EL5123 --- Image Processing

Wavelet Transforms and JPEG2000

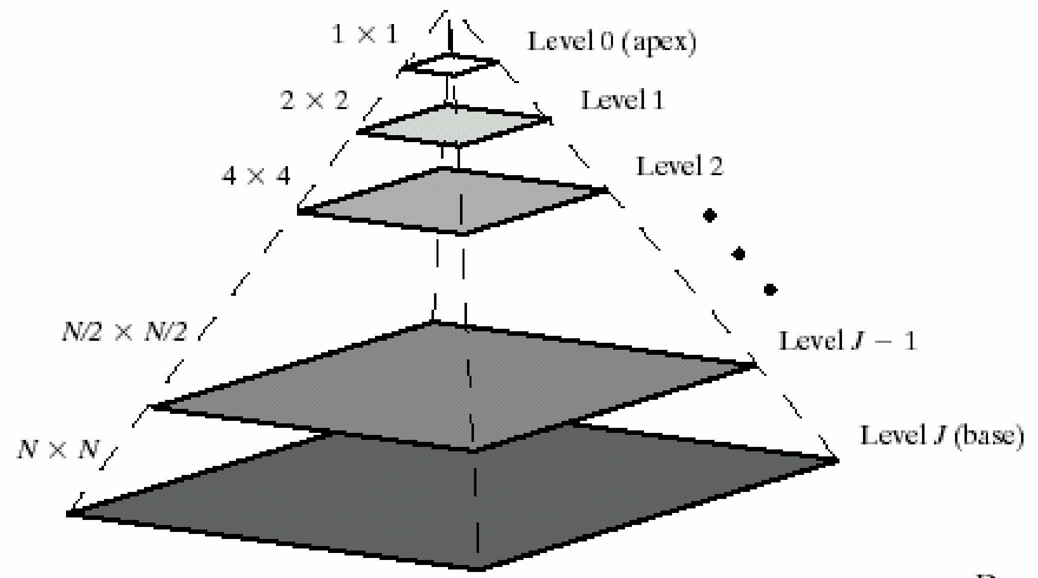
Yao Wang
Polytechnic University, Brooklyn, NY 11201

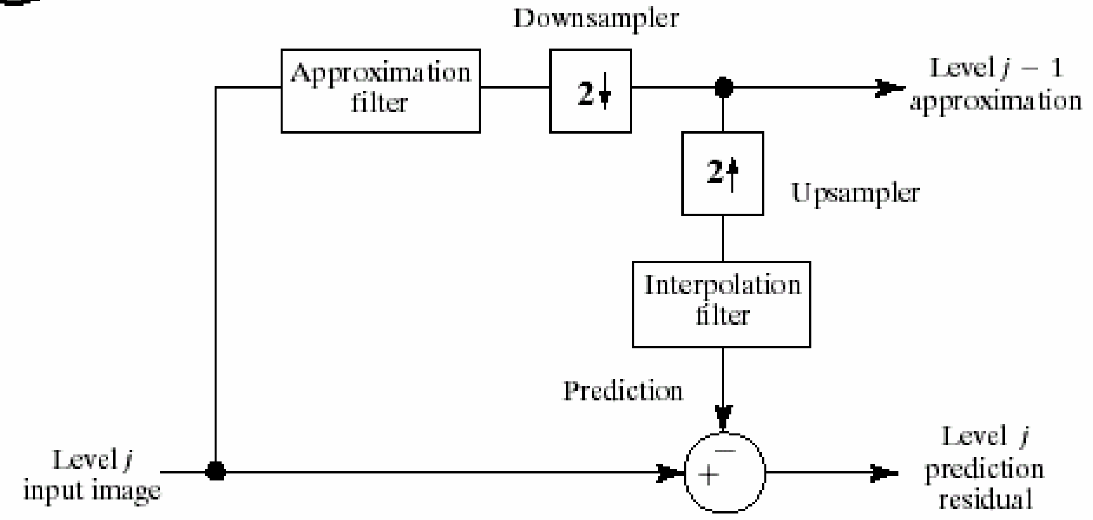
Based on
Gonzalez/Woods, Digital Image Processing, 2ed
A. Skodras, C. Christopoulos, T. Ebrahimi, The JPEG2000 Still Image Compression Standard, IEEE Signal Processing Magazine, Sept. 2001.
[Some figures are extracted from above references]

Lecture Outline

- Introduction
- Multi-resolution representation of images
- Wavelet transform through Iterated Filterbank Implementation
- Basic Ideas in JPEG2000 Codec
- JPEG vs. JPEG2000
- Scalability: why and how

Multi-Resolution Representation (aka Pyramid Representation)





a b

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

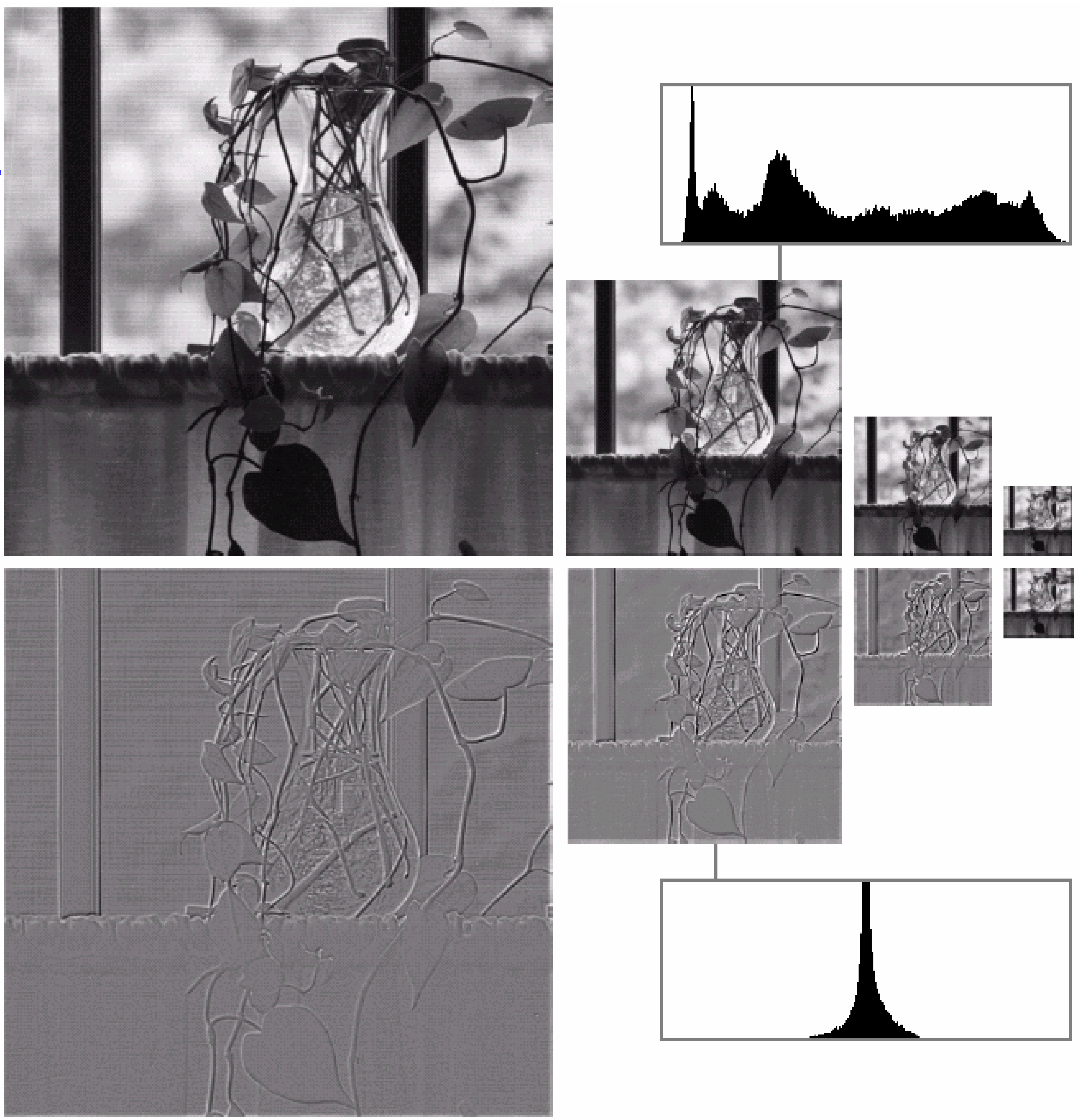


FIGURE 7.3 Two image pyramids and their statistics: (a) a Gaussian (approximation) pyramid and (b) a Laplacian (prediction residual) pyramid.

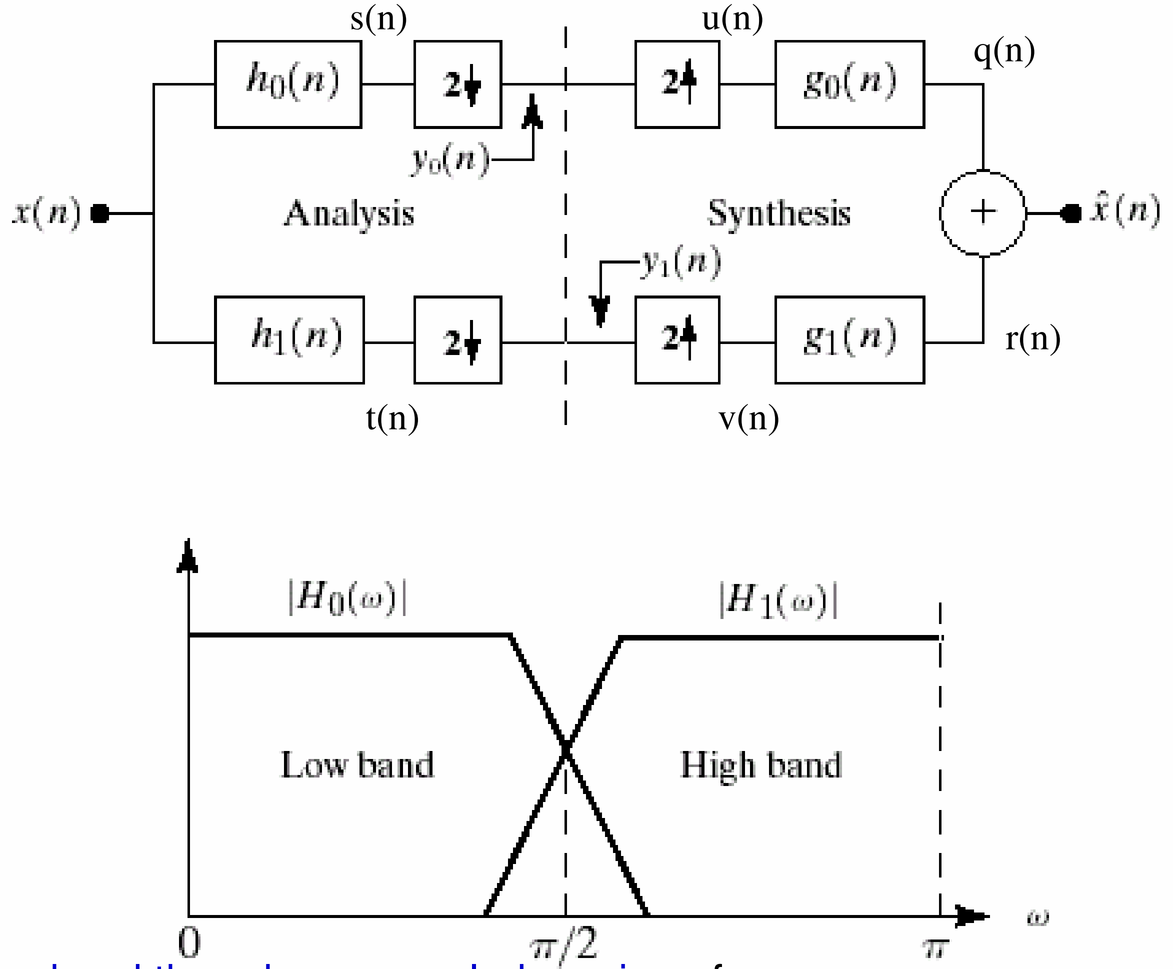
Wavelet vs. Pyramid vs. Subband Decomposition

- 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

Two Band Filterbank

a
b

FIGURE 7.4 (a) A two-band filter bank for one-dimensional subband coding and decoding, and (b) its spectrum splitting properties.



h0: Lowpass filter, y0: a blurred and then down-sampled version of x
h1: Highpass filter, y1: edges in x
When the filters h0,h1, g0, g1 are designed appropriately,
X^=X (perfect reconstruction filterbank)

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)

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

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

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

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

Synthesis:

