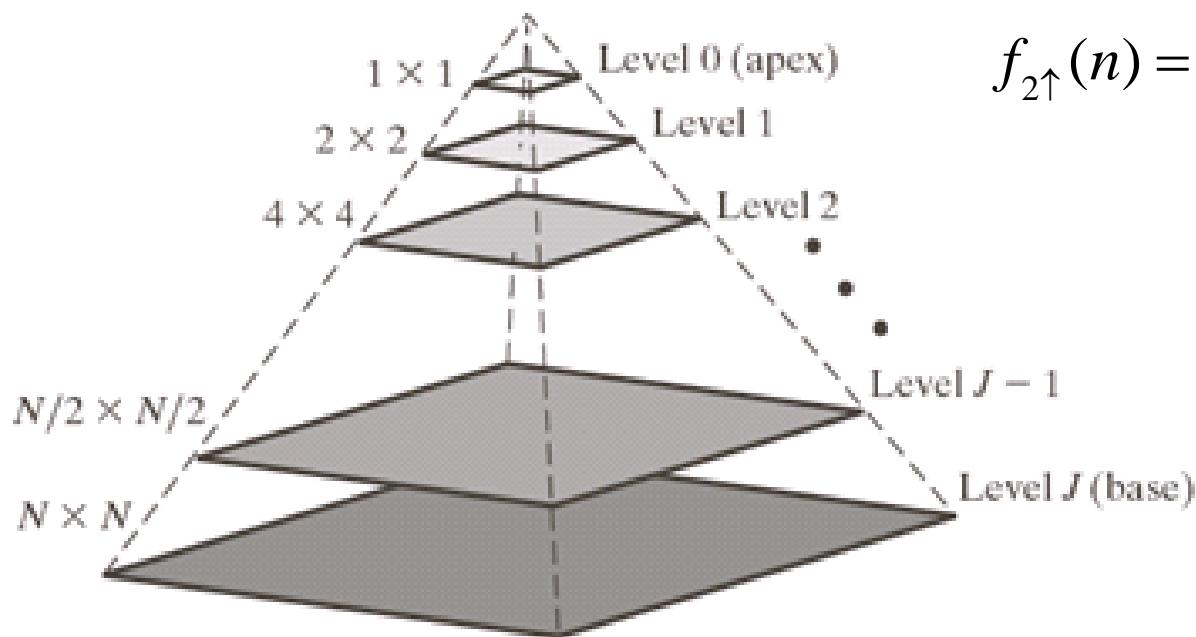


3rd Edition

Image Pyramids (cont...)

Prediction Pyramid:

- A prediction of each high resolution level is obtained by **upsampling** (inserting zeros) the previous low resolution level (prediction pyramid) and **interpolation** (filtering).



$$f_{2\uparrow}(n) = \begin{cases} f(n/2) & \text{if } n \text{ is even} \\ 0 & \text{otherwise} \end{cases}$$

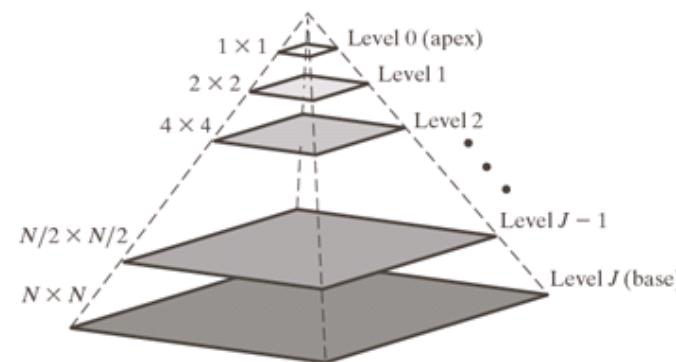
FIGURE 7.2
(a) An image pyramid.

3rd Edition

Image Pyramids (cont...)

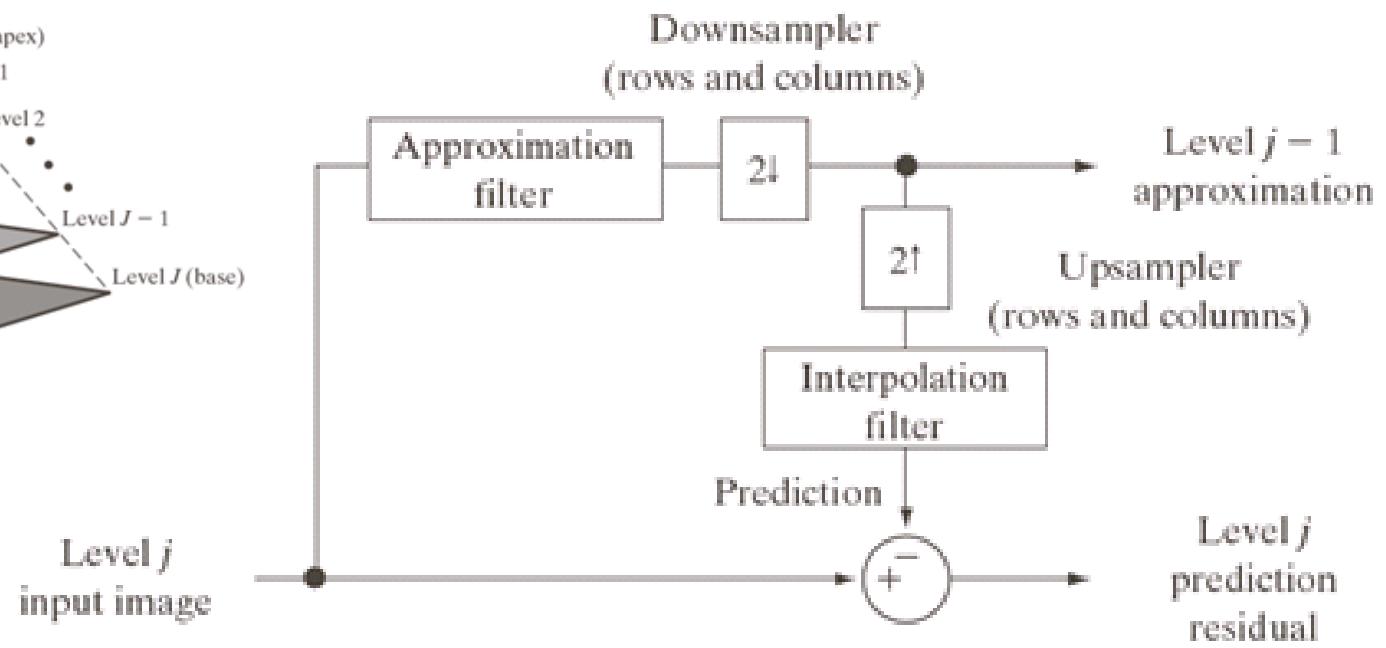
Prediction Residual Pyramid:

