Multi-Resolution Representation (aka Pyramid Representation)

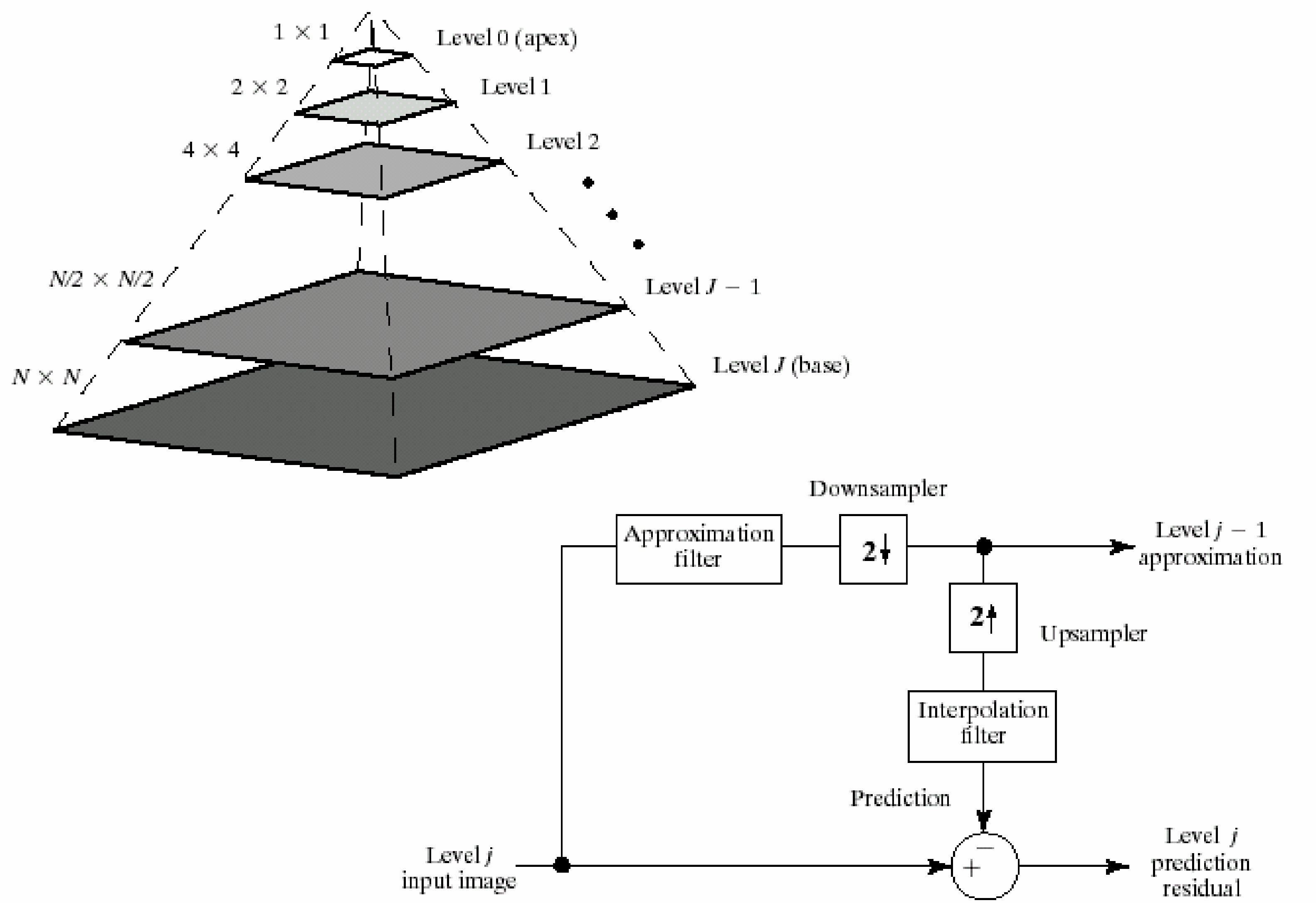


FIGURE 7.2 (a) A pyramidal image structure and (b) system block diagram for creating it.