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

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

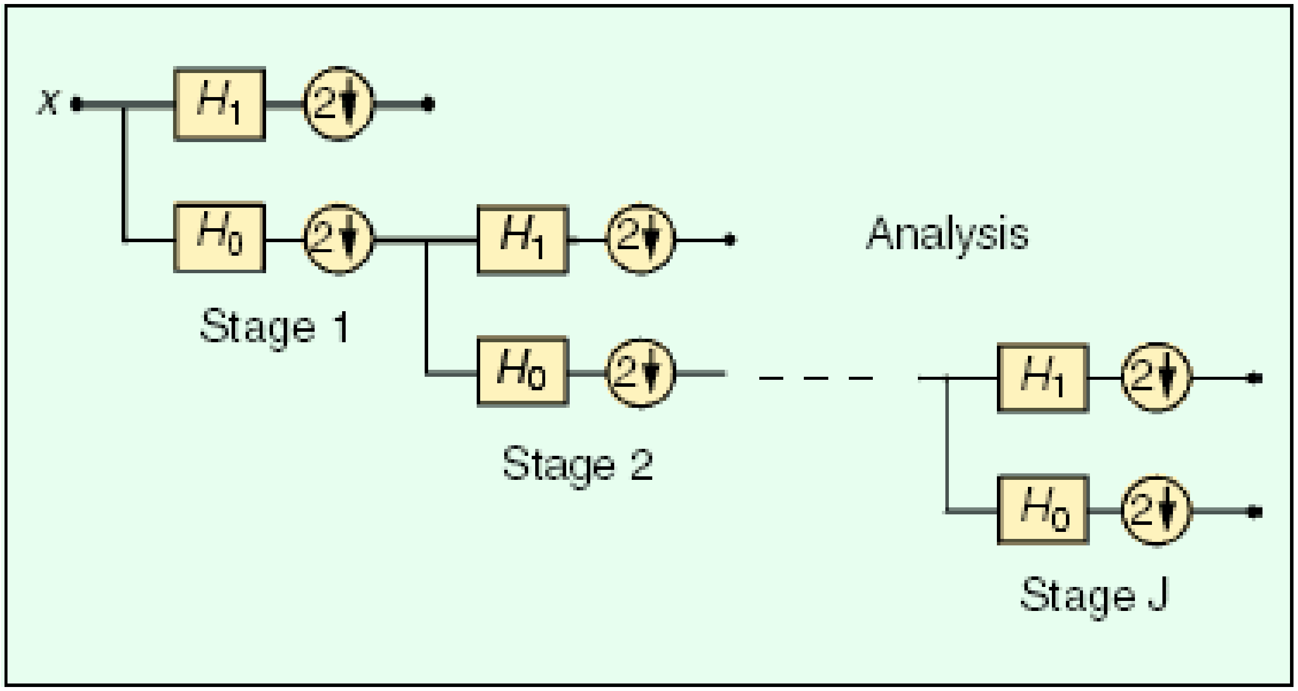
$v = y1 \uparrow 2 = [0, t1, 0, t3, ...]$

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

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

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

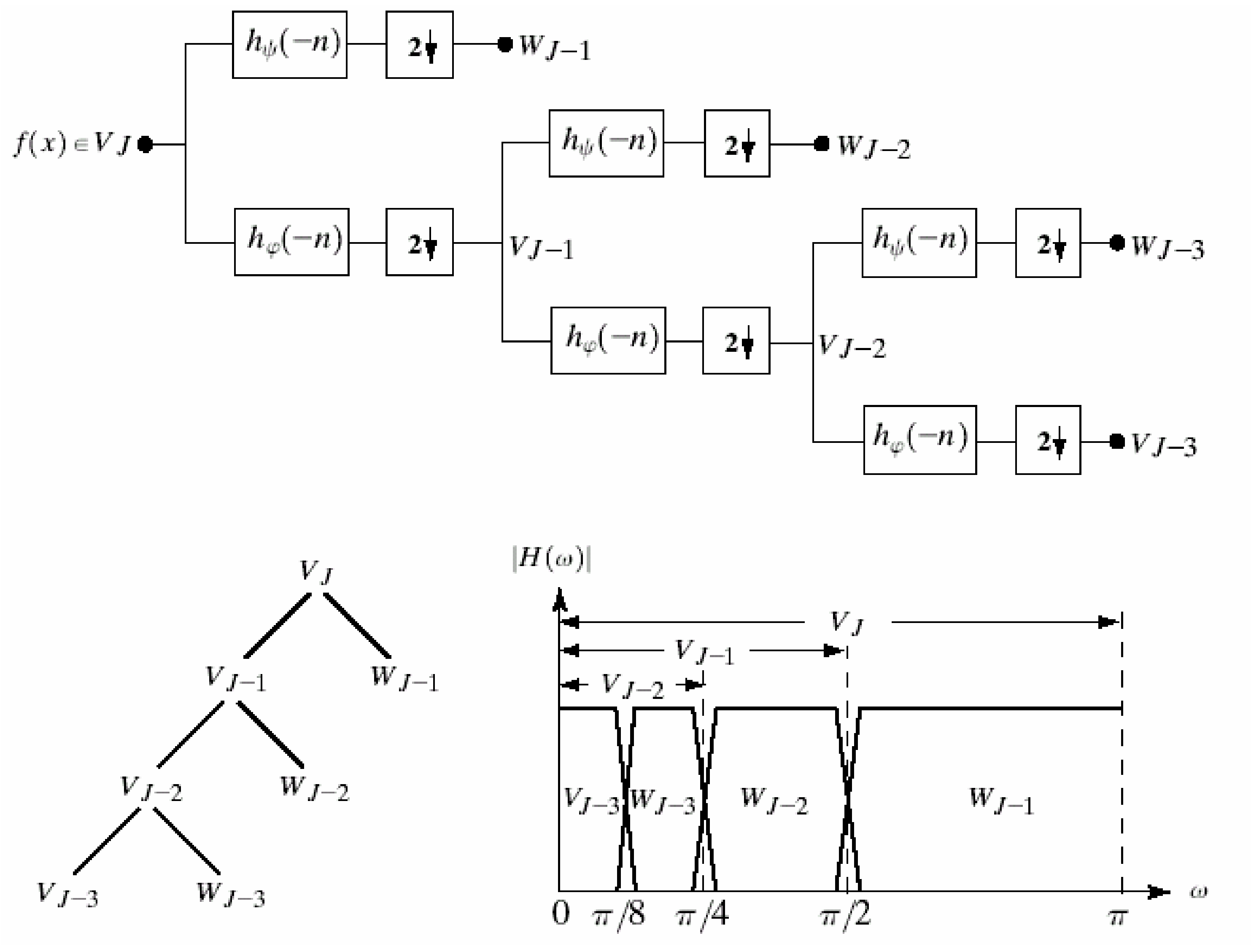
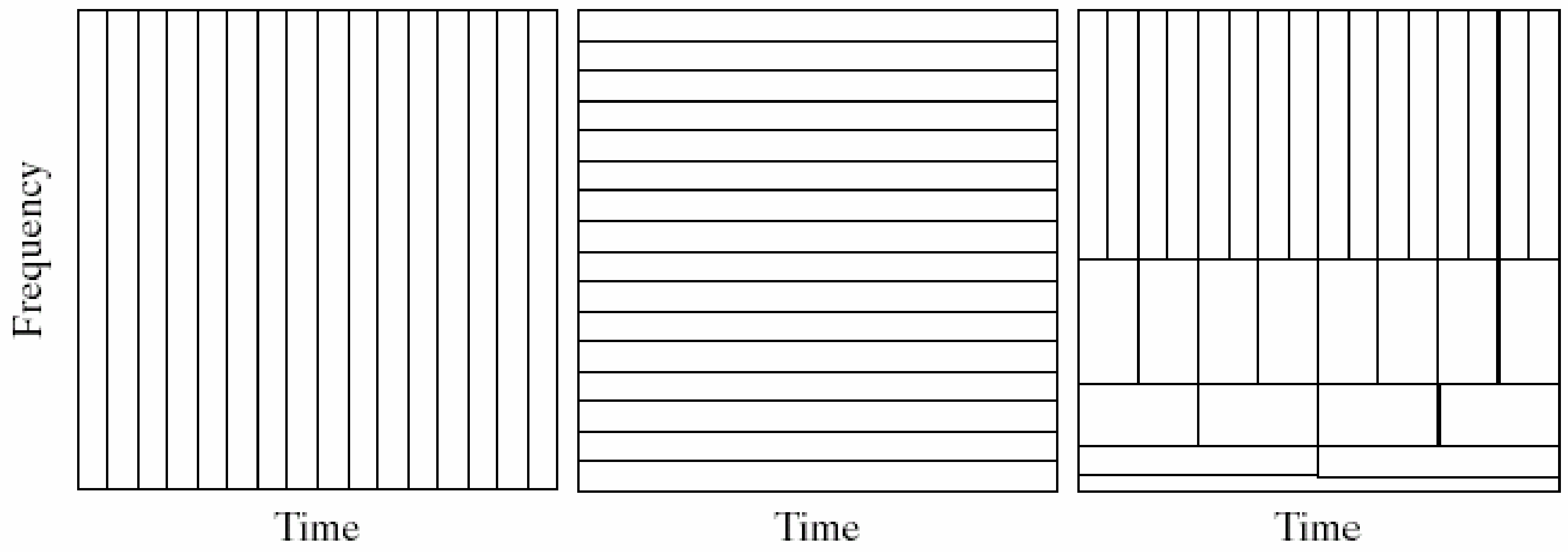


FIGURE 7.28 A three-scale FWT filter bank: (a) block diagram; (b) decomposition space tree; and (c) spectrum splitting characteristics.

Temporal-Frequency Domain Partition



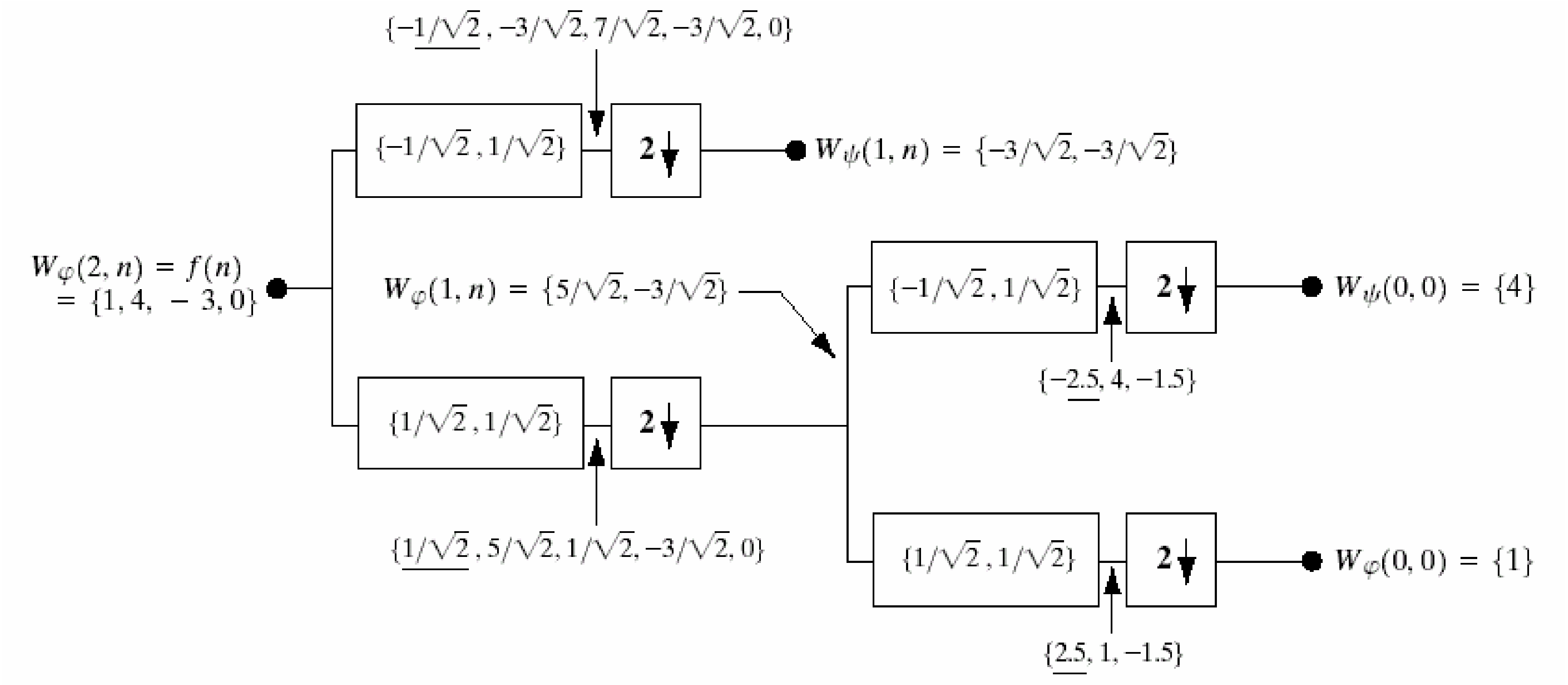
a b c

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



$\{-\underline{1/\sqrt{2}}, -3/\sqrt{2}, 7/\sqrt{2}, -3/\sqrt{2}, 0\}$
$\{-1/\sqrt{2}, 1/\sqrt{2}\}$
$2\downarrow$
$W_\psi(1,n) = \{-3/\sqrt{2}, -3/\sqrt{2}\}$
$W_\varphi(2,n) = f(n) = \{1, 4, -3, 0\}$
$W_\varphi(1,n) = \{5/\sqrt{2}, -3/\sqrt{2}\}$
$\{-1/\sqrt{2}, 1/\sqrt{2}\}$
$2\downarrow$
$W_\psi(0,0) = \{4\}$
$\{-\underline{2.5}, 4, -1.5\}$
$\{1/\sqrt{2}, 1/\sqrt{2}\}$
$2\downarrow$
$\{\underline{1/\sqrt{2}}, 5/\sqrt{2}, 1/\sqrt{2}, -3/\sqrt{2}, 0\}$
$\{1/\sqrt{2}, 1/\sqrt{2}\}$
$2\downarrow$
$W_\varphi(0,0) = \{1\}$
$\{\underline{2.5}, 1, -1.5\}$

