Multiresolution and wavelets

3D Image Processing
Alireza Ghane

Readings

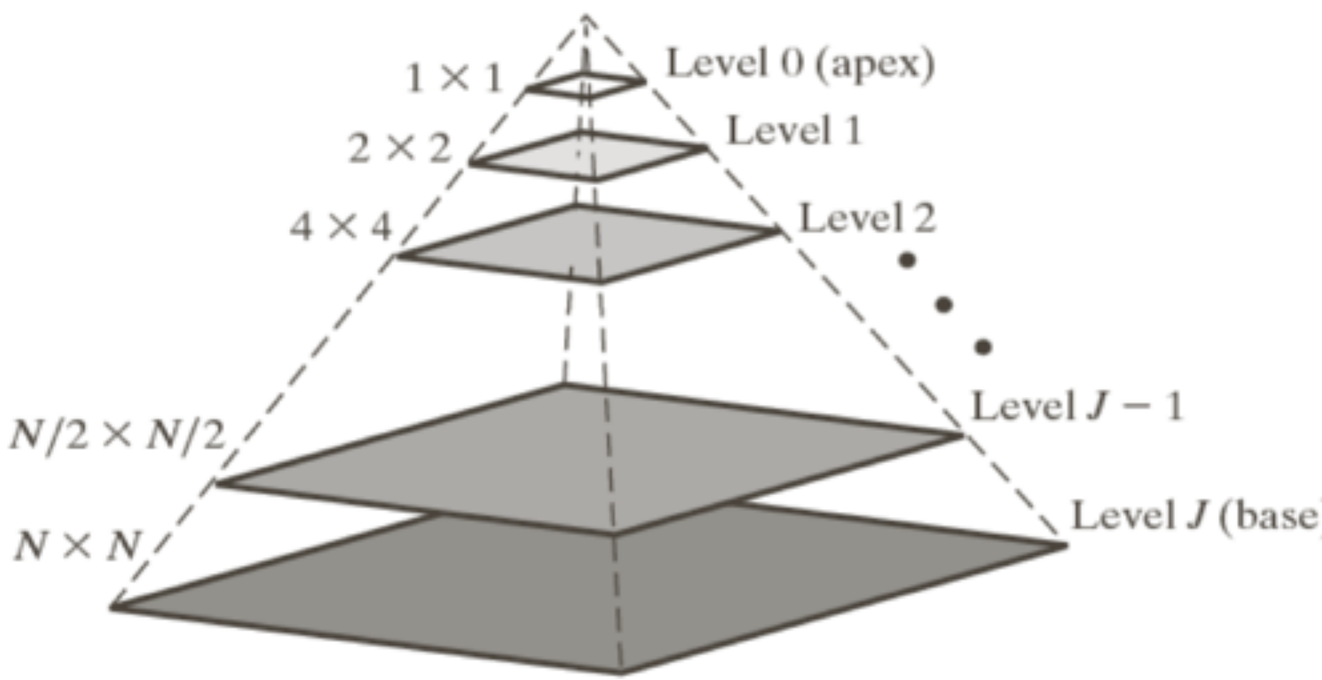
- The Gonzales + Woods, Chapter 7

Overview

- Background
  - image pyramids
  - subband coding
  - Haar
- Multiresolution expansions
- Wavelet transform in 1D
- Fast wavelet transform
- Wavelet transform in 2D
- Wavelet packets

Image pyramids

- sequence of same image of decreasing resolution



Important operators

- $2\downarrow$ — Downsampling:

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

- $2\uparrow$ — Upsampling:

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

Image pyramids

- can be build recursively

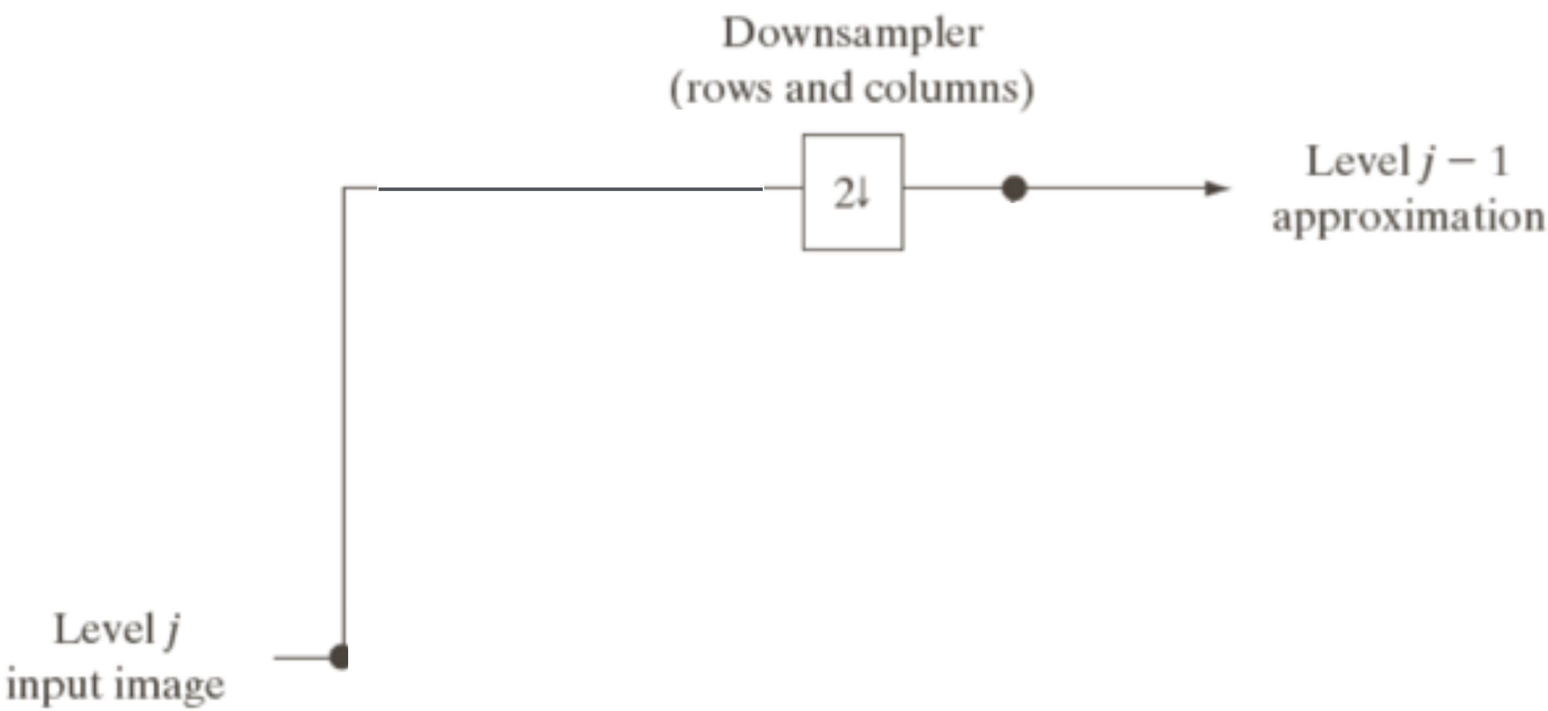


Image pyramids

- can be build recursively

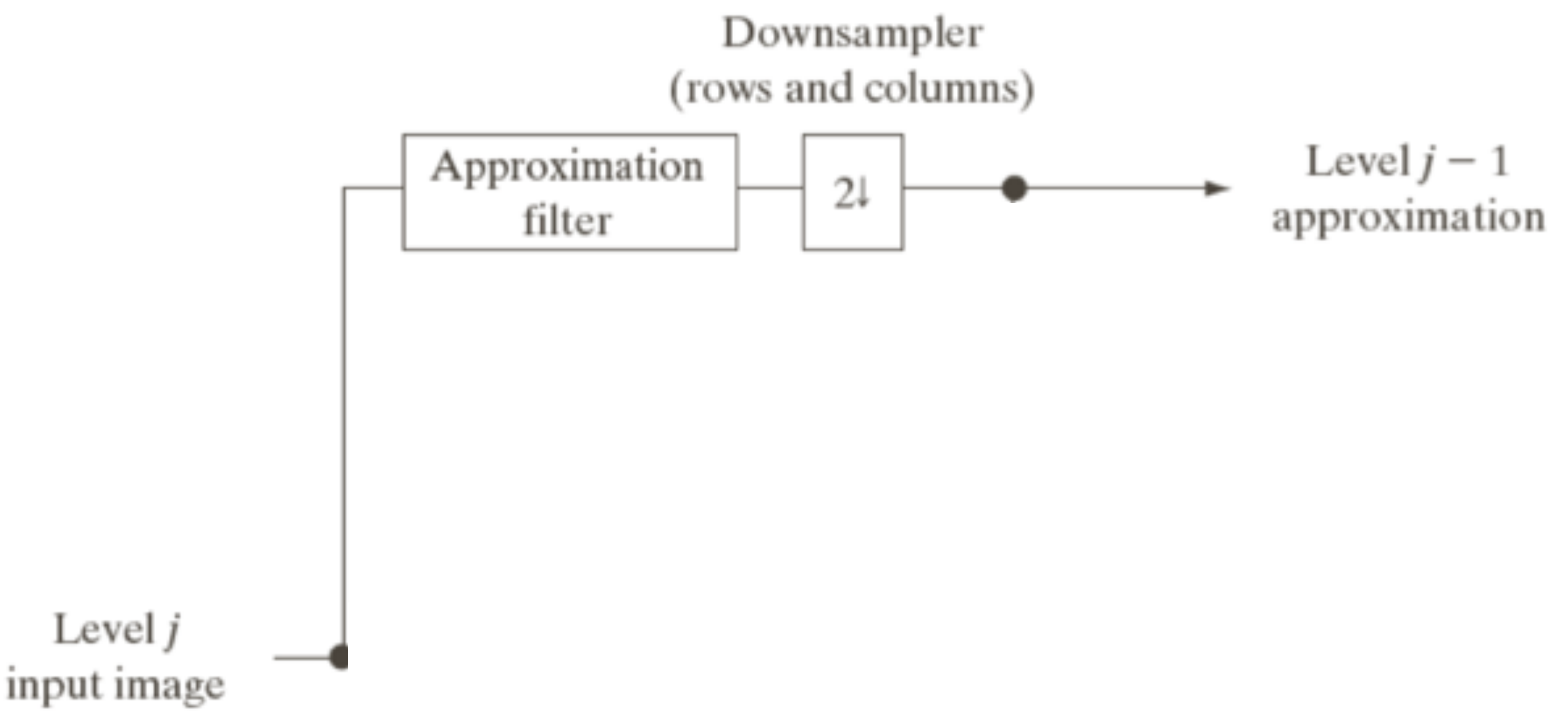


Image pyramids

- can be build recursively

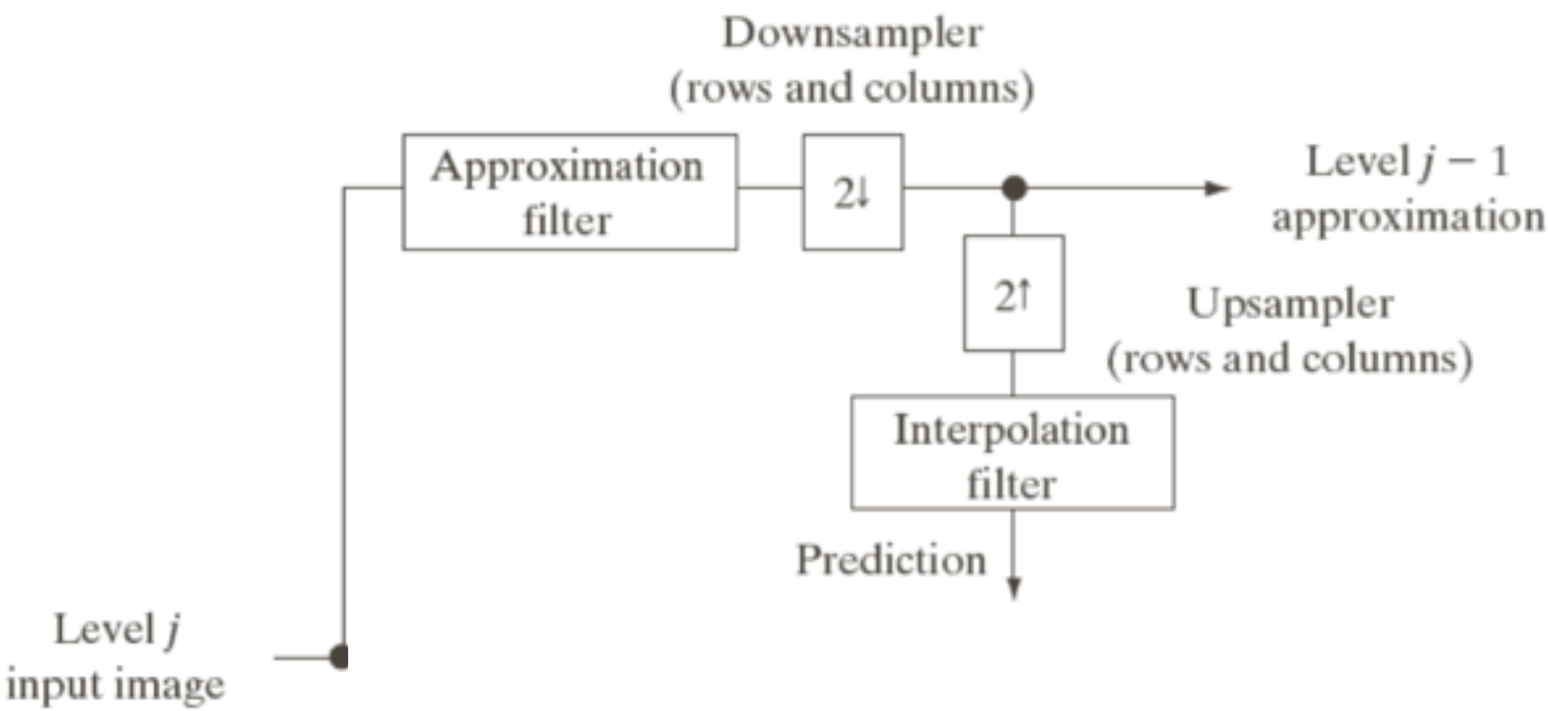


Image pyramids

- can be build recursively

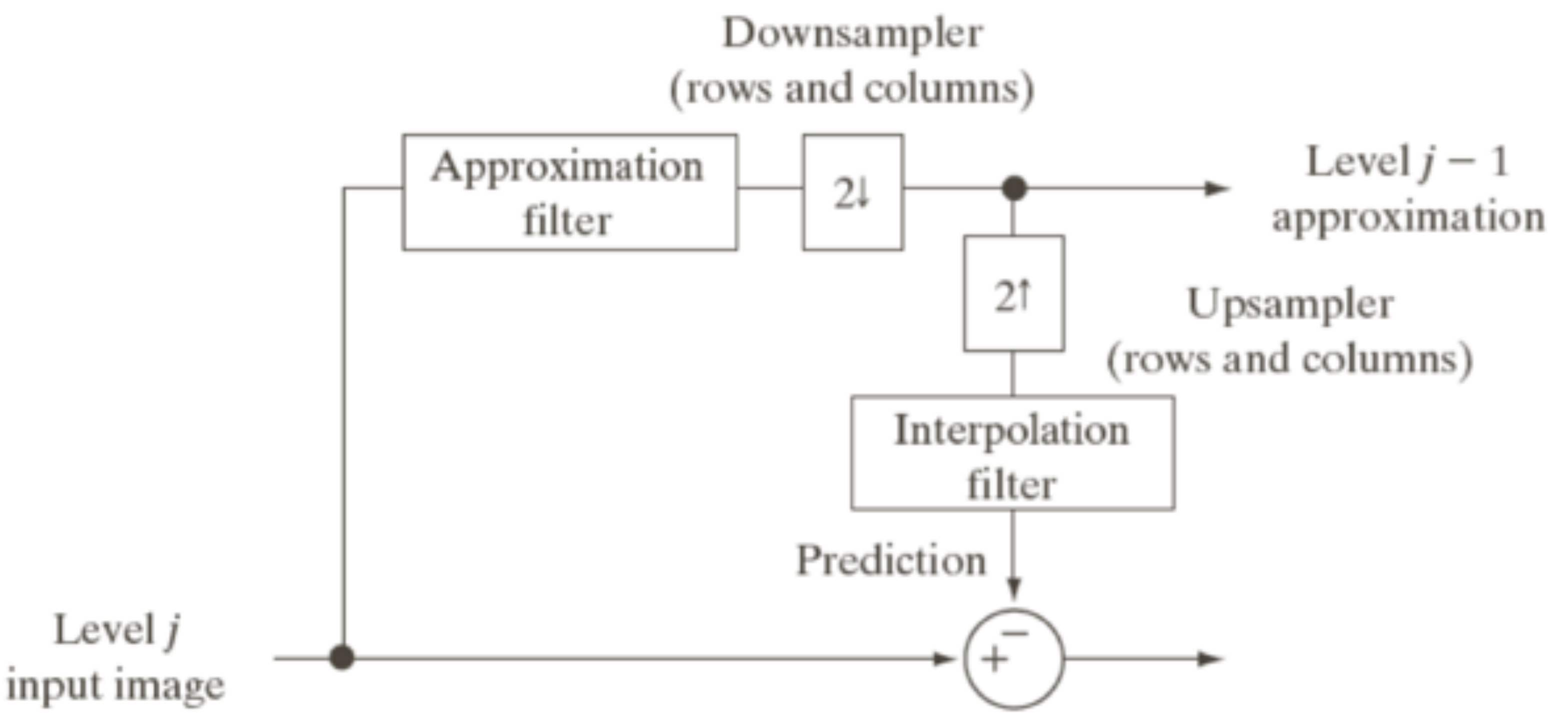


Image pyramids

- can be build recursively

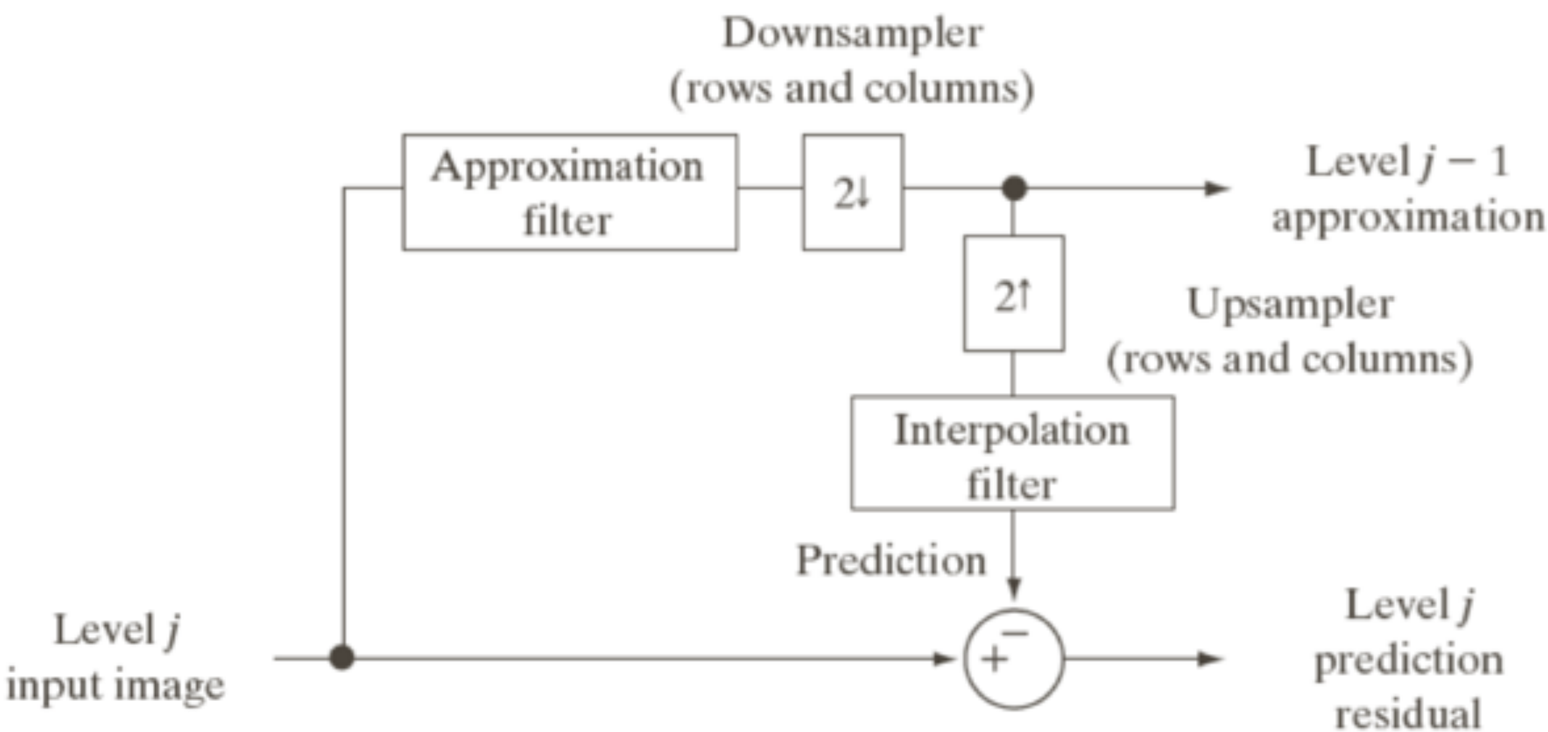


Image pyramids

- can be build recursively
  - approximate level j-1 from level j
    - filter, then
    - downsample (by a factor of 2)
  - compute an estimate of level j from level j-1
    - upsample, then
    - filter
  - compute difference between level j and its estimation

