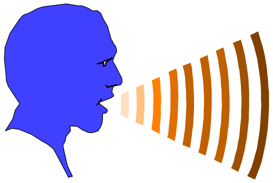


# EE3414
# Multimedia Communication Systems - I

# Image Coding Standards: JPEG and JPEG2000

Yao Wang
Polytechnic University, Brooklyn, NY11201
http://eeweb.poly.edu/~yao

# Outline

- Overview of image coding standards
- JPEG standard
  - Baseline:
    - Run-length coding of DCT coefficients
    - Coding results
  - JPEG for color images
  - JPEG performance
- JPEG2000
  - Scalable coding
  - Wavelet coding

# Image Coding Standards

- G3,G4: facsimile standard (1980)
- JBIG/JBIG2: The next generation facsimile standard (1994 --)
- JPEG: For coding still images or video frames. (1992)
- Lossless JPEG: for medical and archiving applications.
- JPEG2000: An improvement to JPEG, yielding better images at lower bit rate, plus other features (scalability, error resilience)

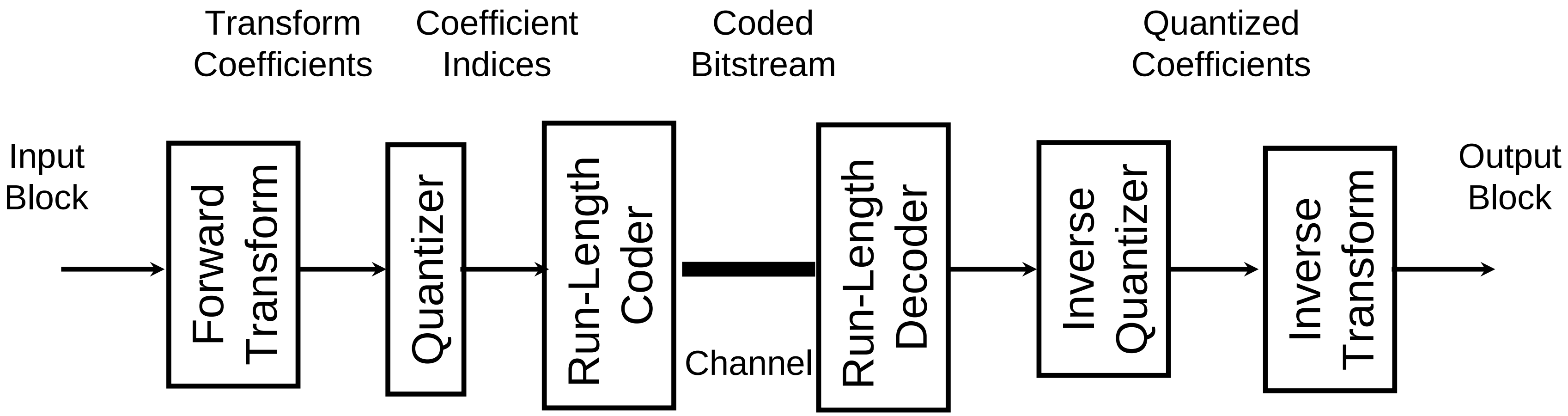
# What is JPEG

- The Joint Photographic Expert Group (JPEG), under both the International Standards Organization (ISO) and the International Telecommunications Union-Telecommunication Sector (ITU-T)
  - www.jpeg.org
- Has published several standards
  - JPEG: lossy coding of continuous tone still images
    - Based on DCT
  - JPEG-LS: lossless and nearlossless coding of continuous tone still images
    - Based on predictive coding and entropy coding
  - JPEG2000: scalable coding of continuous tone still images (from lossy to lossless)
    - Based on wavelet transform

# The 1992 JPEG Standard

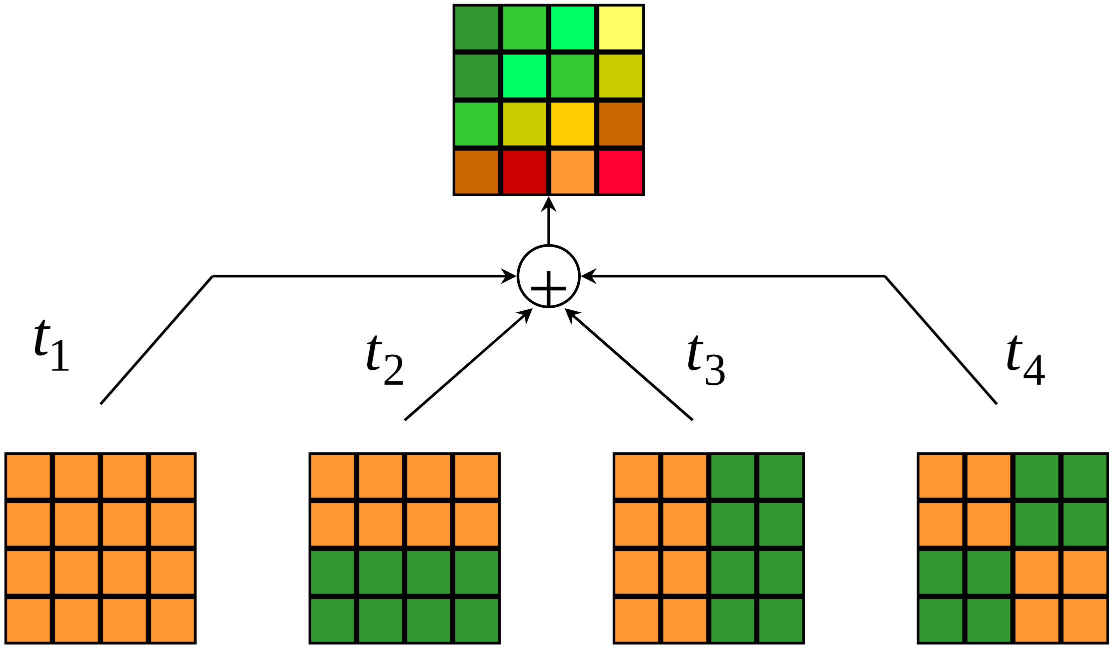
- Contains several modes:
  - Baseline system (what is commonly known as JPEG!): lossy
    - Can handle gray scale or color images, with 8 bits per color component
  - Extended system: can handle higher precision (12 bit) images, providing progressive streams, etc.
  - Lossless version
- Baseline version
  - Each color component is divided into 8x8 blocks
  - For each 8x8 block, three steps are involved:
    - Block DCT
    - Perceptual-based quantization
    - Variable length coding: Runlength and Huffman coding

# Encoder/Decoder Block Diagram

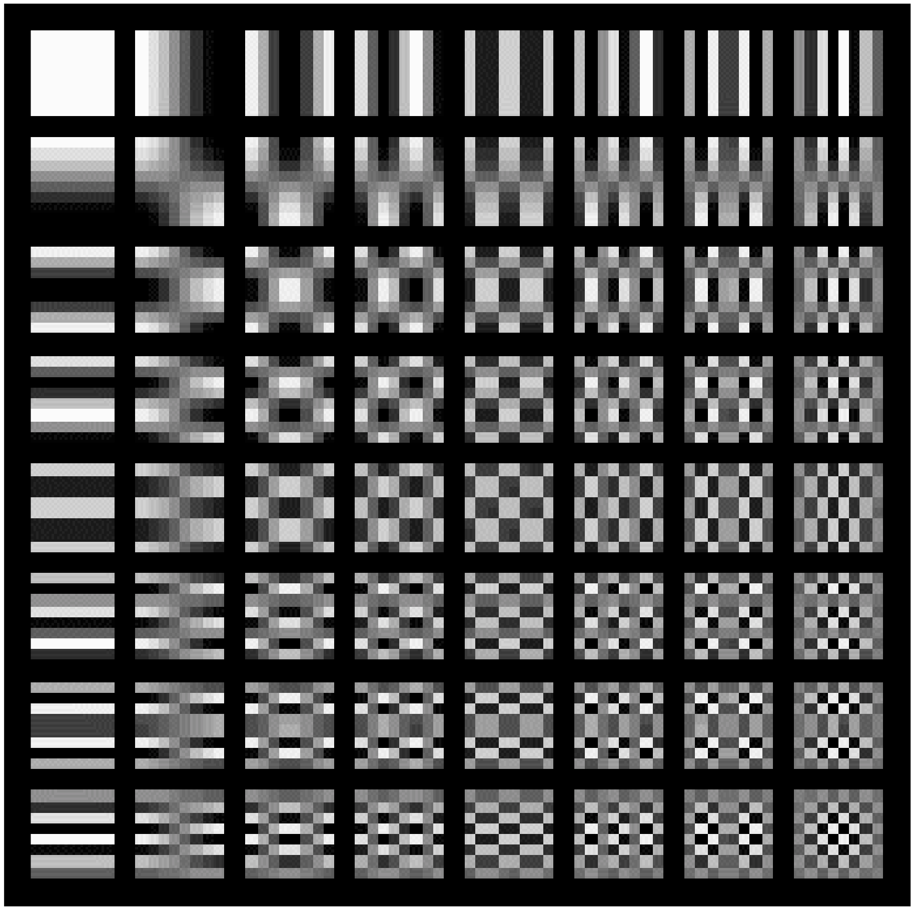


# What does transform do?

- Represent an image as the linear combination of some basis images and specify the linear coefficients.