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

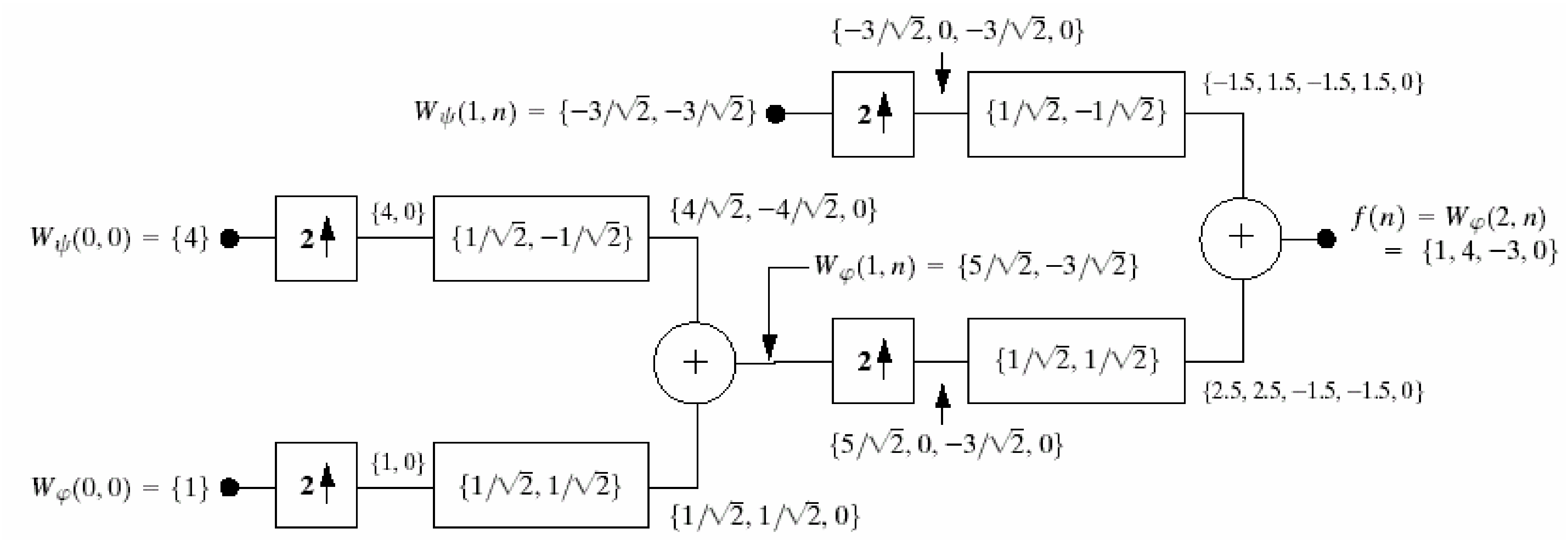

$\{-3/\sqrt{2}, 0, -3/\sqrt{2}, 0\}$
$W_\psi(1,n) = \{-3/\sqrt{2}, -3/\sqrt{2}\}$
$2\uparrow$
$\{1/\sqrt{2}, -1/\sqrt{2}\}$
$\{-1.5, 1.5, -1.5, 1.5, 0\}$
$W_\psi(0,0) = \{4\}$
$\{4, 0\}$
$\{4/\sqrt{2}, -4/\sqrt{2}, 0\}$
$f(n) = W_\varphi(2,n) = \{1, 4, -3, 0\}$
$W_\varphi(1,n) = \{5/\sqrt{2}, -3/\sqrt{2}\}$
$\{1/\sqrt{2}, 1/\sqrt{2}\}$
$\{2.5, 2.5, -1.5, -1.5, 0\}$
$\{5/\sqrt{2}, 0, -3/\sqrt{2}, 0\}$
$W_\varphi(0,0) = \{1\}$
$\{1, 0\}$
$\{1/\sqrt{2}, 1/\sqrt{2}, 0\}$

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?

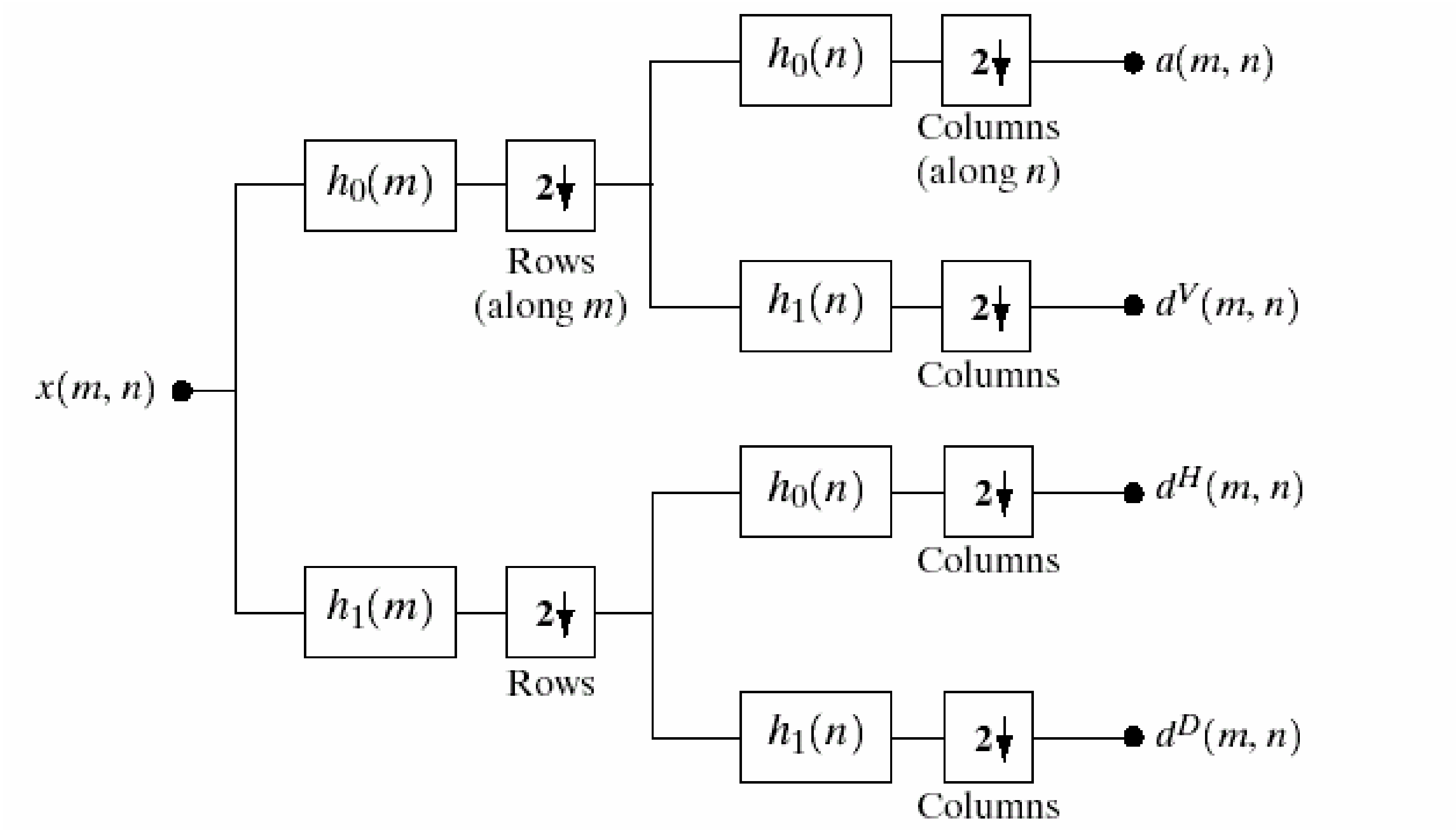


FIGURE 7.5 A two-dimensional, four-band filter bank for subband image coding.

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

1 Stage Decomposition: 4 Subimages

LL

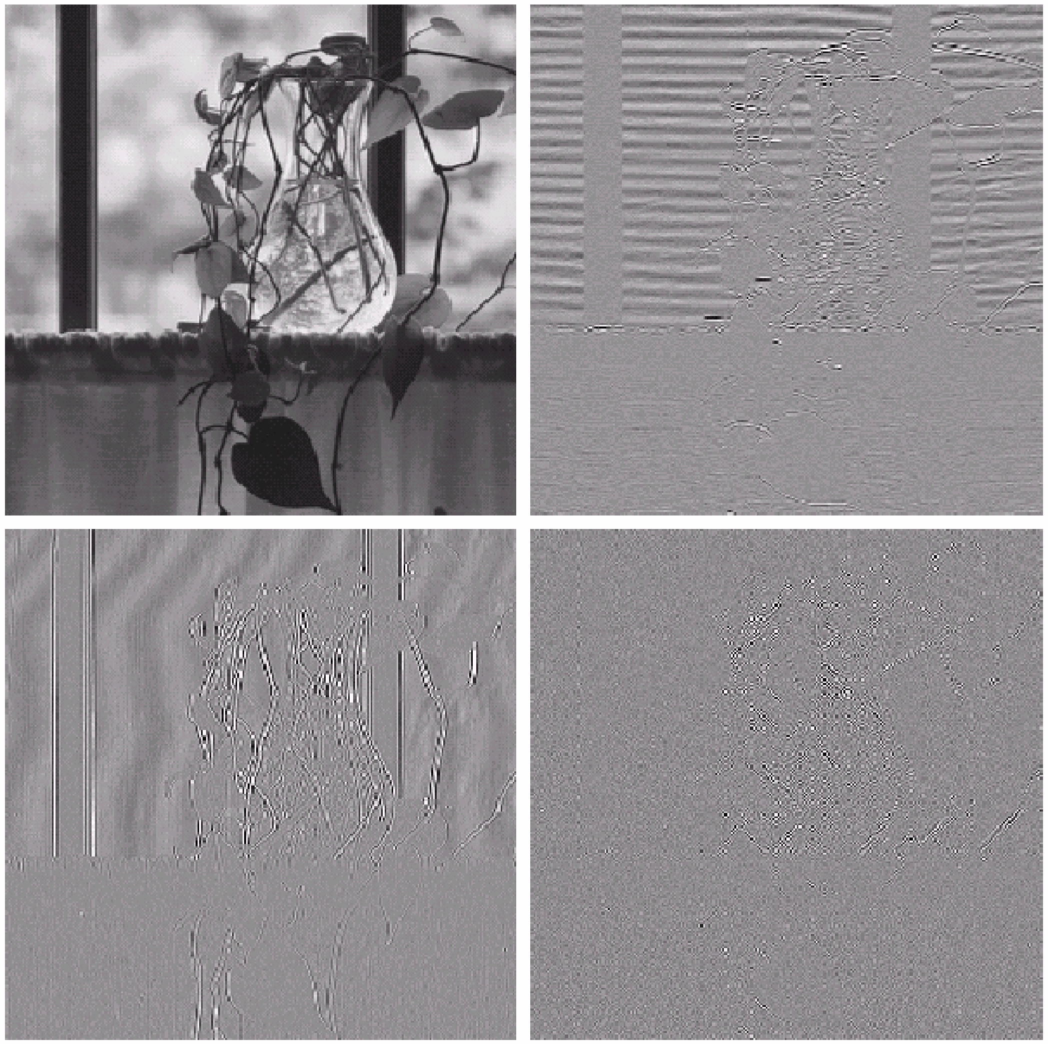


FIGURE 7.7 A four-band split of the vase in Fig. 7.1 using the subband coding system of Fig. 7.5.