- At each resolution level, the prediction error is retained along with the lowest resolution level image.
- The original image may be reconstructed from this information.



a
b

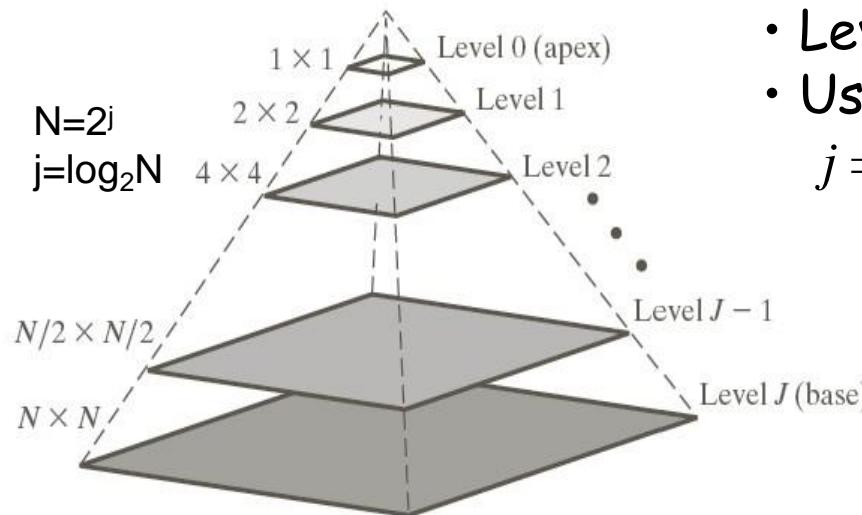
FIGURE 7.2
 (a) An image pyramid. (b) A simple system for creating approximation and prediction residual pyramids.



3rd Edition



7.1.1 Image Pyramids



- Level 0 is of little value
- Usually truncated at some level- P
 $j = J-P, \dots, J \rightarrow (P+1 \text{ levels})$

$$f_{2\uparrow}(n) = \begin{cases} f(n/2) & \text{if } n \text{ is even} \\ 0 & \text{otherwise} \end{cases}$$

$$f_{2\downarrow}(n) = f(2n)$$

- Approximation filter

- Gaussian

- Mean

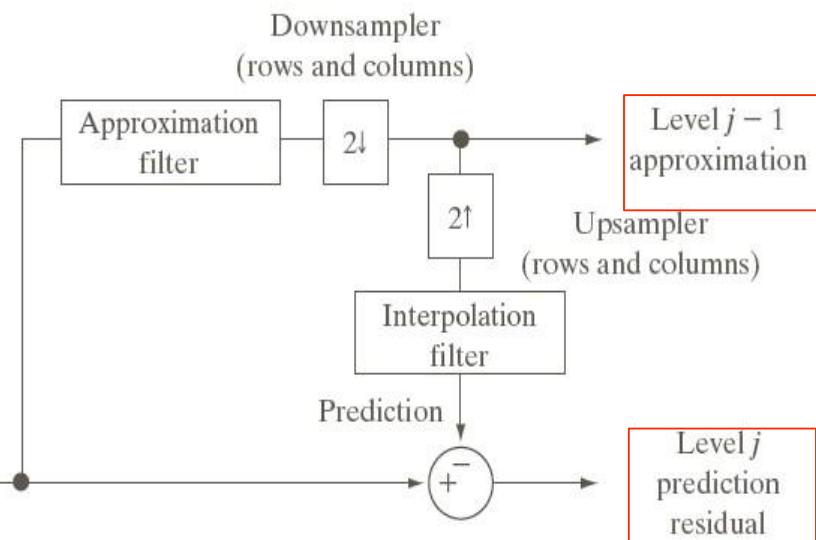
- No filtering

- Interpolation filter

- Nearest neighbor

- Bilinear interpolation

Level j
input image



a
b

FIGURE 7.2
(a) An image pyramid. (b) A simple system for creating approximation and prediction residual pyramids.

3rd Edition

Image Pyramids (cont...)

- Approximation Pyramid

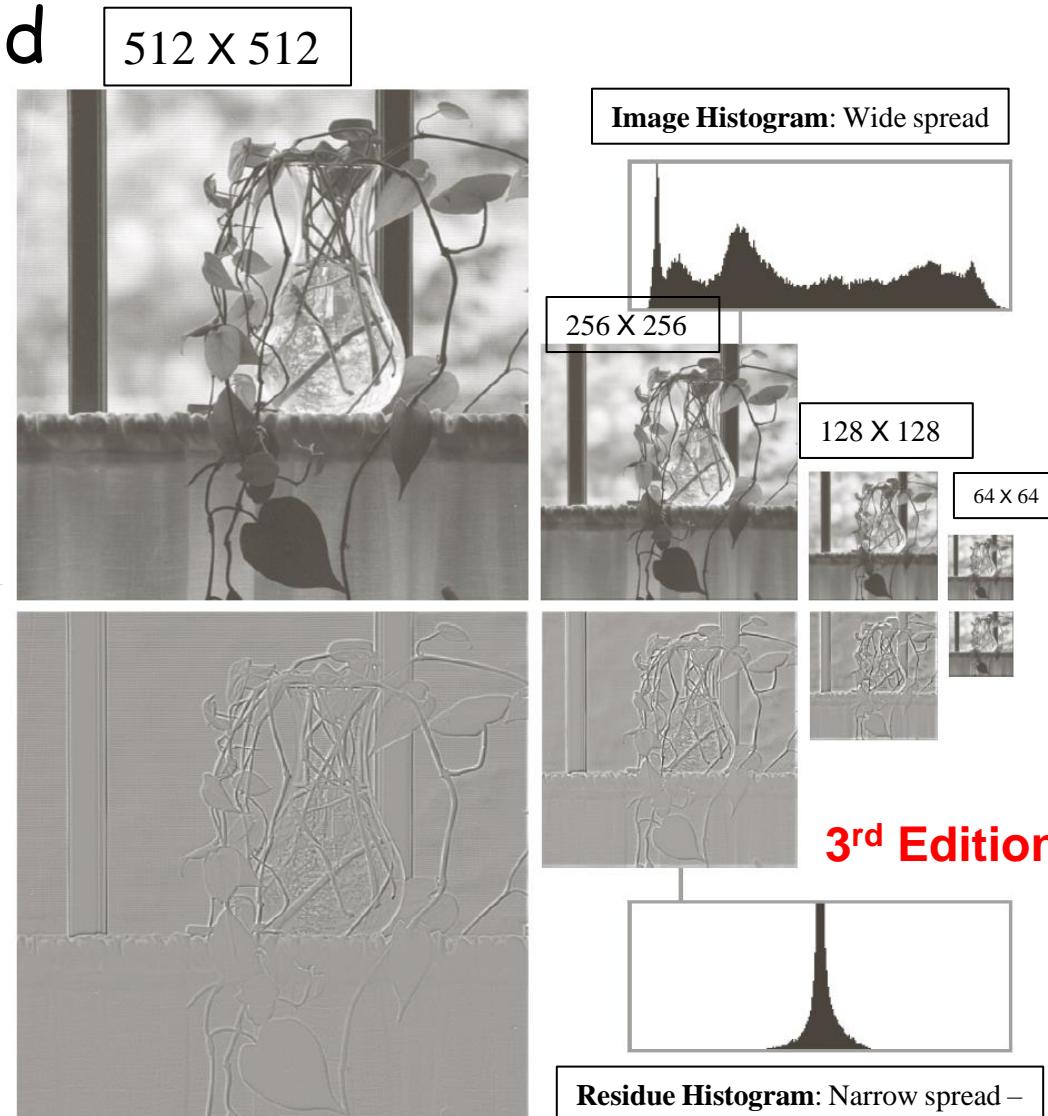
- $512 \times 512 (j=9)$
to $64 \times 64 (j=6)$