# Basis Images of 8x8 DCT



Low-Low

High-Low

Low-High

High-High

# DCT on a Real Image Block

```
>>imblock = lena256(128:135,128:135)-128
imblock=
 54    68    71    73    75    73    71    45
   47    52    48    14    20    24    20    -8
   20   -10    -5   -13   -14   -21   -20   -21
  -13   -18   -18   -16   -23   -19   -27   -28
  -24   -22   -22   -26   -24   -33   -30   -23
  -29   -13     3   -24   -10   -42   -41     5
  -16    26    26   -21    12   -31   -40    23
   17    30    50    -5     4    12    10     5
```

```
>>dctblock =dct2(imblock)
dctblock=
 31.0000   51.7034    1.1673  -24.5837  -12.0000  -25.7508   11.9640    23.2873
  113.5766    6.9743  -13.9045   43.2054   -6.0959   35.5931  -13.3692  -13.0005
  195.5804   10.1395   -8.6657   -2.9380  -28.9833   -7.9396    0.8750     9.5585
   35.8733  -24.3038  -15.5776  -20.7924   11.6485  -19.1072   -8.5366  0.5125
   40.7500  -20.5573  -13.6629   17.0615  -14.2500   22.3828   -4.8940  -11.3606
    7.1918  -13.5722   -7.5971  -11.9452   18.2597  -16.2618   -1.4197    -3.5087
   -1.4562  -13.3225   -0.8750    1.3248   10.3817   16.0762    4.4157    1.1041
   -6.7720   -2.8384    4.1187    1.1118   10.5527   -2.7348   -3.2327    1.5799
```

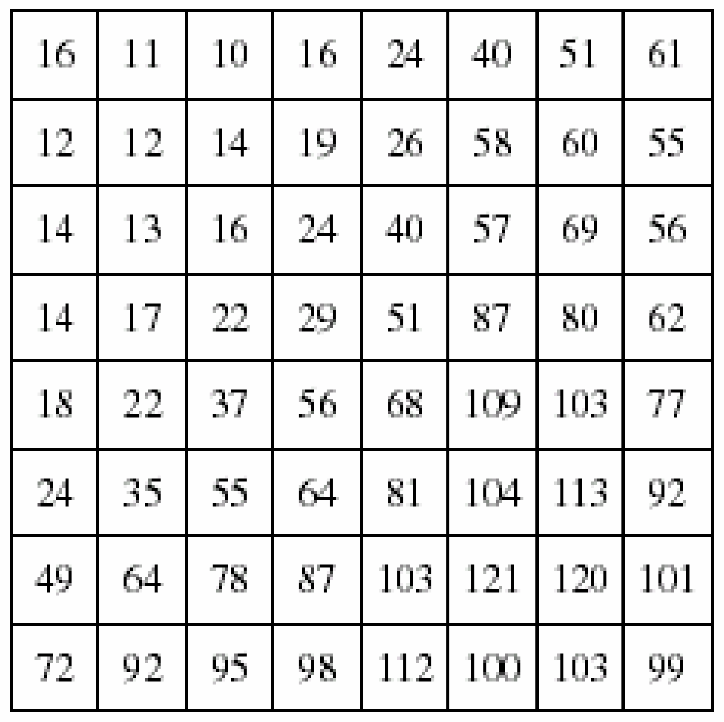
In JPEG, “imblock-128” is done before DCT to shift the mean to zero

# Quantization of DCT Coefficients

- Use uniform quantizer on each coefficient
- Different coefficient is quantized with different step-size (Q):
  - Human eye is more sensitive to low frequency components
  - Low frequency coefficients with a smaller Q
  - High frequency coefficients with a larger Q
  - Specified in a normalization matrix
  - Normalization matrix can then be scaled by a scale factor (QP), which is inversely proportional to a quality factor
    - Quality factor: 100% best quality -> lower quality
  - One can use either the default normalization matrix, or specify an arbitrary matrix
  - For color images, a different normalization matrix can be specified for each color component

# Default Normalization Matrix in JPEG

For Luminance component



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 16 | 11 | 10 | 16 | 24 | 40 | 51 | 61 |
| 12 | 12 | 14 | 19 | 26 | 58 | 60 | 55 |
| 14 | 13 | 16 | 24 | 40 | 57 | 69 | 56 |
| 14 | 17 | 22 | 29 | 51 | 87 | 80 | 62 |
| 18 | 22 | 37 | 56 | 68 | 109 | 103 | 77 |
| 24 | 35 | 55 | 64 | 81 | 104 | 113 | 92 |
| 49 | 64 | 78 | 87 | 103 | 121 | 120 | 101 |
| 72 | 92 | 95 | 98 | 112 | 100 | 103 | 99 |

Actual step size for C(i,j): Q(i,j) = QP*M(i,j)

# Example: Quantized Indices

```
>>dctblock =dct2(imblock)
dctblock=
 31.0000   51.7034    1.1673  -24.5837  -12.0000  -25.7508   11.9640    23.2873
  113.5766    6.9743  -13.9045   43.2054   -6.0959   35.5931  -13.3692  -13.0005
  195.5804   10.1395   -8.6657   -2.9380  -28.9833   -7.9396    0.8750     9.5585
   35.8733  -24.3038  -15.5776  -20.7924   11.6485  -19.1072   -8.5366  0.5125
   40.7500  -20.5573  -13.6629   17.0615  -14.2500   22.3828   -4.8940  -11.3606
    7.1918  -13.5722   -7.5971  -11.9452   18.2597  -16.2618   -1.4197     -3.5087
   -1.4562  -13.3225   -0.8750    1.3248   10.3817   16.0762    4.4157    1.1041
   -6.7720   -2.8384    4.1187    1.1118   10.5527   -2.7348   -3.2327    1.5799
```

```
>>QP=1;
>>QM=Qmatrix*QP;
>>qdct=floor((dctblock+QM/2)./(QM))
qdct =
   2    5    0   -2    0   -1    0    0
   9    1   -1    2    0    1    0    0
  14    1   -1    0   -1    0    0    0
   3   -1   -1   -1    0    0    0    0
   2   -1    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
```

Only 19 coefficients are retained out of 64

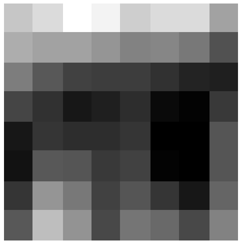
# Example: Quantized Coefficients

```
%dequantized DCT block
>>  iqdct=qdct.*QM
iqdct=
      32      55       0     -32       0      -40   0   0
     108      12     -14      38       0       58   0   0
     196      13     -16       0     -40        0   0   0
      42     -17     -22     -29       0        0   0   0
      36     -22       0       0       0        0   0   0
       0       0       0       0       0        0   0   0
       0       0       0       0       0        0   0   0
       0       0       0       0       0        0   0   0
```

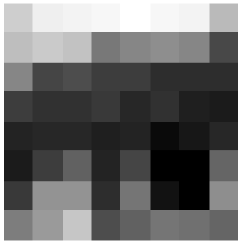
```
Original DCT block
dctblock=
 31.0000   51.7034    1.1673  -24.5837  -12.0000  -25.7508   11.9640    23.2873
  113.5766    6.9743  -13.9045   43.2054   -6.0959   35.5931  -13.3692  -13.0005
  195.5804   10.1395   -8.6657   -2.9380  -28.9833   -7.9396    0.8750     9.5585
   35.8733  -24.3038  -15.5776  -20.7924   11.6485  -19.1072   -8.5366  0.5125
   40.7500  -20.5573  -13.6629   17.0615  -14.2500   22.3828   -4.8940  -11.3606
    7.1918  -13.5722   -7.5971  -11.9452   18.2597  -16.2618   -1.4197     -3.5087
   -1.4562  -13.3225   -0.8750    1.3248   10.3817   16.0762    4.4157    1.1041
   -6.7720   -2.8384    4.1187    1.1118   10.5527   -2.7348   -3.2327    1.5799
```