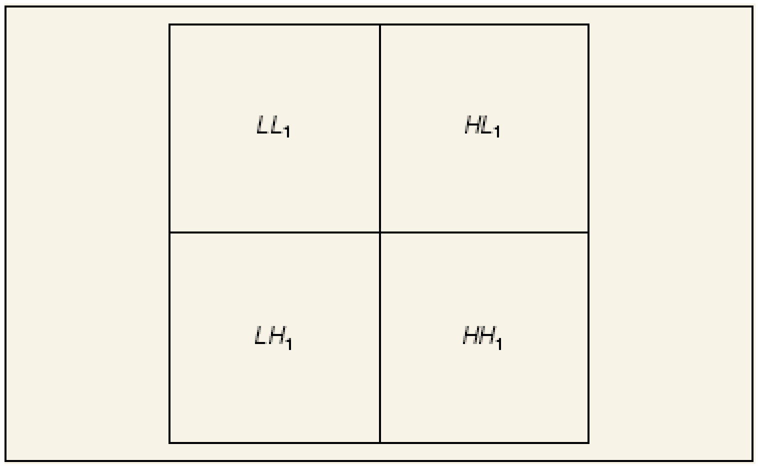
HL

LH

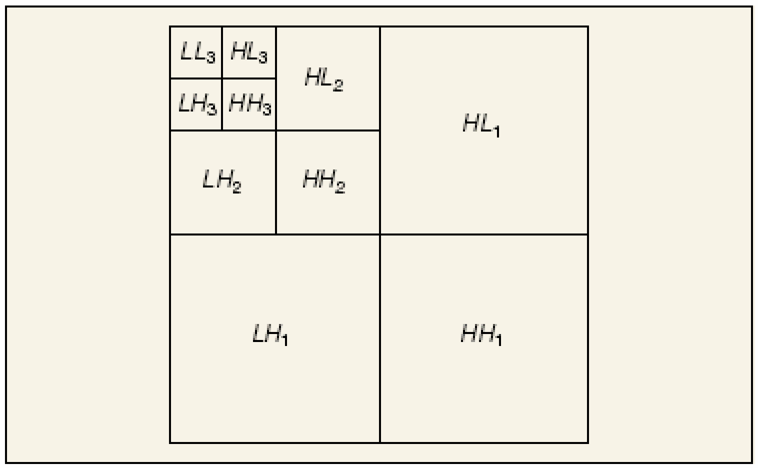
HH

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]

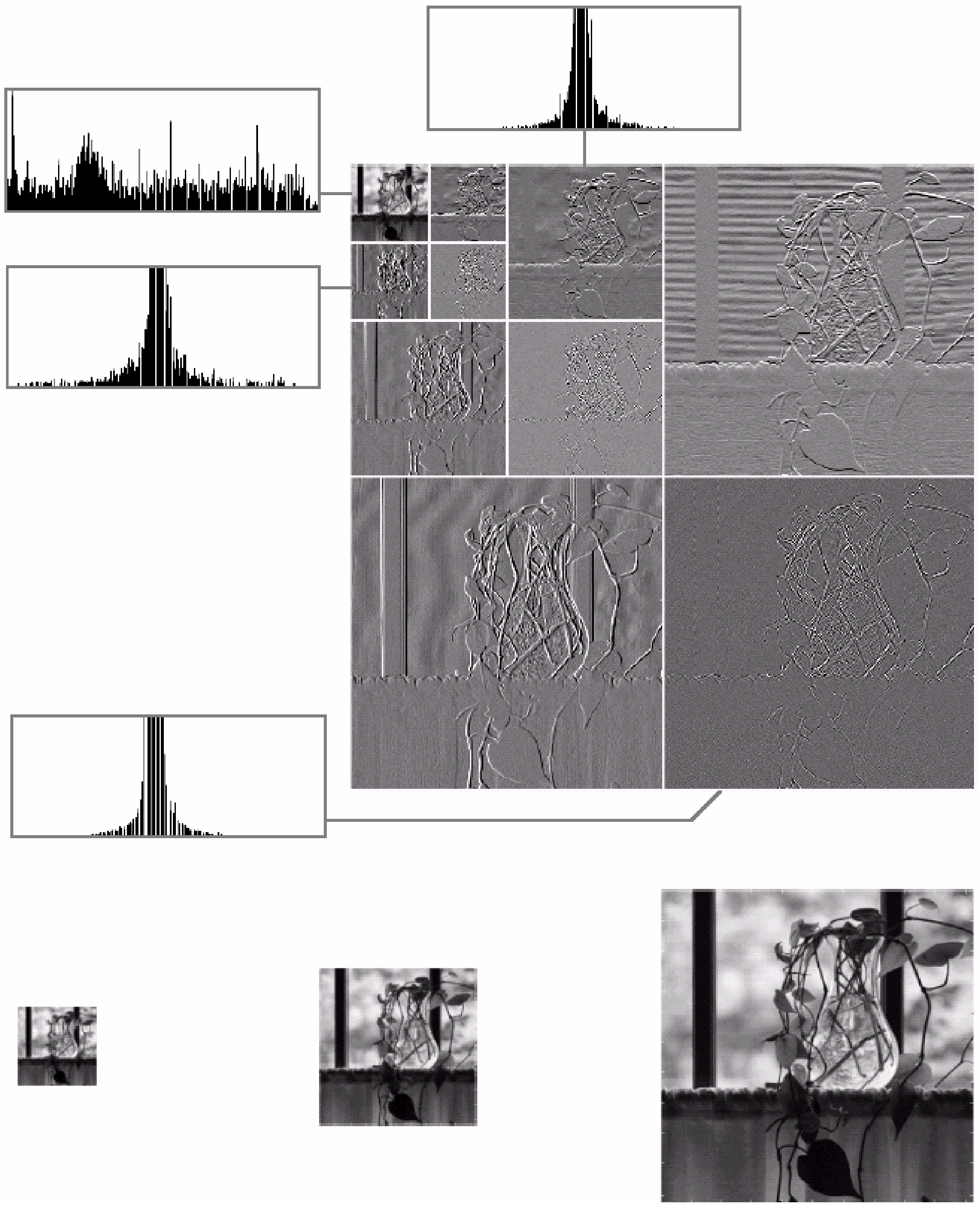


FIGURE 7.8 (a) A discrete wavelet transform using Haar basis functions. Its local histogram variations are also shown; (b)–(d) Several different approximations (64×64, 128×128, and 256×256) that can be obtained from (a).

Common Wavelet Filters

- Haar: simplest, orthogonal, not very good
- Daubechies 8/8: orthogonal
- Daubechies 9/7: bi-orthogonal, most commonly used if numerical reconstruction errors are acceptable
- LeGall 5/3: bi-orthogonal, integer operation, can be implemented with integer operations only, used for lossless image coding
- Differ in energy compaction capability

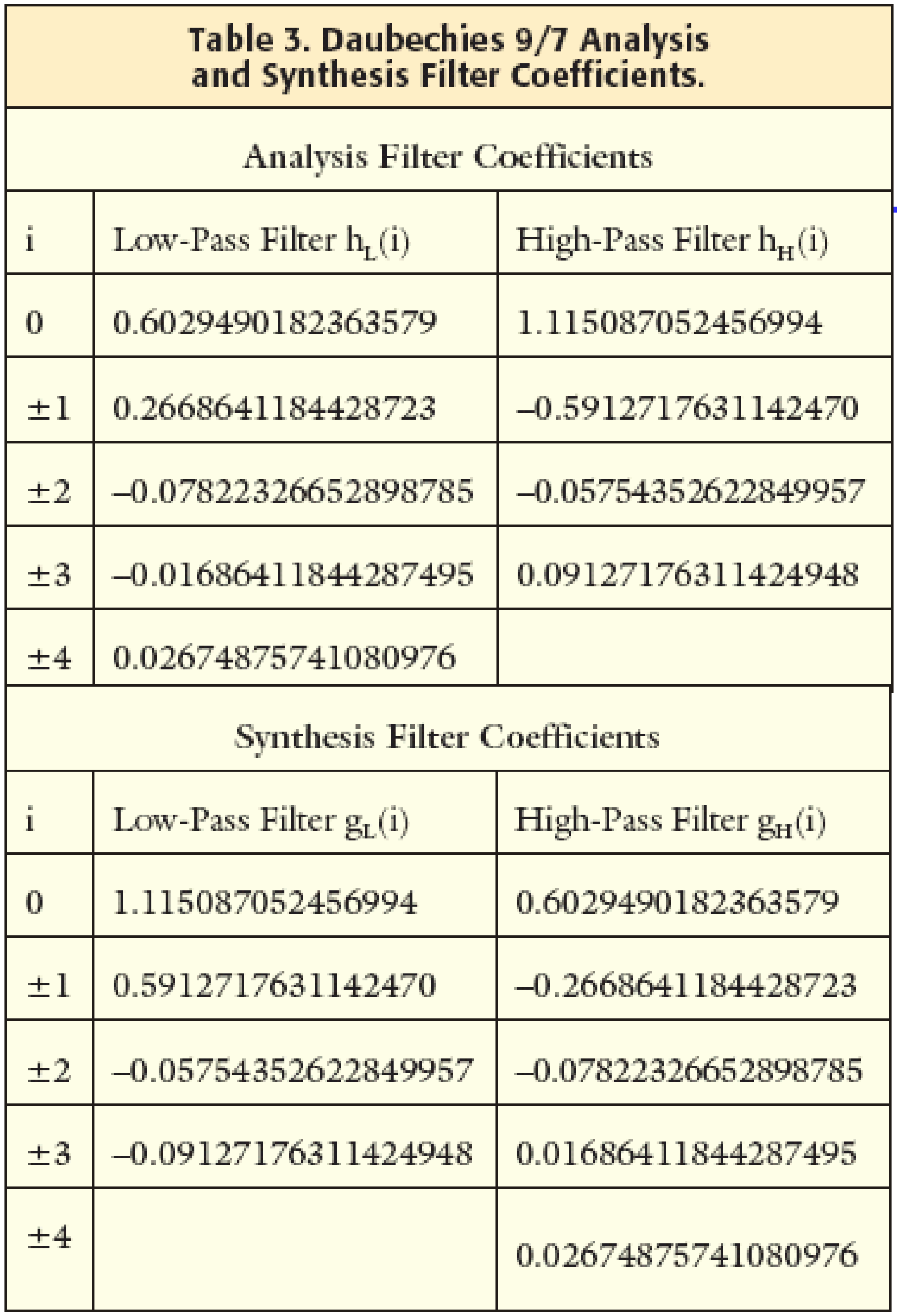


Table 3. Daubechies 9/7 Analysis and Synthesis Filter Coefficients.

Analysis Filter Coefficients		
i	Low-Pass Filter $h_L(i)$	High-Pass Filter $h_H(i)$
0	0.6029490182363579	1.115087052456994
±1	0.2668641184428723	–0.5912717631142470
±2	–0.07822326652898785	–0.05754352622849957
±3	–0.01686411844287495	0.09127176311424948
±4	0.02674875741080976	

Synthesis Filter Coefficients		
i	Low-Pass Filter $g_L(i)$	High-Pass Filter $g_H(i)$
0	1.115087052456994	0.6029490182363579
±1	0.5912717631142470	–0.2668641184428723
±2	–0.05754352622849957	–0.07822326652898785
±3	–0.09127176311424948	0.01686411844287495
±4		0.02674875741080976

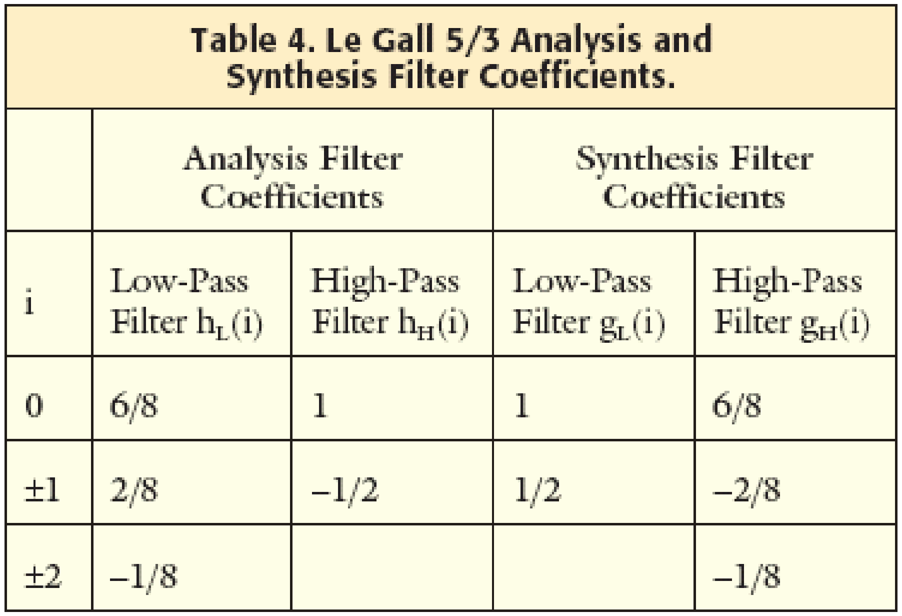


Table 4. Le Gall 5/3 Analysis and Synthesis Filter Coefficients.