- Prediction Residual pyramid

- 64×64 approximation image at top of pyramid, rest are residuals
- Higher compressibility (fewer bits)

a
b

FIGURE 7.3
Two image pyramids and their histograms:
(a) an approximation pyramid;
(b) a prediction residual pyramid.

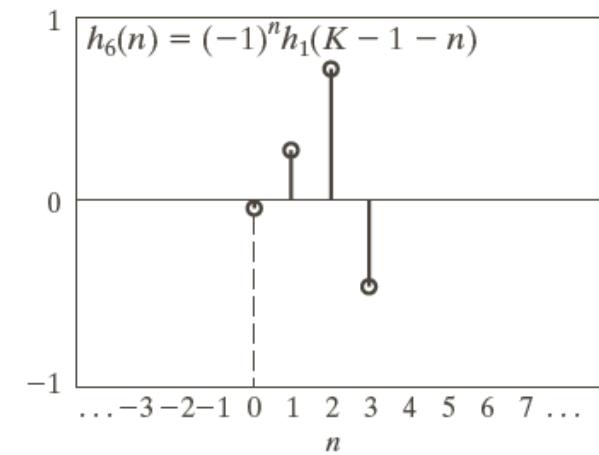
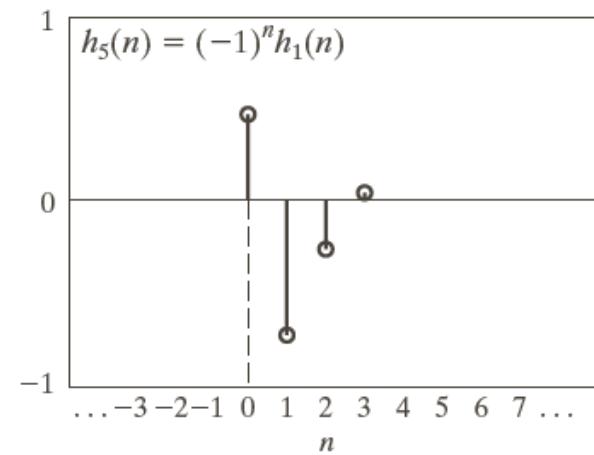
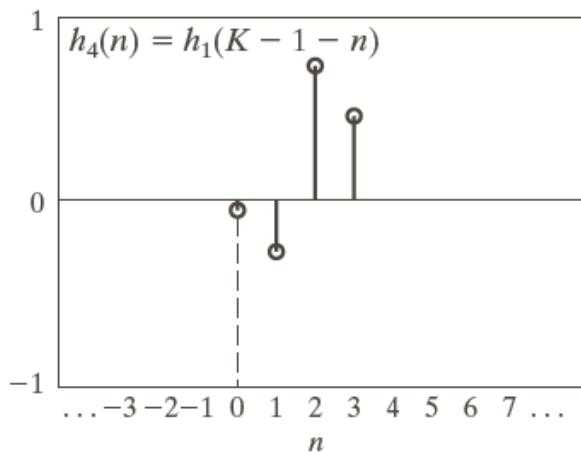
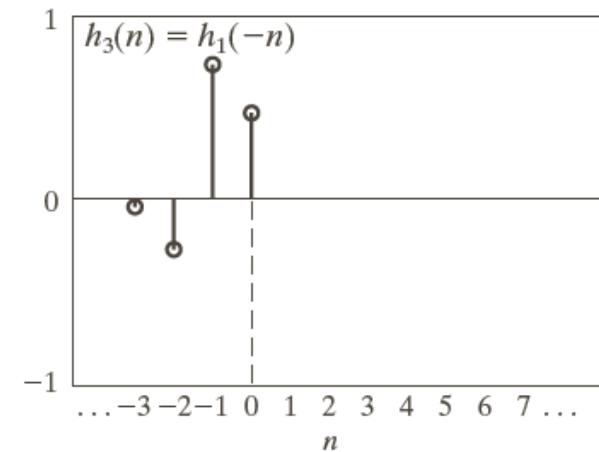
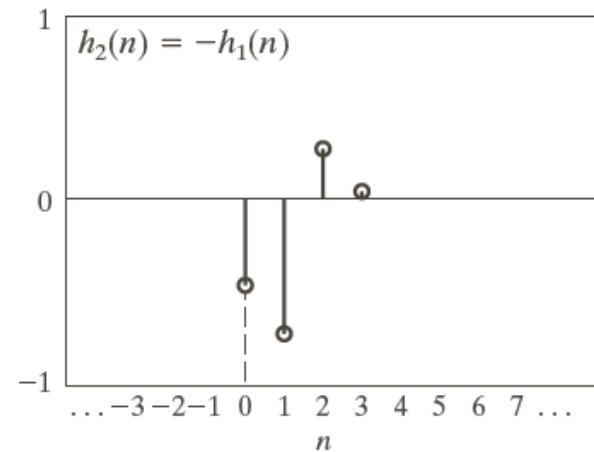
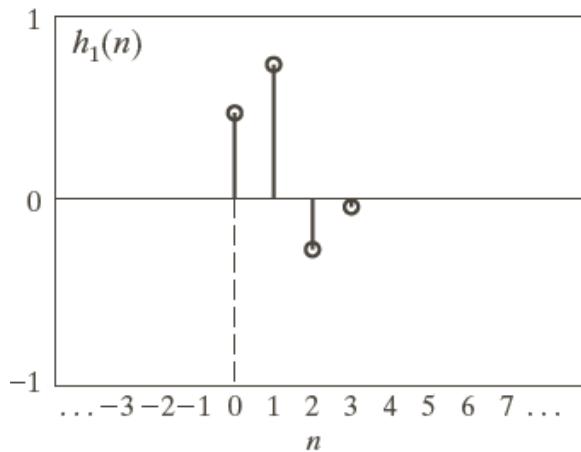