# Example: Reconstructed Image

```
%reconstructed image block
>>  qimblock=round(idct2(iqdct))
qimblock=
   58   68   85   79   61   68   67   38
   45   38   39   33   22   24   19   -2
   21    2  -11  -12  -13  -19  -24  -27
   -8  -19  -31  -26  -20  -35  -37  -15
  -31  -17  -21  -20  -16  -39  -41    0
  -33    3   -1  -14  -11  -37  -44    1
  -16   32   18  -10    1  -16  -30    8
    3   54   30   -6   16   11   -7   23
```



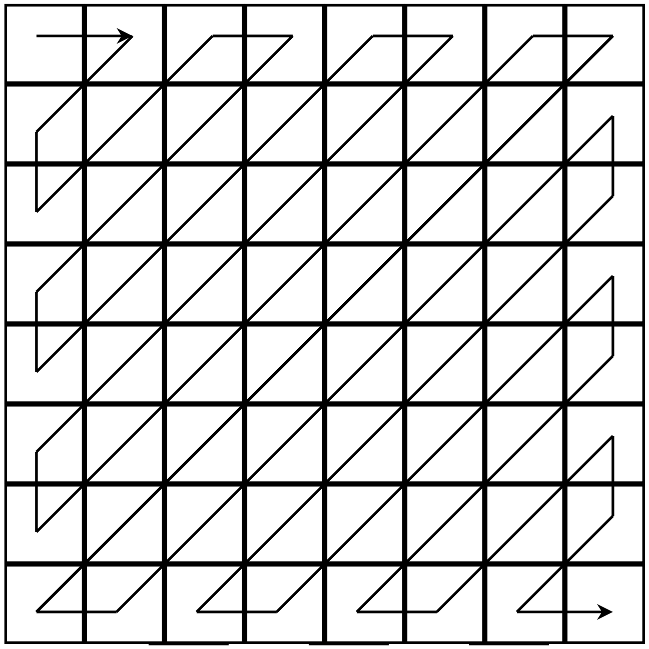
```
Original image block
imblock=
 54   68   71   73   75   73   71   45
   47   52   48   14   20   24   20   -8
   20  -10   -5  -13  -14  -21  -20  -21
  -13  -18  -18  -16  -23  -19  -27  -28
  -24  -22  -22  -26  -24  -33  -30  -23
  -29  -13    3  -24  -10  -42  -41    5
  -16   26   26  -21   12  -31  -40   23
   17   30   50   -5    4   12   10    5
```



# Coding of Quantized DCT Coefficients

- DC coefficient: Predictive coding
  - The DC value of the current block is predicted from that of the previous block, and the error is coded using Huffman coding
- AC Coefficients: Runlength coding
  - Many high frequency AC coefficients are zero after first few low-frequency coefficients
  - Runlength Representation:
    - Ordering coefficients in the zig-zag order
    - Specify how many zeros before a non-zero value
    - Each symbol=(length-of-zero, non-zero-value)
  - Code all possible symbols using Huffman coding
    - More frequently appearing symbols are given shorter codewords
    - For more details on the actual coding table, see Handout (Sec.8.5.3 in [Gonzalez02]
- One can use default Huffman tables or specify its own tables.
- Instead of Huffman coding, arithmetic coding can be used to achieve higher coding efficiency at an added complexity.

# Zig-Zag Ordering of DCT Coefficients



Zig-Zag ordering: converting a 2D matrix into a 1D array, so that the frequency (horizontal+vertical) increases in this order, and the coefficient variance (average of magnitude square) decreases in this order.

# Example

```
qdct =

   2     5     0    -2     0    -1     0     0
     9     1    -1     2     0     1     0     0
   14     1    -1     0    -1     0     0     0
     3    -1    -1    -1     0     0     0     0
     2    -1     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
```

Run-length symbol representation:
{2,(0,5),(0,9),(0,14),(0,1),(1,-2),(0,-1),(0,1),(0,3),(0,2),(0,-1),(0,-1),(0,2),(1,-1),(2,-1), (0,-1), (4,-1),(0,-1),(0,1),EOB}

EOB: End of block, one of the symbol that is assigned a short Huffman codeword

# Coding of DC Symbols

- Example:
  - Current quantized DC index: 2
  - Previous block DC index: 4
  - Prediction error: -2
  - The prediction error is coded in two parts:
    - Which category it belongs to (Table of JPEG Coefficient Coding Categories), and code using a Huffman code (JPEG Default DC Code)
      - DC= -2 is in category “2”, with a codeword “100”
    - Which position it is in that category, using a fixed length code, length=category number
      - “-2” is the number 1 (starting from 0) in category 2, with a fixed length code of “10”.
      - The overall codeword is “10010”

# JPEG Tables for Coding DC

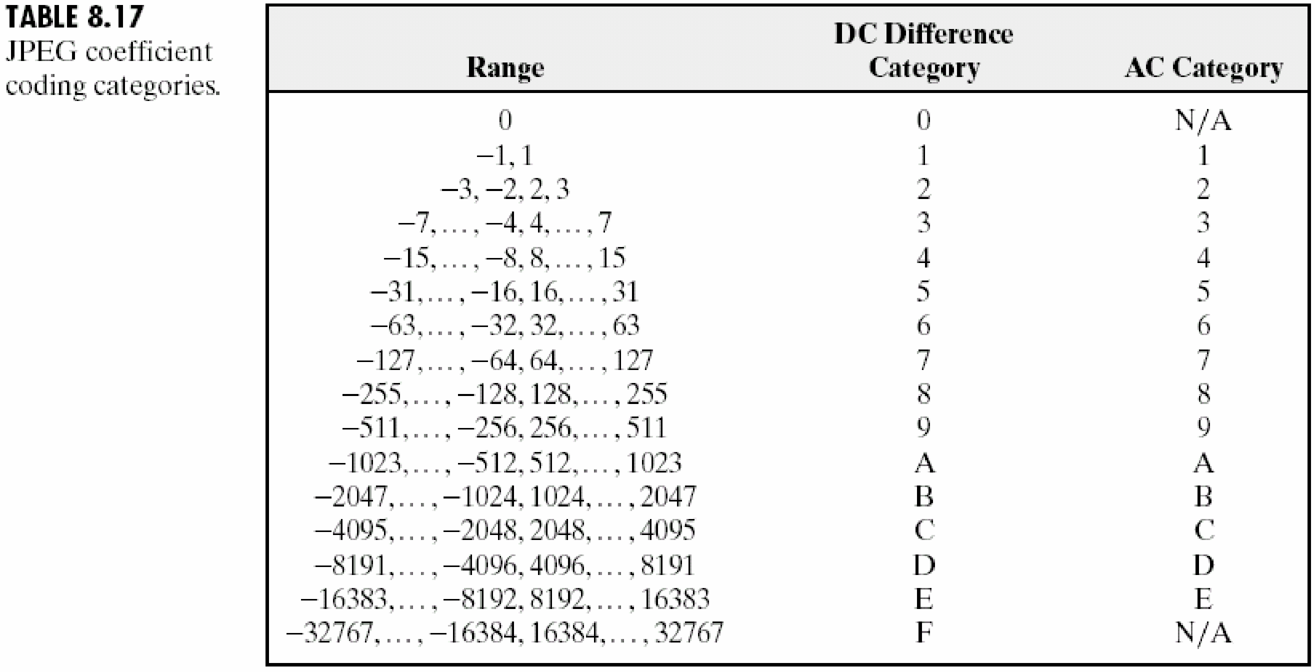


**TABLE 8.17** JPEG coefficient coding categories.

| Range | DC Difference Category | AC Category |
|---|---|---|
| 0 | 0 | N/A |
| −1, 1 | 1 | 1 |
| −3, −2, 2, 3 | 2 | 2 |
| −7, . . . , −4, 4, . . . , 7 | 3 | 3 |
| −15, . . . , −8, 8, . . . , 15 | 4 | 4 |
| −31, . . . , −16, 16, . . . , 31 | 5 | 5 |
| −63, . . . , −32, 32, . . . , 63 | 6 | 6 |
| −127, . . . , −64, 64, . . . , 127 | 7 | 7 |
| −255, . . . , −128, 128, . . . , 255 | 8 | 8 |
| −511, . . . , −256, 256, . . . , 511 | 9 | 9 |
| −1023, . . . , −512, 512, . . . , 1023 | A | A |
| −2047, . . . , −1024, 1024, . . . , 2047 | B | B |
| −4095, . . . , −2048, 2048, . . . , 4095 | C | C |
| −8191, . . . , −4096, 4096, . . . , 8191 | D | D |
| −16383, . . . , −8192, 8192, . . . , 16383 | E | E |
| −32767, . . . , −16384, 16384, . . . , 32767 | F | N/A |