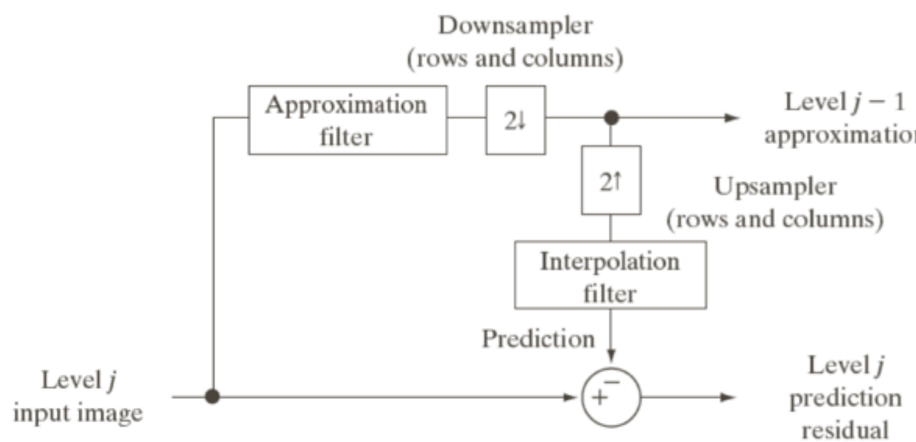
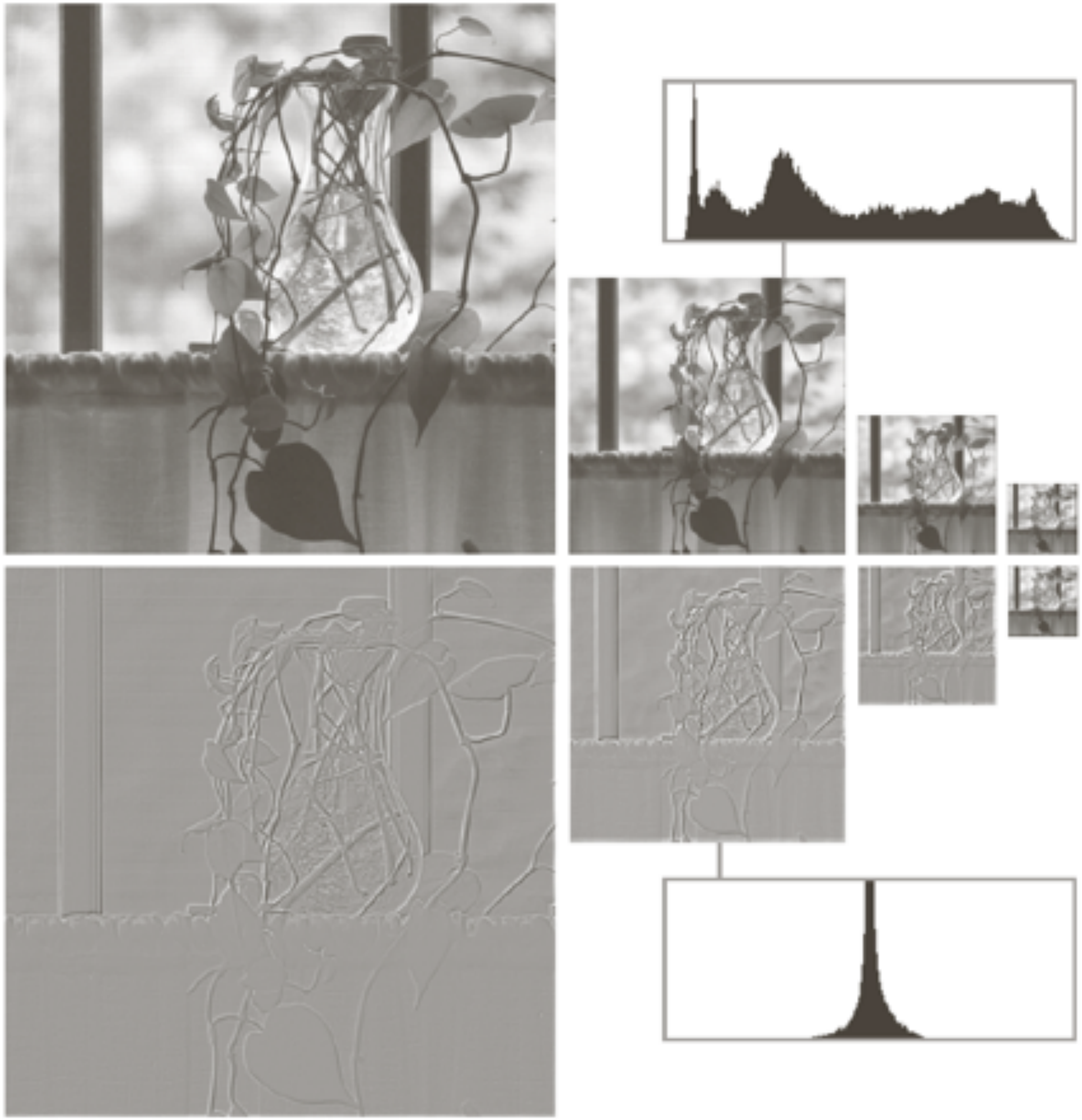


Image pyramids



a
b

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

Image pyramids

a
b

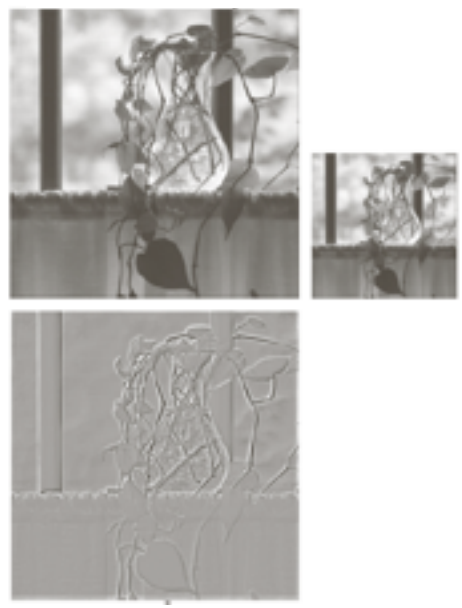
FIGURE 7.3 Two image pyramids and their histograms: (a) an approximation pyramid; (b) a prediction residual pyramid.

Image pyramids

a
b

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

Image pyramids



a
b

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

Image pyramids

a
b

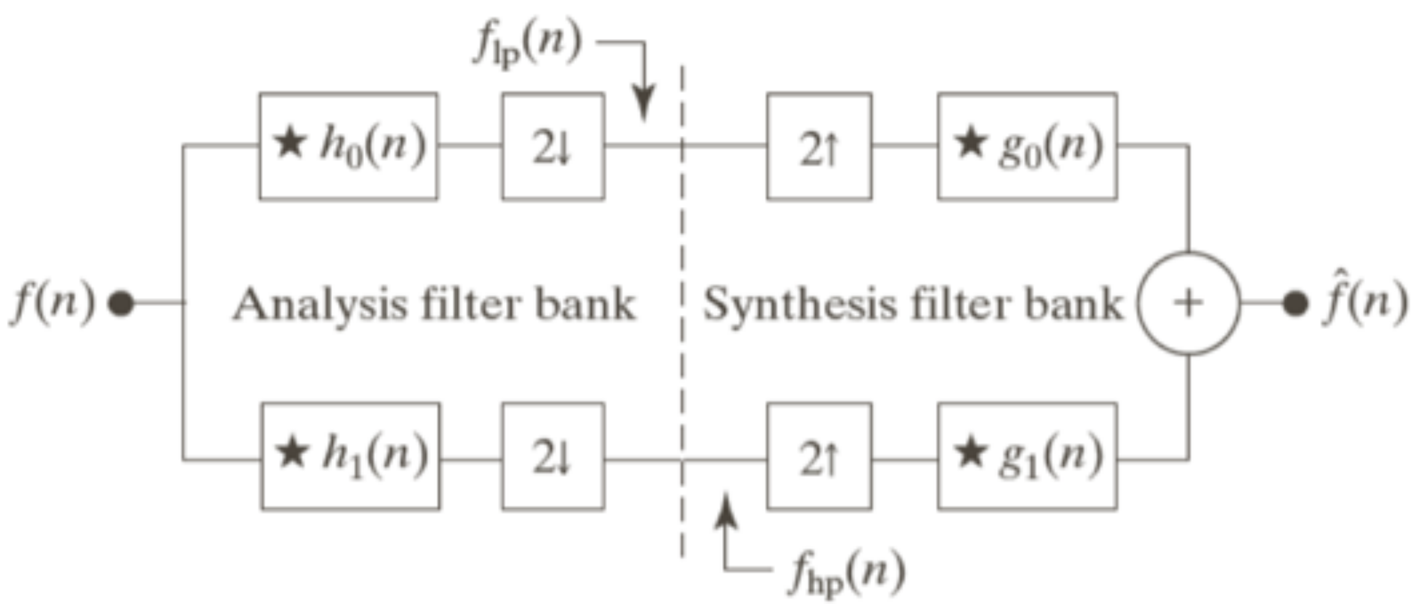
FIGURE 7.3
Two image pyramids and their histograms: (a) an approximation pyramid; (b) a prediction residual pyramid.

Image pyramids

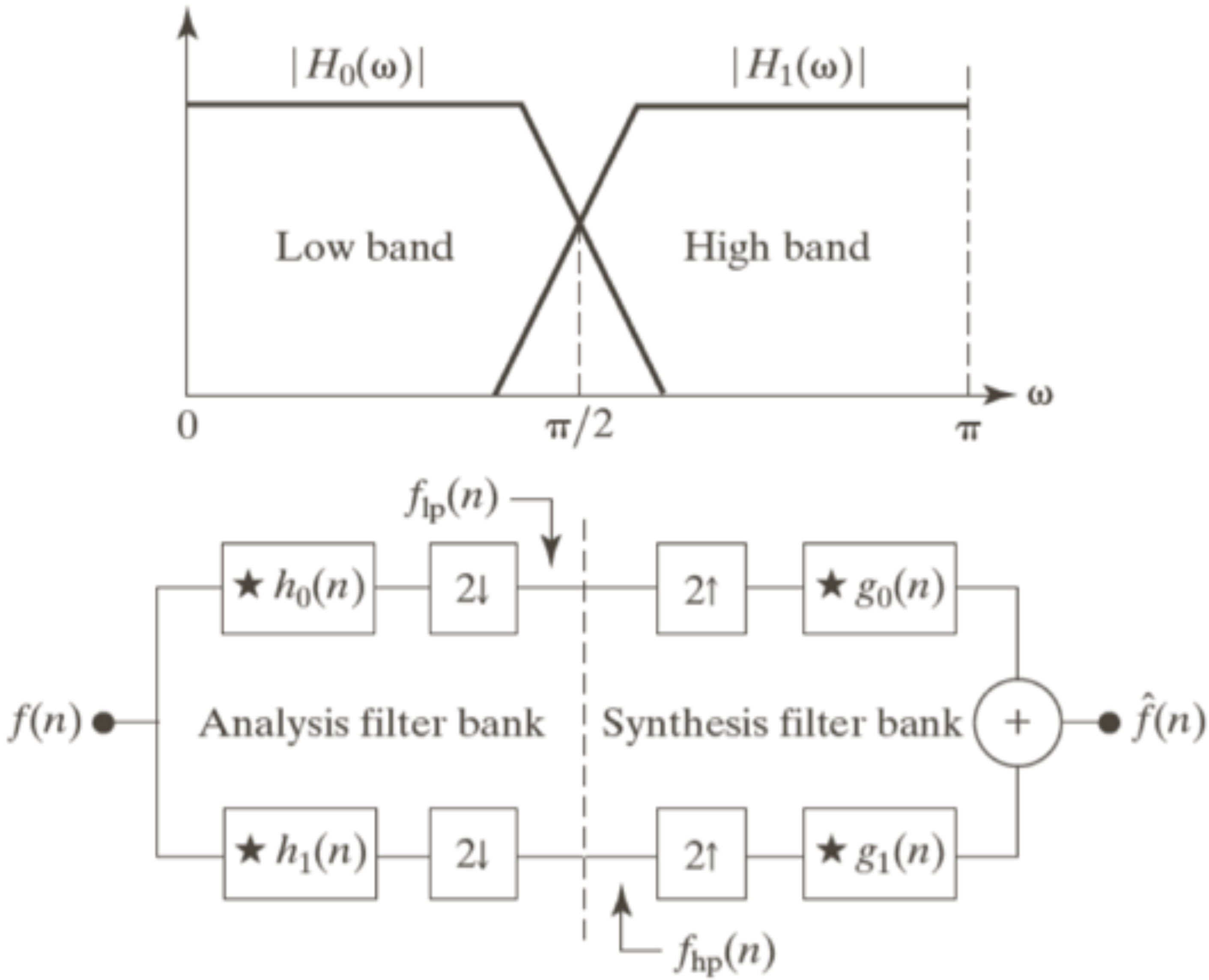
FIGURE 7.3 Two image pyramids and their histograms: (a) an approximation pyramid; (b) a prediction residual pyramid.

Subband coding

- in image is decomposed into a set of bandlimited components
- try to keep all information to reassemble image from its components



Subband coding



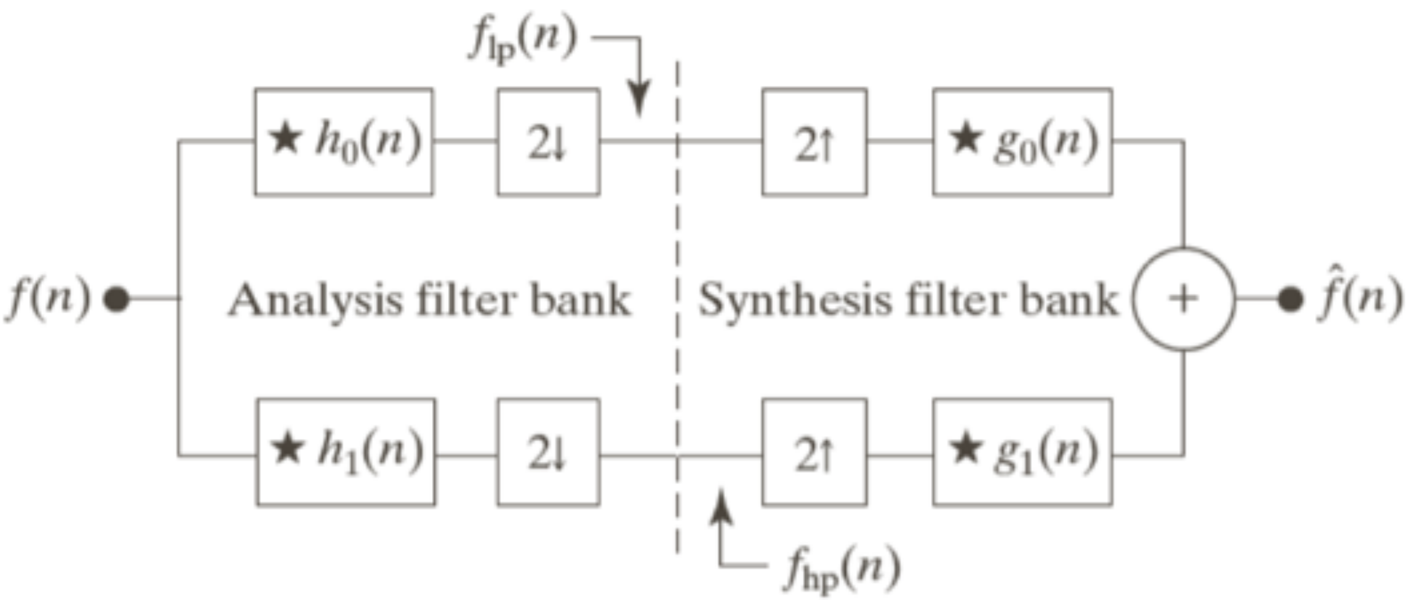
Subband coding

- building 'perfect reconstruction filter banks':

$$g_0(n) = (-1)^n h_1(n)$$
$$g_1(n) = (-1)^{n+1} h_0(n)$$

or

$$g_0(n) = (-1)^{n+1} h_1(n)$$
$$g_1(n) = (-1)^n h_0(n)$$



Subband coding

- building 'perfect reconstruction filter banks':

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

or

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

- fulfills biorthogonality:

$$\langle h_i(2n-k), g_j(k)\rangle = \delta(i-j)\delta(n)$$

- desire also orthonormality:

$$\langle g_i(n), g_j(n+2m)\rangle = \delta(i-j)\delta(m)$$

Subband coding

- orthonomality requires:

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

$$h_i(n) = g_i(K_{\text{even}} - 1 - n)$$

Subband coding in 2D

- doing things first by row and then by column

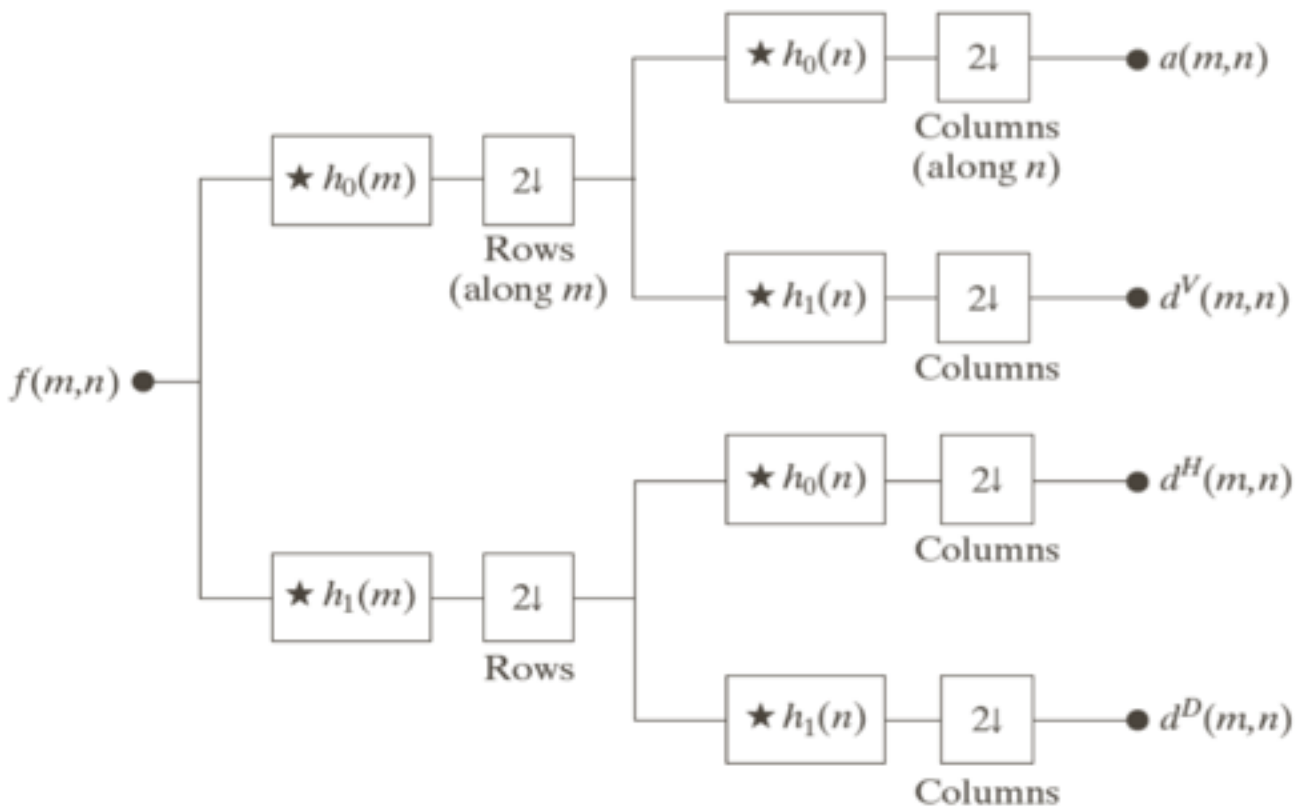
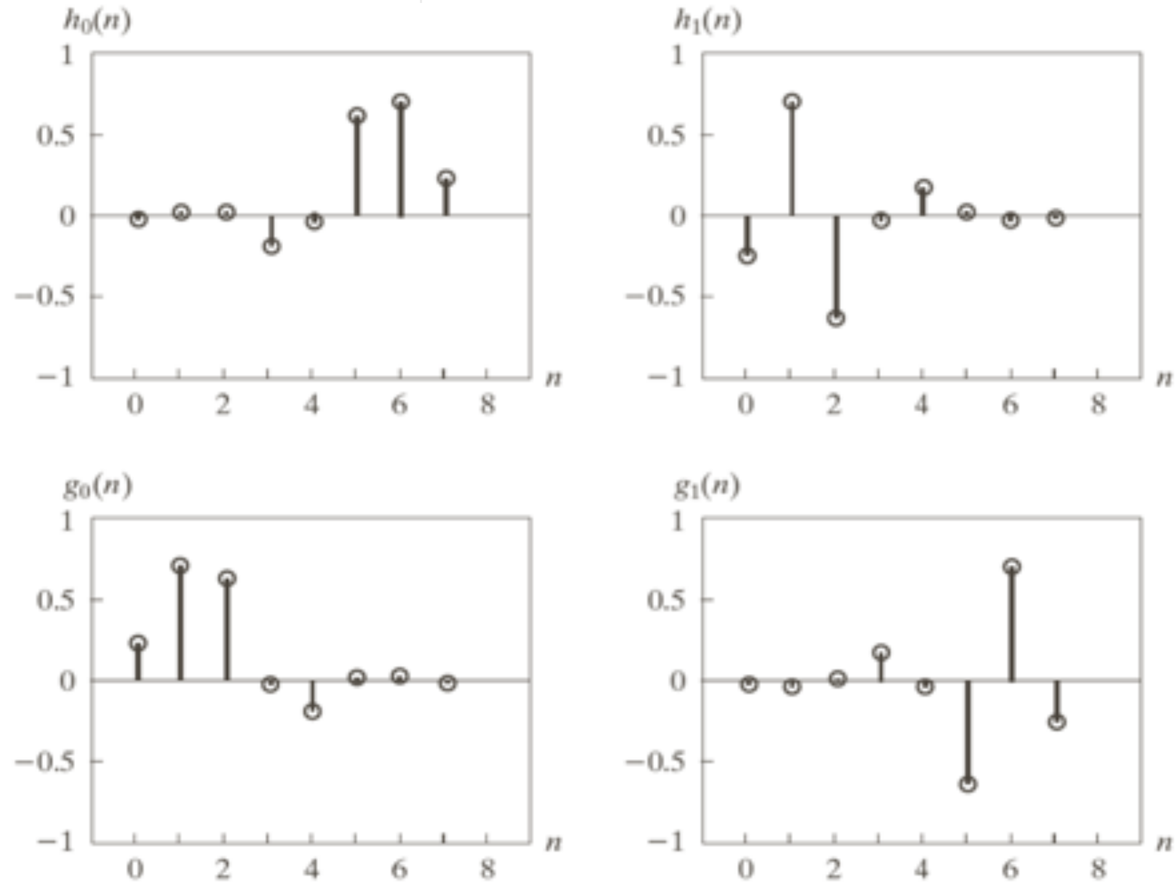


FIGURE 7.7 A two-dimensional, four-band filter bank for subband image coding.