7.1.2 Subband Coding

- An image is decomposed into a set of bandlimited components (subbands).
- The decomposition is carried out by digital filtering (orthogonal or bi-orthogonal) and downsampling.
- If the filters are properly selected the image may be reconstructed without error by filtering and upsampling.

3rd Edition

Functionally Related Filters



a	b	c
d	e	f

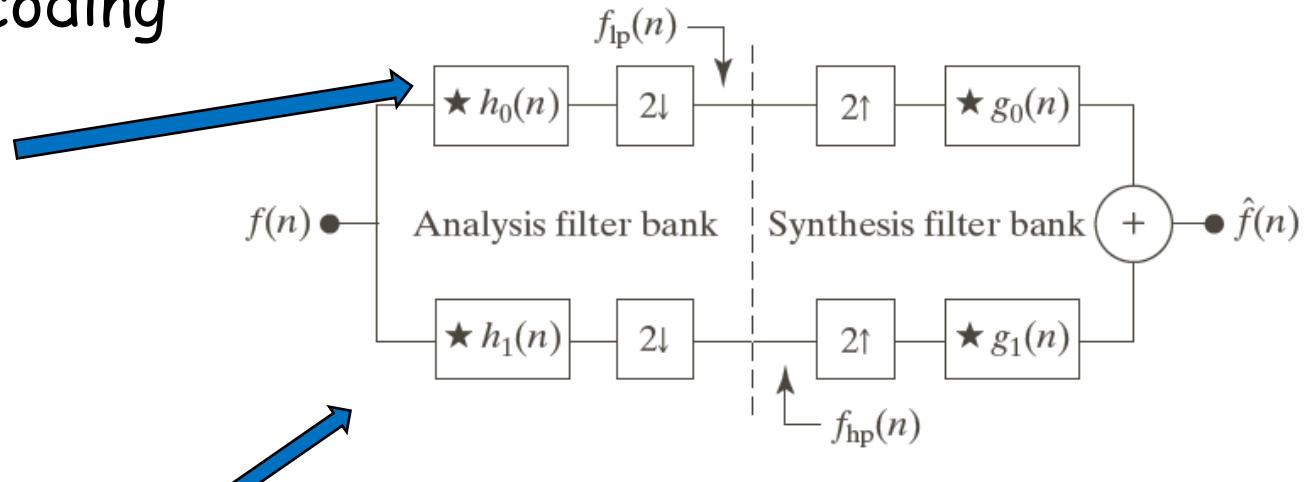
FIGURE 7.5 Six functionally related filter impulse responses: (a) reference response; (b) sign reversal; (c) and (d) order reversal (differing by the delay introduced); (e) modulation; and (f) order reversal and modulation.

3rd Edition

Two-Band Subband Coding

Approximation
filter (low pass)

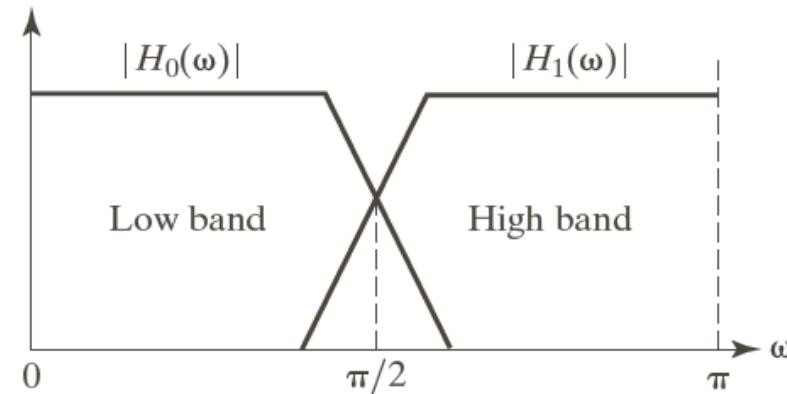
A two-band subband
coding



Detail filter (high pass)

a
b

FIGURE 7.6
(a) A two-band subband coding and decoding system, and (b) its spectrum splitting properties.



3rd Edition



7.1.2 Subband Coding

- Perfect Reconstruction filters

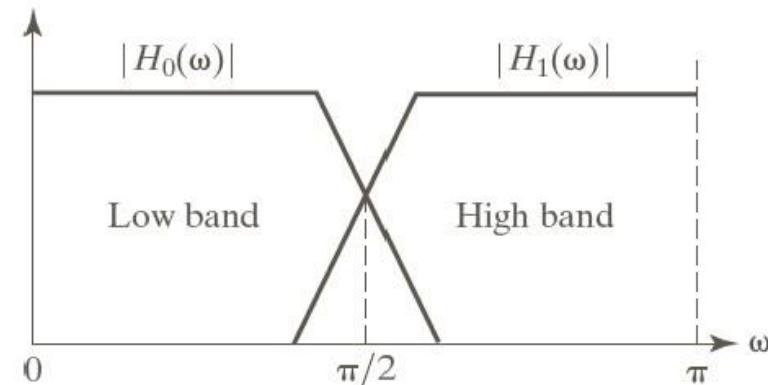
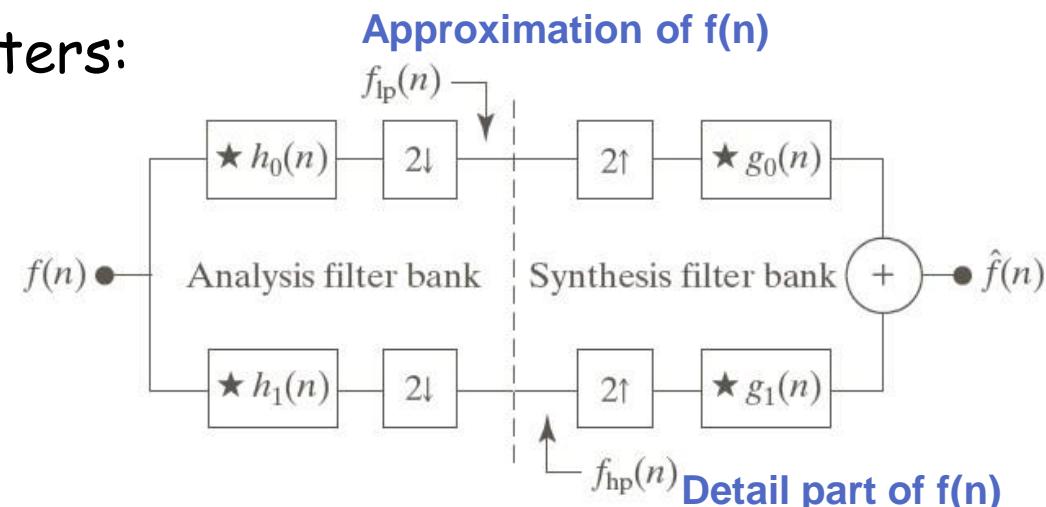
$$\hat{f}(n) = f(n)$$