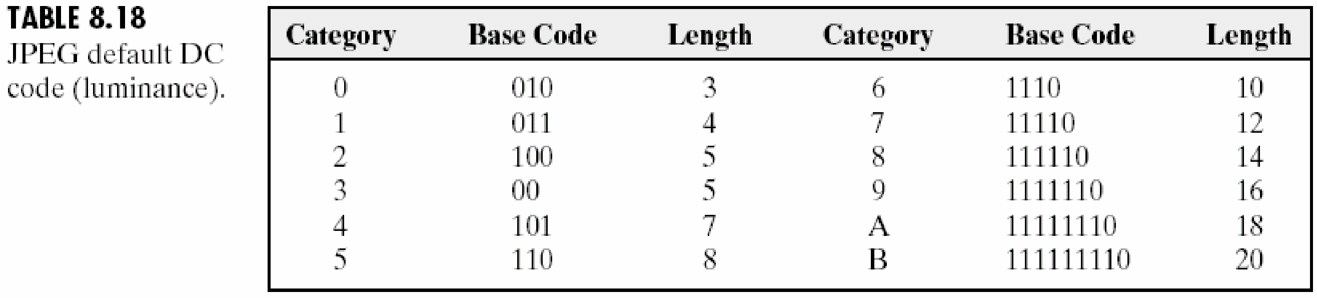


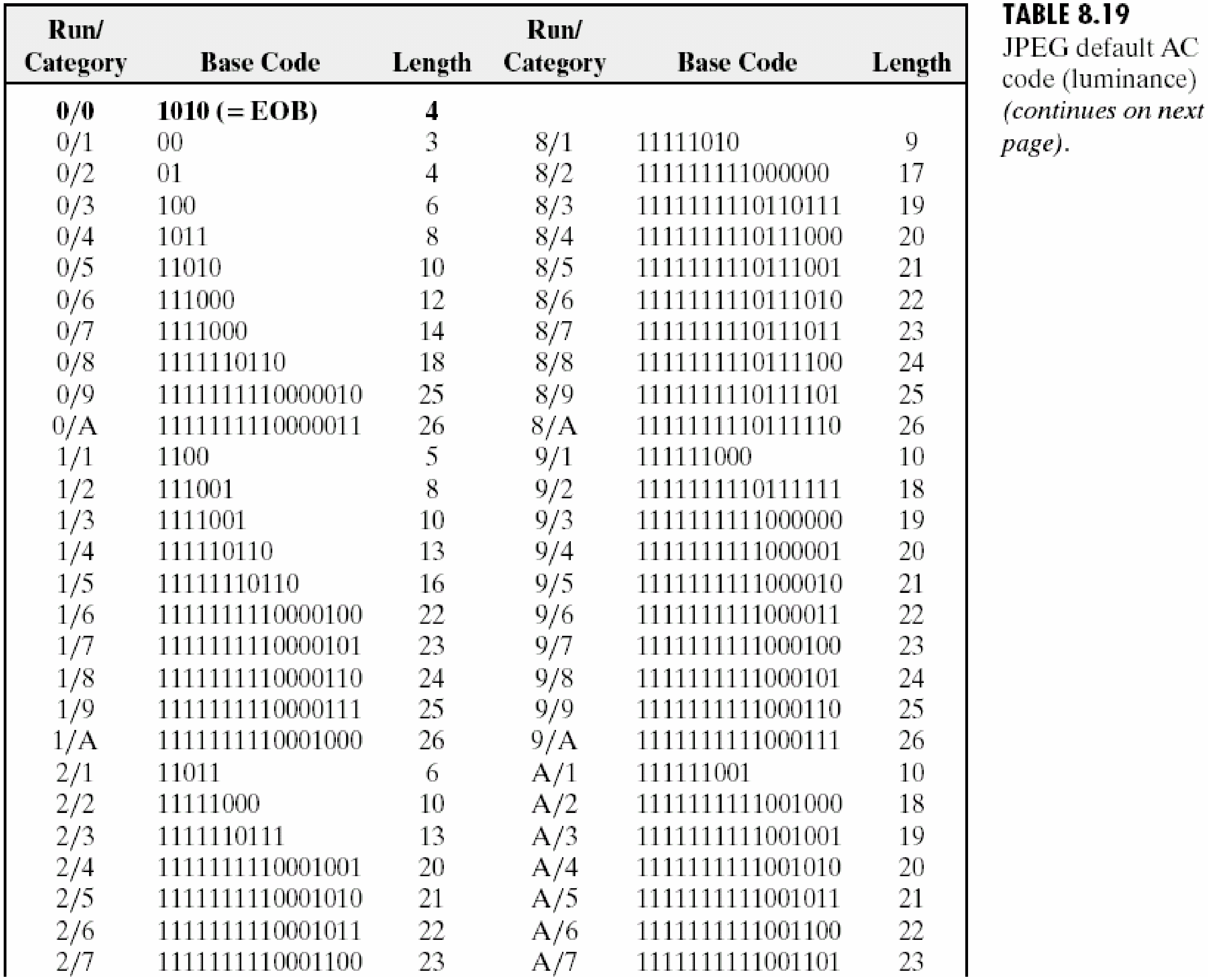
**TABLE 8.18** JPEG default DC code (luminance).

| Category | Base Code | Length | Category | Base Code | Length |
|---|---|---|---|---|---|
| 0 | 010 | 3 | 6 | 1110 | 10 |
| 1 | 011 | 4 | 7 | 11110 | 12 |
| 2 | 100 | 5 | 8 | 111110 | 14 |
| 3 | 00 | 5 | 9 | 1111110 | 16 |
| 4 | 101 | 7 | A | 11111110 | 18 |
| 5 | 110 | 8 | B | 111111110 | 20 |

# Coding of AC Coefficients

- Example:
  - First symbol (0,5)
    - The value ‘5’ is represented in two parts:
    - Which category it belongs to (Table of JPEG Coefficient Coding Categories), and code the “(runlength,category)” using a Huffman code (JPEG Default AC Code)
      - AC=5 is in category “3”,
      - Symbol (0,3) has codeword “100”
    - Which position it is in that category, using a fixed length code, length=category number
      - “5” is the number 5 (starting from 0) in category 3, with a fixed length code of “101”.
      - The overall codeword for (0,5) is “100101”
  - Second symbol (0,9)
    - ‘9’ in category ‘4’, (0,4) has codeword ‘1011’,’9’ is number 9 in category 4 with codeword ‘1001’ -> overall codeword for (0,9) is ‘10111001’
  - ETC