	Analysis Filter Coefficients		Synthesis Filter Coefficients	
i	Low-Pass Filter $h_L(i)$	High-Pass Filter $h_H(i)$	Low-Pass Filter $g_L(i)$	High-Pass Filter $g_H(i)$
0	6/8	1	1	6/8
±1	2/8	–1/2	1/2	–2/8
±2	–1/8			–1/8

Comparison of Different Filters

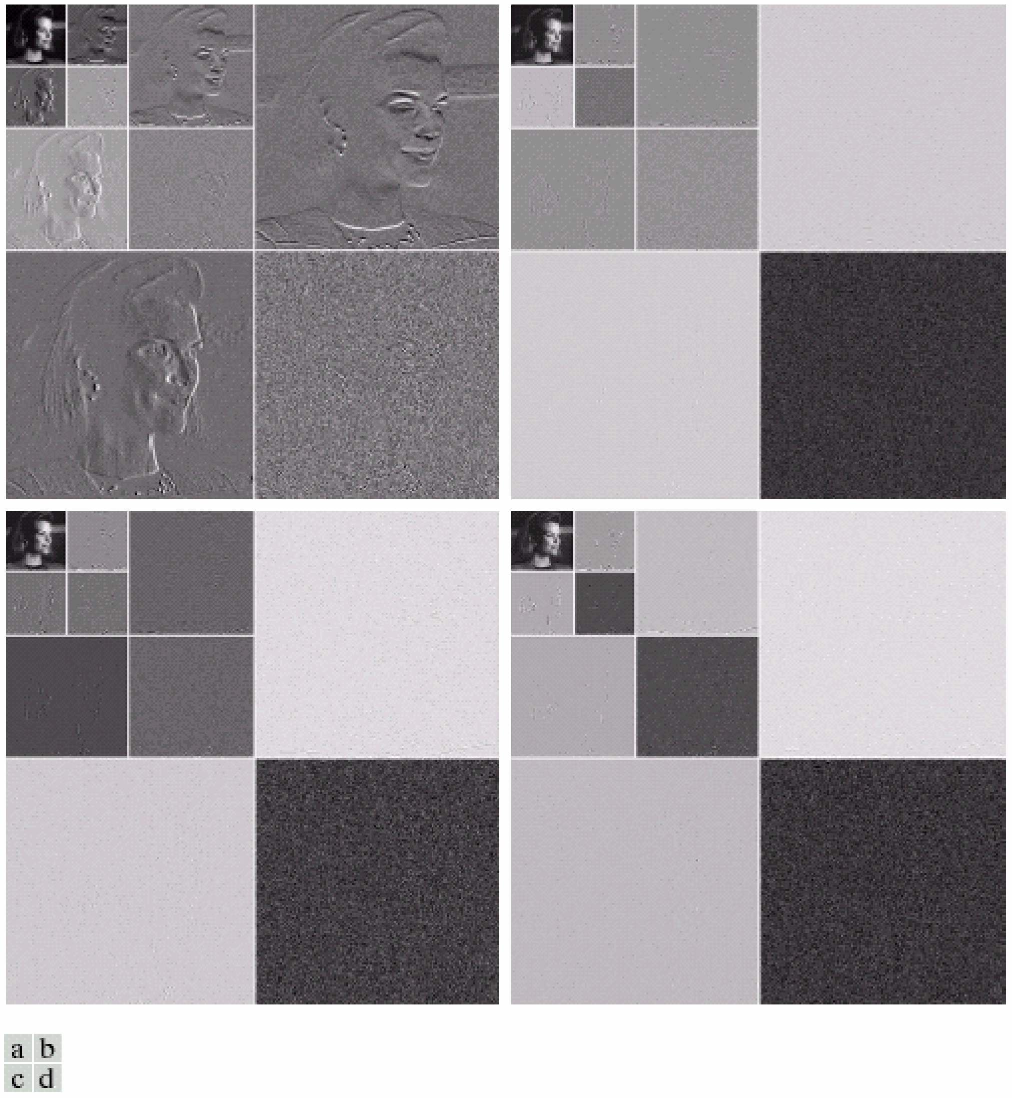
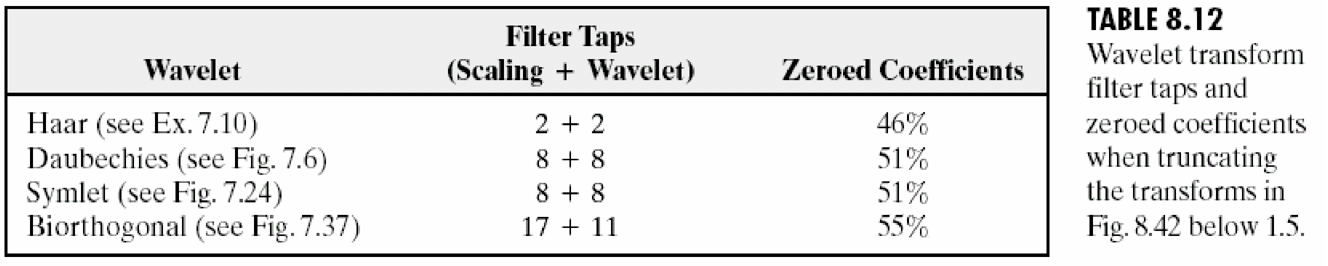


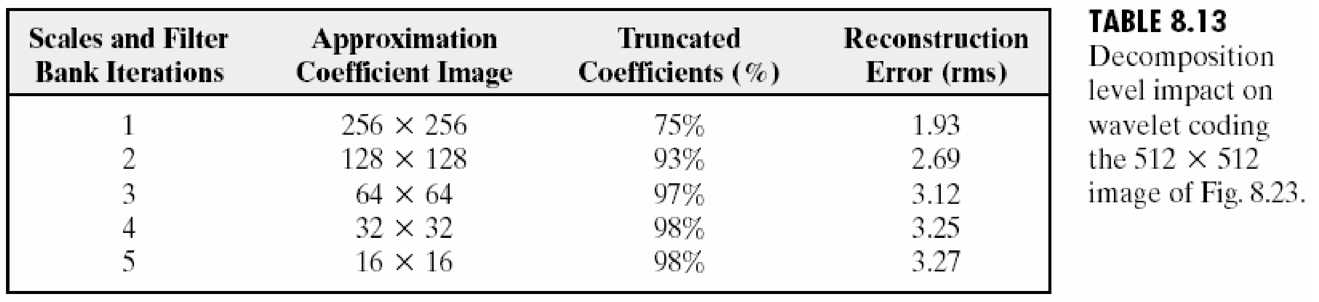
FIGURE 8.42 Wavelet transforms of Fig. 8.23 with respect to (a) Haar wavelets, (b) Daubechies wavelets, (c) symlets, and (d) Cohen-Daubechies-Feauveau biorthogonal wavelets.

Impact of Filters and Decomposition Levels



Wavelet	Filter Taps (Scaling + Wavelet)	Zeroed Coefficients
Haar (see Ex. 7.10)	2 + 2	46%
Daubechies (see Fig. 7.6)	8 + 8	51%
Symlet (see Fig. 7.24)	8 + 8	51%
Biorthogonal (see Fig. 7.37)	17 + 11	55%

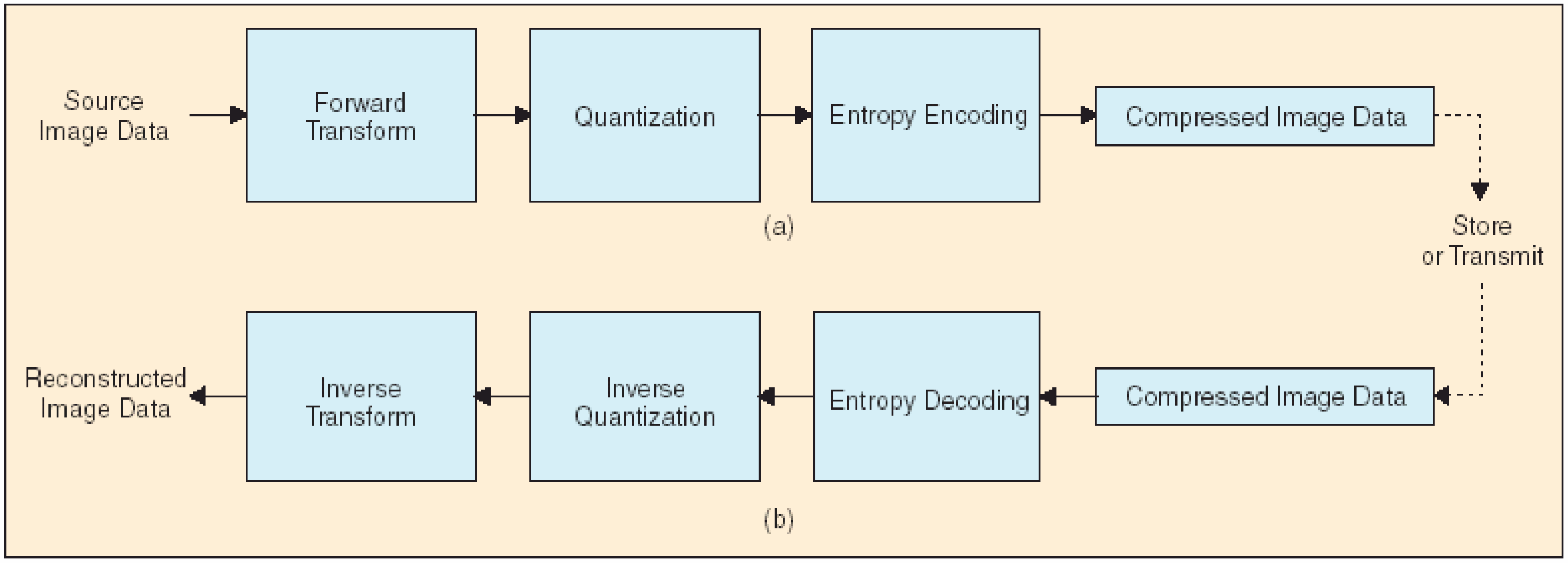
TABLE 8.12 Wavelet transform filter taps and zeroed coefficients when truncating the transforms in Fig. 8.42 below 1.5.



Scales and Filter Bank Iterations	Approximation Coefficient Image	Truncated Coefficients (%)	Reconstruction Error (rms)
1	256×256	75%	1.93
2	128×128	93%	2.69
3	64×64	97%	3.12
4	32×32	98%	3.25
5	16×16	98%	3.27

TABLE 8.13 Decomposition level impact on wavelet coding the 512×512 image of Fig. 8.23.

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.

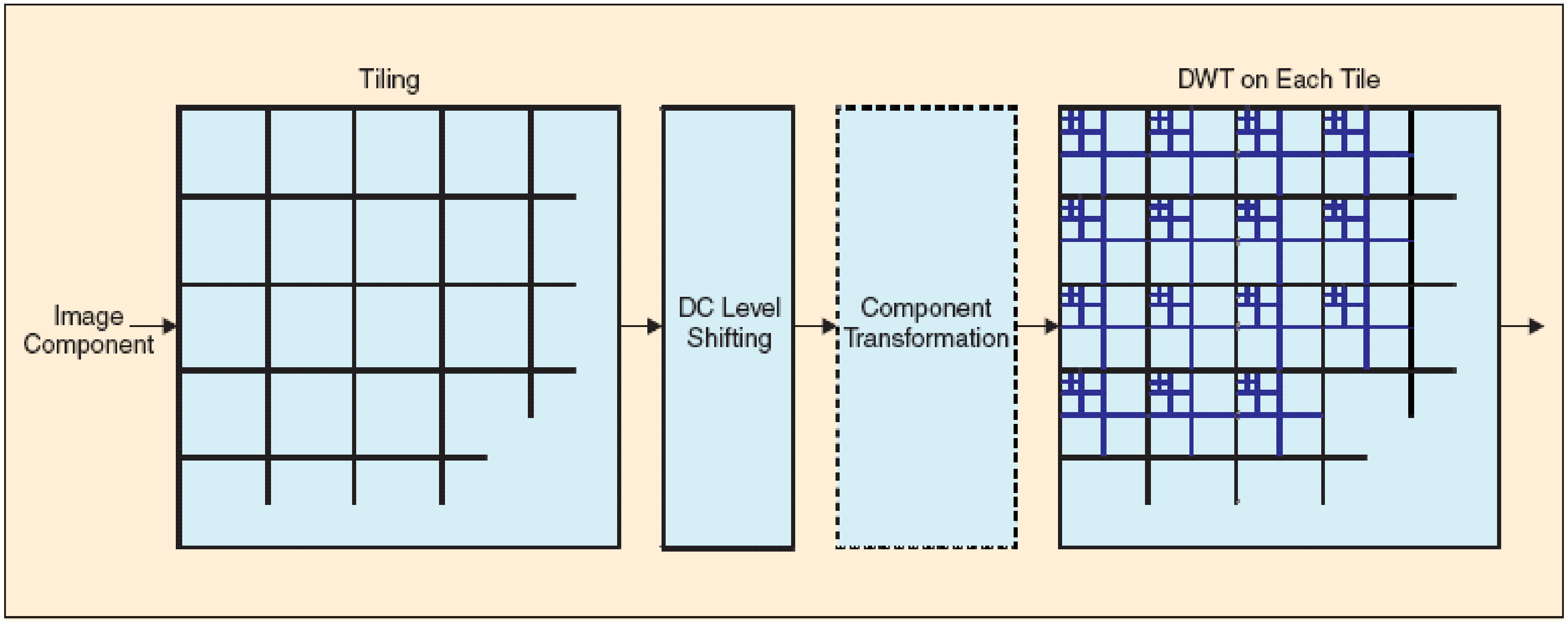
 •Uses sophisticated context-based arithmetic coding