- Biorthogonal filters
 - Two prototypes required

$$g_0(n) = (-1)^n h_1(n)$$

$$g_1(n) = (-1)^{n+1} h_0(n)$$

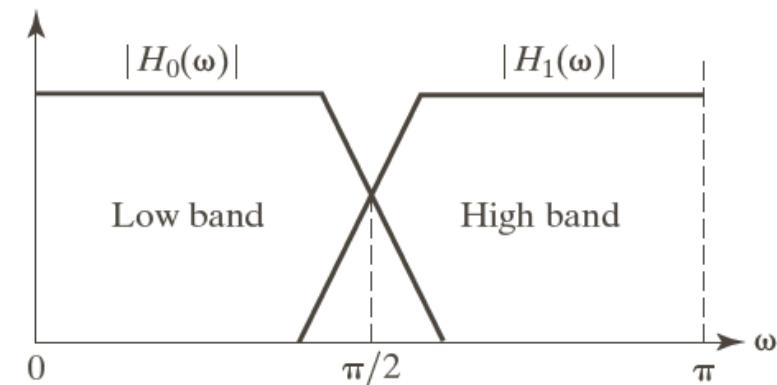
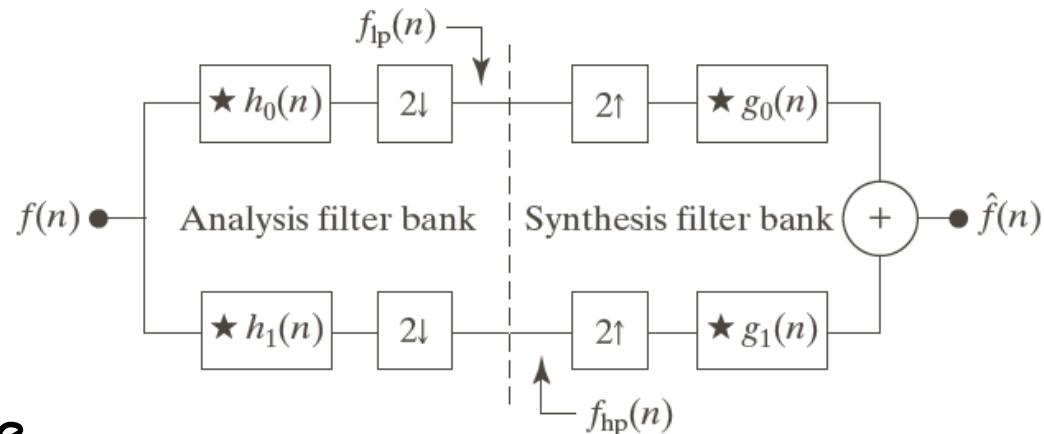
- Orthonormal filters
 - Given a single prototype filter, remaining three can be computed to satisfy the orthonormality constraint



3rd Edition

Subband Coding (cont...)

- The goal of subband coding is to select the analysis and synthesis filters in order to have *perfect reconstruction* of the signal.
- It may be shown that the synthesis filters should be modulated versions of the analysis filters with one (and only one) synthesis filter being sign reversed of an analysis filter.



3rd Edition

Subband Coding (cont...)

The analysis and synthesis filters should be related in one of the two ways:

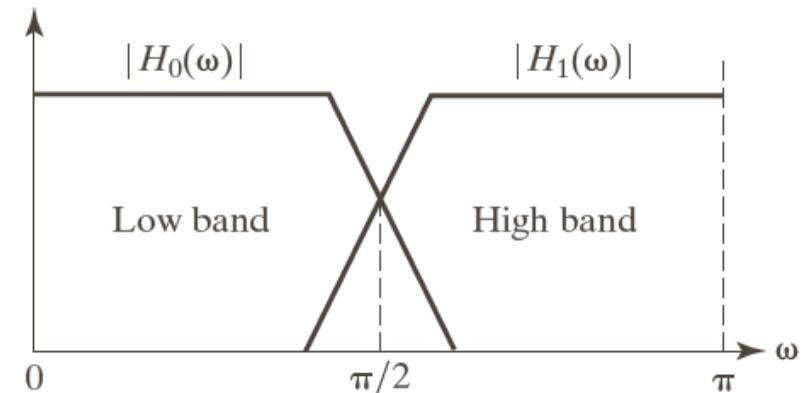
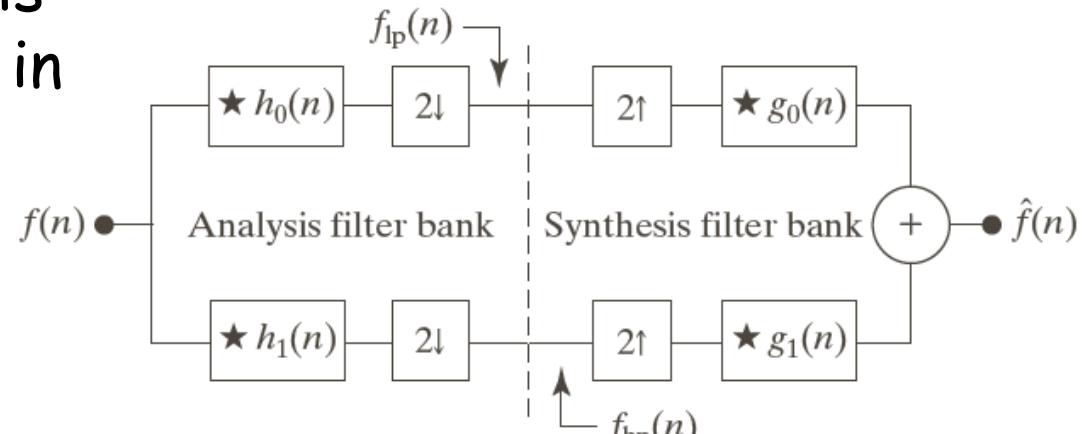
$$g_0(n) = (-1)^n h_1(n)$$