# JPEG Tables for Coding AC (Run,Category) Symbols

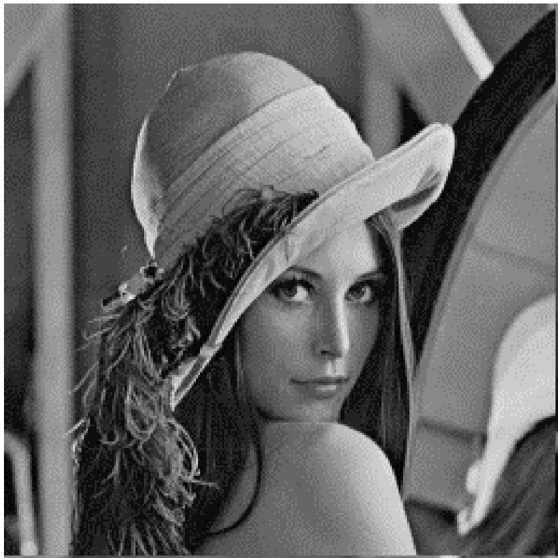


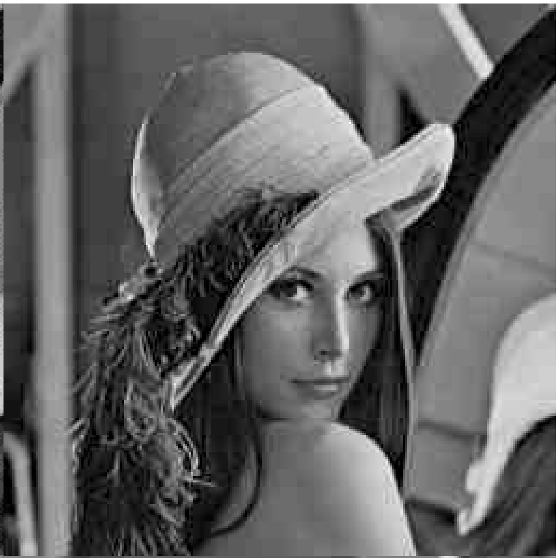
| Run/<br>Category | Base Code | Length | Run/<br>Category | Base Code | Length |
|---|---|---|---|---|---|
| **0/0** | **1010 (= EOB)** | **4** | | | |
| 0/1 | 00 | 3 | 8/1 | 11111010 | 9 |
| 0/2 | 01 | 4 | 8/2 | 111111111000000 | 17 |
| 0/3 | 100 | 6 | 8/3 | 1111111110110111 | 19 |
| 0/4 | 1011 | 8 | 8/4 | 1111111110111000 | 20 |
| 0/5 | 11010 | 10 | 8/5 | 1111111110111001 | 21 |
| 0/6 | 111000 | 12 | 8/6 | 1111111110111010 | 22 |
| 0/7 | 1111000 | 14 | 8/7 | 1111111110111011 | 23 |
| 0/8 | 1111110110 | 18 | 8/8 | 1111111110111100 | 24 |
| 0/9 | 1111111110000010 | 25 | 8/9 | 1111111110111101 | 25 |
| 0/A | 1111111110000011 | 26 | 8/A | 1111111110111110 | 26 |
| 1/1 | 1100 | 5 | 9/1 | 111111000 | 10 |
| 1/2 | 111001 | 8 | 9/2 | 1111111110111111 | 18 |
| 1/3 | 1111001 | 10 | 9/3 | 1111111111000000 | 19 |
| 1/4 | 111110110 | 13 | 9/4 | 1111111111000001 | 20 |
| 1/5 | 11111110110 | 16 | 9/5 | 1111111111000010 | 21 |
| 1/6 | 1111111110000100 | 22 | 9/6 | 1111111111000011 | 22 |
| 1/7 | 1111111110000101 | 23 | 9/7 | 1111111111000100 | 23 |
| 1/8 | 1111111110000110 | 24 | 9/8 | 1111111111000101 | 24 |
| 1/9 | 1111111110000111 | 25 | 9/9 | 1111111111000110 | 25 |
| 1/A | 1111111110001000 | 26 | 9/A | 1111111111000111 | 26 |
| 2/1 | 11011 | 6 | A/1 | 111111001 | 10 |
| 2/2 | 11111000 | 10 | A/2 | 1111111111001000 | 18 |
| 2/3 | 1111110111 | 13 | A/3 | 1111111111001001 | 19 |
| 2/4 | 1111111110001001 | 20 | A/4 | 1111111111001010 | 20 |
| 2/5 | 1111111110001010 | 21 | A/5 | 1111111111001011 | 21 |
| 2/6 | 1111111110001011 | 22 | A/6 | 1111111111001100 | 22 |
| 2/7 | 1111111110001100 | 23 | A/7 | 1111111111001101 | 23 |

**TABLE 8.19**
JPEG default AC code (luminance) *(continues on next page)*.

# JPEG Performance for B/W images

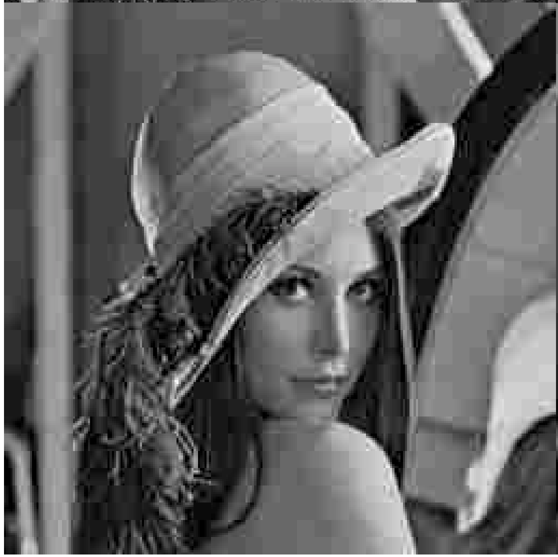
65536 Bytes
8 bpp

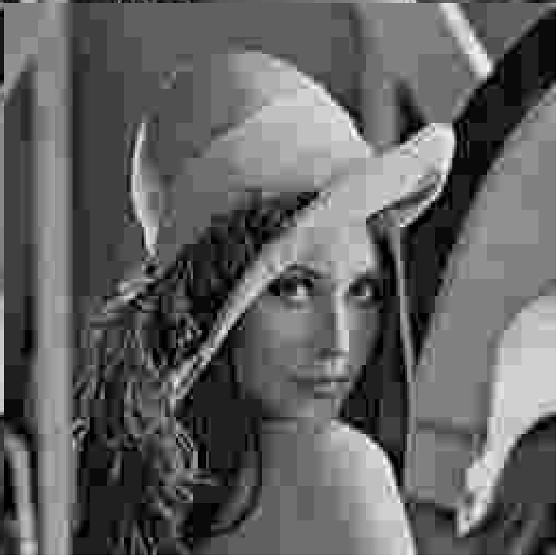




4839 Bytes
0.59 bpp
CR=13.6

3037 Bytes
0.37 bpp
CR=21.6





1818 Bytes
0.22 bpp
CR=36.4

# JPEG for Color Images

- Color images are typically stored in (R,G,B) format
- JPEG standard can be applied to each component separately
  - Does not make use of the correlation between color components
  - Does not make use of the lower sensitivity of the human eye to chrominance samples
- Alternate approach
  - Convert (R,G,B) representation to a YCbCr representation
    - Y: luminance, Cb, Cr: chrominance
  - Down-sample the two chrominance components
    - Because the peak response of the eye to the luminance component occurs at a higher frequency (3-10 cpd) than to the chrominance components (0.1-0.5 cpd).
- JPEG standard can handle an image consisting of many (up to 100) components

# RGB <-> YCbCr Conversion

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

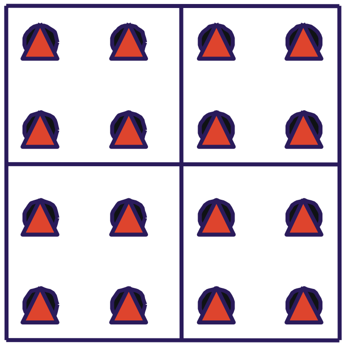
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & -0.001 & 1.402 \\ 1.000 & -0.344 & -0.714 \\ 1.000 & 1.772 & 0.001 \end{bmatrix} \begin{bmatrix} Y \\ C_b - 128 \\ C_r - 128 \end{bmatrix}$$

Note: Cb ~ Y-B, Cr ~ Y-R, are known as color difference signals.

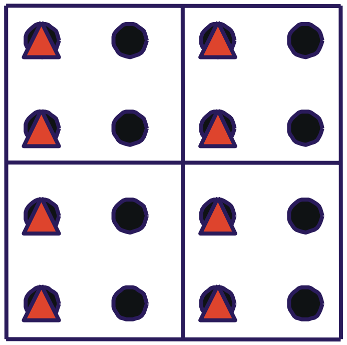
# Example

- What is the (Y,Cb,Cr) value for (R,G,B)=(255,128,64)?
- What is the (R,G,B) value for (Y,Cb,Cr)=(128,25,43)?