• **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

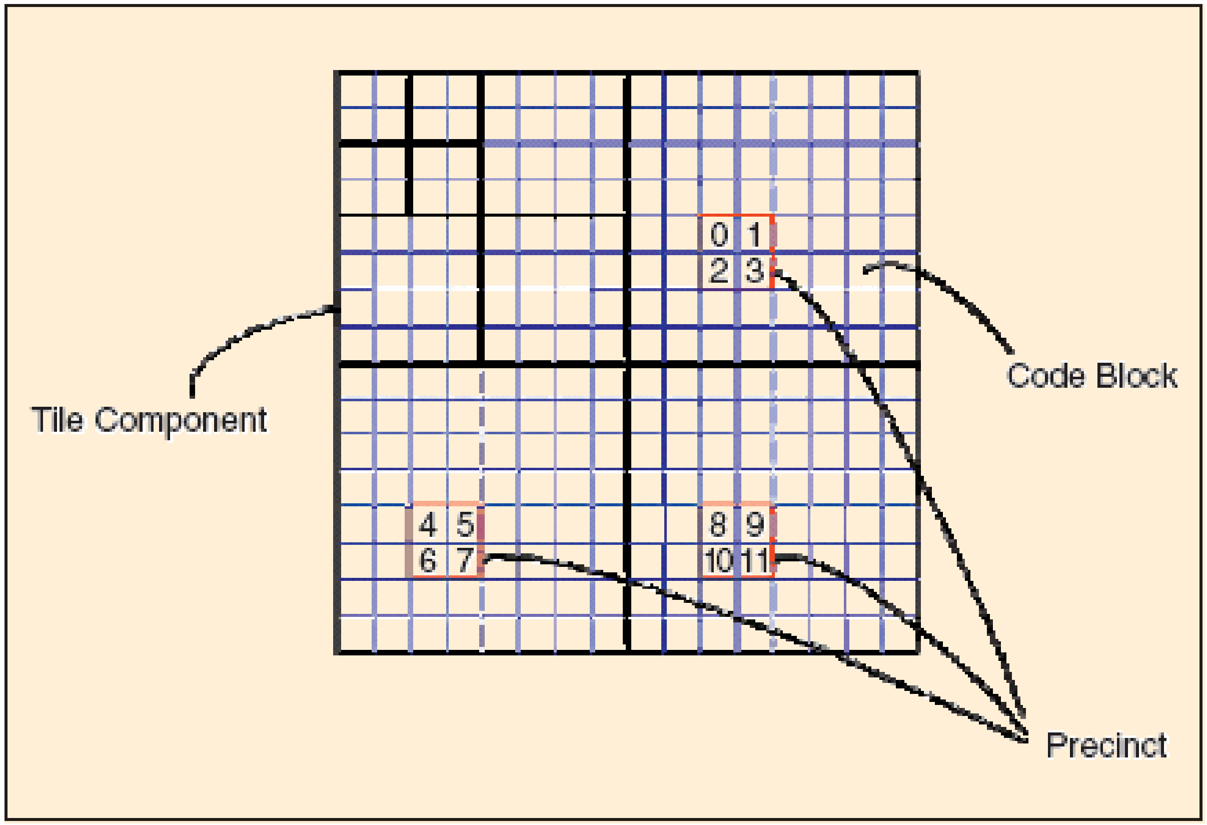
Preprocessing Steps



▲ *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

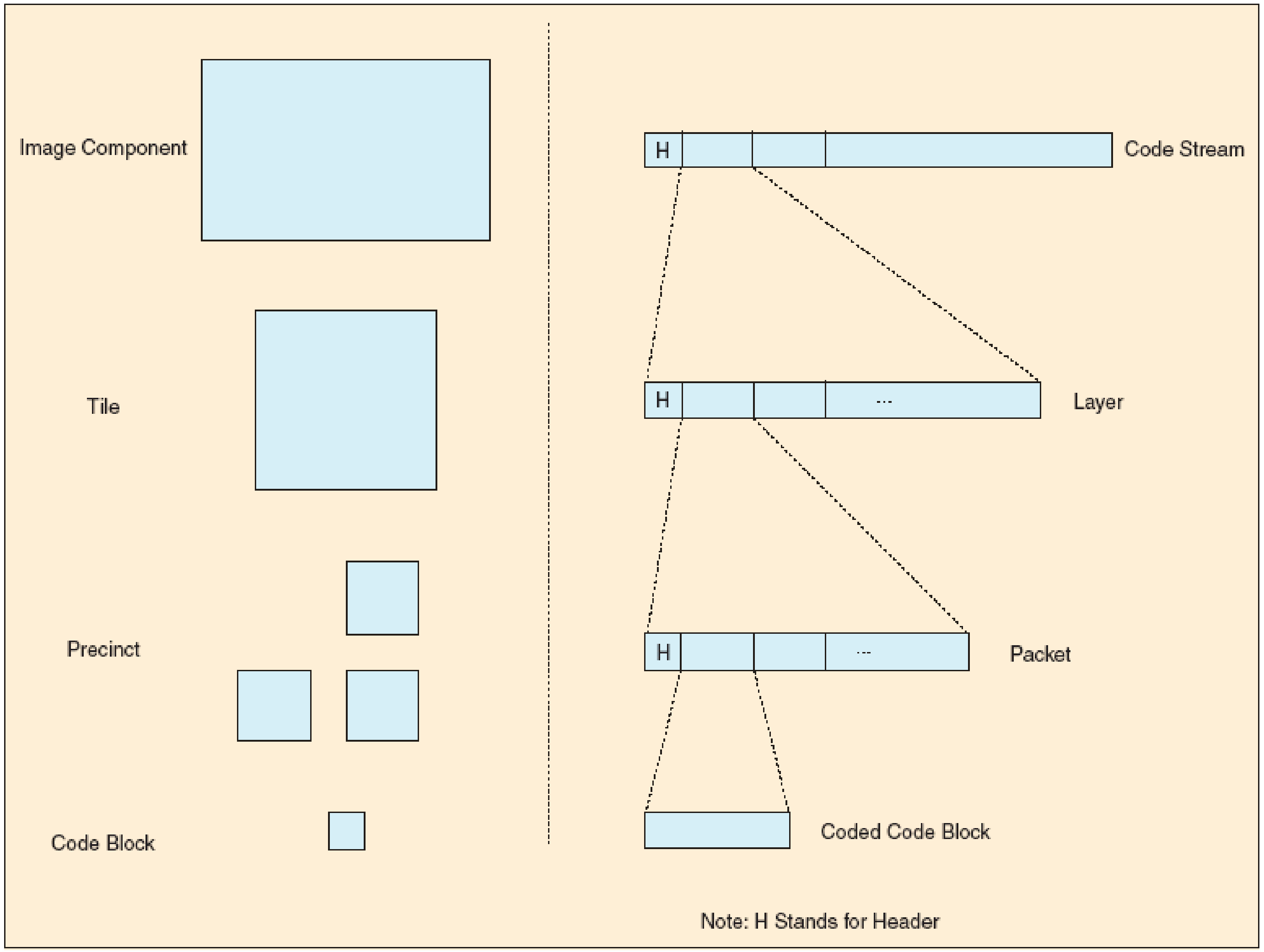
Dividing Each Resolution into Precints



▲ *9. Partition of a tile component into code blocks and precincts.*

- Each precint is divided into many code blocks, each coded independently.
- Bits for all code blocks in the same precint are put into one packet.

Scalable Bit Stream Formation



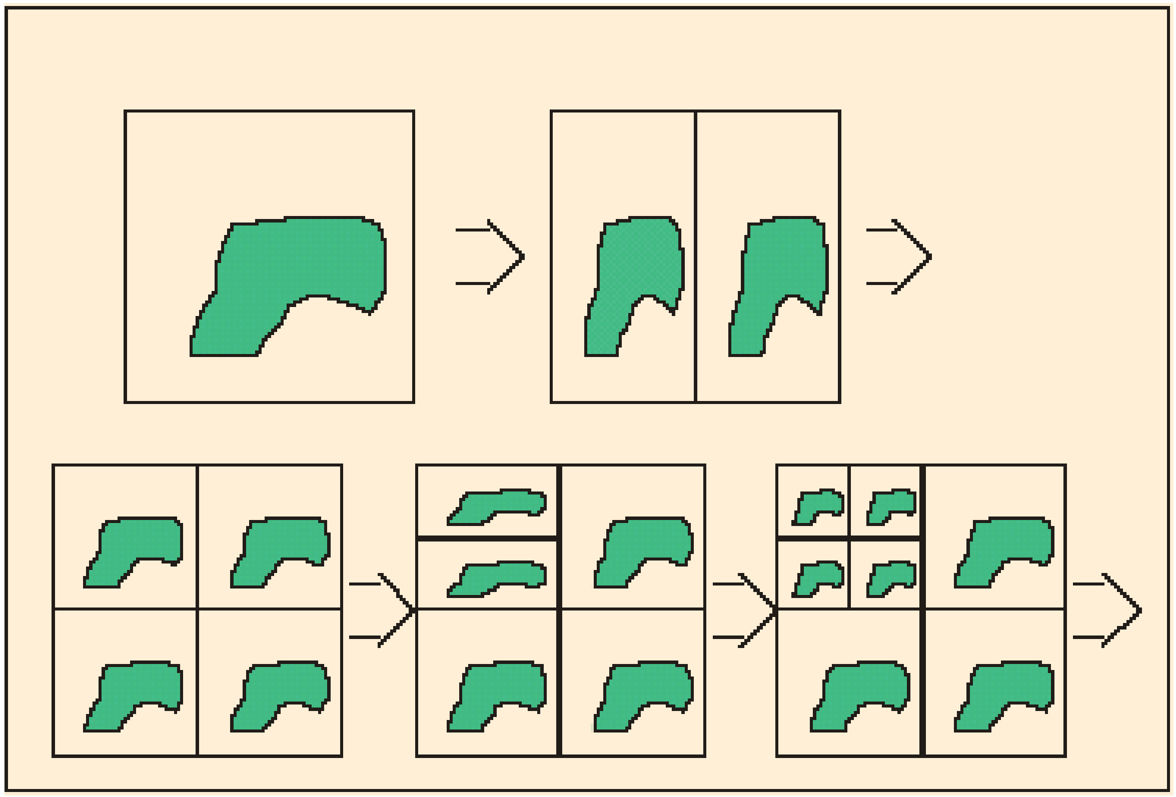
▲ *11. Conceptual correspondence between the spatial and the bit stream representations.*

Coding Steps for a Code Block

- The bit planes of each code block are coded sequentially, from the most significant to the least significant
- Each bit plane is coded in three passes
  - Significance propagation: code location of insignificant bits with significant neighbors
  - Magnitude refinement: code current bit plane of coefficients which become significant in previous bit planes
  - Clean up: code location of insignificant bits with insignificant neighbors
- Each pass is coded using Context-Based Arithmetic Coding
  - The bit of a current coefficient depends on the bits of its neighboring coefficients (context)
  - The current bit is coded based on the conditional probability of this bit given its context