$$g_1(n) = (-1)^{n+1} h_0(n)$$

or

$$g_0(n) = (-1)^{n+1} h_1(n)$$

$$g_1(n) = (-1)^n h_0(n)$$



These filters are called *cross-modulated*. (because related by modulation with diagonal opposite in block-diagram)

Subband Coding (cont...)

- These filters are also *biorthogonal* (See Sec 7.2.1):

$$\langle h_i(2n-k), g_j(k) \rangle = \delta(i-j)\delta(n), \quad i, j = \{0,1\}$$

- Of special interest in subband coding are filters that go beyond biorthogonality and require to be *orthonormal* also (for FWT, and perfect Reconstruction; See 7.4) :

$$\langle g_i(n), g_j(n+2m) \rangle = \delta(i-j)\delta(m), \quad i, j = \{0,1\}$$

- Orthonormal filters satisfy the following conditions:

$$g_1(n) = (-1)^n g_0(K_{\text{even}} - 1 - n)$$

$$h_i(n) = g_i(K_{\text{even}} - 1 - n), \quad i = \{0,1\}$$

- Subscript "even" means that the size of the filter should be even

3rd Edition

Summary Notes:

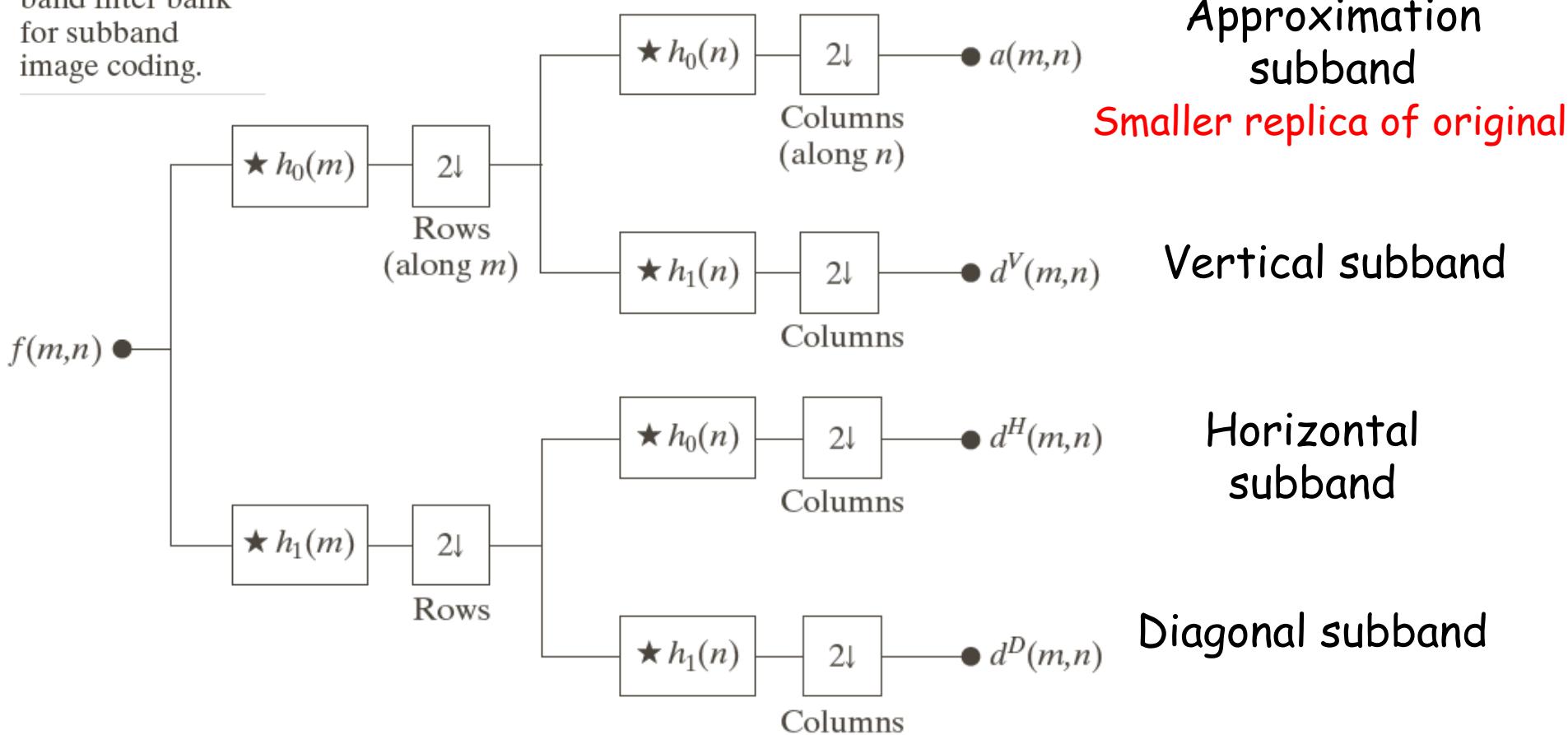
- Synthesis filters (g_0 and g_1): Related by order reversal and modulation.
- Analysis filters (h_0 and h_1): Both order reversed versions of the synthesis filters.
- An orthonormal filter bank may be constructed by starting with the impulse response of g_0 which is called the *prototype*.
- Biorthogonal filter banks require two prototypes, h_0 and h_1 . Other two found using 7.1-10 or 7.1-11
- 1-D orthonormal filters may be used as 2-D separable filters for subband image coding.

3rd Edition

Subband Coding (cont...)