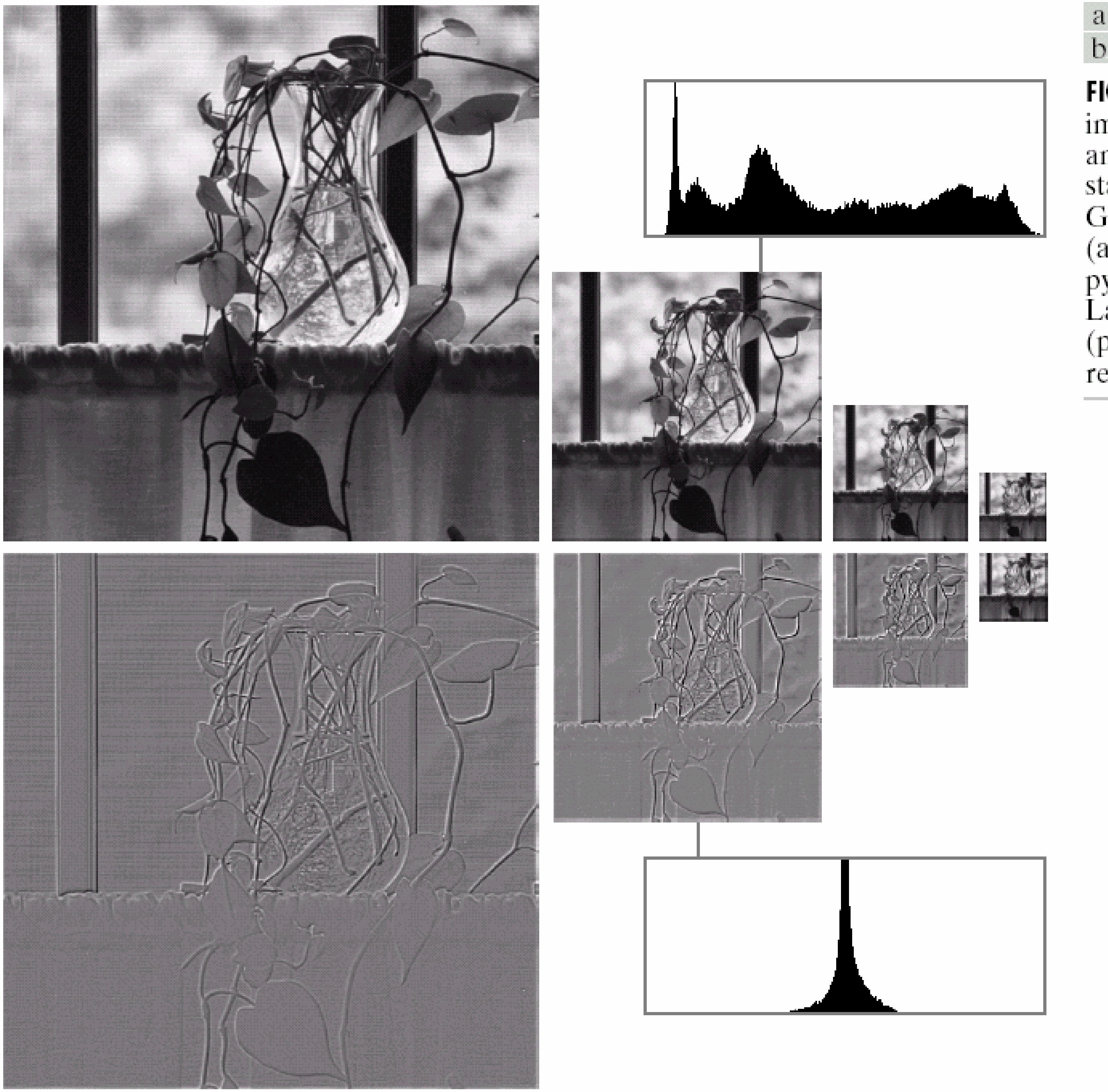


FIGURE 7.3 Two image pyramids and their statistics: (a) a Gaussian (approximation) pyramid and (b) a Laplacian (prediction residual) pyramid.

Wavelet vs. Pyramid vs. Subband Decomposition

- Wavelet transform is a particular way of generating the Laplacian pyramid
- There are many ways to interpret wavelet transform. Here we describe the generation of discrete wavelet transform using the tree-structured subband decomposition (aka iterated filterbank) approach
  - 1D 2-band decomposition
  - 1D tree-structured subband decomposition
  - Harr wavelet as an example
  - Extension to 2D by separable processing

Example: Haar Filter

$h0: \text{averaging, } [1,1]/\sqrt{2}; \quad h1: \text{difference, } [1,\text{-}1]/\sqrt{2};$

$g0 = [1,1]/\sqrt{2}; \quad g1 = [\text{-}1,1]/\sqrt{2}$

Input sequence : $[x1, x2, x3, x4, ...]$

Analysis(Assuming samples outside the boundaries are 0. remember to flip the filter when doing convolution)

$s = x * h0 = [s0, s1, s2, s3, s4, ...],\ s0 = (x1+0)/\sqrt{2}, s1 = (x2+x1)/\sqrt{2}, s2 = (x3+x2)/\sqrt{2}, s3 = (x4+x3)/\sqrt{2}...$

$y0 = s \downarrow 2 = [s1, s3, ...,]$

$t = x * h1 = [t0, t1, t2, t3, t4, ...], t0 = [x1 - 0]/\sqrt{2}, t1 = [x2 - x1]/\sqrt{2}, t2 = [x3 - x2]/\sqrt{2}, t3 = [x4 - x3]/\sqrt{2}, ...$

$y1 = t \downarrow 2 = [t1, t3, ...]$

Synthesis :

$u = y0 \uparrow 2 = [0, s1, 0, s3, ...]$

$q = u * g0 = [q1, q2, q3, q4, ...], q1 = (s1+0)/\sqrt{2} = (x1+x2)/2, q2 = (0+s1)/\sqrt{2} = (x1+x2)/2, q3 = (s3+0)/\sqrt{2} = (x3+x4)/2$

$v = y1 \uparrow 2 = [0, t1, 0, t3, ...]$

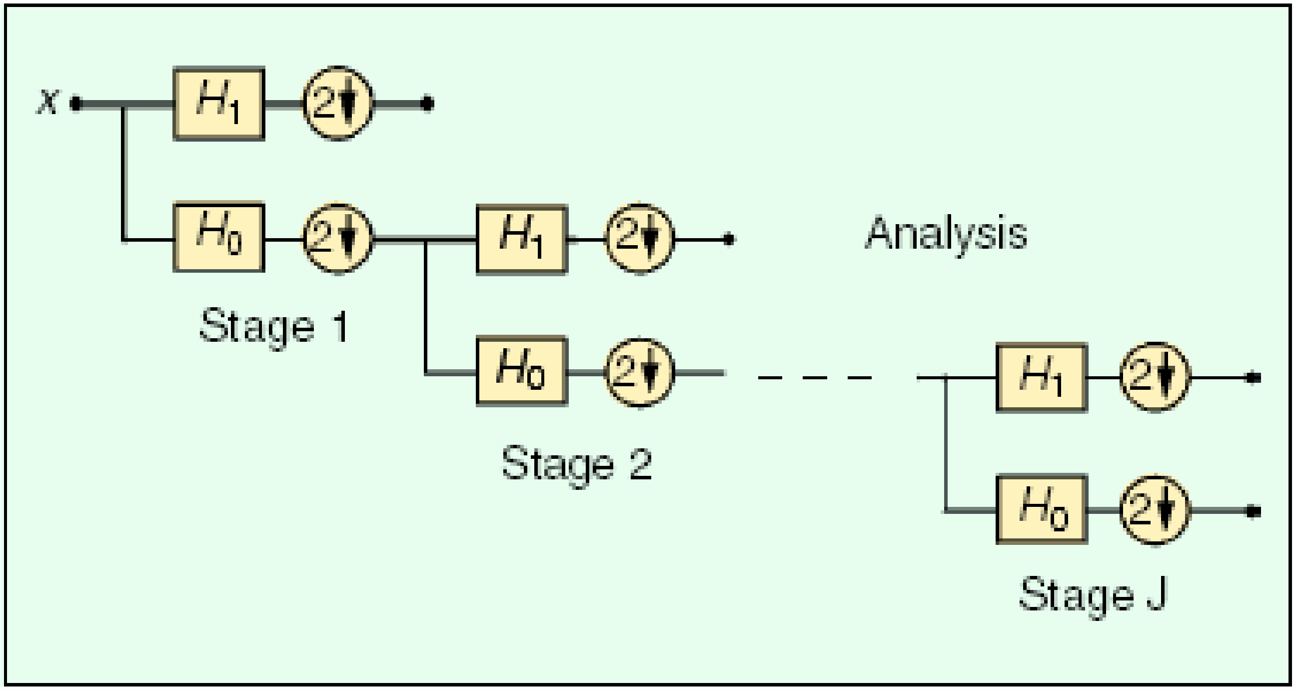
$r = v * g1 = [r1, r2, r3, r4, ...], r1 = (\text{-}t1+0)/\sqrt{2} = (x1-x2)/2, r2 = (\text{-}0+t1)/\sqrt{2} = (\text{-}x1+x2)/2, r3 = (\text{-}t3+0)/\sqrt{2} = (x3-x4)/2,$

$\hat{x} = q + r = [q1+r1, q2+r2, ...] = [x1, x2, x3, ...]$

Note with Haar wavelet, the lowpass subband essentially takes the average of every two samples, L=(x1+x2)/sqrt(2), and the highpass subband takes the difference of every two samples, H=(x1-x2)/sqrt(2).