Region of Interests

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

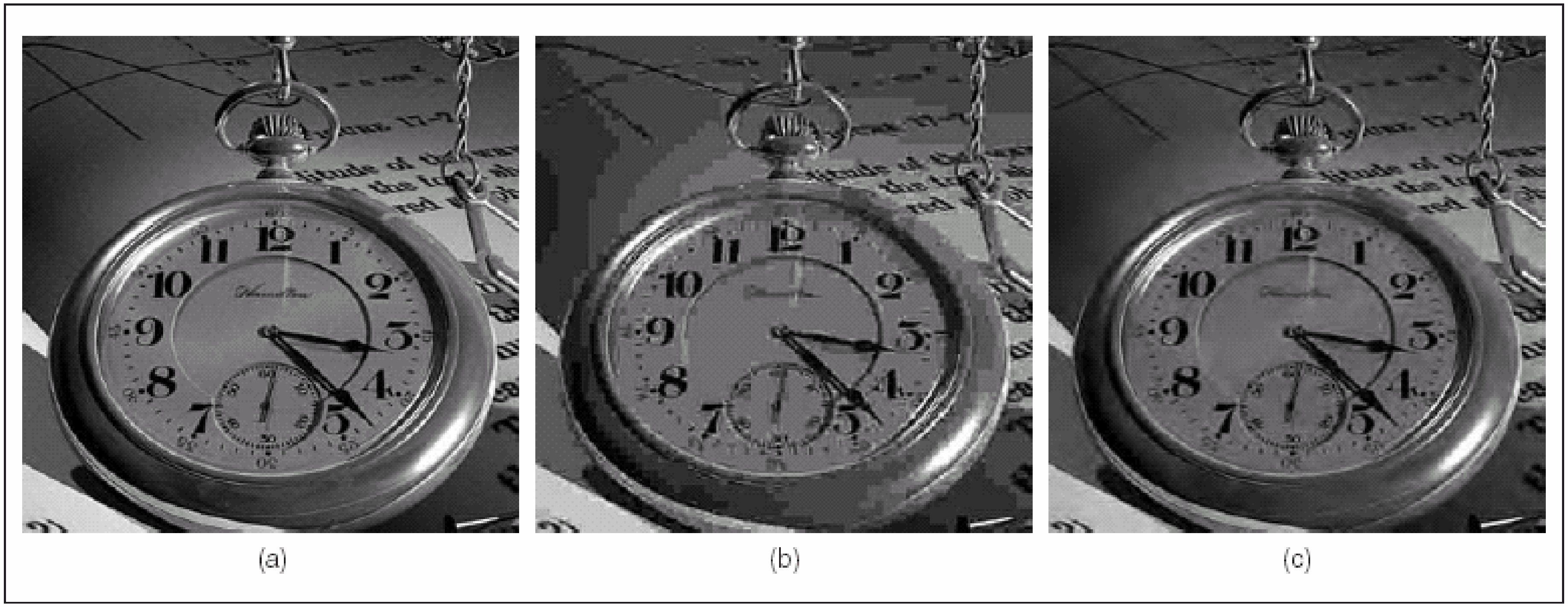


▲ *13. Wavelet domain ROI mask generation.*

Error Resilience

- By adding resynchronization codewords at the beginning of each packet, transmission errors in one packet will not affect following received packets
- The context model for each coding pass in a codeblock can be reset to enhance error resilience
- Packet size and codeblock size and context model reset periods can control tradeoff between coding efficiency and error resilience

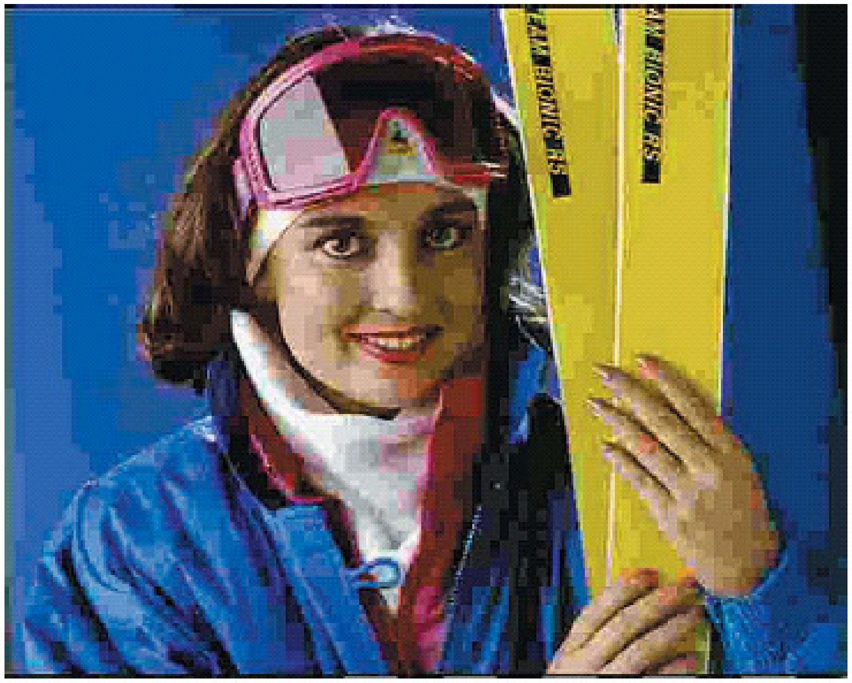
Coding Results: JPEG vs. JPEG2K



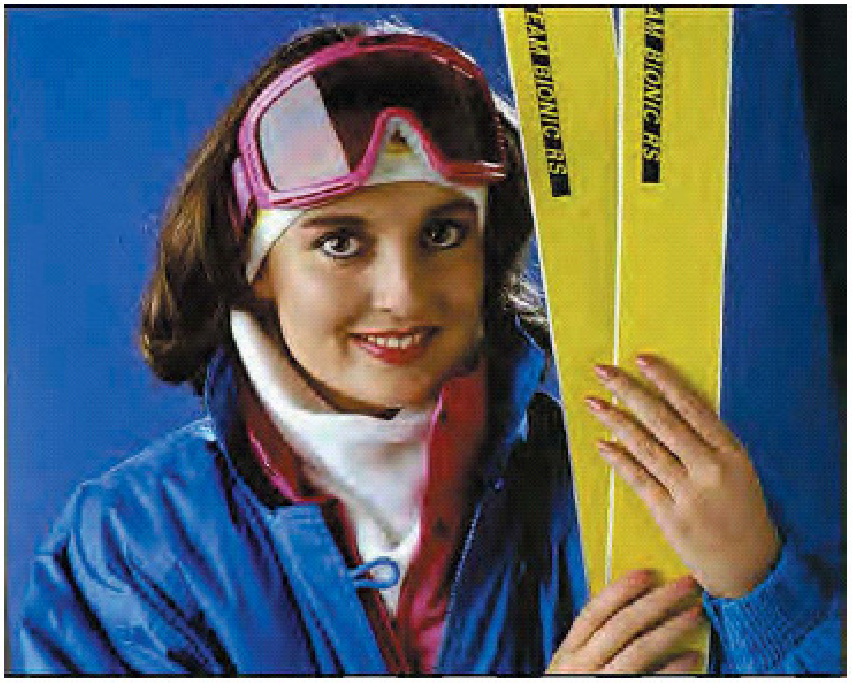
▲ 20. Image "watch" of size 512 × 512 (courtesy of Kevin Odhner): (a) original, and reconstructed after compression at 0.2 b/p by means of (b) JPEG and (c) JPEG 2000.

From [skodras01]

Another Example



(a)

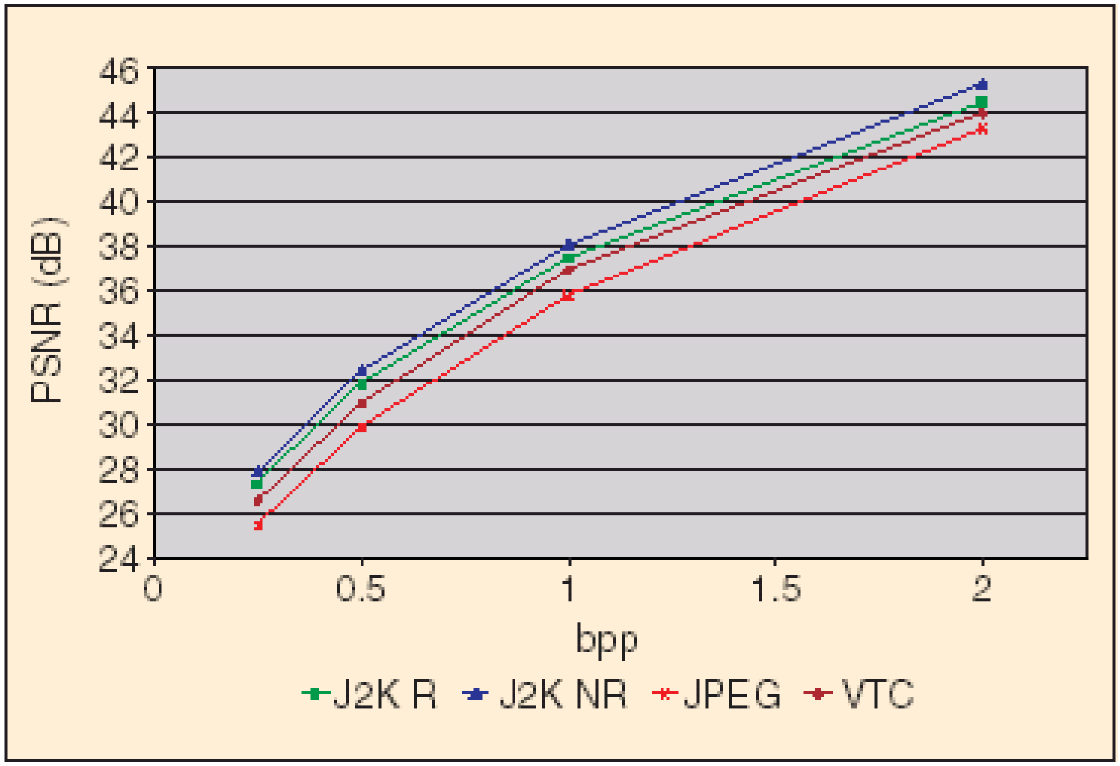


(b)

▲ *21. Reconstructed image "ski" after compression at 0.25 b/p by means of (a) JPEG and (b) JPEG 2000.*

From [skodras01]

JPEG2000 vs. JPEG: Coding Efficiency



▲ 19. *PSNR results for the lossy compression of a natural image by means of different compression standards.*

From [skodras01]

J2K R: Using reversible wavelet filters; J2K NR: Using non-reversible filter; VTC: Visual texture coding for MPEG-4 video

JPEG Pros and Cons

- Pros
  - Low complexity
  - Memory efficient
  - Reasonable coding efficiency
- Cons
  - Single resolution
  - Single quality
  - No target bit rate
  - Blocking artifacts at low bit rate
  - No lossless capability
  - Poor error resilience
  - No tiling
  - No regions of interest

JPEG2000 Features

- Improved coding efficiency
- Full quality scalability
  - From lossless to lossy at different bit rate
- Spatial scalability
- Improved error resilience
- Tiling
- Region of interests
- More demanding in memory and computation time

Why do we want scalability

- The same image may be accessed by users with different access links or different display capability
  - High resolution monitor through High speed Corporate Intranet
  - Small portable device through Wireless modem
- Non-scalable:
  - Have different versions for each desirable bit rate and image size
- Scalable
  - A single bit stream that can be accessed and decoded partially

What is Scalability?