FIGURE 7.7

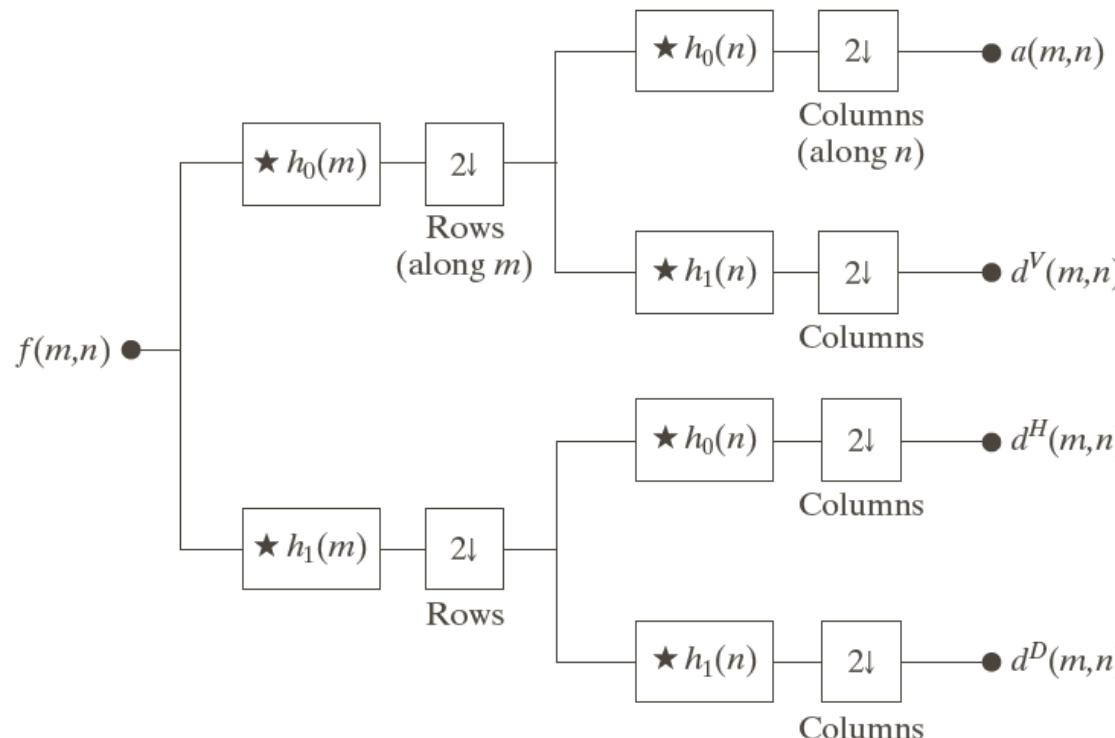
A two-dimensional, four-band filter bank for subband image coding.



3rd Edition

Subband Coding (cont...)

- The subbands may be subsequently further split into smaller subbands.
- Image synthesis is obtained by reversing the procedure.

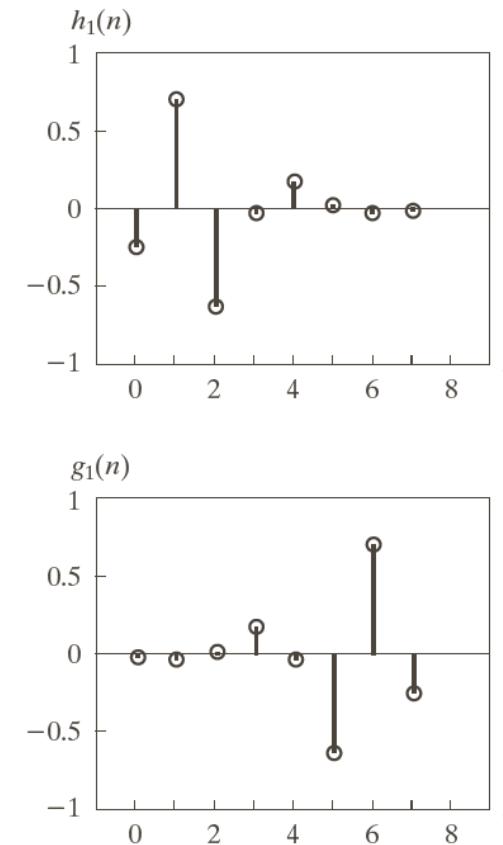
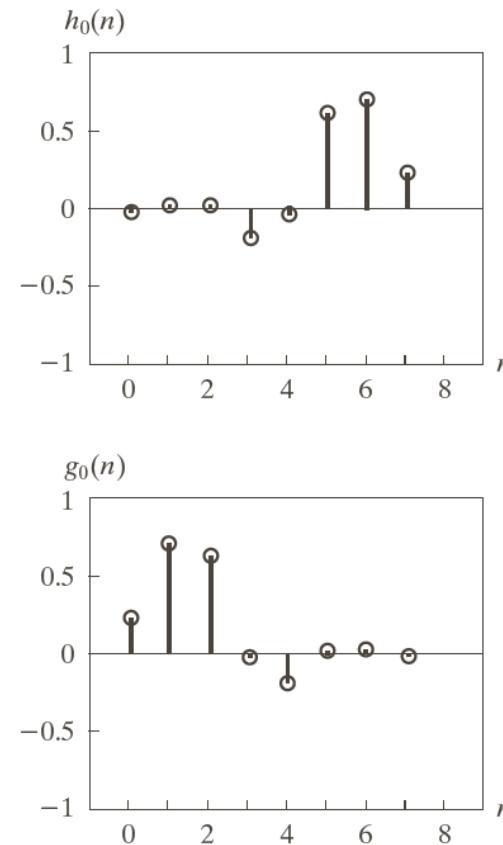


3rd Edition

➤ Both Orthonormal and Biorthogonal

n	$g_0(n)$
0	0.23037781
1	0.71484657
2	0.63088076
3	-0.02798376
4	-0.18703481
5	0.03084138
6	0.03288301
7	-0.01059740

TABLE 7.1
 Daubechies 8-tap
 orthonormal filter
 coefficients for
 $g_0(n)$ (Daubechies
 [1992]).