For synthesis, you take the sum of the lowpass and high pass signal to recover first sample A=(L+H)/sqrt(2), and you take the difference to recover the second sample B=(L-H)/sqrt(2).

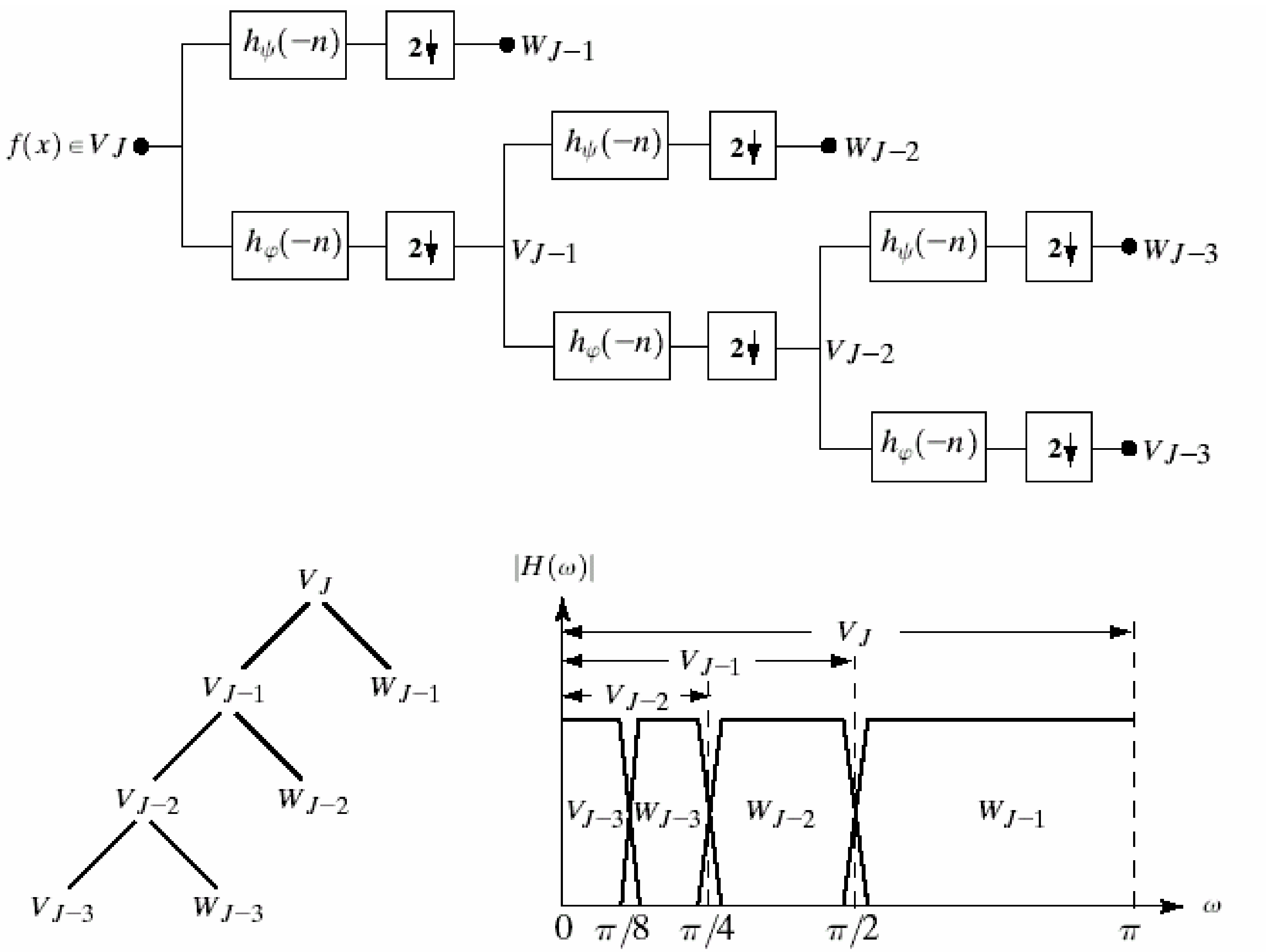
Iterated Filter Bank



▲ 3. Iterated filter bank. The lowpass branch gets split repeatedly to get a discrete-time wavelet transform.

From [Vetterli01]

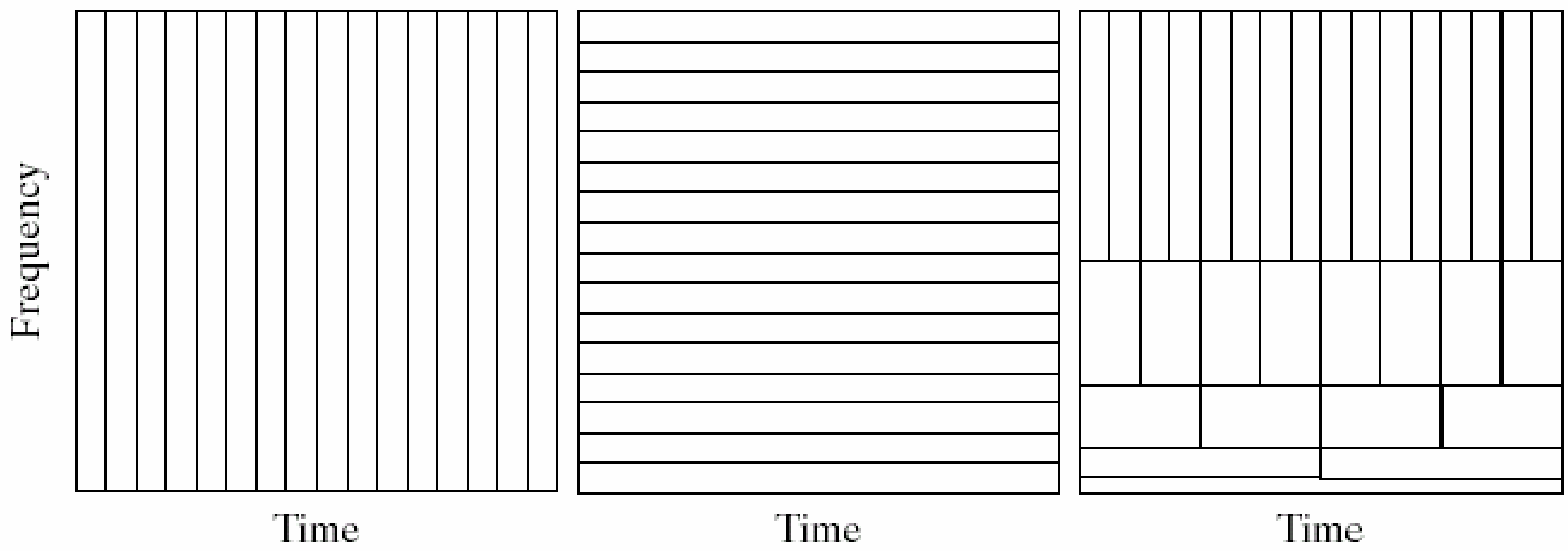
Discrete Wavelet Transform = Iterated Filter Bank



a
b c

FIGURE 7.28 A three-scale FWT filter bank: (a) block diagram; (b) decomposition space tree; and (c) spectrum splitting characteristics.