Example

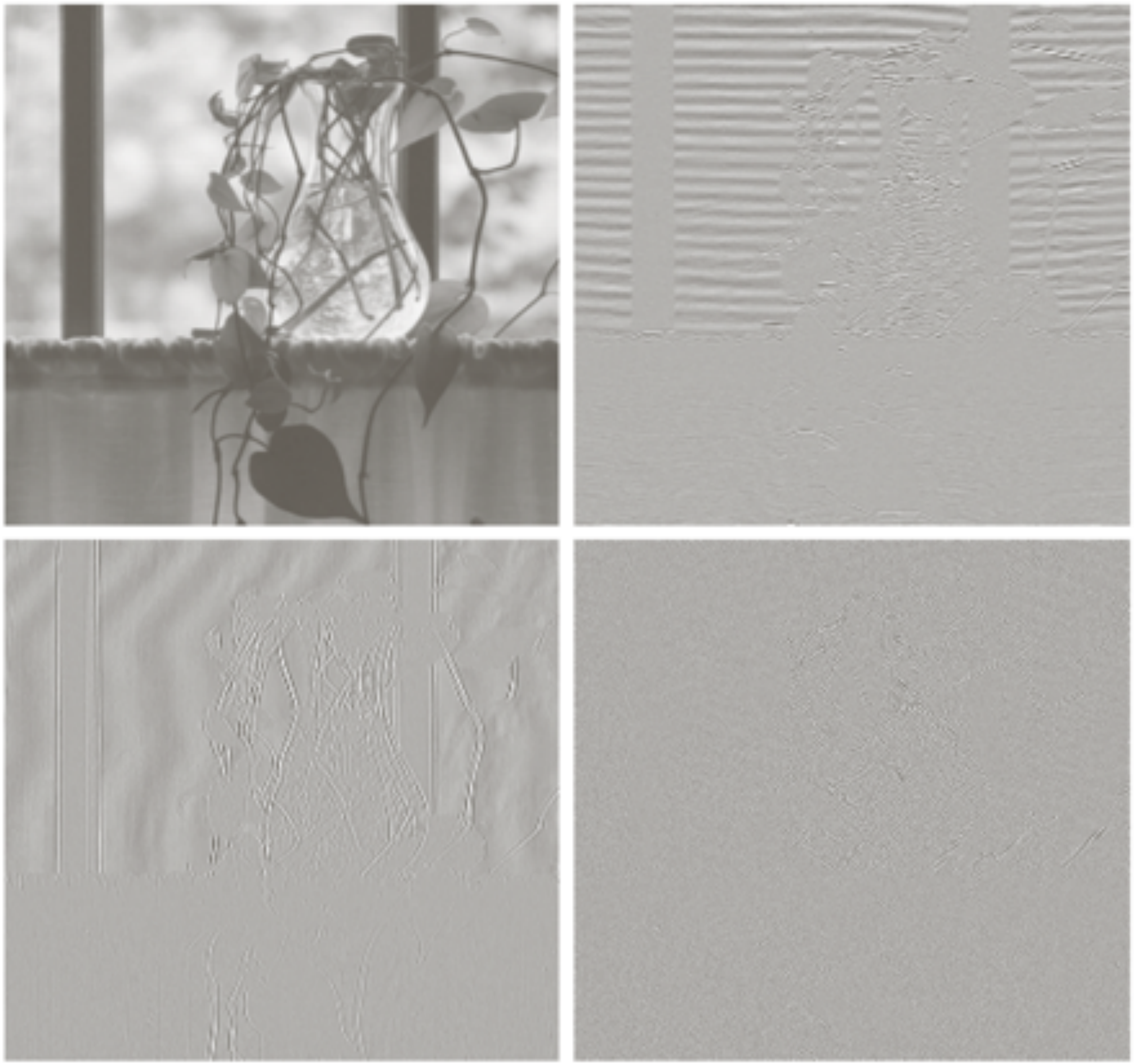
FIGURE 7.8
The impulse responses of four 8-tap Daubechies orthonormal filters. See Table 7.1 for the values of $g_0(n)$ for $0 \le n \le 7$.



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]).

Example



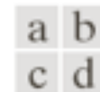


FIGURE 7.9 A four-band split of the vase in Fig. 7.1 using the subband coding system of Fig. 7.7. The four subbands that result are the (a) approximation, (b) horizontal detail, (c) vertical detail, and (d) diagonal detail subbands.

Haar

- perhaps the oldest transform (1910)
- certainly the most simplest

$$H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$H_4 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

Haar transform example

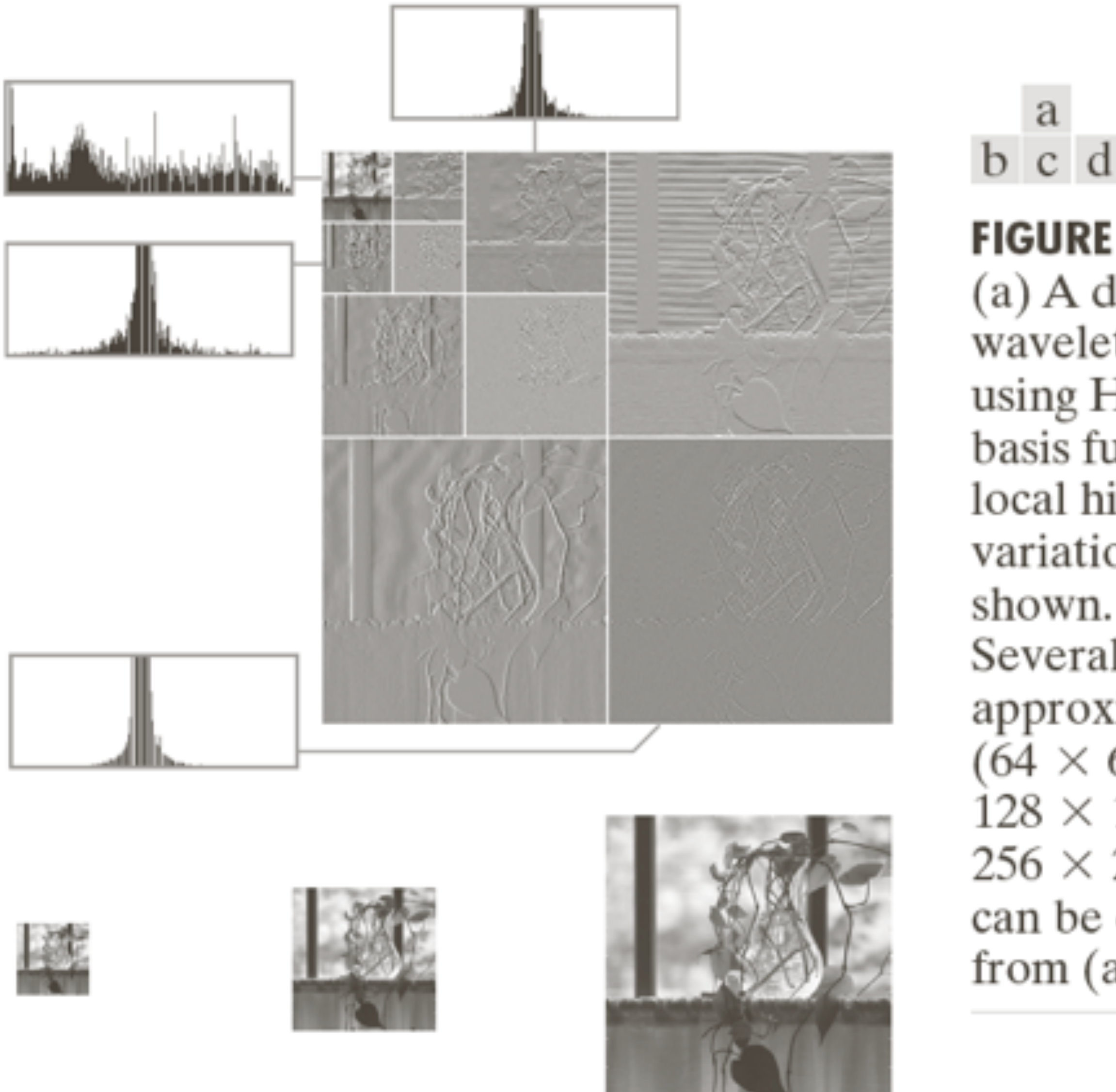


FIGURE 7.10 (a) A discrete wavelet transform using Haar $\mathbf{H}_2$ basis functions. Its local histogram variations are also shown. (b)–(d) Several different approximations (64×64, 128×128, and 256×256) that can be obtained from (a).

Overview

- Background
  - image pyramids
  - subband coding
  - Haar
- Multiresolution expansions
- Wavelet transform in 1D
- Fast wavelet transform
- Wavelet transform in 2D
- Wavelet packets

Overview

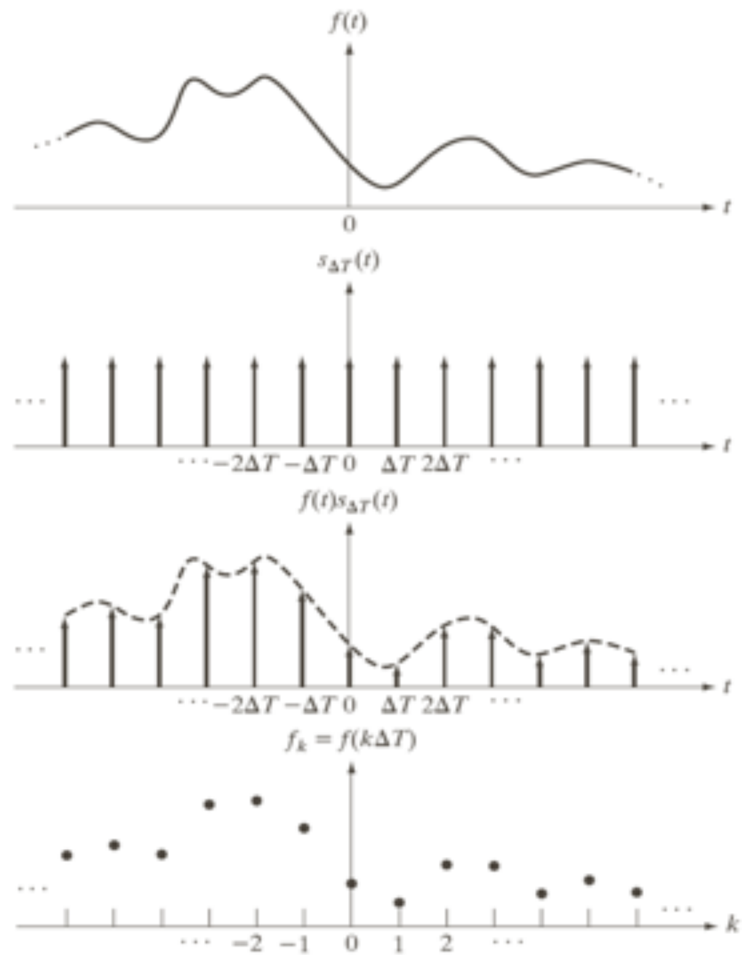
- Background
- Multiresolution expansions
  - understanding spaces
  - scaling functions
  - wavelet functions
- Wavelet transform in 1D
- Fast wavelet transform
- Wavelet transform in 2D
- Wavelet packets

What is sampling?

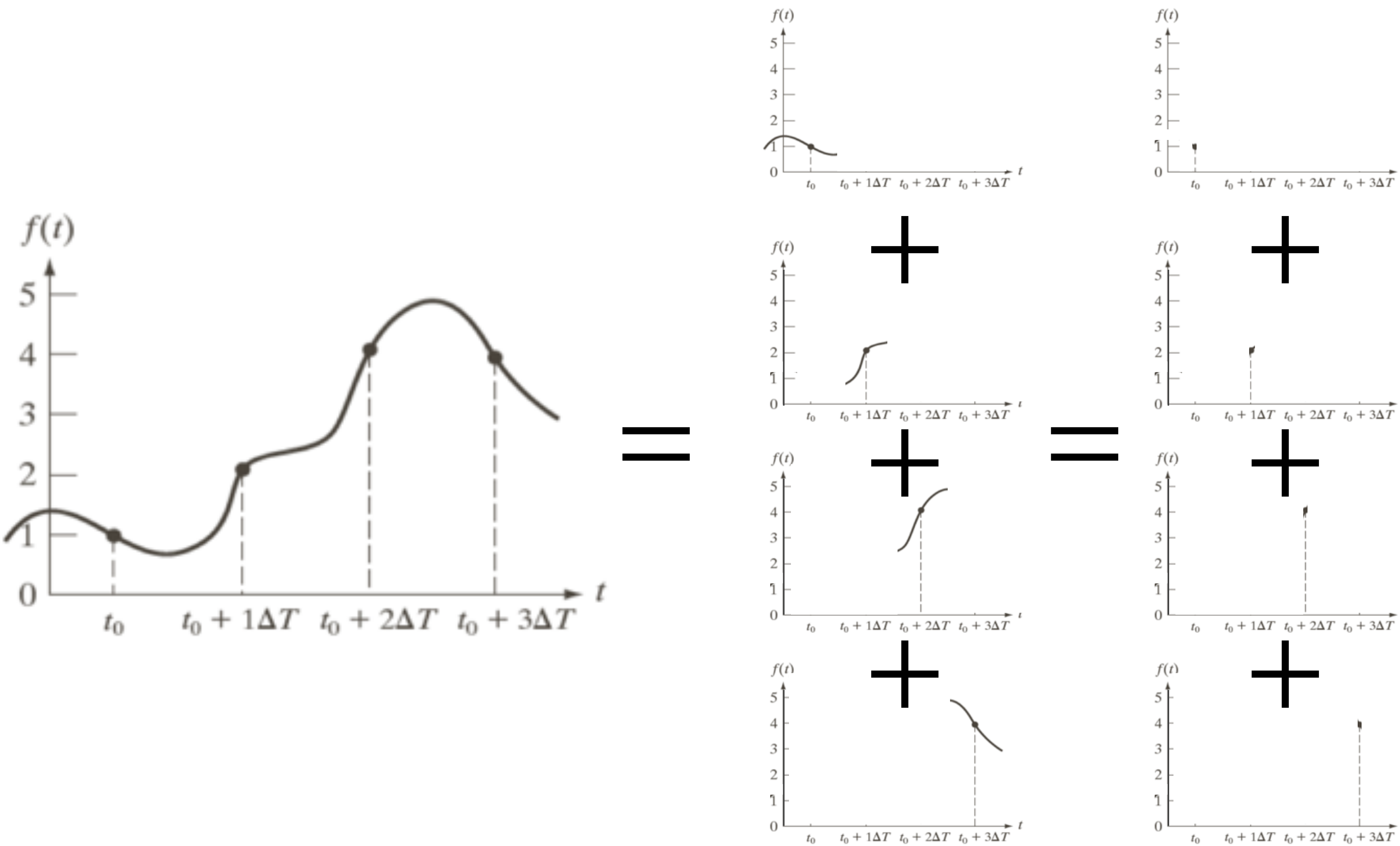
$$f(t) \times s_{\Delta T}(t) =$$

$$\sum_{-\infty}^{\infty} f(n\Delta T)\delta(t - n\Delta T)$$

$$\sum_{-\infty}^{\infty} f[n]\delta(t - n\Delta T)$$

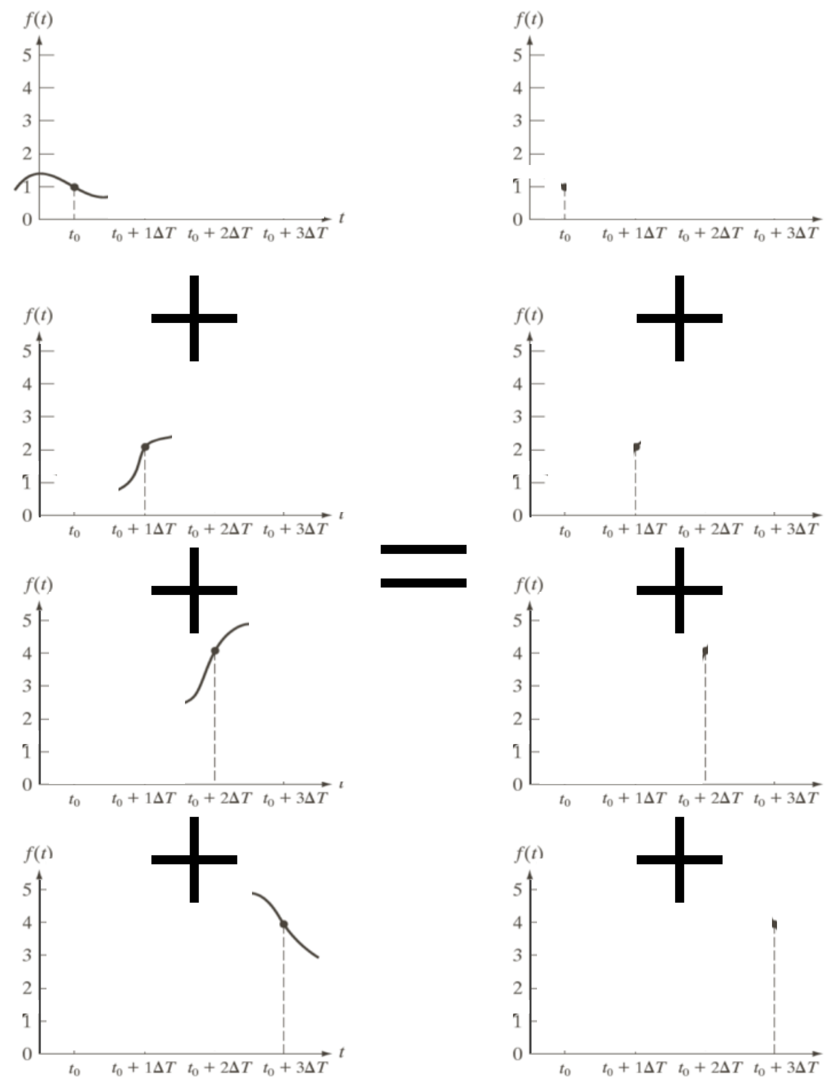


Sampling: step by step



Sampling: step by step

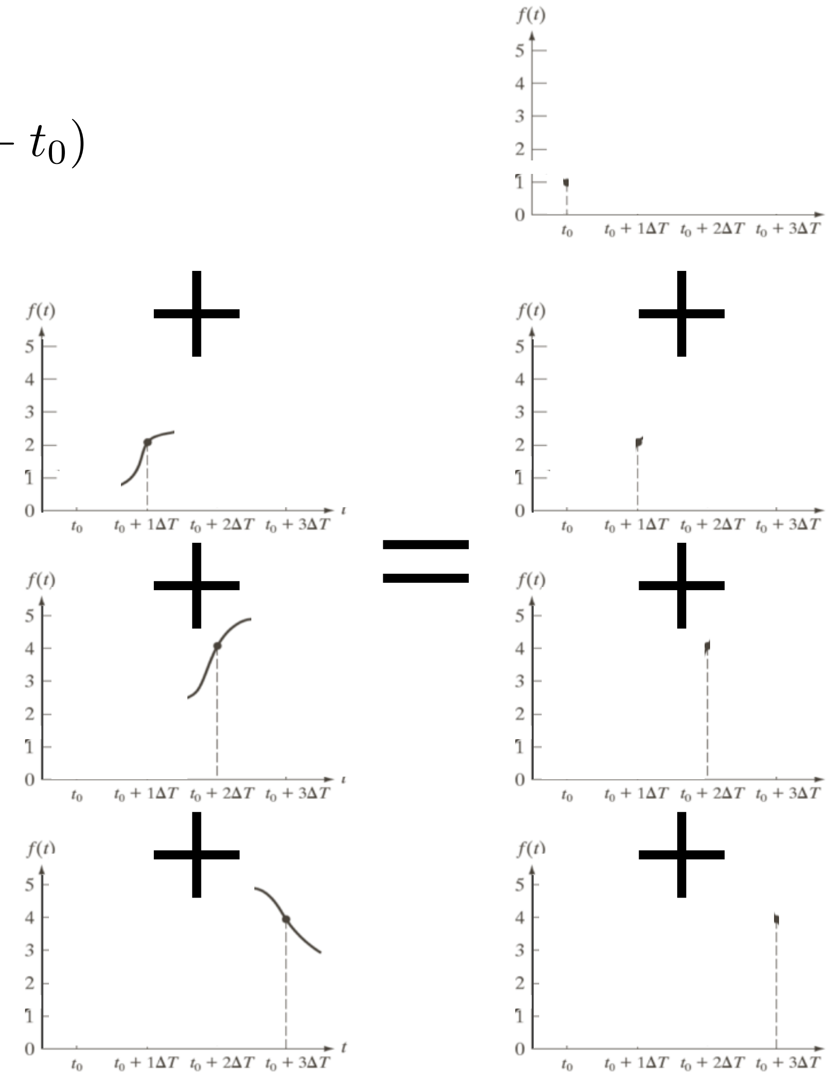
$$f(t) =$$



Sampling: step by step

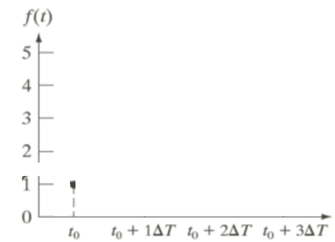
$$\left(\int_{-\infty}^{\infty} f(t)\delta(t - t_0)dt \right) \delta(t - t_0)$$