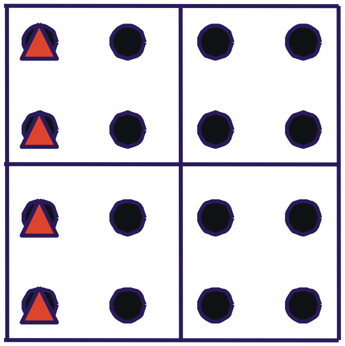
# Chrominance Subsampling



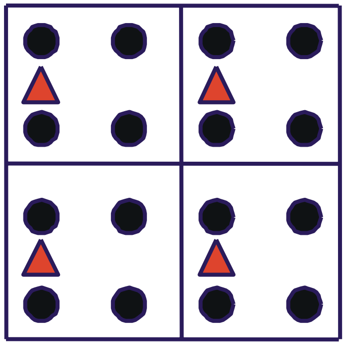
4:4:4
For every 2x2 Y Pixels
4 Cb & 4 Cr Pixel
(No subsampling)



4:2:2
For every 2x2 Y Pixels
2 Cb & 2 Cr Pixel
(Subsampling by 2:1
horizontally only)



4:1:1
For every 4x1 Y Pixels
1 Cb & 1 Cr Pixel
(Subsampling by 4:1
horizontally only)

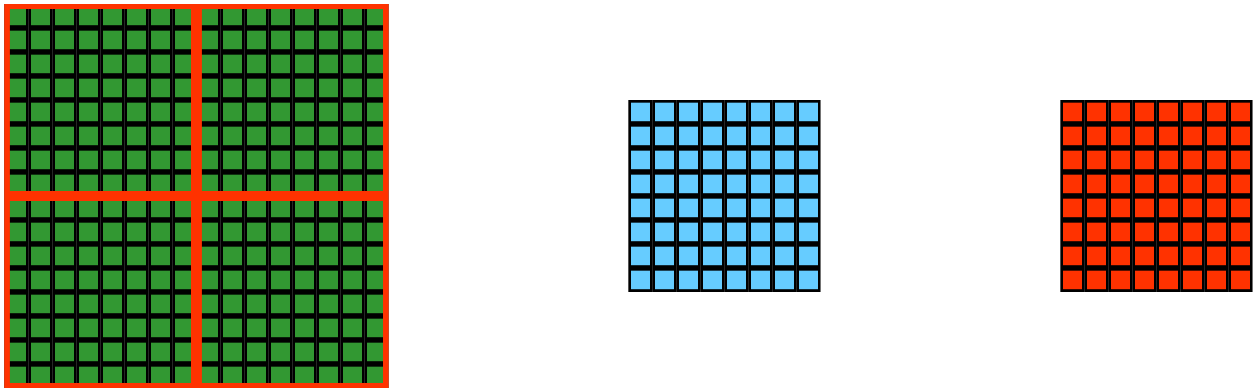


4:2:0
For every 2x2 Y Pixels
1 Cb & 1 Cr Pixel
(Subsampling by 2:1 both
horizontally and vertically)

● Y Pixel ▲ Cb and Cr Pixel

4:2:0 is the most common format

# Coding Unit in JPEG

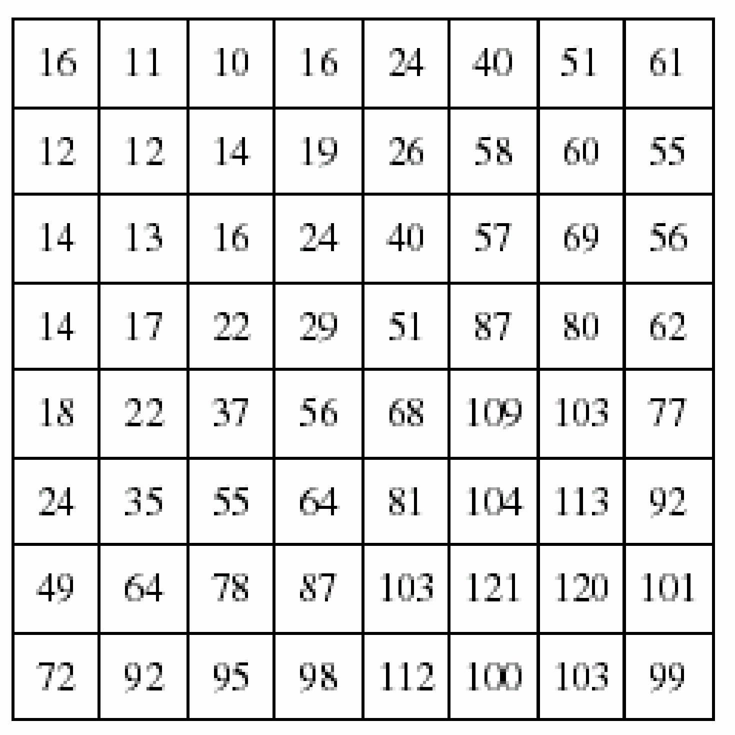


4 8x8 Y blocks 1 8x8 Cb blocks 1 8x8 Cr blocks

Each basic coding unit (called a data unit) is a 8x8 block in any color component. In the interleaved mode, 4 Y blocks and 1 Cb and 1 Cr blocks are processed as a group (called a minimum coding unit or MCU) for a 4:2:0 image.

# Default Quantization Table

For luminance



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 16 | 11 | 10 | 16 | 24 | 40 | 51 | 61 |
| 12 | 12 | 14 | 19 | 26 | 58 | 60 | 55 |
| 14 | 13 | 16 | 24 | 40 | 57 | 69 | 56 |
| 14 | 17 | 22 | 29 | 51 | 87 | 80 | 62 |
| 18 | 22 | 37 | 56 | 68 | 109 | 103 | 77 |
| 24 | 35 | 55 | 64 | 81 | 104 | 113 | 92 |
| 49 | 64 | 78 | 87 | 103 | 121 | 120 | 101 |
| 72 | 92 | 95 | 98 | 112 | 100 | 103 | 99 |

For chrominance

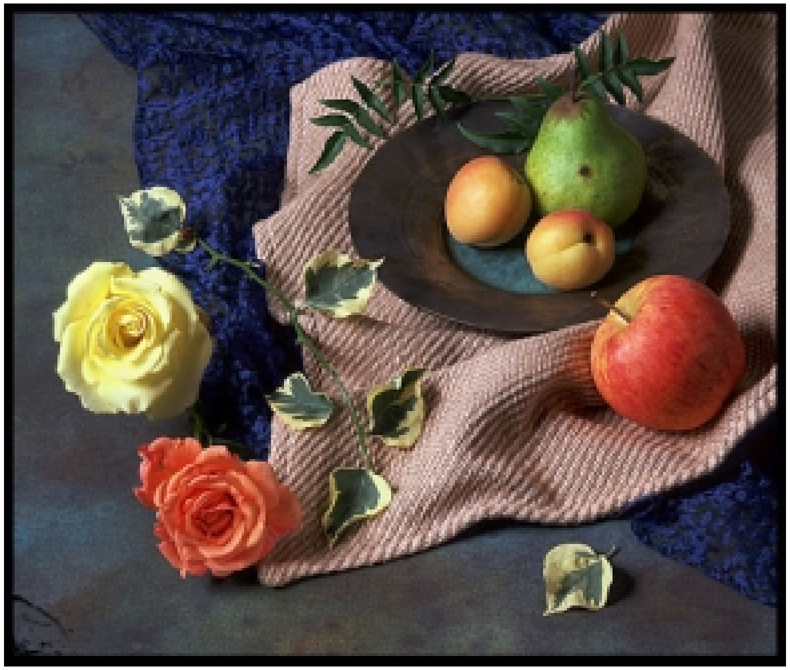
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 17 | 18 | 24 | 47 | 99 | 99 | 99 | 99 |
| 18 | 21 | 26 | 66 | 99 | 99 | 99 | 99 |
| 24 | 26 | 56 | 99 | 99 | 99 | 99 | 99 |
| 47 | 66 | 99 | 99 | 99 | 99 | 99 | 99 |
| 99 | 99 | 99 | 99 | 99 | 99 | 99 | 99 |
| 99 | 99 | 99 | 99 | 99 | 99 | 99 | 99 |
| 99 | 99 | 99 | 99 | 99 | 99 | 99 | 99 |
| 99 | 99 | 99 | 99 | 99 | 99 | 99 | 99 |

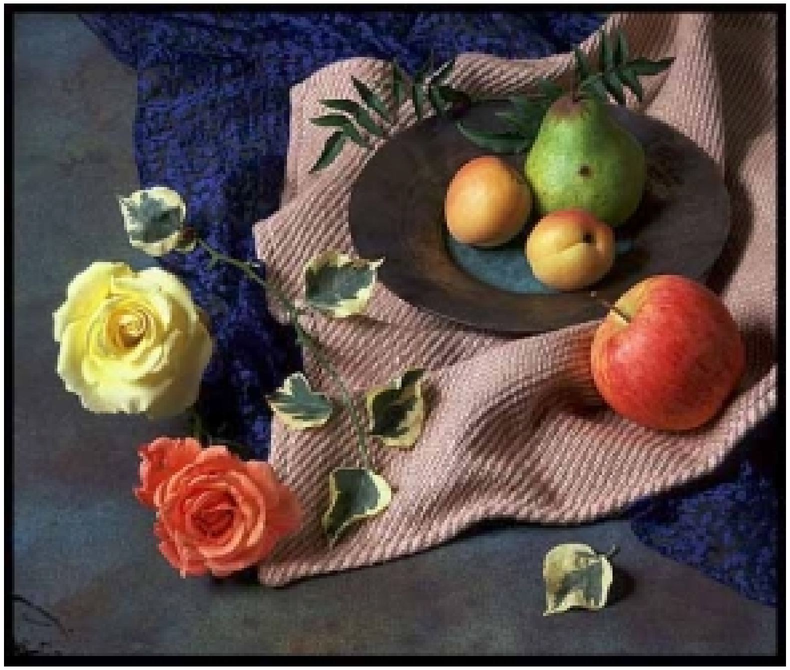
The encoder can specify the quantization tables different from the default ones as part of the header information