Temporal-Frequency Domain Partition



a b c

FIGURE 7.21 Time-frequency tilings for (a) sampled data, (b) FFT, and (c) FWT basis functions.

Wavelet Transform vs. Fourier Transform

- Fourier transform:
  - Basis functions cover the entire signal range, varying in frequency only
- Wavelet transform
  - Basis functions vary in frequency (called “scale”) as well as spatial extend
    - High frequency basis covers a smaller area
    - Low frequency basis covers a larger area
    - Non-uniform partition of frequency range and spatial range
    - More appropriate for non-stationary signals

Haar Wavelet: Analysis

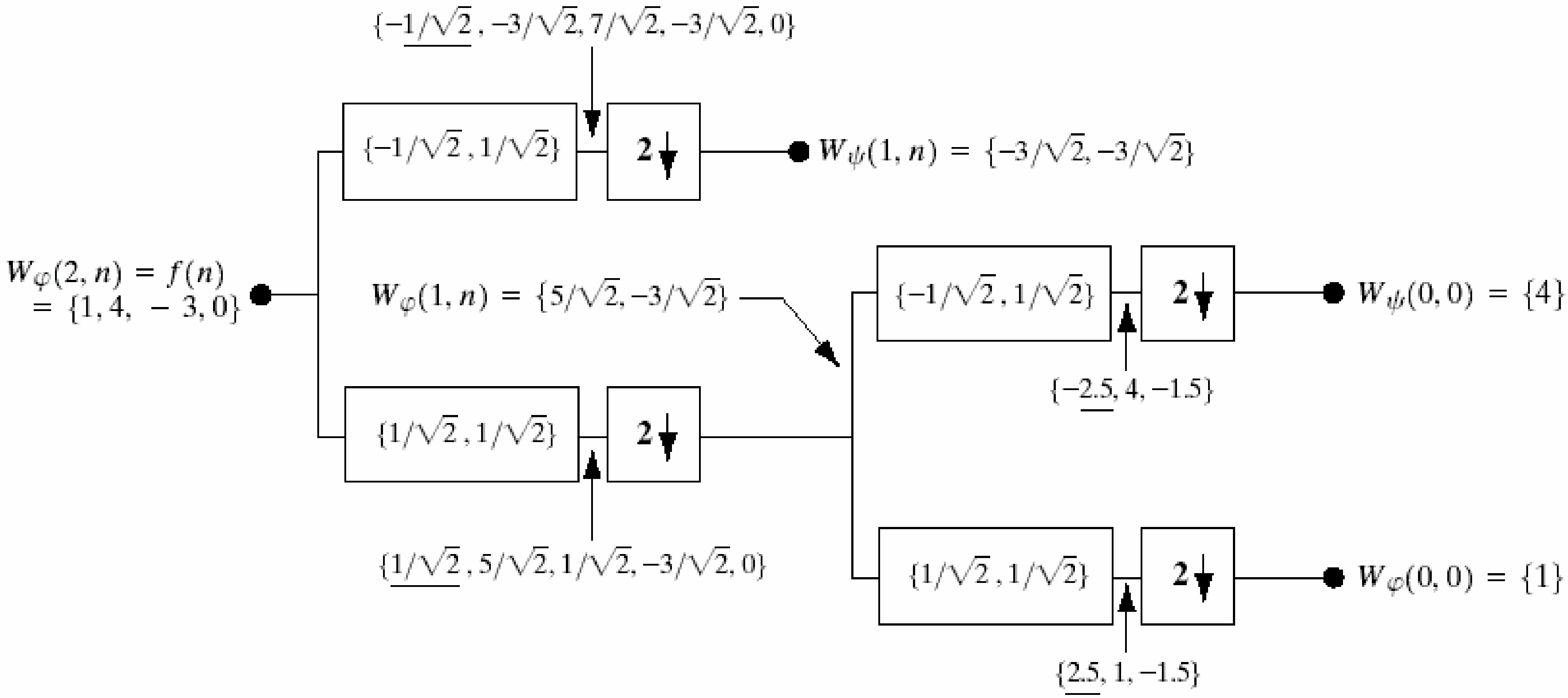


FIGURE 7.17 Computing a two-scale fast wavelet transform of sequence {1, 4, –3, 0} using Haar scaling and wavelet vectors.

Note that the assumed high pass filter in this example has a factor “-1” difference from our previous example. Similarly the synthesis filter is off by the same factor. Both are OK.