a b
c d

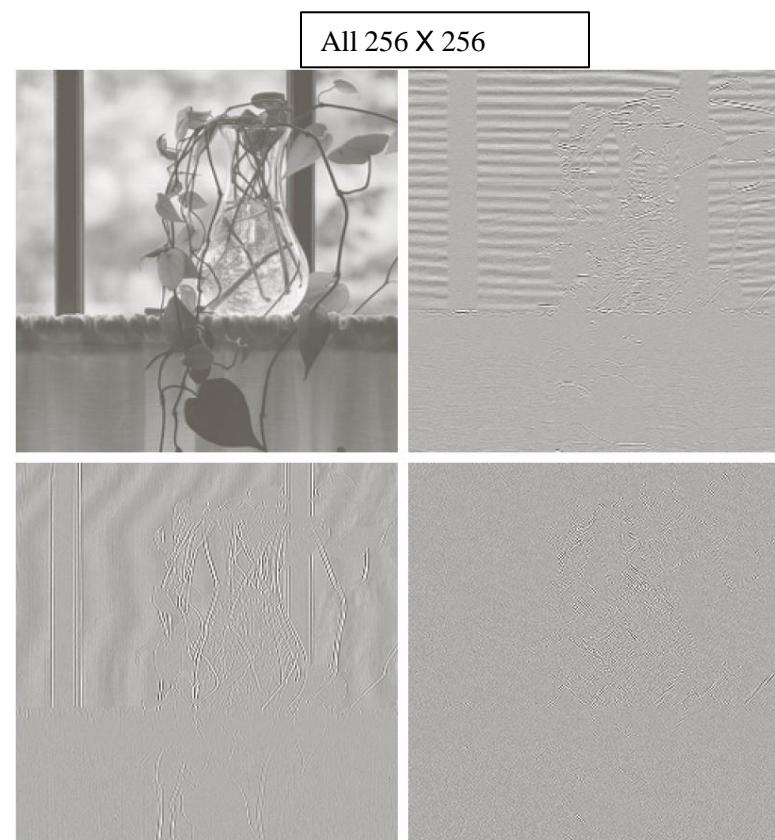
FIGURE 7.8
 The impulse
 responses of four
 8-tap Daubechies
 orthonormal
 filters. See
 Table 7.1 for the
 values of $g_0(n)$ for
 $0 \leq n \leq 7$.

3rd Edition

Subband Coding (cont...)

- The wavy lines are due to aliasing of the barely discernible window screen.
- Despite the aliasing, the image may be perfectly reconstructed.
- Original Figure 7.1
 512×512
- Subband sizes
 256×256

Key Fact: Subbands are compressible with fewer bits

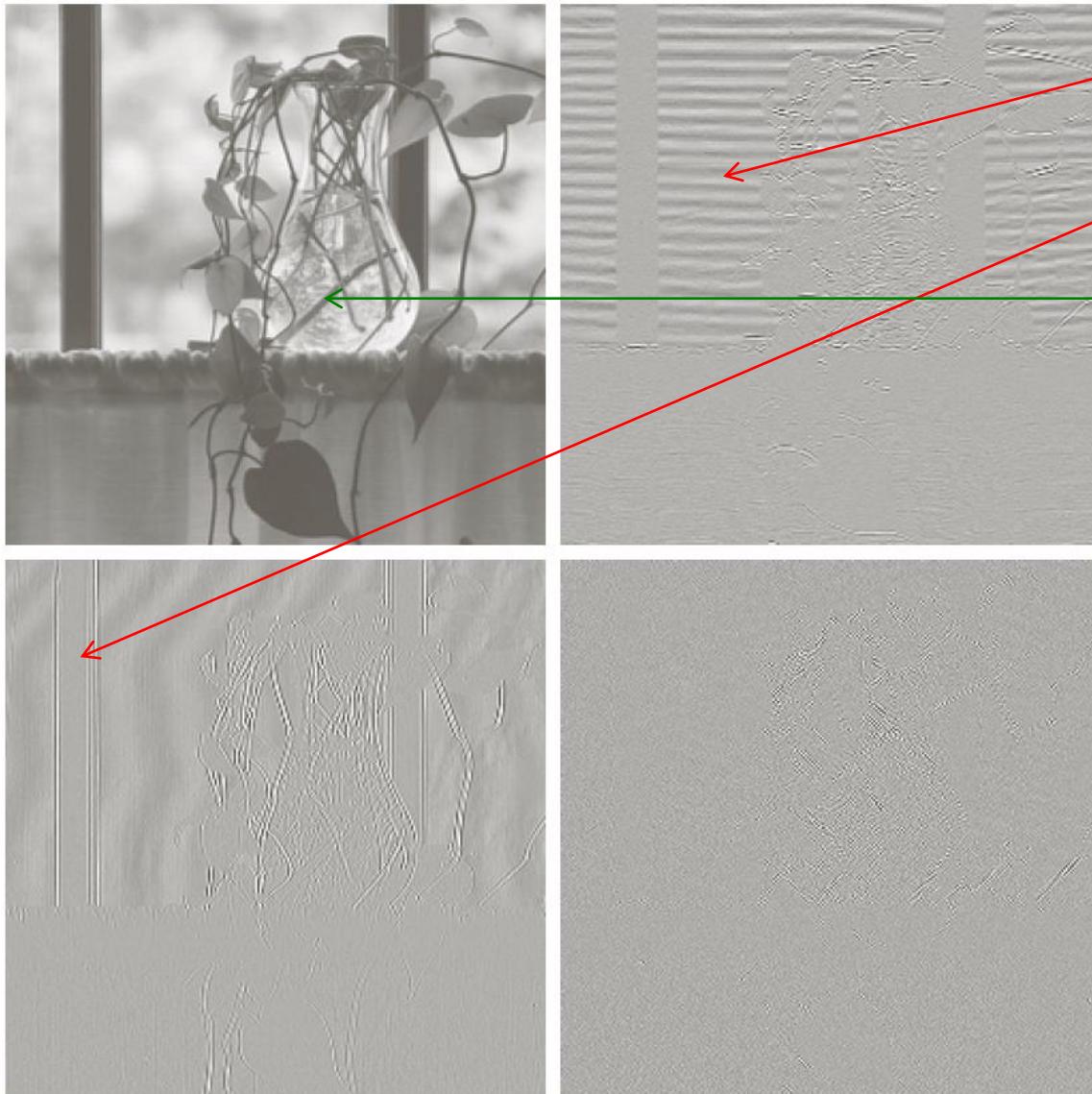


a	b
c	d

FIGURE 7.9
A four-band split of the vase in Fig. 7.1 using the subband coding system of Fig. 7.7. The four subbands that result are the (a) approximation, (b) horizontal detail, (c) vertical detail, and (d) diagonal detail subbands.

3rd Edition

Subband Coding (cont...)



- Horizontal Details
- Subsampled → Aliased

- Vertical Details

Smaller replica of original

a	b
c	d

FIGURE 7.9
A four-band split of the vase in Fig. 7.1 using the subband coding system of Fig. 7.7. The four subbands that result are the (a) approximation, (b) horizontal detail, (c) vertical detail, and (d) diagonal detail subbands.