# Performance of JPEG

- For color images at 24 bits/pixel (bpp)
  - 0.25-0.5 bpp: moderate to good
  - 0.5-0.75 bpp: good to very good
  - 0.75-1.5 bpp: excellent, sufficient for most applications
  - 1.5-2 bpp: indistinguishable from original
  - From: G. K. Wallace: The JPEG Still picture compression standard, *Communications of ACM*, April 1991.
- For grayscale images at 8 bpp
  - 0.5 bpp: excellent quality

# JPEG Performance





487x414 pixels,
Uncompressed, 600471 Bytes,24 bpp
85502 Bytes, 3.39 bpp, CR=7

487x414 pixels
41174 Bytes, 1.63 bpp, CR=14.7

# 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

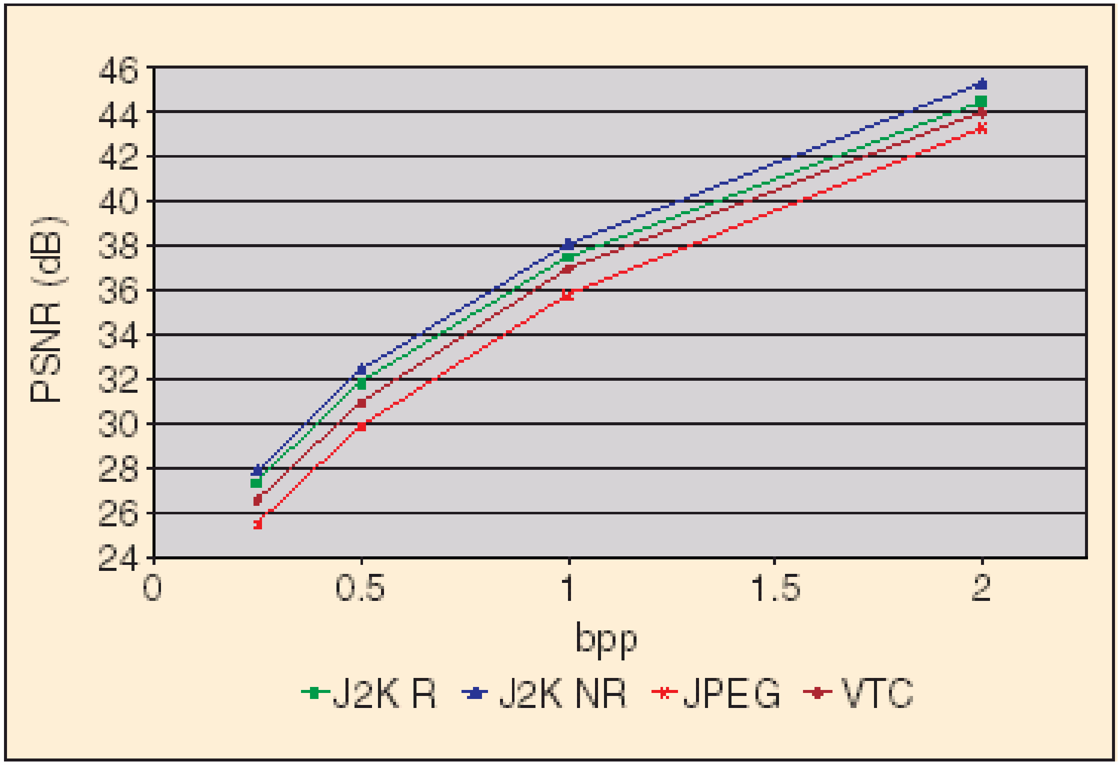
# JPEG2000

- Improved coding efficiency
- Full quality scalability
  - From lossless to lossy at different bit rate
- Spatial scalability
- Improved error resilience
- Tiling
- Region of interests

# 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

# JPEG2000 vs. JPEG: Coding Efficiency

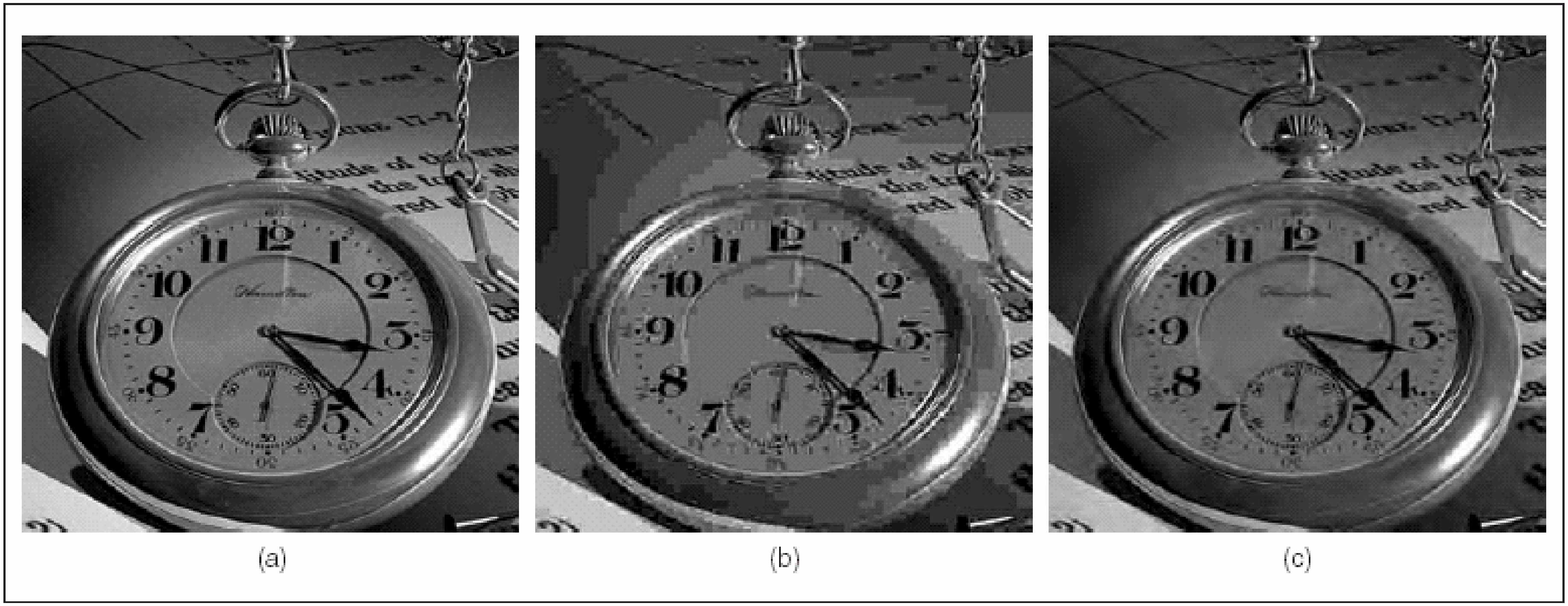


▲ 19. PSNR results for the lossy compression of a natural image by means of different compression standards.