Haar Wavelet: Synthesis

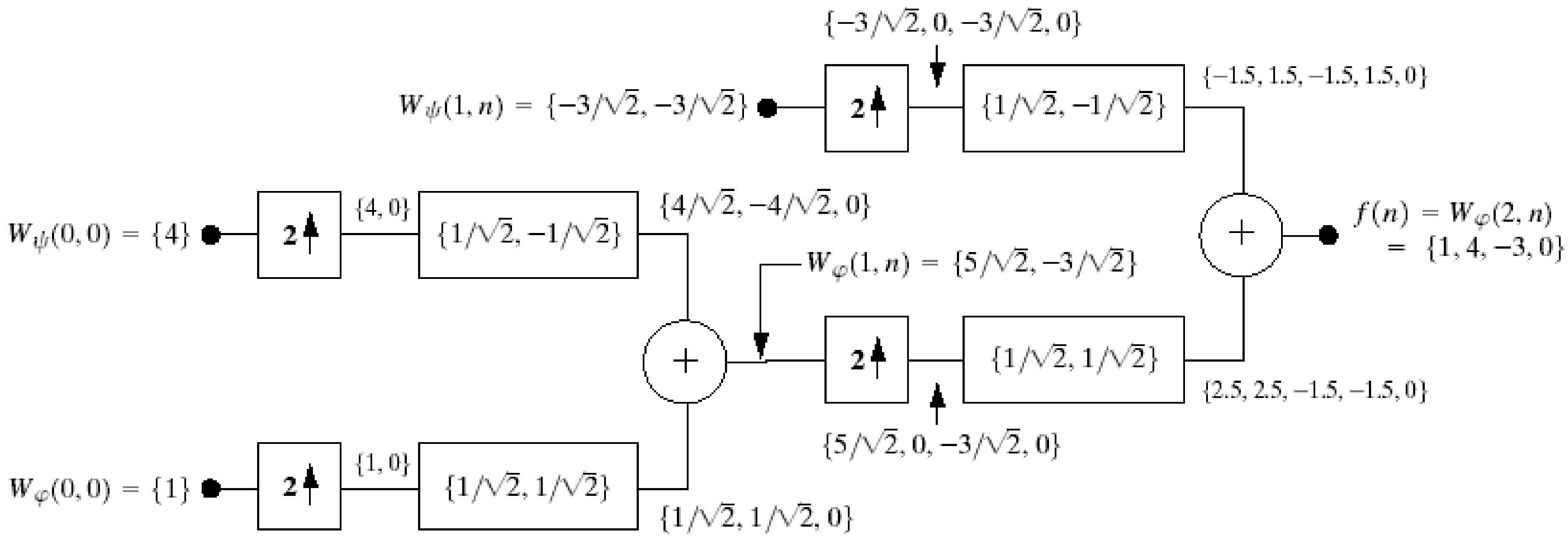


FIGURE 7.20 Computing a two-scale inverse fast wavelet transform of sequence $\{1, 4, -1.5\sqrt{2}, -1.5\sqrt{2}\}$ with Haar scaling and wavelet vectors.

How to Apply Filterbank to Images?