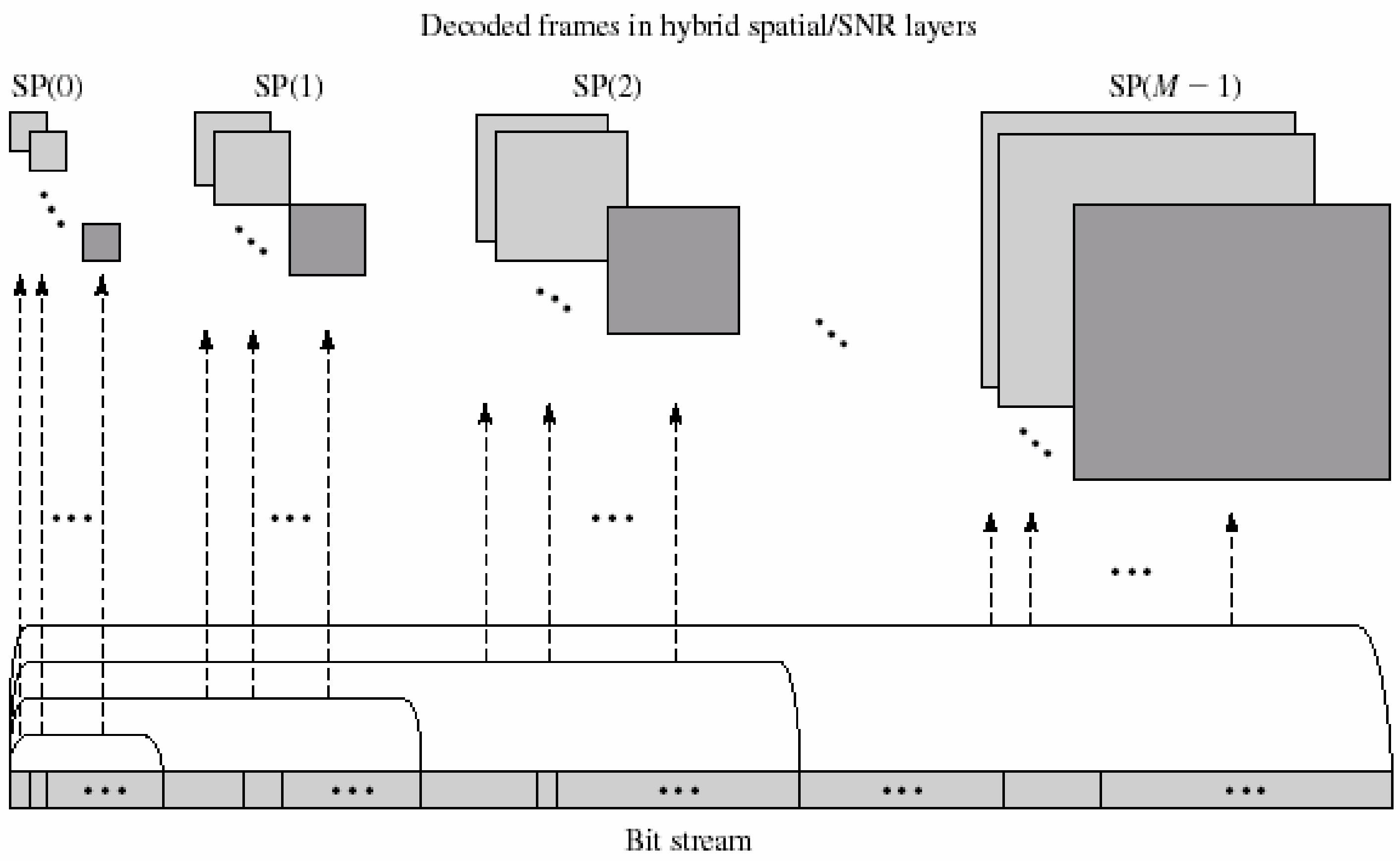
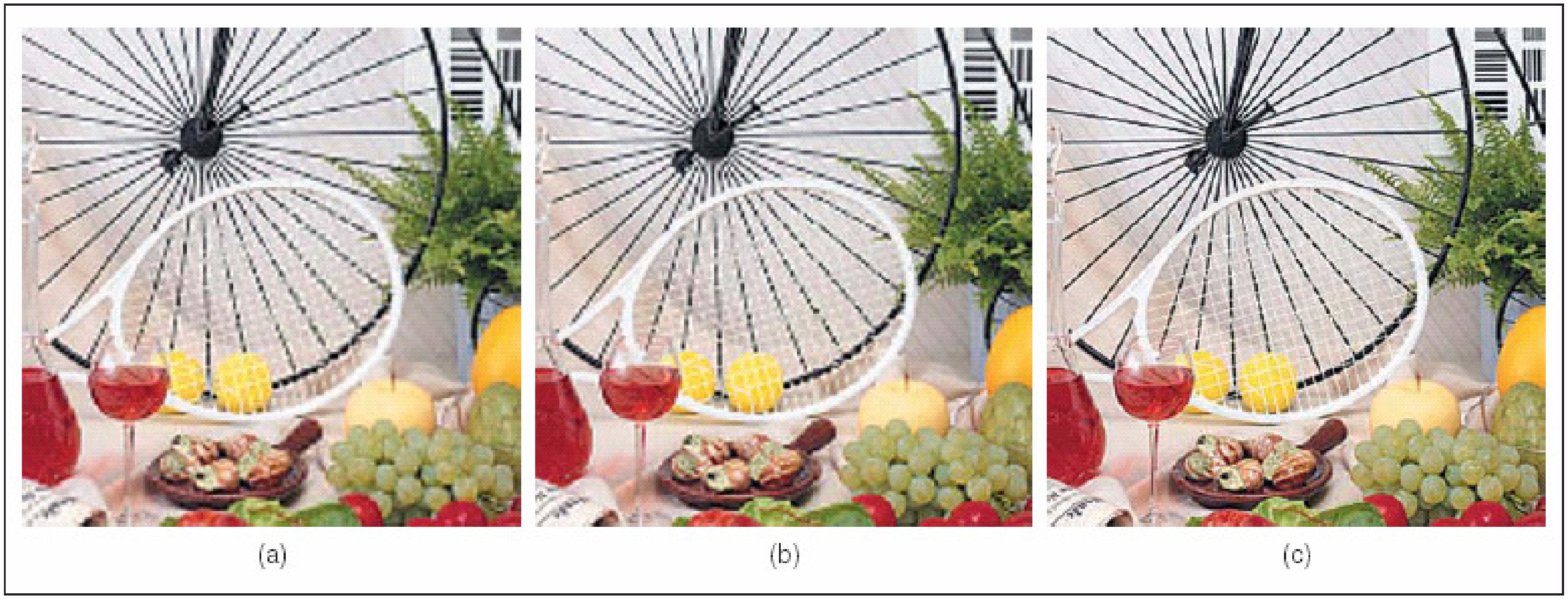


Figure 11.7 $N \times M$ layers of combined spatial/quality scalability. Reprinted from I. Sodagar, H.-J. Lee, P. Hatrack, and Y.-Q. Zhang, Scalable wavelet coding for synthetic/natural hybrid images, *IEEE Trans. Circuits Syst. for Video Technology* (March 1999), 9:244–54. Copyright 1999 IEEE.

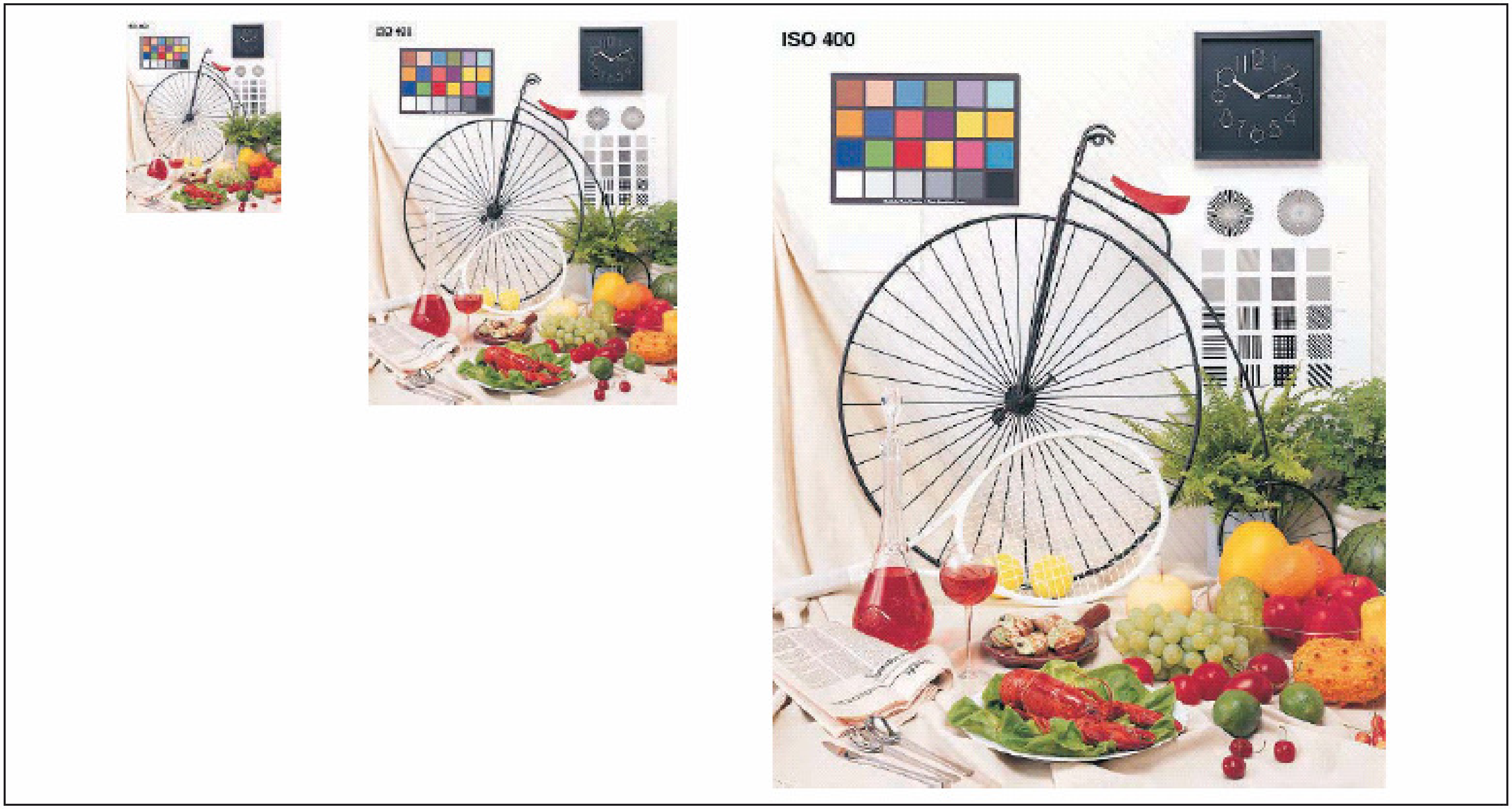
Quality Scalability of JPEG2000



▲ *17. Example of SNR scalability. Part of the decompressed image "bike" at (a) 0.125 b/p, (b) 0.25 b/p, and (c) 0.5 b/p.*

Figures in this slide are extracted from: A. Skodras, C. Christopoulos, T. Ebrahimi, The JPEG2000 Still Image Compression Standard, IEEE Signal Processing Magazine, Sept. 2001.

Spatial Scalability of JPEG2000



▲ *18. Example of the progressive-by-resolution decoding for the color image "bike."*

From [skodras01]

How J2K Achieves Scalability?

- Core: Wavelet transform
  - Yields a multi-resolution representation of an original image
- Still a transform coder
  - Block DCT is replaced by a full frame wavelet transform
    - Also known as subband or wavelet coder
  - Wavelet coefficients are coded bit plane by bit plane
  - Spatial scalability can be achieved by reconstructing from only low resolution (coarse scale) wavelet coefficients
  - Quality scalability can be achieved by decoding only partial bit planes

Homework

1. Given an image of size NxN, where N=2^J, what is the maximum number of levels you can have in an approximation pyramid representation? (The maximum level is reached when the coarsest level has only 1 pixel). What is the total number of pixels in the pyramid (i.e. including pixels at all pyramid levels). How does this number compare with the original number of pixels in the image? Since this number is larger than the original pixel number, what are some of the benefits of using the approximation pyramid? (give some examples).
2. Repeat the above for the prediction residual pyramid.
3. For the given image below, manually compute a 3-level approximation pyramid and corresponding prediction residual pyramid. Use 2x2 averaging for the approximation filter, and use pixel replication for the interpolation filter.

$$F = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

4. For the same image above, manually compute the wavelet transform (with 3-level) using the Haar analysis filters. Use separable processing, i.e., at each level, do row transform first, followed by column transform. Comment on the differences between the pyramids generated in Prob. 3 with the ones generated here.
5. Quantize all the wavelet coefficients created in Prob. 4 by a stepsize of 2. Then reconstruct the 4x4 image from the quantized wavelet coefficients using Haar synthesis filter.
6. Using MATLAB to derive the frequency response of the low-pass and high-pass filters used in the following wavelet transforms: Haar, Daubechies 9/7, and LeGall 5/3. Plot the magnitude response of each and comment on their pros and cons.

Computer Assignment

1. Develop a MATLAB code (or C-code) that implements one-stage subband decomposition of an image using the Haar wavelet. Show the decomposed sub-images.
2. Develop a MATLAB code that reconstructs an image from its one stage subband decomposition using the Haar wavelet. Show the reconstructed image. $\hat{f}$
3. Set each of the four sub-images into all zero and reconstruct using the program in 2. Show the reconstructed images and explain your results (i.e., what is the impact of zeroing each of the sub-image).
4. Quantize the wavelet coefficients using a user-given step size, and then reconstruct the image from quantized coefficients using the program in 2. Show the reconstructed images with two different quantization stepsizes, 4 and 16.
5. (Optional) Develop a MATLAB code that implements a 2-stage subband decomposition and reconstruction using the Haar wavelet. (You should call the basic functions developed in 1 and 2). Show the decomposed images and reconstructed images at different stages.

note you can simplify your programming by making use of the special properties of the Haar decomposition and synthesis filters. Your program does NOT have to be general so that it can use any filters.

References

- A. Skodras, C. Christopoulos, T. Ebrahimi, The JPEG2000 Still Image Compression Standard, IEEE Signal Processing Magazine, vol. 18, pp. 36-58, Sept. 2001. (An excellent tutorial of JPEG2K)
- R. Gonzalez, “Digital Image Processing,” Chap 7 (Wavelet transforms), Chap 8 (Image compression)
- M. Vetterli, “Wavelets, approximation and compression,” *IEEE Signal Processing Mag.*, vol. 18, pp. 59-73, Sept. 2001
- B.E. Usevitch, “A tutorial on modern lossy wavelet image compression: Foundations of JPEG 2000,” *IEEE Signal Processing Mag.*, vol. 18, pp.22-35, Sept. 2001.