$$f(t) =$$



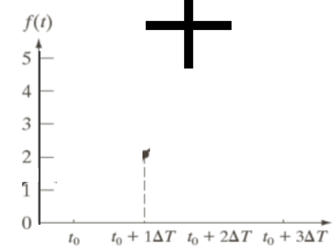
Sampling: step by step

$$\left(\int_{-\infty}^{\infty} f(t)\delta(t-t_0)dt\right)\delta(t-t_0)$$



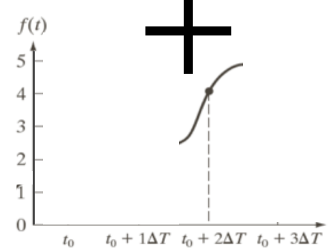
$+$

$$\left(\int_{-\infty}^{\infty} f(t)\delta(t-(t_0+\Delta T))dt\right)\delta(t-(t_0+\Delta T))$$



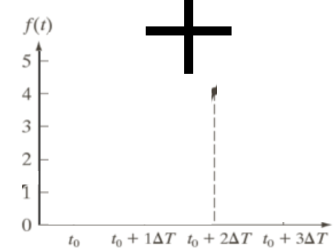
$+$

$$f(t) =$$

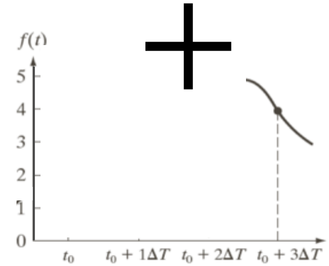


$+$

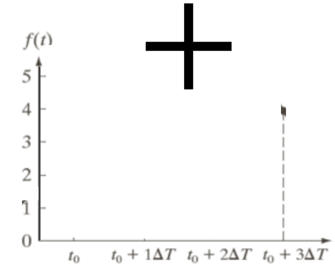
$=$



$+$



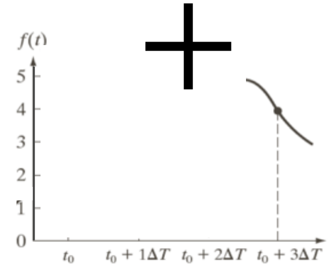
$+$



$+$

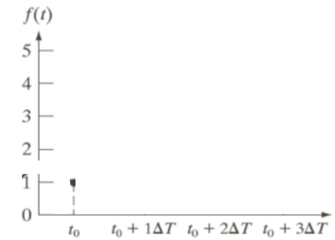
Sampling: step by step

$$f(t) = \left(\int_{-\infty}^{\infty} f(t)\delta(t-t_0)dt\right)\delta(t-t_0) + \left(\int_{-\infty}^{\infty} f(t)\delta(t-(t_0+\Delta T))dt\right)\delta(t-(t_0+\Delta T)) + \left(\int_{-\infty}^{\infty} f(t)\delta(t-(t_0+2\Delta T))dt\right)\delta(t-(t_0+2\Delta T)) +$$

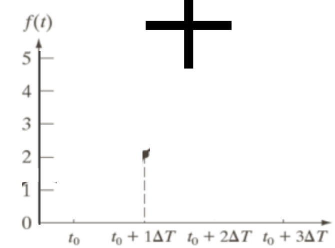


$+$

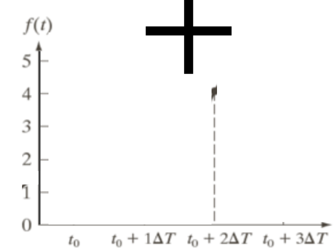
$=$



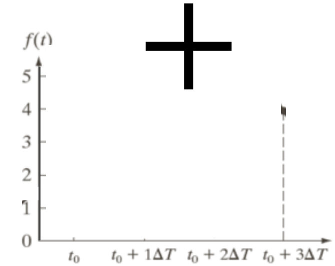
$+$



$+$



$+$

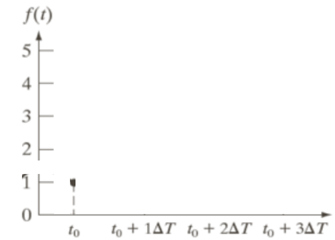


$+$

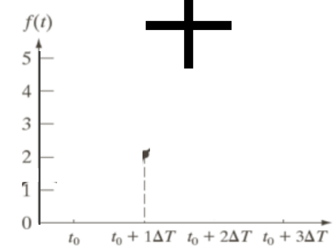
Sampling: step by step

$$
\begin{aligned}
f(t) = \; & \left(\int_{-\infty}^{\infty} f(t)\delta(t-t_0)dt\right)\delta(t-t_0) \\
+ \; & \left(\int_{-\infty}^{\infty} f(t)\delta(t-(t_0+\Delta T))dt\right)\delta(t-(t_0+\Delta T)) \\
+ \; & \left(\int_{-\infty}^{\infty} f(t)\delta(t-(t_0+2\Delta T))dt\right)\delta(t-(t_0+2\Delta T)) \\
+ \; & \left(\int_{-\infty}^{\infty} f(t)\delta(t-(t_0+3\Delta T))dt\right)\delta(t-(t_0+3\Delta T))
\end{aligned}
$$

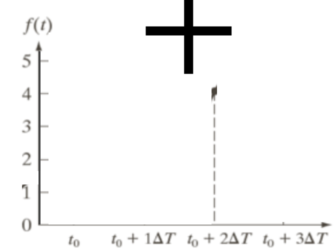
=



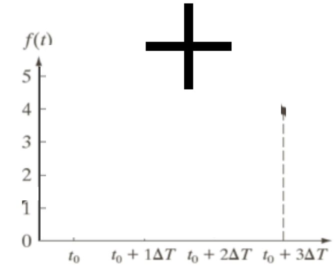
+



+



+



Sampling: step by step

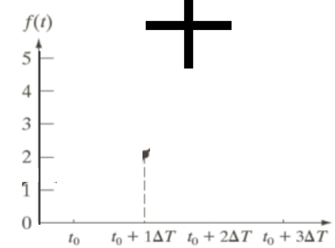
$$f(t) = \left(\int_{-\infty}^{\infty} f(t)\delta(t-t_0)dt\right)\delta(t-t_0)$$

$$+ \left(\int_{-\infty}^{\infty} f(t)\delta(t-(t_0+\Delta T))dt\right)\delta(t-(t_0+\Delta T))$$

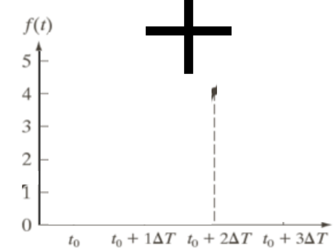
$$+ \left(\int_{-\infty}^{\infty} f(t)\delta(t-(t_0+2\Delta T))dt\right)\delta(t-(t_0+2\Delta T))$$

$$+ \left(\int_{-\infty}^{\infty} f(t)\delta(t-(t_0+3\Delta T))dt\right)\delta(t-(t_0+3\Delta T))$$

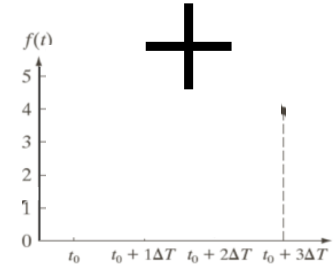
$$= f[0]\delta(t-t_0) +$$



$+$



$+$

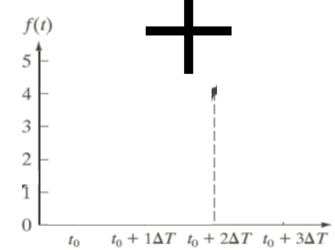


$+$

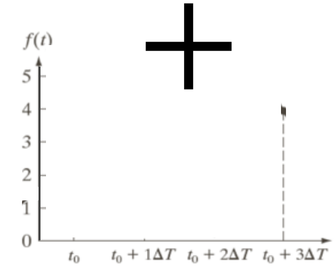
Sampling: step by step

$$
f(t) = \left(\int_{-\infty}^{\infty} f(t)\delta(t-t_0)dt\right)\delta(t-t_0) + \left(\int_{-\infty}^{\infty} f(t)\delta(t-(t_0+\Delta T))dt\right)\delta(t-(t_0+\Delta T)) + \left(\int_{-\infty}^{\infty} f(t)\delta(t-(t_0+2\Delta T))dt\right)\delta(t-(t_0+2\Delta T)) + \left(\int_{-\infty}^{\infty} f(t)\delta(t-(t_0+3\Delta T))dt\right)\delta(t-(t_0+3\Delta T))
$$

$$
= f[0]\delta(t-t_0) + f[1]\delta(t-(t_0+\Delta T)) +
$$

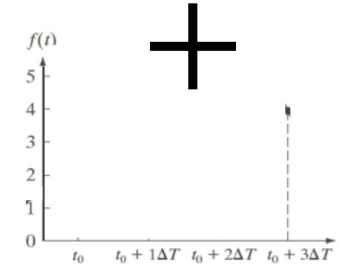


$+$



Sampling: step by step

$$f(t) = \begin{array}{c} \left(\int_{-\infty}^{\infty} f(t)\delta(t-t_0)dt\right)\delta(t-t_0) \\ + \\ \left(\int_{-\infty}^{\infty} f(t)\delta(t-(t_0+\Delta T))dt\right)\delta(t-(t_0+\Delta T)) \\ + \\ \left(\int_{-\infty}^{\infty} f(t)\delta(t-(t_0+2\Delta T))dt\right)\delta(t-(t_0+2\Delta T)) \\ + \\ \left(\int_{-\infty}^{\infty} f(t)\delta(t-(t_0+3\Delta T))dt\right)\delta(t-(t_0+3\Delta T)) \end{array} = \begin{array}{c} f[0]\delta(t-t_0) \\ + \\ f[1]\delta(t-(t_0+\Delta T)) \\ + \\ f[2]\delta(t-(t_0+2\Delta T)) \\ + \end{array}$$



Sampling: step by step

$$
f(t) = \begin{array}{c}
\left(\int_{-\infty}^{\infty} f(t)\delta(t-t_0)dt\right)\delta(t-t_0) \\
+ \\
\left(\int_{-\infty}^{\infty} f(t)\delta(t-(t_0+\Delta T))dt\right)\delta(t-(t_0+\Delta T)) \\
+ \\
\left(\int_{-\infty}^{\infty} f(t)\delta(t-(t_0+2\Delta T))dt\right)\delta(t-(t_0+2\Delta T)) \\
+ \\
\left(\int_{-\infty}^{\infty} f(t)\delta(t-(t_0+3\Delta T))dt\right)\delta(t-(t_0+3\Delta T))
\end{array}
= \begin{array}{c}
f[0]\delta(t-t_0) \\
+ \\
f[1]\delta(t-(t_0+\Delta T)) \\
+ \\
f[2]\delta(t-(t_0+2\Delta T)) \\
+ \\
f[3]\delta(t-(t_0+3\Delta T))
\end{array}
$$

A more general view on sampling

- in summary:

$$f(t) = \sum_{n=-\infty}^{\infty} \left(\int_{-\infty}^{\infty} f(t)\delta(t - n\Delta T)dt \right) \delta(t - n\Delta T)$$

$$f(t) = \sum_{n=-\infty}^{\infty} f[n]\delta(t - n\Delta T)$$

- more generally:

$$f(t) = \sum_{n=-\infty}^{\infty} \left(\int_{-\infty}^{\infty} f(t)\psi(t - n\Delta T)dt \right) \delta(t - n\Delta T)$$

$$f(t) = \sum_{n=-\infty}^{\infty} c[n]\delta(t - n\Delta T)$$

A more general view on sampling

- in summary:

$$f(t) = \sum_{n=-\infty}^{\infty} \left(\int_{-\infty}^{\infty} f(t)\delta(t - n\Delta T)dt \right) \delta(t - n\Delta T)$$

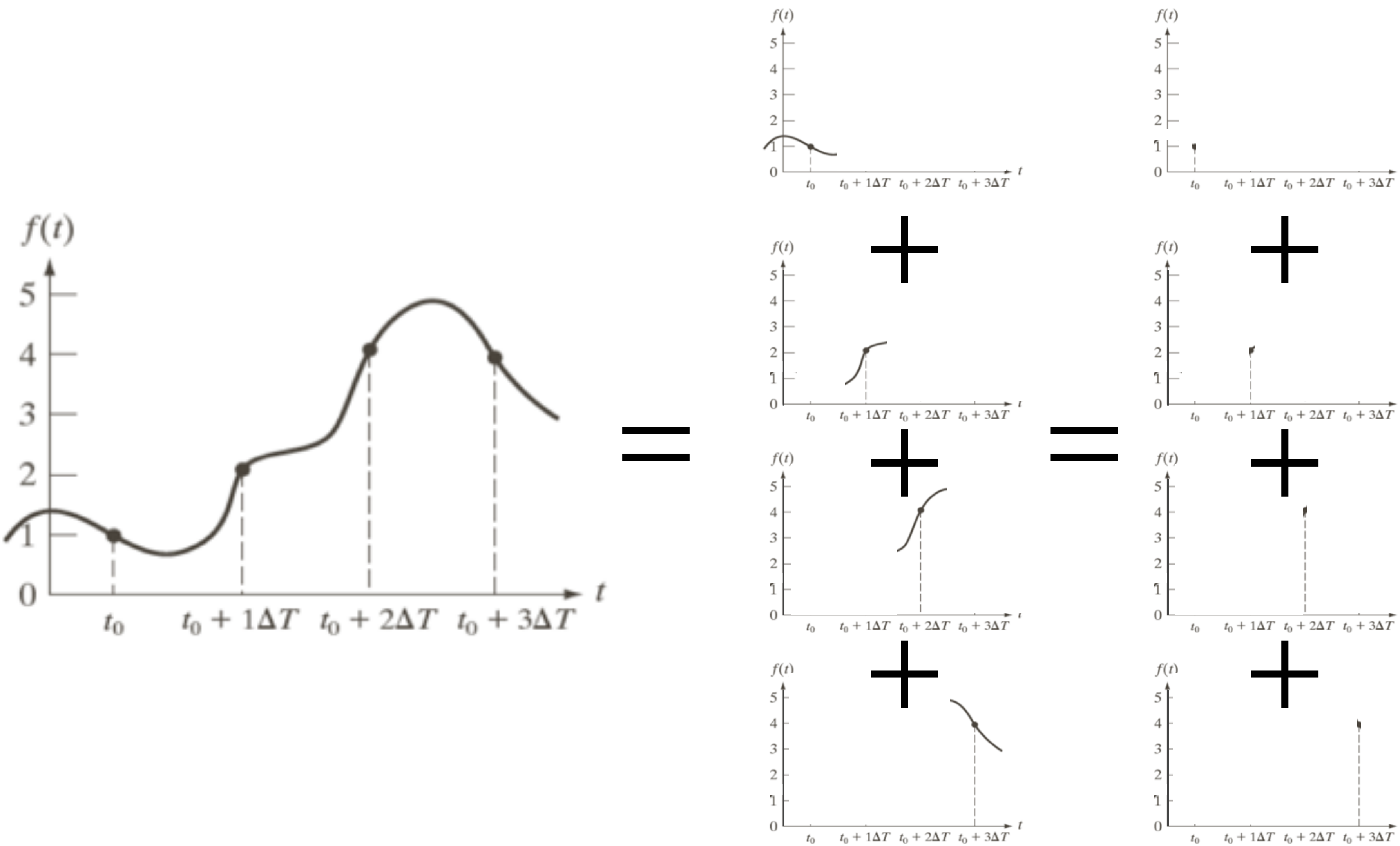
$$f(t) = \sum_{n=-\infty}^{\infty} f[n]\delta(t - n\Delta T)$$

- even more general:

$$f(t) = \sum_{n=-\infty}^{\infty} \left(\int_{-\infty}^{\infty} f(t)\psi(t - n\Delta T)dt \right) \phi(t - n\Delta T)$$

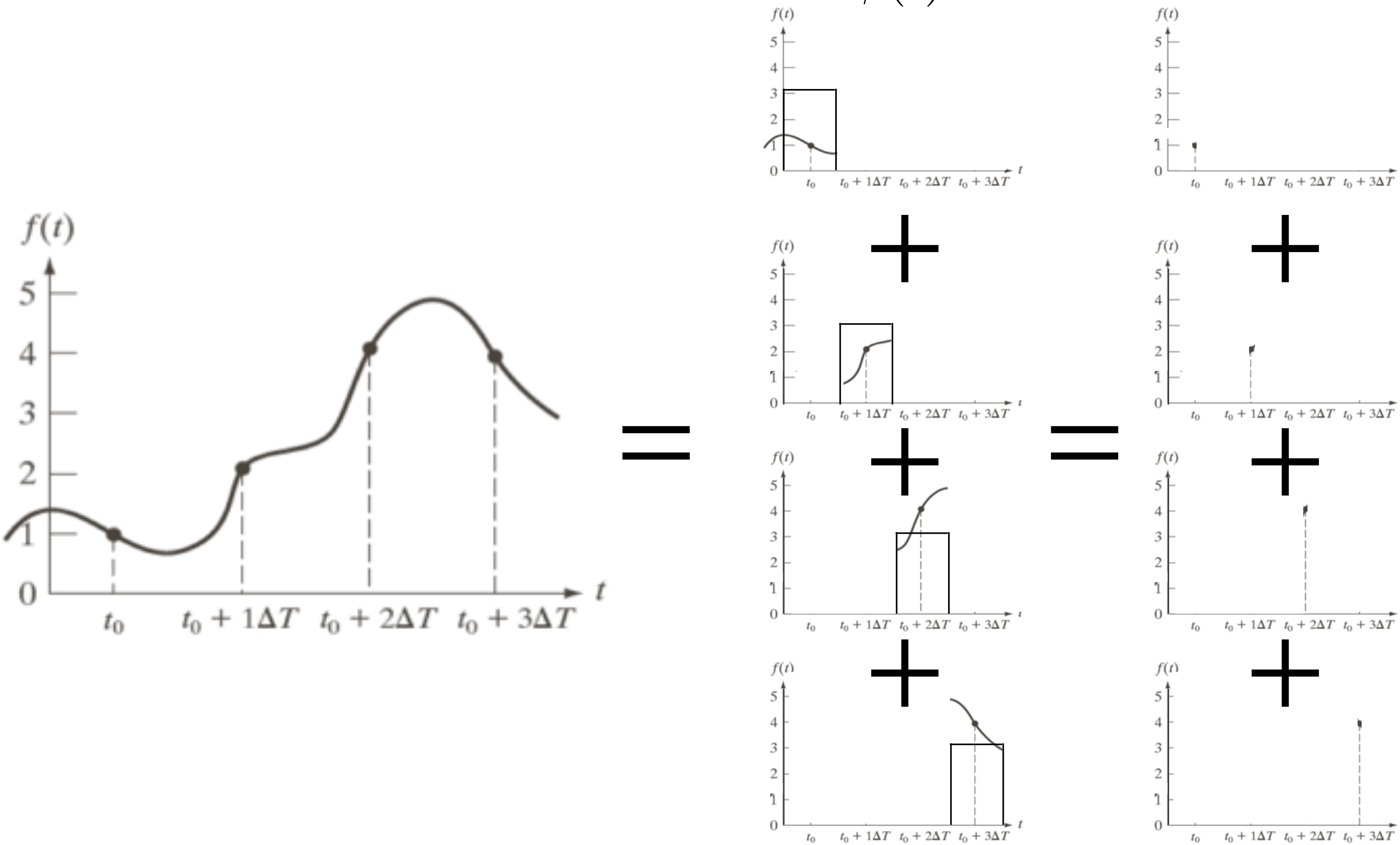
$$f(t) = \sum_{n=-\infty}^{\infty} c[n]\phi(t - n\Delta T)$$