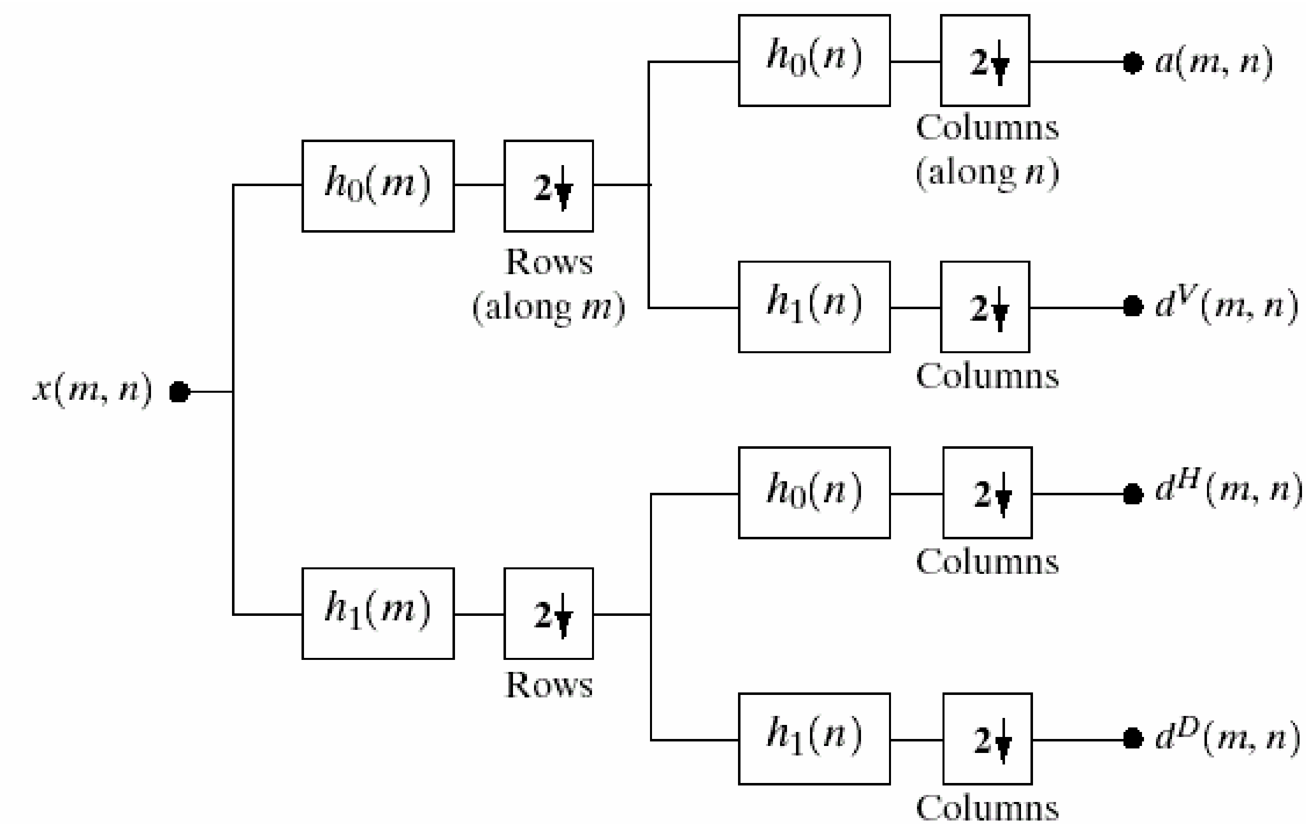


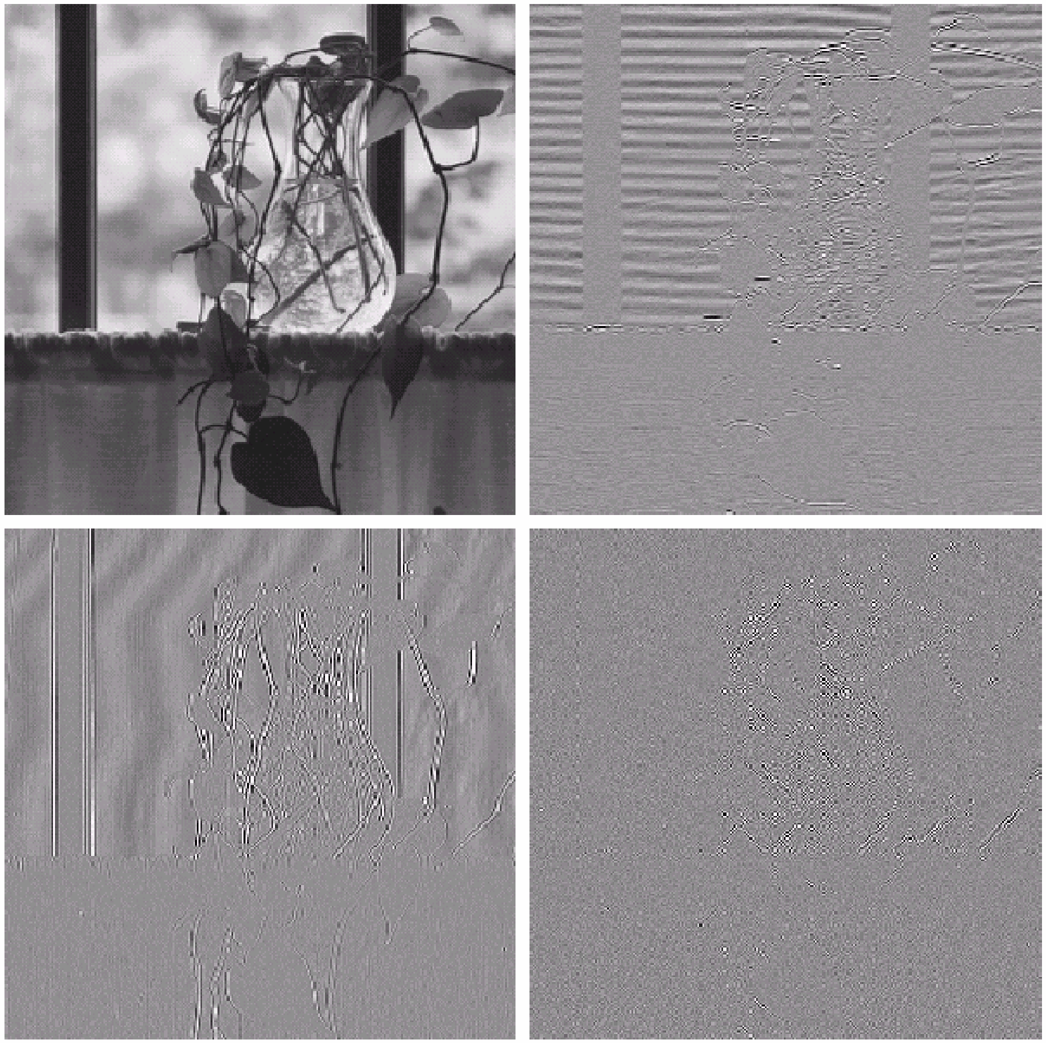
FIGURE 7.5 A two-dimensional, four-band filter bank for subband image coding.

2D decomposition is accomplished by applying the 1D decomposition along rows of an image first, and then columns.

1 Stage Decomposition: 4 Subimages

LL

HL



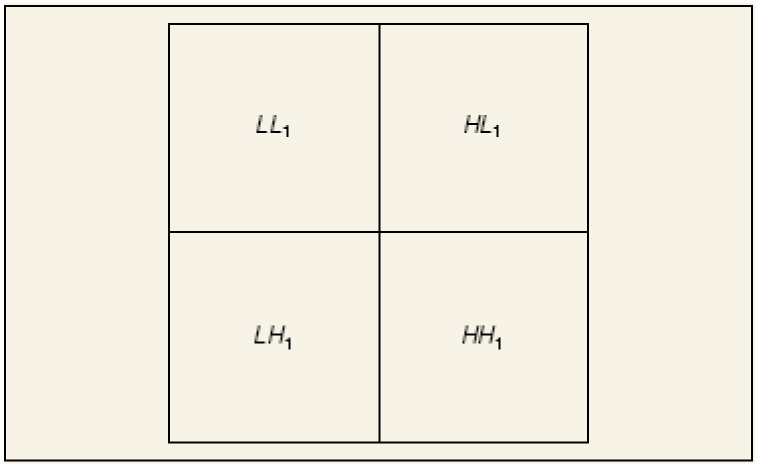
LH

HH

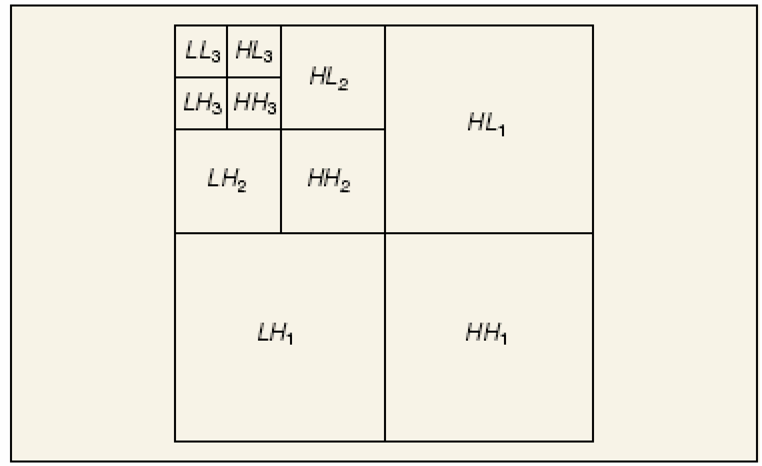
FIGURE 7.7 A four-band split of the vase in Fig. 7.1 using the subband coding system of Fig. 7.5.

With Harr filter, you can work on every 2x2 blocks in an image, [A,B;C,D]. LL=(A+B+C+D)/2;LH=(A+B-C-D)/2; HL=(A-B+C-D)/2; HH=(A+D-B-C)/2. For synthesis, A=(LL+LH+HL+HH)/2,B=((LL+LH)-(HL+HH))/2; C=((LL+HL)-(LH+HH))/2;D=((LL+HH)-(LH+HL))/2;

Wavelet Transform for Images



▲ 4. The subband labeling scheme for a one-level, 2-D wavelet transform.



▲ 6. The subband labeling scheme for a three-level, 2-D wavelet transform.

From [Usevitch01]