Sampling: generalization I

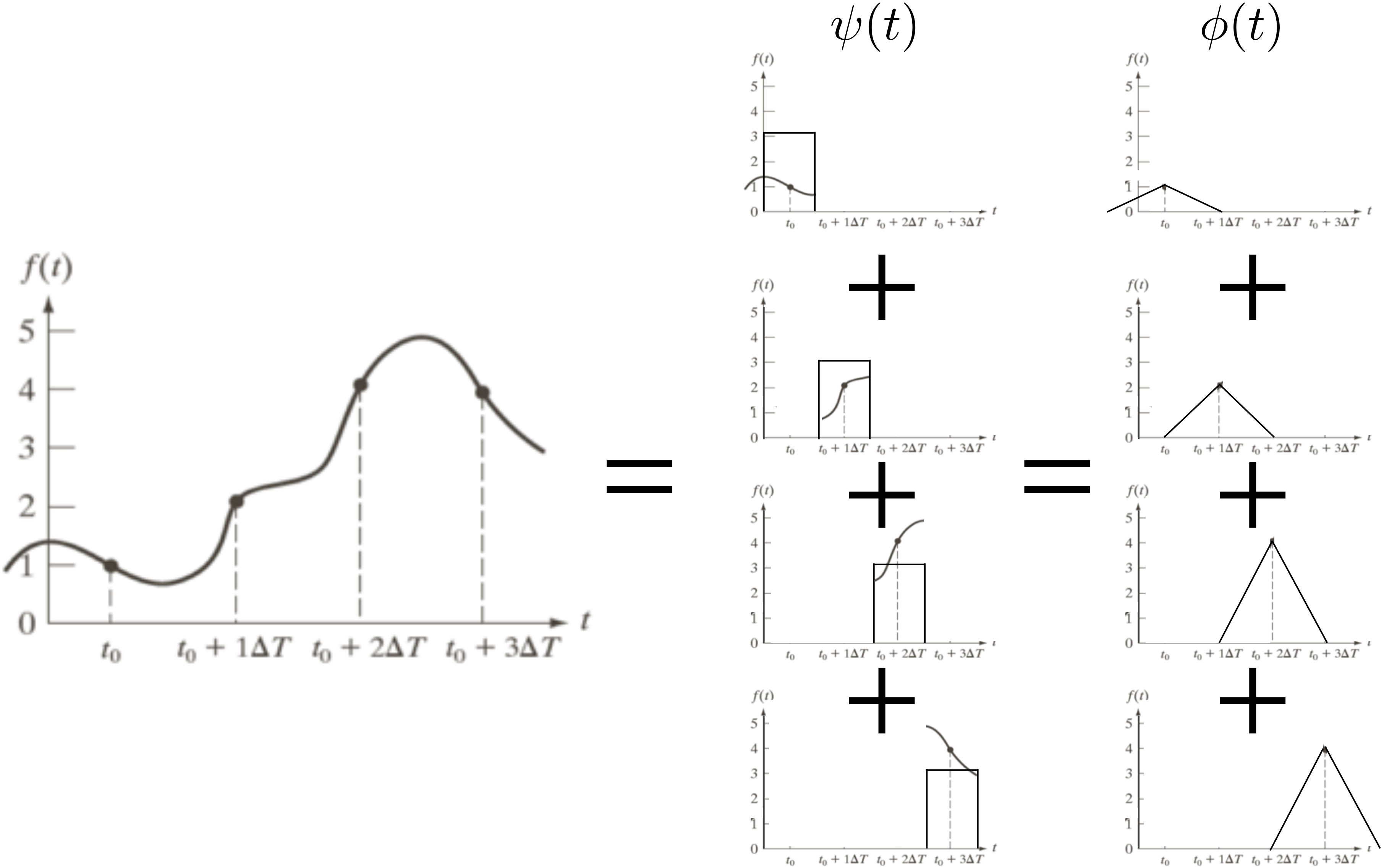


Sampling: generalization I

$\psi(t)$



Sampling: generalization II



A more general view on sampling

- in summary:

$$f(t) \approx \sum_{n=-\infty}^{\infty} \left(\int_{-\infty}^{\infty} f(t)\psi(t - n\Delta T)dt \right) \phi(t - n\Delta T)$$

$$f(t) \approx \sum_{n=-\infty}^{\infty} c[n]\phi(t - n\Delta T)$$

- **y** also known as a “point-spread function”; common for image acquisition
- **f** also known as the reconstruction function

Function spaces

- in summary:

$$f(t) \approx \sum_{n=-\infty}^{\infty} \left(\int_{-\infty}^{\infty} f(t)\psi(t - n\Delta T)dt \right) \phi(t - n\Delta T)$$

$$f(t) \approx \sum_{n=-\infty}^{\infty} c[n]\phi(t - n\Delta T)$$

- Perhaps the most general we will get (reverting to the notation in the book):

$$f(x) = \sum_n \alpha_n \phi_n(x)$$

$$\alpha_n = \langle \tilde{\phi}_n(x), f(x) \rangle = \int \tilde{\phi}_n^*(x) f(x) dx$$

Function spaces

- in summary:

$$f(t) \approx \sum_{n=-\infty}^{\infty} \left(\int_{-\infty}^{\infty} f(t) \psi(t - n\Delta T) dt \right) \phi(t - n\Delta T)$$

$$f(t) \approx \sum_{n=-\infty}^{\infty} c[n] \phi(t - n\Delta T)$$

- Perhaps the most general we will get (reverting to the notation in the book):

$$f(x) = \sum_n \alpha_n \phi_n(x)$$

$$\alpha_n = \langle \tilde{\phi}_n(x), f(x) \rangle = \int \tilde{\phi}_n^{\,*}(x) f(x) dx$$

Function spaces

- Given an expansion with expansion set $\{\phi_n(x)\}$: $$f(x) = \sum_n \alpha_n \phi_n(x)$$
- If the expansion is unique, then $\{\phi_n(x)\}$ is a basis spanning a function space V:

$$V = \overline{\mathrm{Span}\{\phi_n(x)\}}$$

Ortonormal Basis

- if $\{\phi_n(x)\}$ are orthonormal, i.e.

$$\langle \phi_j, \phi_k \rangle = \delta_{jk} = \begin{cases} 0 & j \neq k \\ 1 & j = k \end{cases}$$

- then the dual basis is equal to the actual basis:

$$\phi_k = \tilde{\phi}_k$$

- and hence: $\alpha_n = \langle \phi_n, f \rangle$

Bi-orthogonal basis

- For a bi-orthogonal basis, we have:

$$\langle \phi_j , \widetilde{\phi_k} \rangle = \delta_{jk} = \begin{cases} 0 & j \neq k \\ 1 & j = k \end{cases}$$

- and hence, we get:

$$\alpha_n = \langle \widetilde{\phi_n} , f \rangle$$

(tight) frames

- the expansion is NOT a basis (overcomplete), but they form a frame:

$$A||f(x)||^2 \leq \sum_k |\langle \phi_k(x), f(x) \rangle|^2 \leq B||f(x)||^2$$

- tight, when A=B, and then we have:

$$f(x) = \frac{1}{A} \sum_k \langle \phi_k(x), f(x) \rangle \phi_k(x)$$

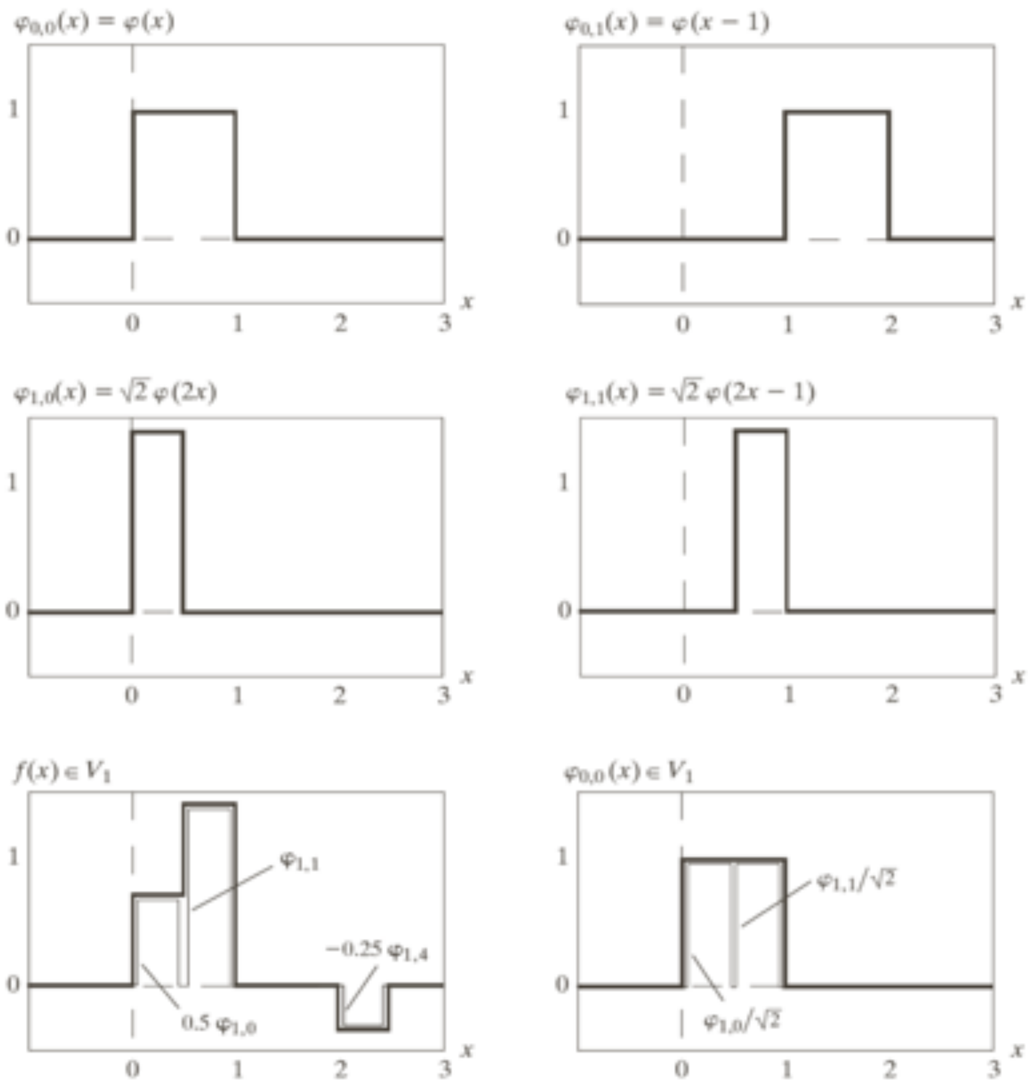
Special MR spaces! Scaling functions

- we would like to create MR spaces with nice properties, what basis functions should they have?
- key idea: self-similar functions:

$$\phi_{j,k}(x) = 2^{j/2}\phi(2^j x - k)$$

- a simple example

Haar scaling functions



Scaling functions

- key idea: self-similar functions:

$$\phi_{j,k}(x) = 2^{j/2}\phi(2^j x - k)$$

- k is just an offset, but j determines scale (resolution, accuracy)
- hence, looking at the spaces they span:

$$V_j = \overline{\mathrm{Span}\{\phi_{j,k}(x)\}}$$

- we really want them to be Matryoshkas:

$$V_{-\infty} \subset \cdots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots V_\infty$$

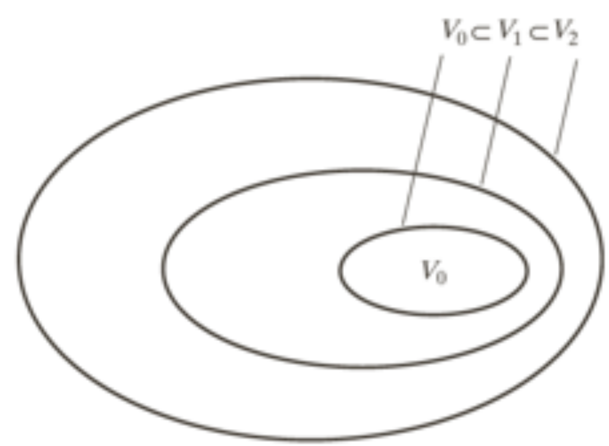
MR Scaling functions

- key idea: self-similar functions:

$$\phi_{j,k}(x) = 2^{j/2}\phi(2^j x - k)$$

$$V_{-\infty} \in \cdots \in V_{-1} \in V_0 \in V_1 \in V_2 \in \cdots V_\infty$$

- with $V_{-\infty} = \{0\}$
- and $V_\infty = \{L^2(\mathcal{R})\}$



MR Scaling functions

- to make it all work, we need 4 conditions
  1. The scaling function is orthogonal to its integer translations.
  2. The subspaces spanned by the scaling function at low scales are nested within those spanned at higher scales.
  3. The only function that is common to all Vj is f(x) = 0.
  4. Any function can be represented with arbitrary precision.

Refinement equation

- with this, we have:

$$\phi_{j,n}(x) = \sum_n \alpha_n \phi_{j+1,n}(x)$$

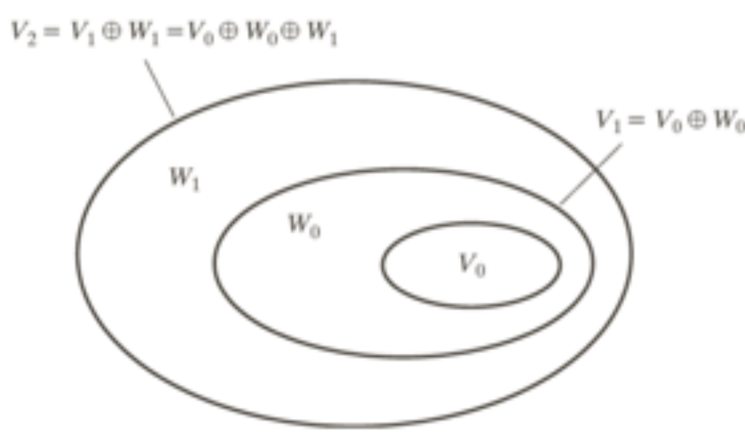
$$\phi_{j,n}(x) = \sum_n h_\phi(n) 2^{(j+1)/2} \phi(2^{(j+1)} x - n)$$

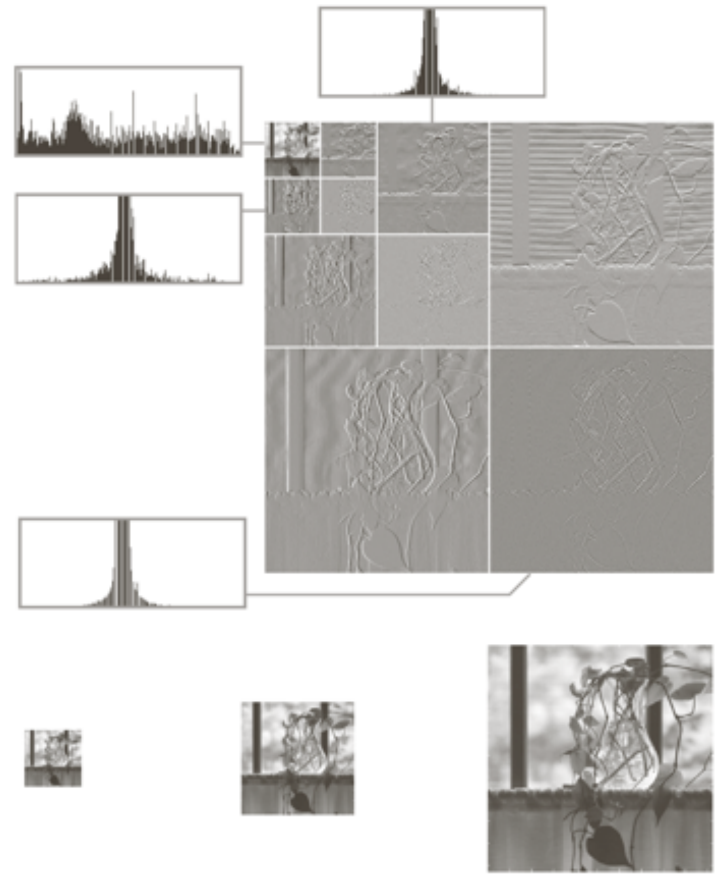
- which we call the refinement/MRA/ dilation equation:

$$\phi(x) = \sum_n h_\phi(n) \sqrt{2} \phi(2x - n)$$

Wavelet space

- there is a space inbetween the resolutions! remember:
- hence:





Wavelet space

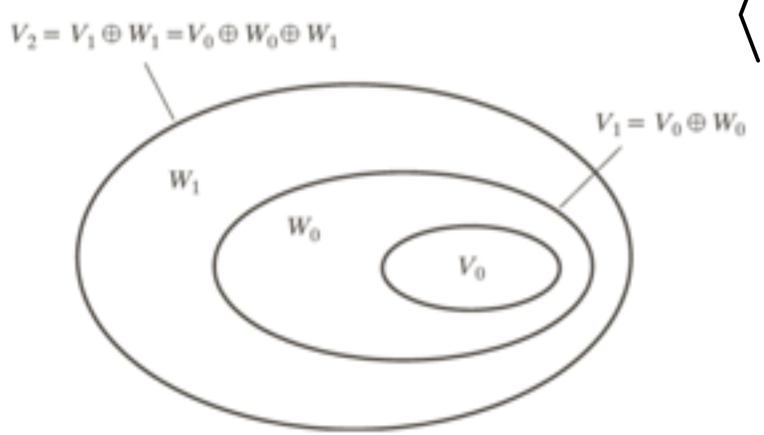
- We call these spaces W_j with basis **y**:

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

$$W_j = \overline{\text{Span}\{\psi_{j,k}(x)\}}$$

- orthogonality between the spaces:

$$\langle \phi_{j,k}(x), \psi_{j,l}(x) \rangle = 0$$



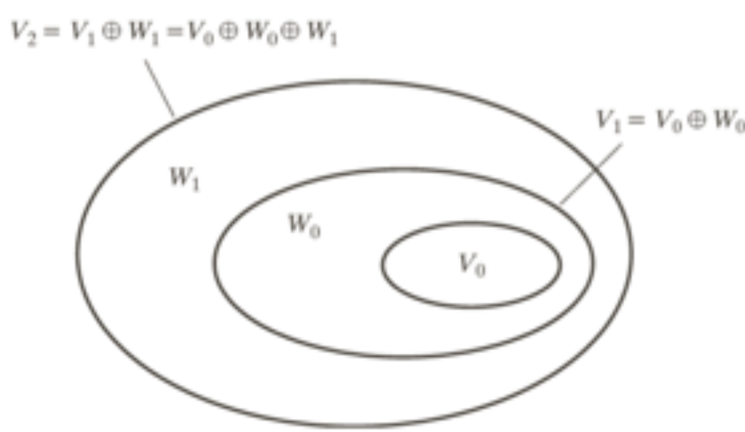
Wavelet space

- Conceptually:

$$V_{-\infty} \subset \cdots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots V_\infty$$

- which means:

$$L^2(\mathcal{R}) = V_\infty = V_0 \oplus W_0 \oplus W_1 \oplus \cdots$$



Wavelet spaces

- similar to the dilation equation, we have:

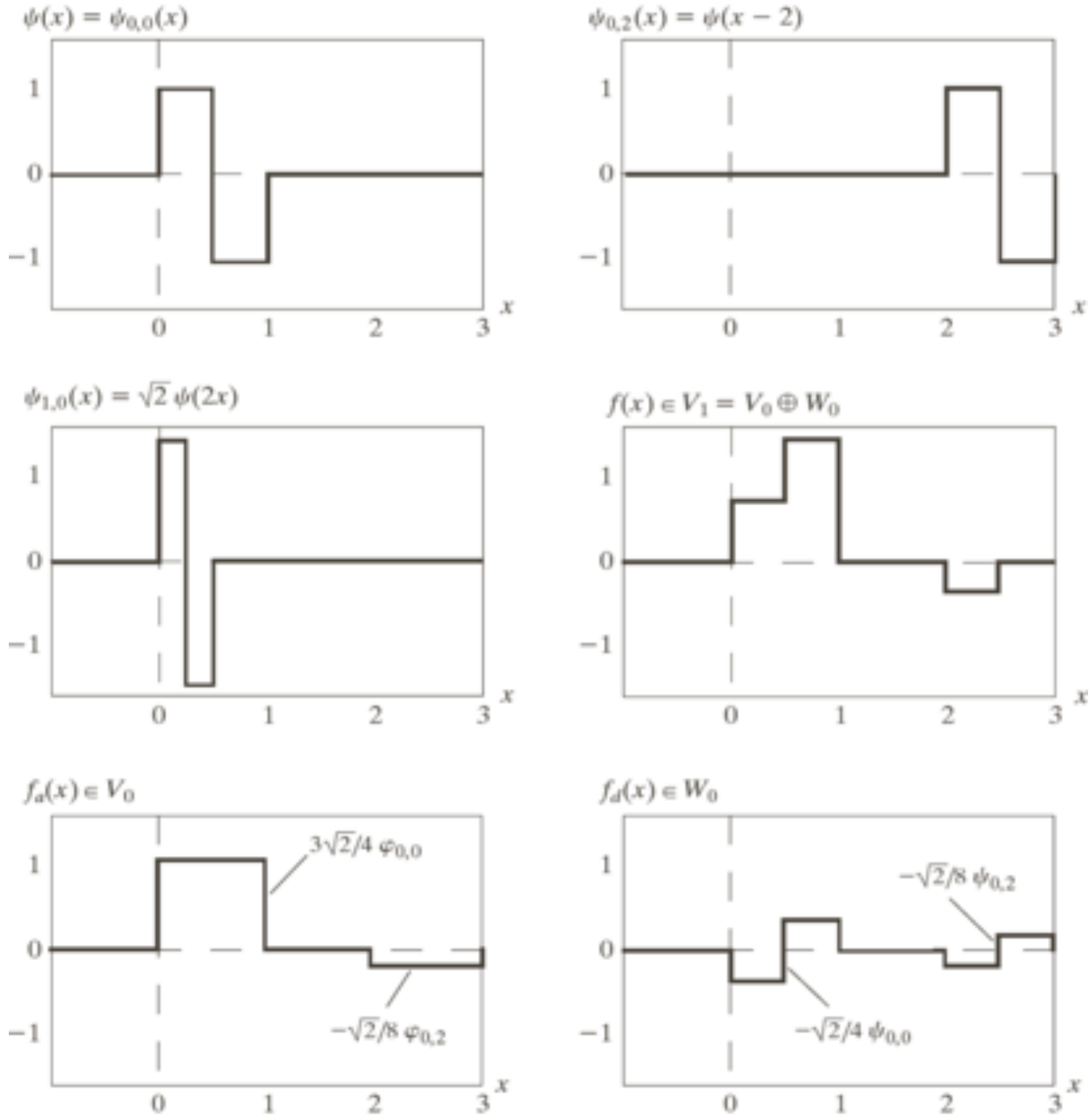
$$\psi(x) = \sum_n h_\psi(n)\sqrt{2}\phi(2x-n)$$

- if integer wavelet translates are orthogonal, it can be shown that:

$$h_\psi(n) = (-1)^n h_\phi(1-n)$$

- Let's look at another example

Wavelet — Example



Overview

- Background
- Multiresolution expansions
  – understanding spaces
  – scaling functions
  – wavelet functions
- Wavelet transform in 1D
- Fast wavelet transform
- Wavelet transform in 2D
- Wavelet packets

Overview

- Background
- Multiresolution expansions
- Wavelet transform in 1D
  - wavelet series expansion
  - discrete wavelet transform
  - continuous wavelet transform
- Fast wavelet transform
- Wavelet transform in 2D
- Wavelet packets

Wavelet series expansion

- starting with
  $$L^2(\mathcal{R}) = V_\infty = V_0 \oplus W_0 \oplus W_1 \oplus \cdots$$
- actual formula:

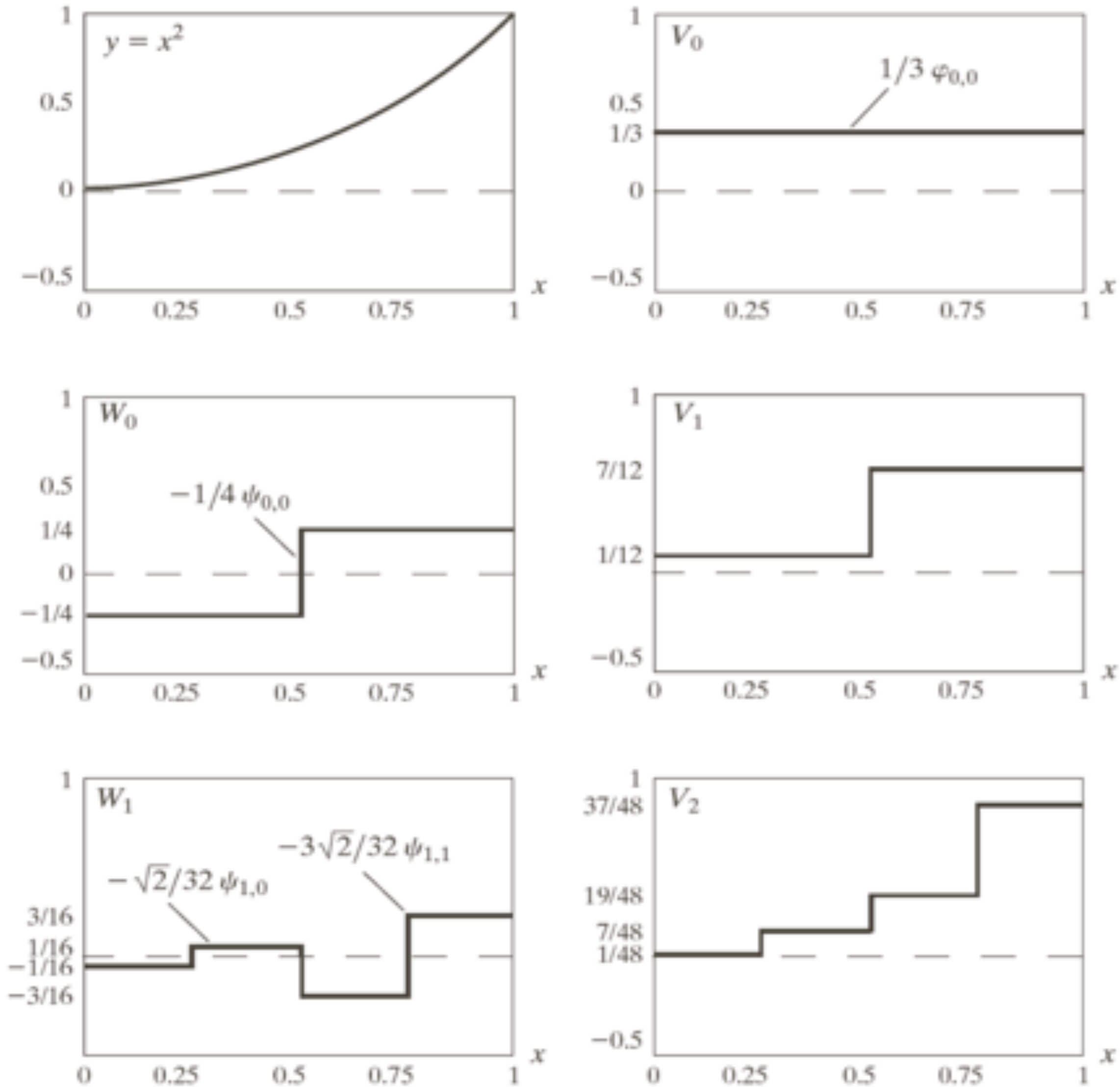
$$f(x) = \sum_k c_0(k)\phi_{0,k}(x) + \sum_{j=0}^{\infty}\sum_k d_j(k)\psi_{j,k}(x)$$

- where

$$c_0(k) = \langle f(x), \phi_{0,k}(x)\rangle = \int f(x)\phi_{0,k}(x)dx$$

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

Example



$y = x^2$
V_0
$1/3\ \varphi_{0,0}$
W_0
$-1/4\ \psi_{0,0}$
V_1
7/12
1/12
W_1
$-\sqrt{2}/32\ \psi_{1,0}$
$-3\sqrt{2}/32\ \psi_{1,1}$
3/16
1/16
−1/16
−3/16
V_2
37/48
19/48
7/48
1/48
1
0.5
1/3
1/4
0
−1/4
−0.5
0
0.25
0.5
0.75
1
x

Discrete wavelet transform

- assuming (n=0, 1, 2, ..., M-1):

$$f(n) = f(x_0 + n\Delta x)$$

- we have

$$f(n) = \frac{1}{\sqrt{M}} \sum_k c_0(k)\phi_{0,k}(n) + \frac{1}{\sqrt{M}} \sum_{j=0}^{\infty} \sum_k d_j(k)\psi_{j,k}(n)$$

- where:

$$c_0(k) = \frac{1}{\sqrt{M}} \sum_n f(n)\phi_{0,k}(n)$$

$$d_j(n) = \frac{1}{\sqrt{M}} \sum_n f(n)\psi_{j,k}(n)$$

Continuous wavelet transform

- a little more complicated:

$$f(x) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty W_\psi(s,\tau) \frac{\psi_{s,\tau}(x)}{s^2} d\tau ds$$

- where

$$C_\psi = \int_{-\infty}^\infty \frac{|\Psi(\mu)|^2}{|\mu|} d\mu \qquad W_\psi(s,\tau) = \int_{-\infty}^\infty f(x) \psi_{s,\tau}(x) dx$$

- with $\Psi(\mu)$ being the Fourier transf. of $\psi(x)$

Overview

- Background
- Multiresolution expansions
- Wavelet transform in 1D
  - wavelet series expansion
  - discrete wavelet transform
  - continuous wavelet transform
- Fast wavelet transform
- Wavelet transform in 2D
- Wavelet packets

The FWT

- computing the wavelet transform seemed hard:

$$c_0(k) = \langle f(x), \phi_{0,k}(x) \rangle = \int f(x)\phi_{0,k}(x)dx$$

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

$$c_0(k) = \frac{1}{\sqrt{M}} \sum_n f(n)\phi_{0,k}(n)$$

$$d_j(n) = \frac{1}{\sqrt{M}} \sum_n f(n)\psi_{j,k}(n)$$

FWT

- ... but it is really VERY easy, start with:

$$\phi(x) = \sum_n h_\phi(n)\sqrt{2}\phi(2x - n)$$

- and then:

$$\begin{aligned}\phi(2^j x - k) &= \sum_n h_\phi(n)\sqrt{2}\phi(2(2^j x - k) - n) \\ &= \sum_n h_\phi(n)\sqrt{2}\phi(2^{j+1} x - 2k - n) \\ &= \sum_m h_\phi(m - 2k)\sqrt{2}\phi(2^{j+1} x - m)\end{aligned}$$

- similarly: $\psi(2^j x - k) = \sum_m h_\psi(m - 2k)\sqrt{2}\phi(2^{j+1} x - m)$

FWT, derivation cont.

- with $\psi(2^j x - k) = \sum_m h_\psi(m-2k)\sqrt{2}\phi(2^{j+1}x - m)$
- we have

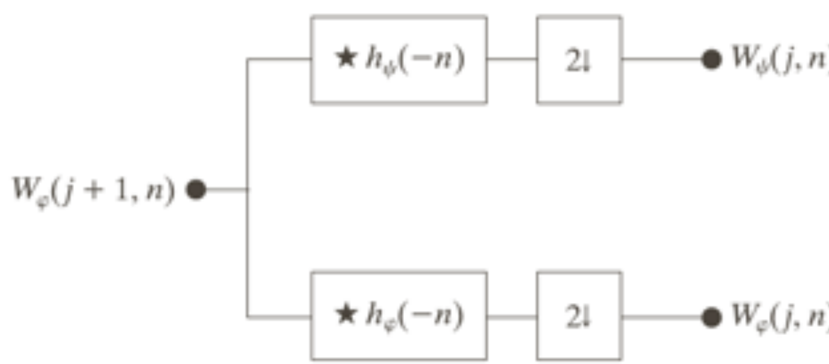
$$\begin{aligned} d_j(k) &= \int f(x) 2^{j/2} \psi(2^j x - k)(x) dx \\ &= \int f(x) 2^{j/2} \left[\sum_m h_\psi(m-2k)\sqrt{2}\phi(2^{j+1}x - m) \right] dx \\ &= \sum_m h_\psi(m-2k) \left[\int f(x) 2^{(j+1)/2} \phi(2^{j+1}x - m) dx \right] \\ &= \sum_m h_\psi(m-2k) c_{j+1}(m) \end{aligned}$$

FWT

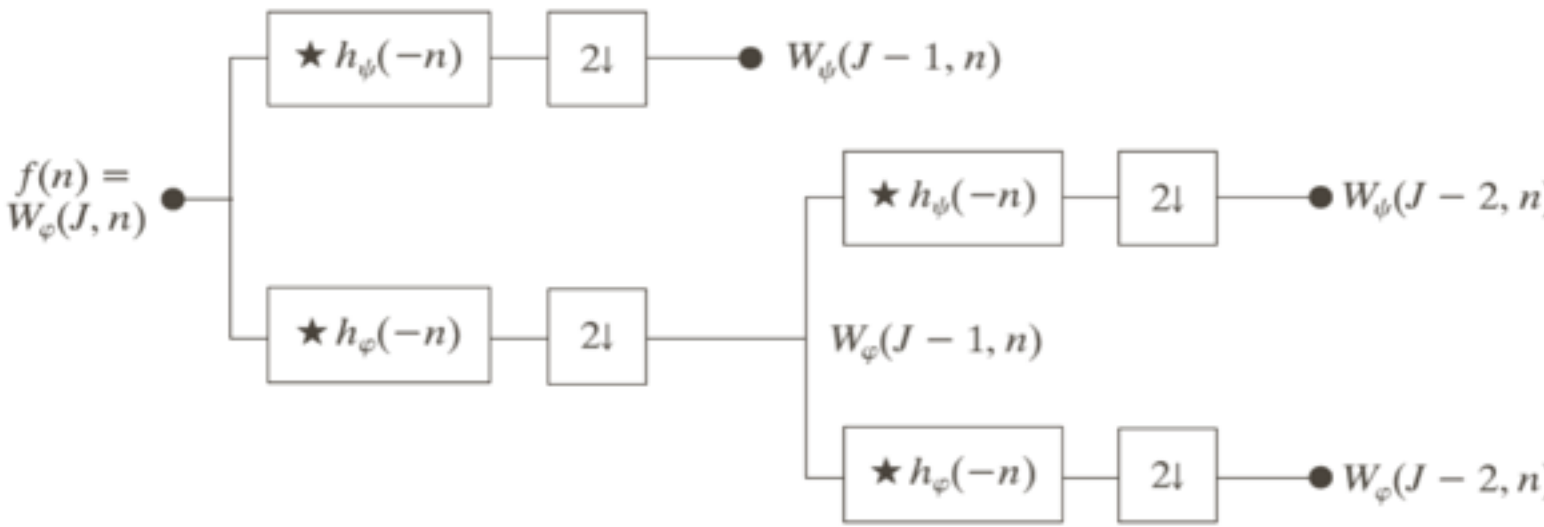
- essentially, we have:

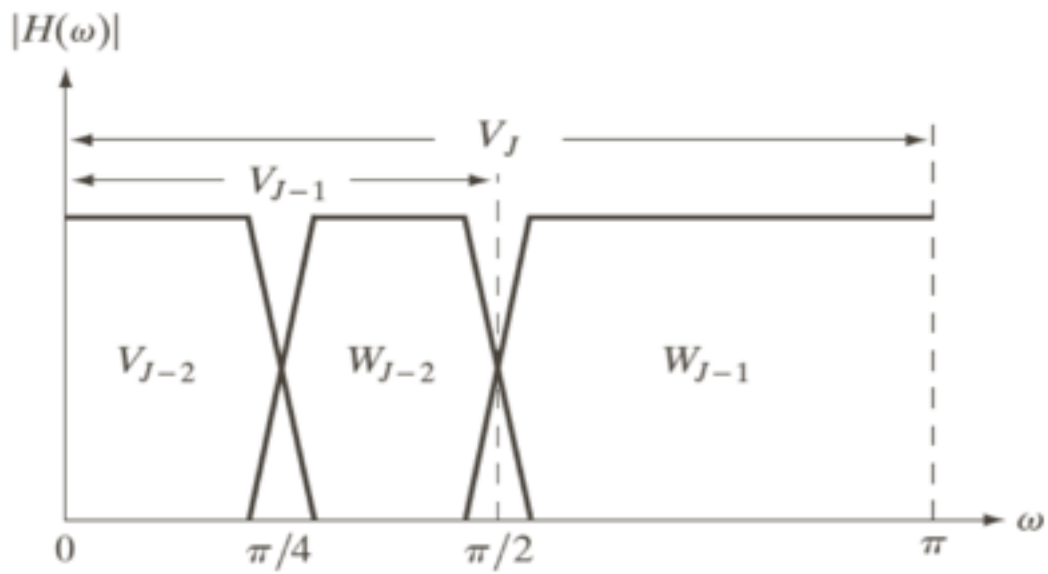
$$d_j(k) = \sum_m h_\psi(m - 2k)c_{j+1}(m)$$