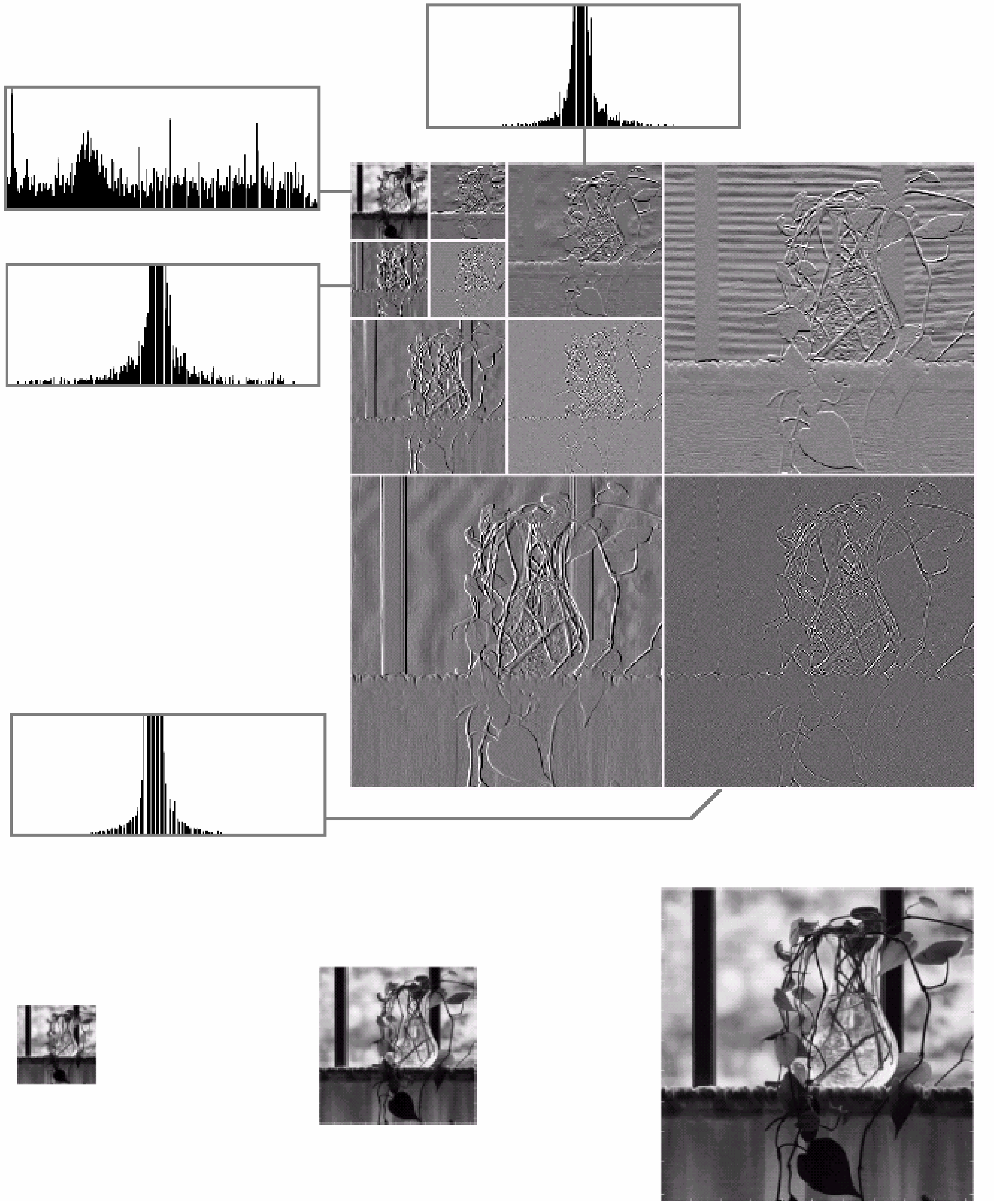
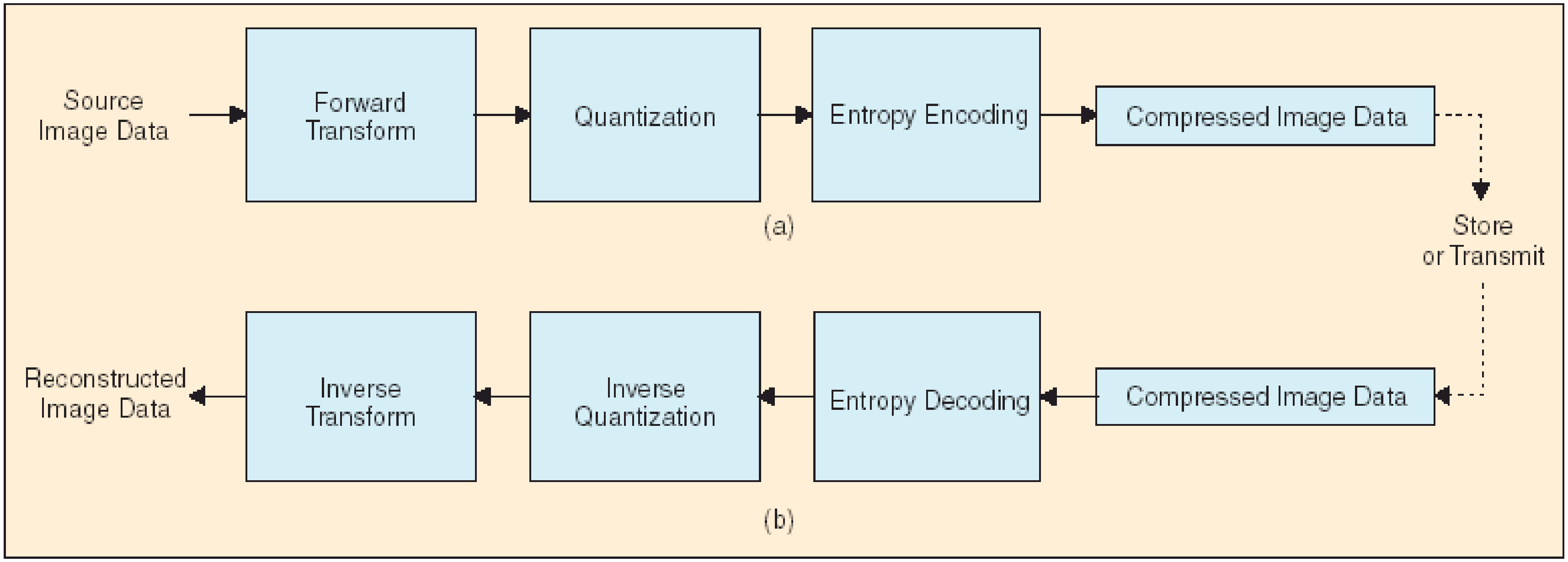


FIGURE 7.8 (a) A discrete wavelet transform using Haar basis functions. Its local histogram variations are also shown; (b)–(d) Several different approximations (64 × 64, 128 × 128, and 256 × 256) that can be obtained from (a).