$$c_j(k) = \sum_m h_\phi(m - 2k)c_{j+1}(m)$$

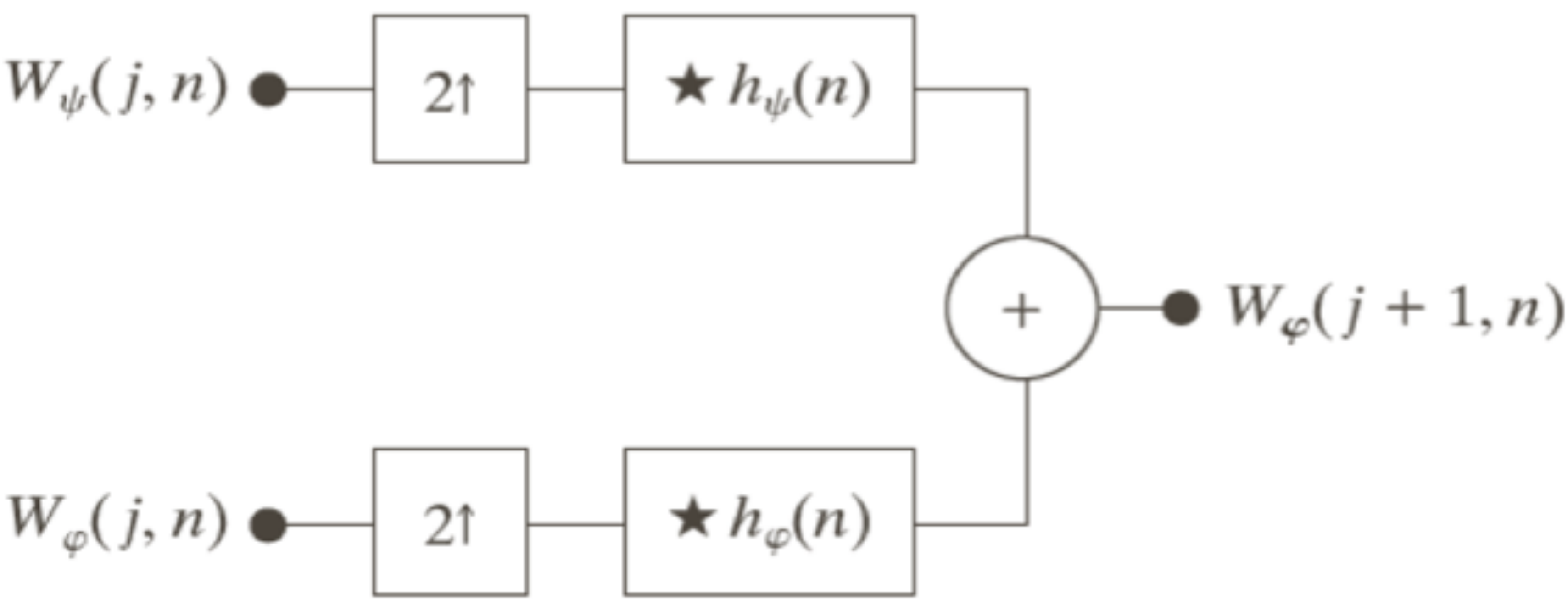


FWT, pictorially

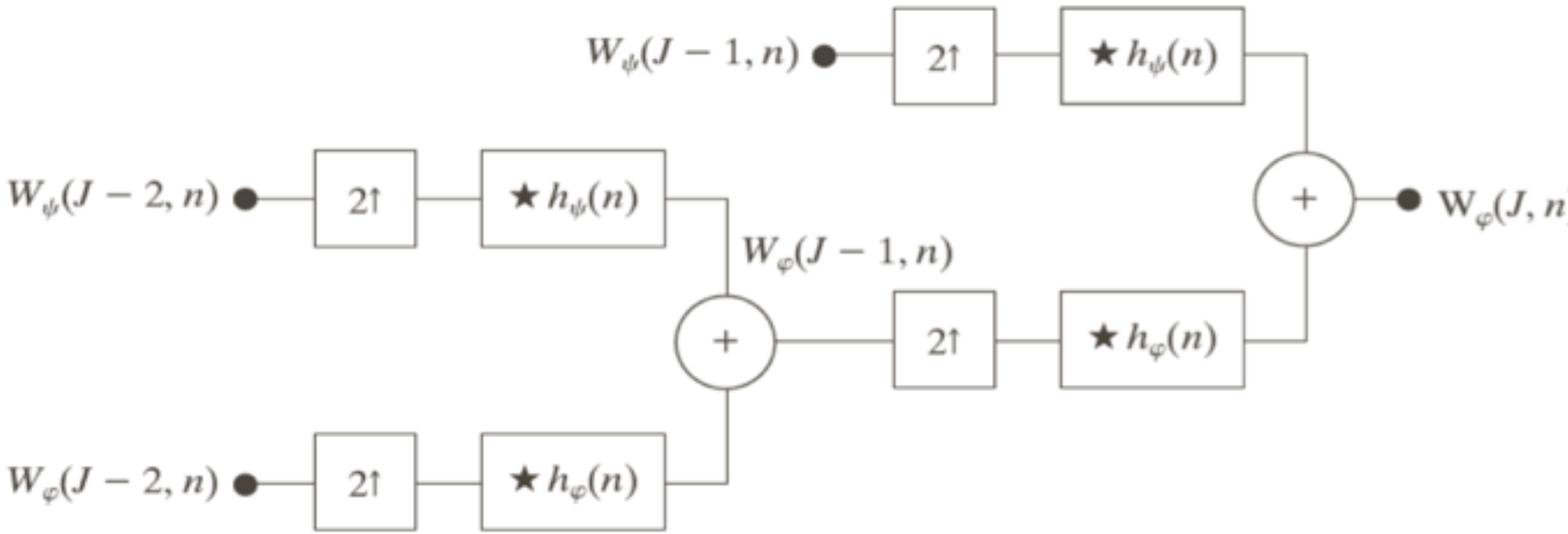




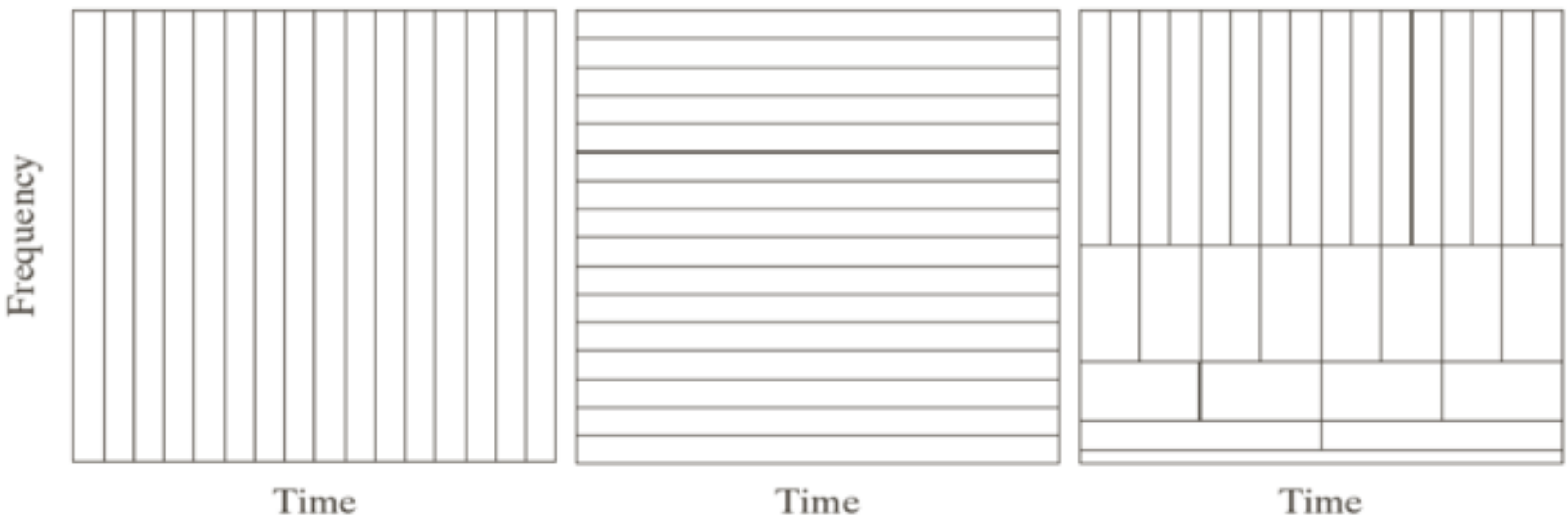
FWT^{-1}, synthesis part



FWT^{-1}, synthesis part



FWT, conceptually



a b c

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.

Overview

- Background
- Multiresolution expansions
- Wavelet transform in 1D
- Fast wavelet transform
- Wavelet transform in 2D
- Wavelet packets

2D (+3D) wavelet transform

- standard appraoch is to use a separable transform:

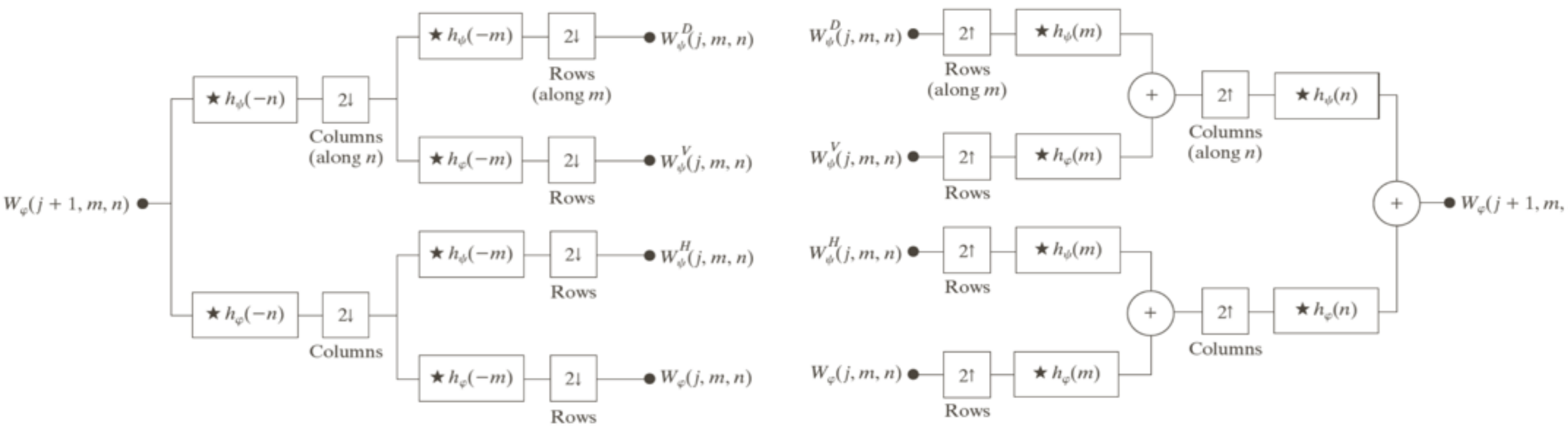
$$\phi(x,y) = \phi(x)\phi(y)$$

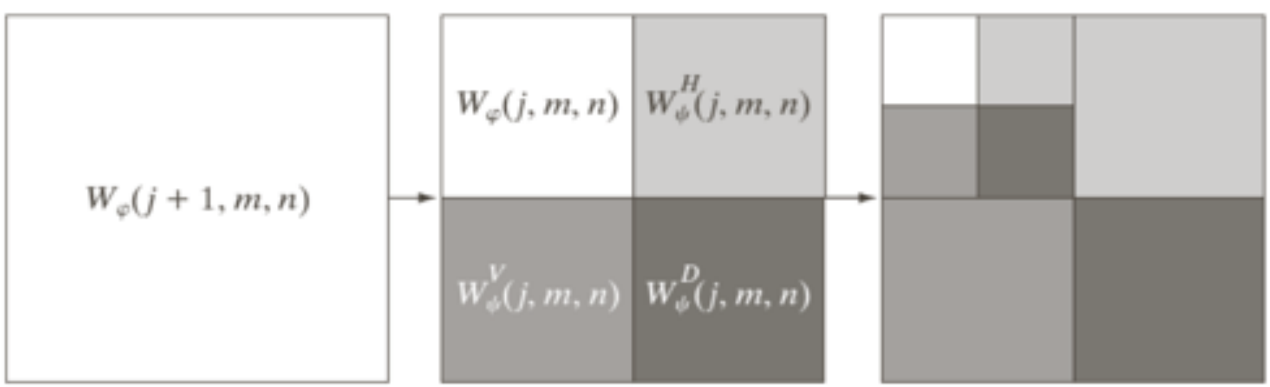
$$\psi_{LH}(x,y) = \phi(x)\psi(y)$$

$$\psi_{HL}(x,y) = \psi(x)\phi(y)$$

$$\psi_{HH}(x,y) = \psi(x)\psi(y)$$

2D wavelet transform





Example

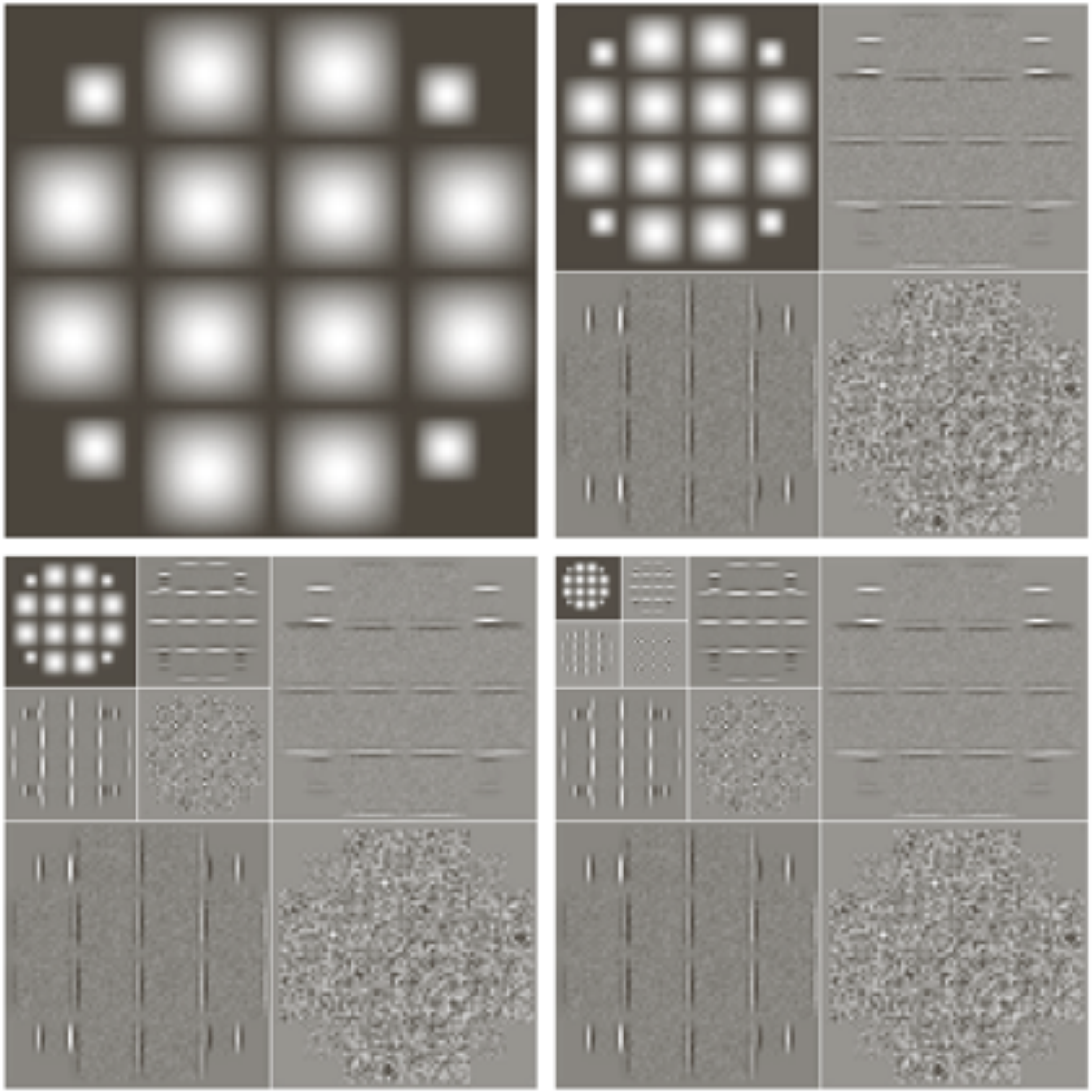
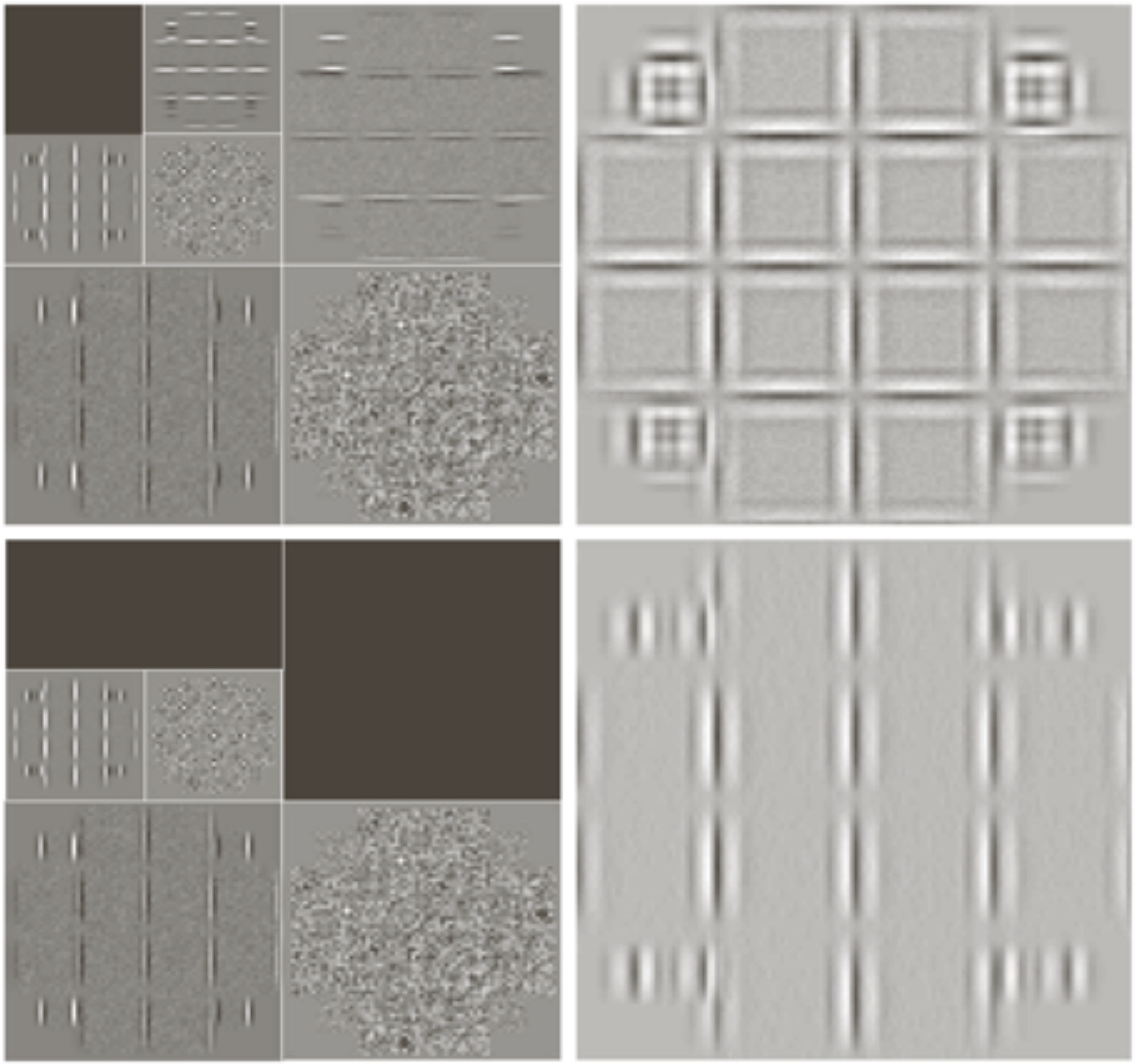


FIGURE 7.25 Computing a 2-D three-scale FWT: (a) the original image; (b) a one-scale FWT; (c) a two-scale FWT; and (d) a three-scale FWT.

How to work with FWT

- Processing similar to Fourier transforms:
  1. Compute the FWT (of an image, signal, volume)
  2. Alter the transform
  3. Compute the inverse transform
- most successful applications include compression, denoising and the like

Example 1



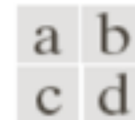
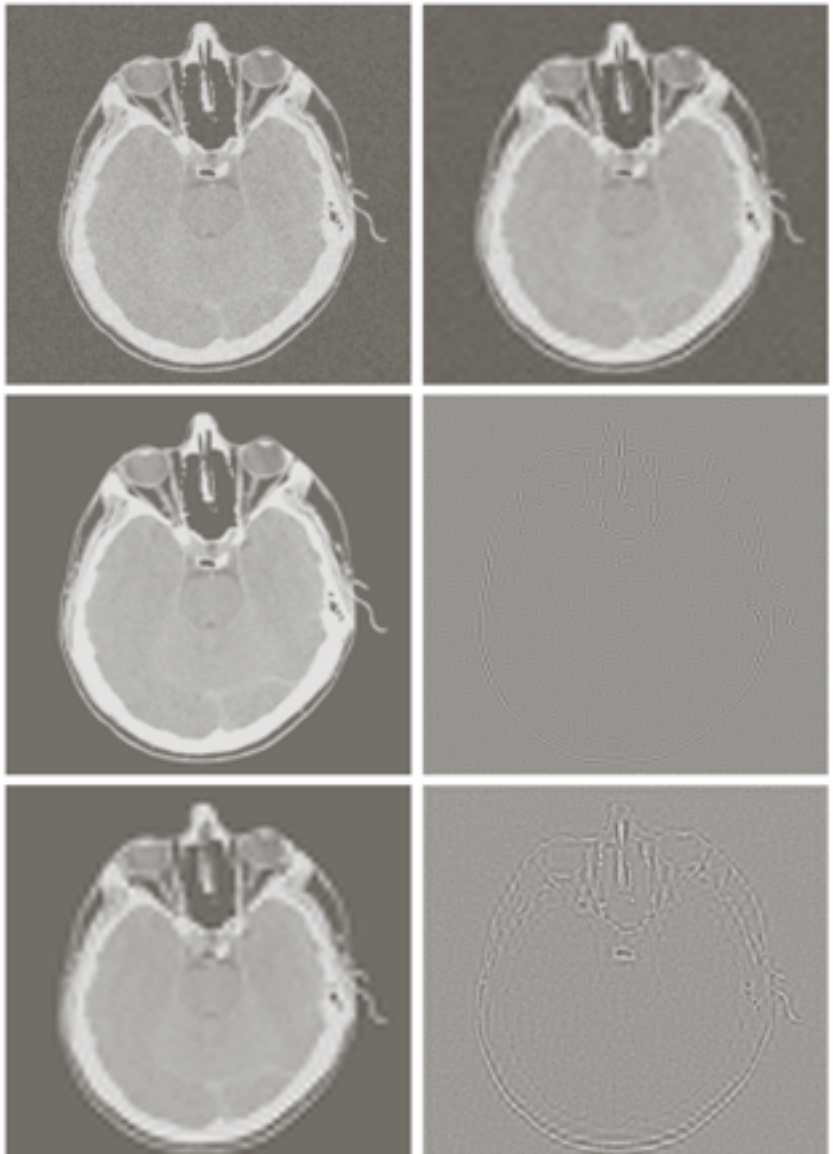


FIGURE 7.27 Modifying a DWT for edge detection: (a) and (c) two-scale decompositions with selected coefficients deleted; (b) and (d) the corresponding reconstructions.

Example 2



a	b
c	d
e	f

FIGURE 7.28 Modifying a DWT for noise removal: (a) a noisy CT of a human head; (b), (c) and (e) various reconstructions after thresholding the detail coefficients; (d) and (f) the information removed during the reconstruction of (c) and (e). (Original image courtesy Vanderbilt University Medical Center.)

Overview

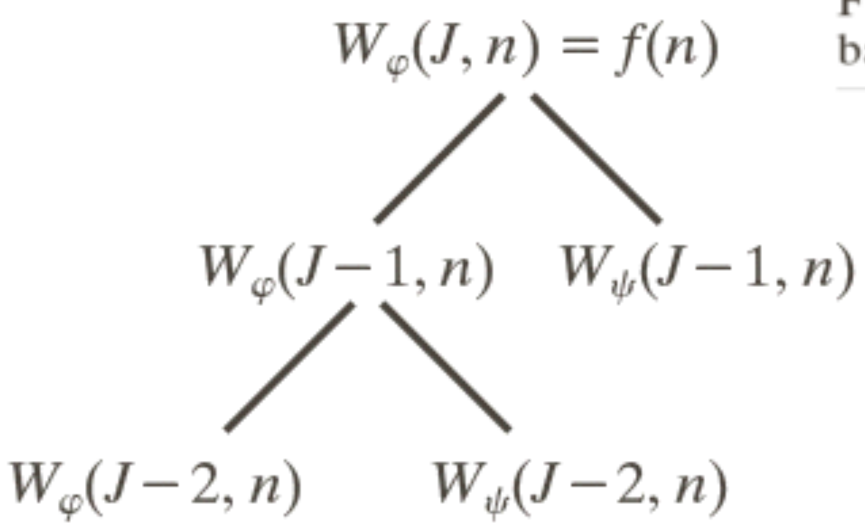
- Background
- Multiresolution expansions
- Wavelet transform in 1D
- Fast wavelet transform
- Wavelet transform in 2D
- Wavelet packets

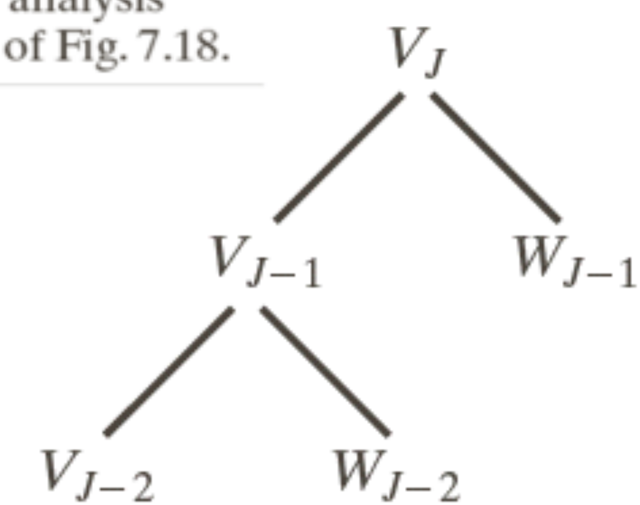
Wavelet packets

- Thus far, we have developed the pyramid one-sided:

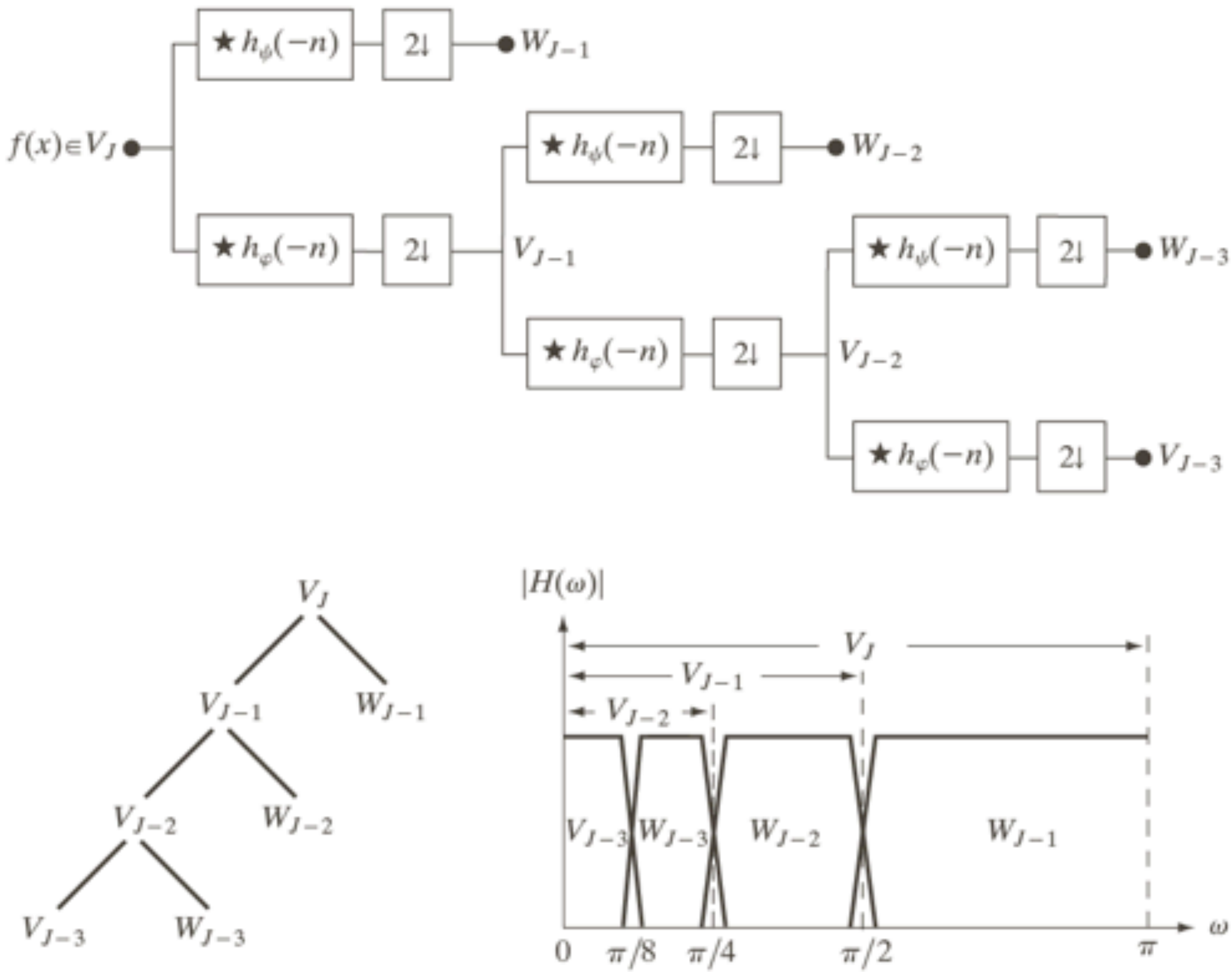
a b

FIGURE 7.29
An (a) coefficient tree and (b) analysis tree for the two-scale FWT analysis bank of Fig. 7.18.





Wavelet packets — a frequency view



a
b c

FIGURE 7.30 A three-scale FWT filter bank: (a) block diagram; (b) decomposition space tree; and (c) spectrum splitting characteristics.

Packets — a natural extension

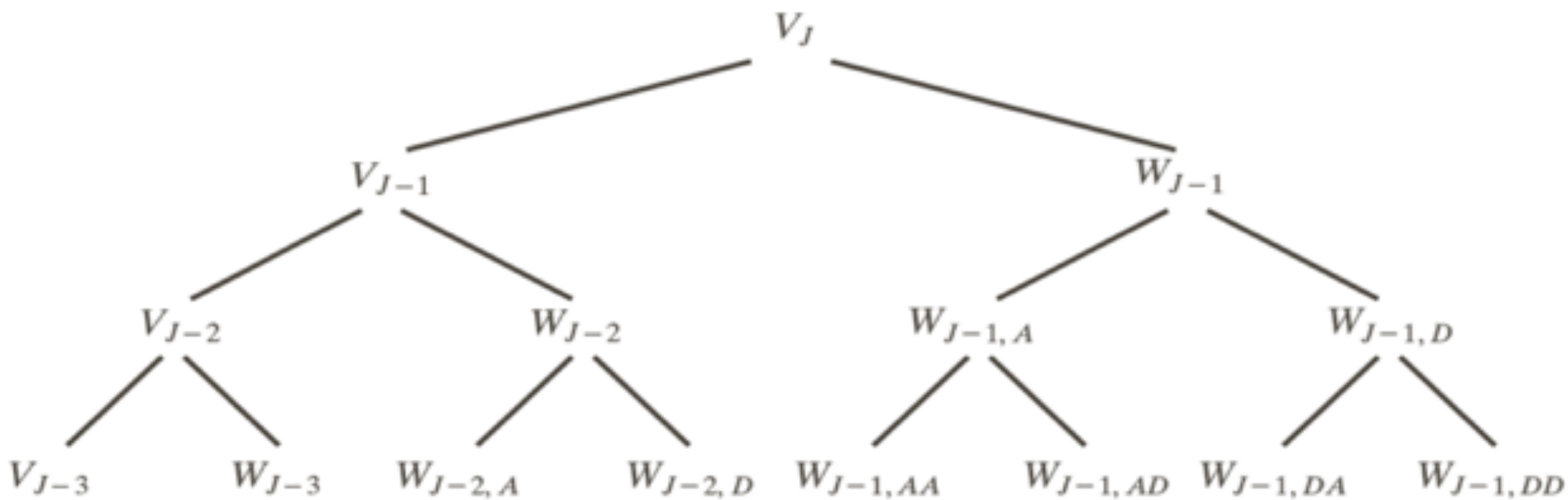
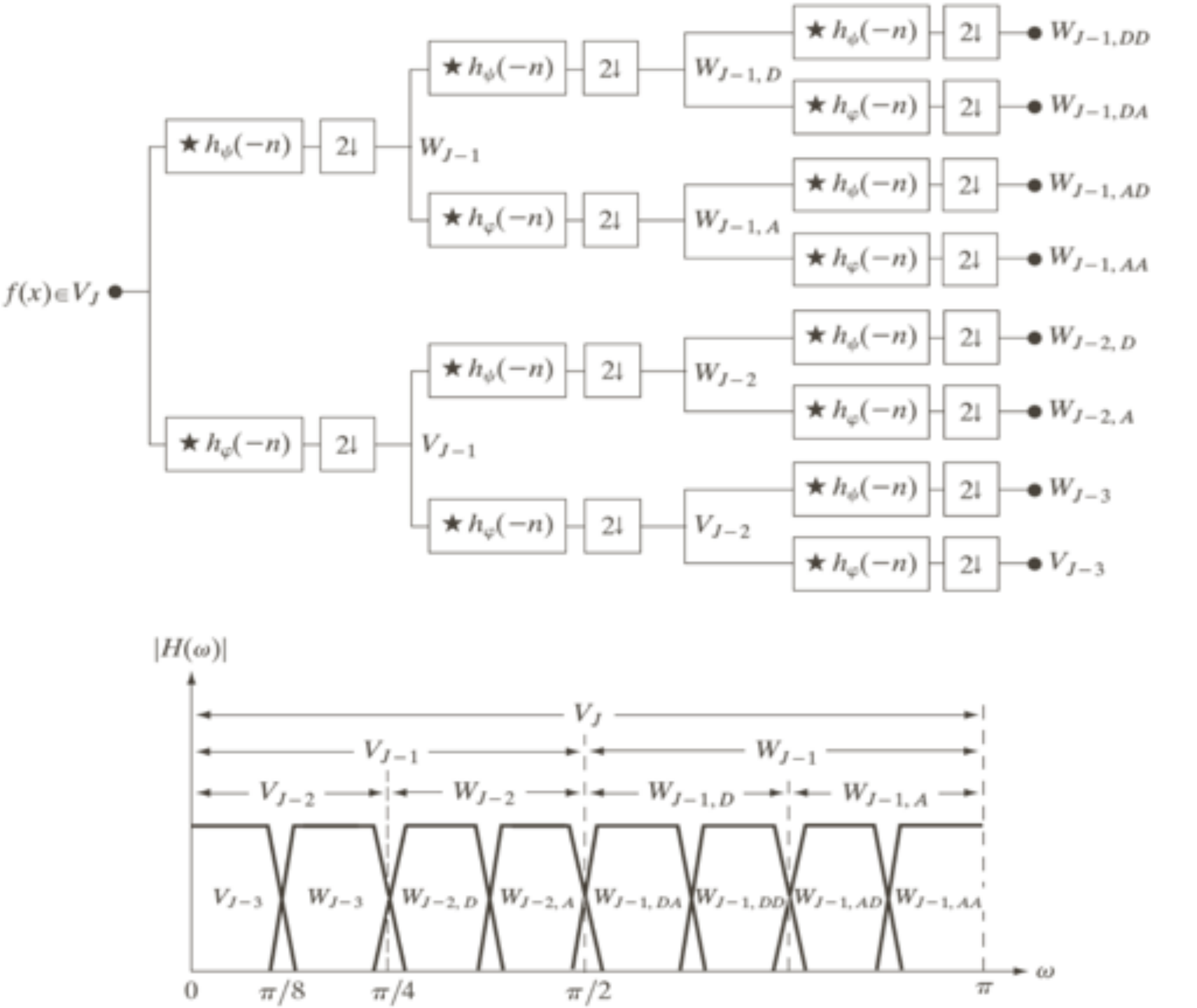


FIGURE 7.31
A three-scale wavelet packet analysis tree.

Packets — a natural extension



a
b

FIGURE 7.32
The (a) filter bank and (b) spectrum splitting characteristics of a three-scale full wavelet packet analysis tree.

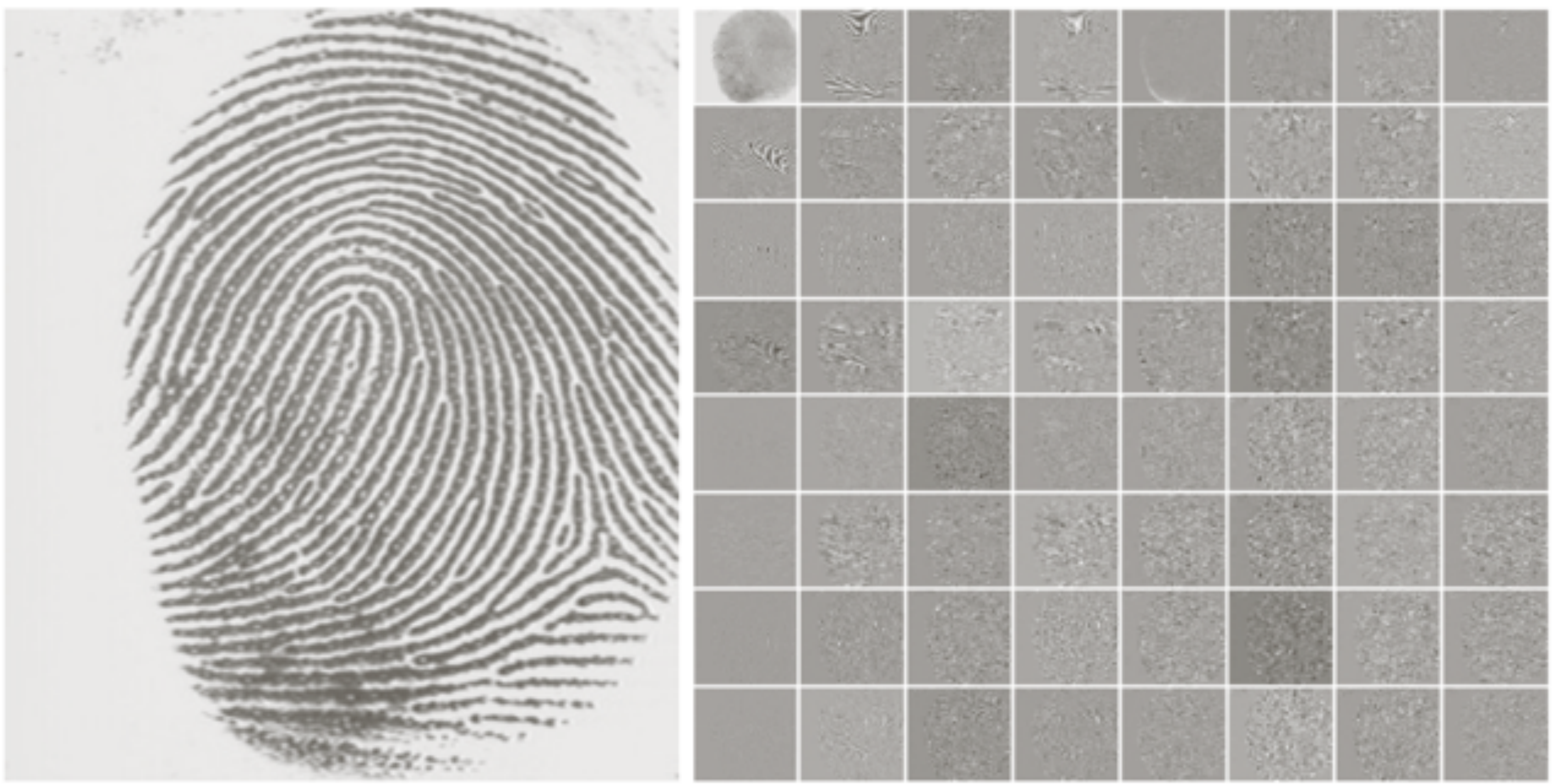
Packets — which decompostion to pick?

- which subtree should we pick in the complete tree for an efficient encoding of a signal?
- e.g. pick an additive cost function:

$$E(f) = \sum_{m,n} |f(m,n)|$$

- expand into children, only if children are 'cheaper'
- Example FBI fingerprinting data base

Fingerprints



a b

FIGURE 7.36 (a) A scanned fingerprint and (b) its three-scale, full wavelet packet decomposition. (Original image courtesy of the National Institute of Standards and Technology.)

Fingerprints

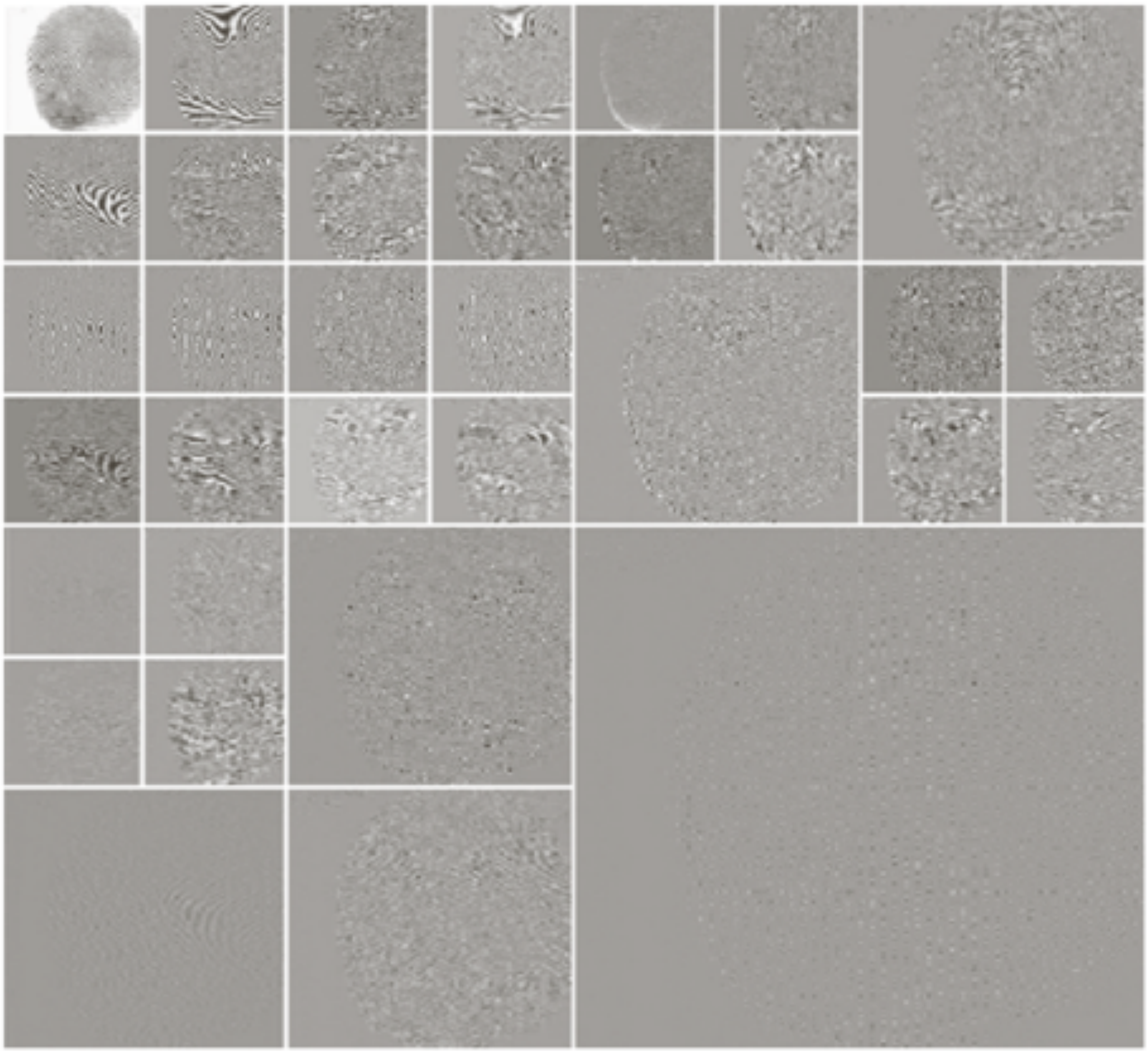


FIGURE 7.37 An optimal wavelet packet decomposition for the fingerprint of Fig. 7.36(a).