JPEG2000 Codec Block Diagram



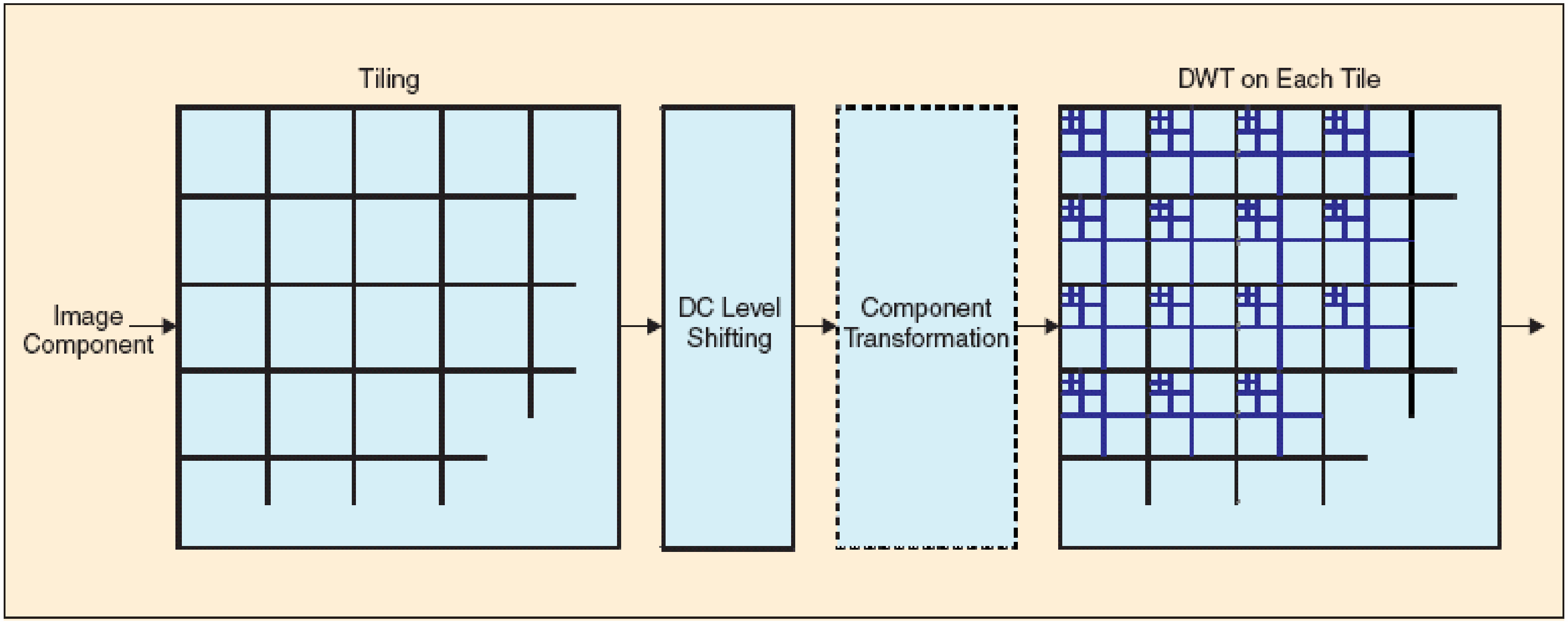
▲ 2. *General block diagram of the JPEG 2000 (a) encoder and (b) decoder.*

- **Quantization**: Each subband may use a different step-size. Quantization can be skipped to achieve lossless coding
- **Entropy coding**: Bit plane coding is used, the most significant bit plane is coded first.
  - Uses sophisticated context-based arithmetic coding
- **Quality scalability** is achieved by decoding only partial bit planes, starting from the MSB. Skipping one bit plane while decoding = Increasing quantization stepsize by a factor of 2.

Lossless vs. Lossy

- Lossless
  - Use LeGall 5/3 filter
  - Use lifting implementation
  - Use an integer version of the RGB->YCbCr transformation
  - No quantization of coefficients
- Lossy
  - Use Daubechies 9/7 filter
  - Use the conventional RGB->YCbCr transformation

Preprocessing Steps

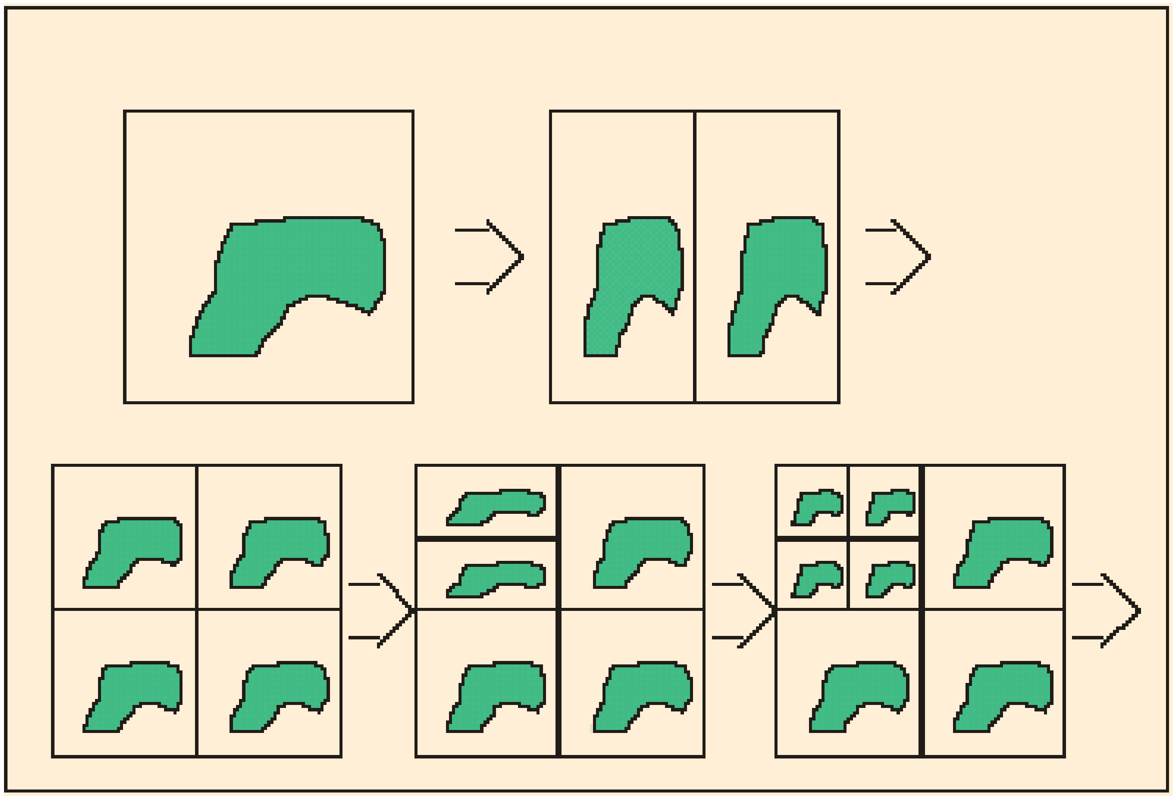


3. Tiling, dc-level shifting, color transformation (optional) and DWT of each image component.

- An image is divided into tiles, and each tile is processed independently
- Tiling can reduce the memory requirement and computation complexity
- Tiling also enable random access of different parts of an image
- The tile size controls trade-off between coding efficiency and complexity

Region of Interests

- Allows selected regions be coded with higher accuracy
  - Ex: faces



▲ 13. Wavelet domain ROI mask generation.