From [skodras01]

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

# Example Image

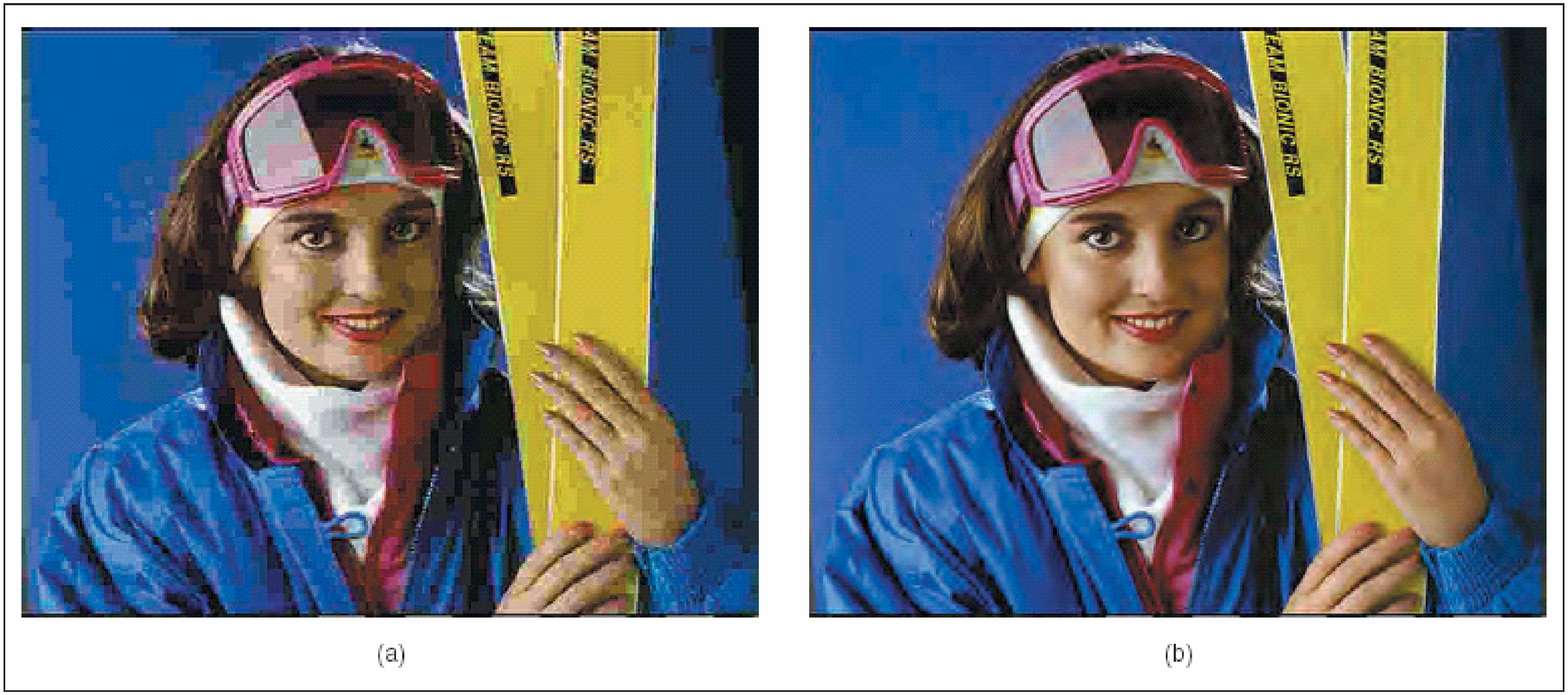


(a) (b) (c)

▲ 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]

# What is Scalability?

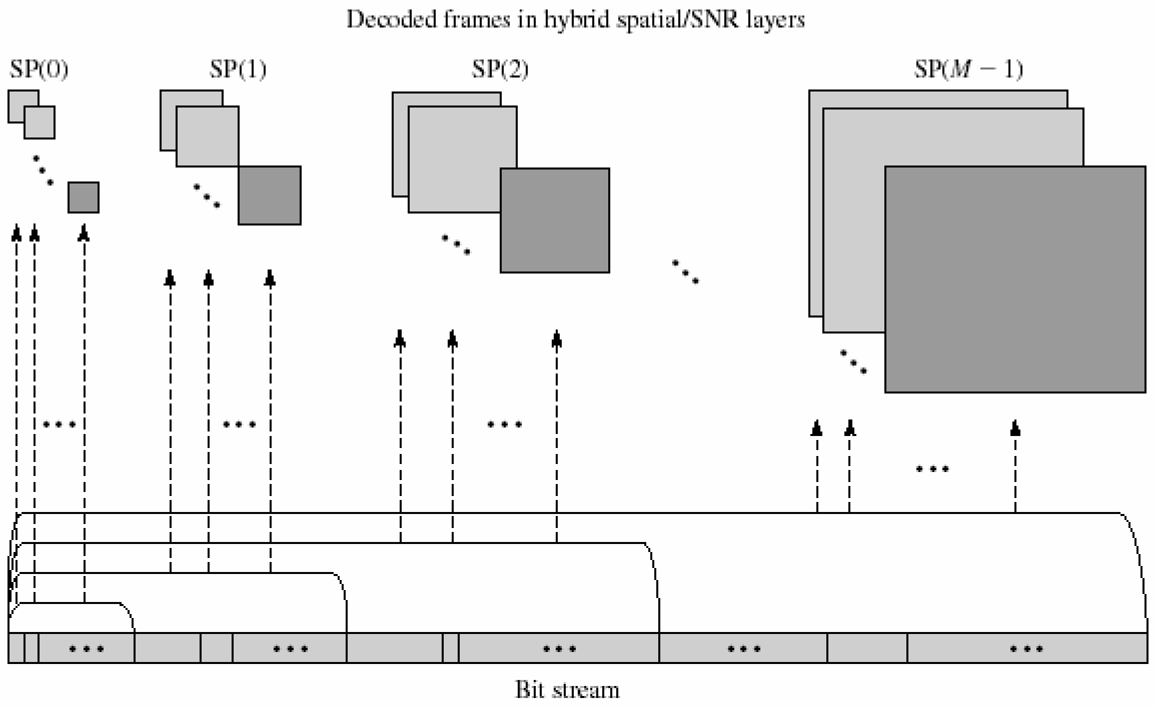
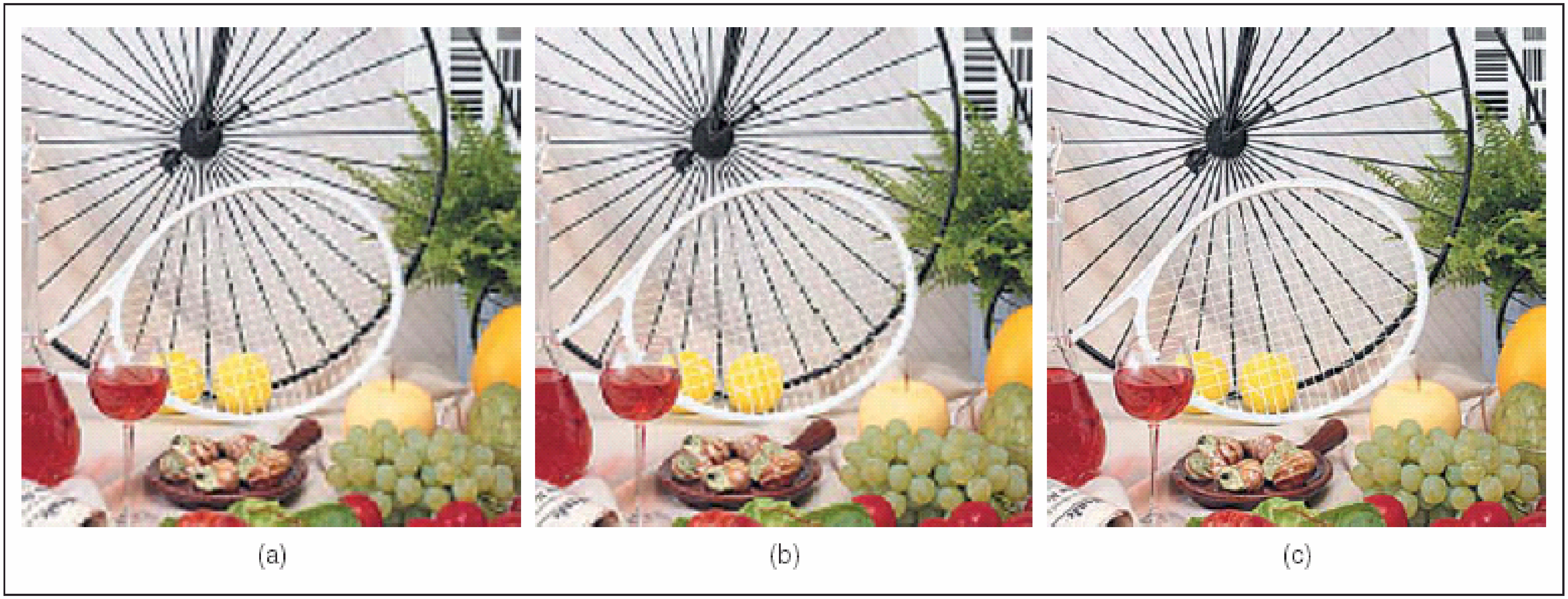


**Figure 11.7** $N \times M$ layers of combined spatial/quality scalability. Reprinted from I. Sodagar, H.-J. Lee, P. Hatrack, and Y.-Q. Zhang, Scalable wavelet coding for synthetic/natural hybrid images, *IEEE Trans. Circuits Syst. for Video Technology* (March 1999), 9:244–54. Copyright 1999 IEEE.

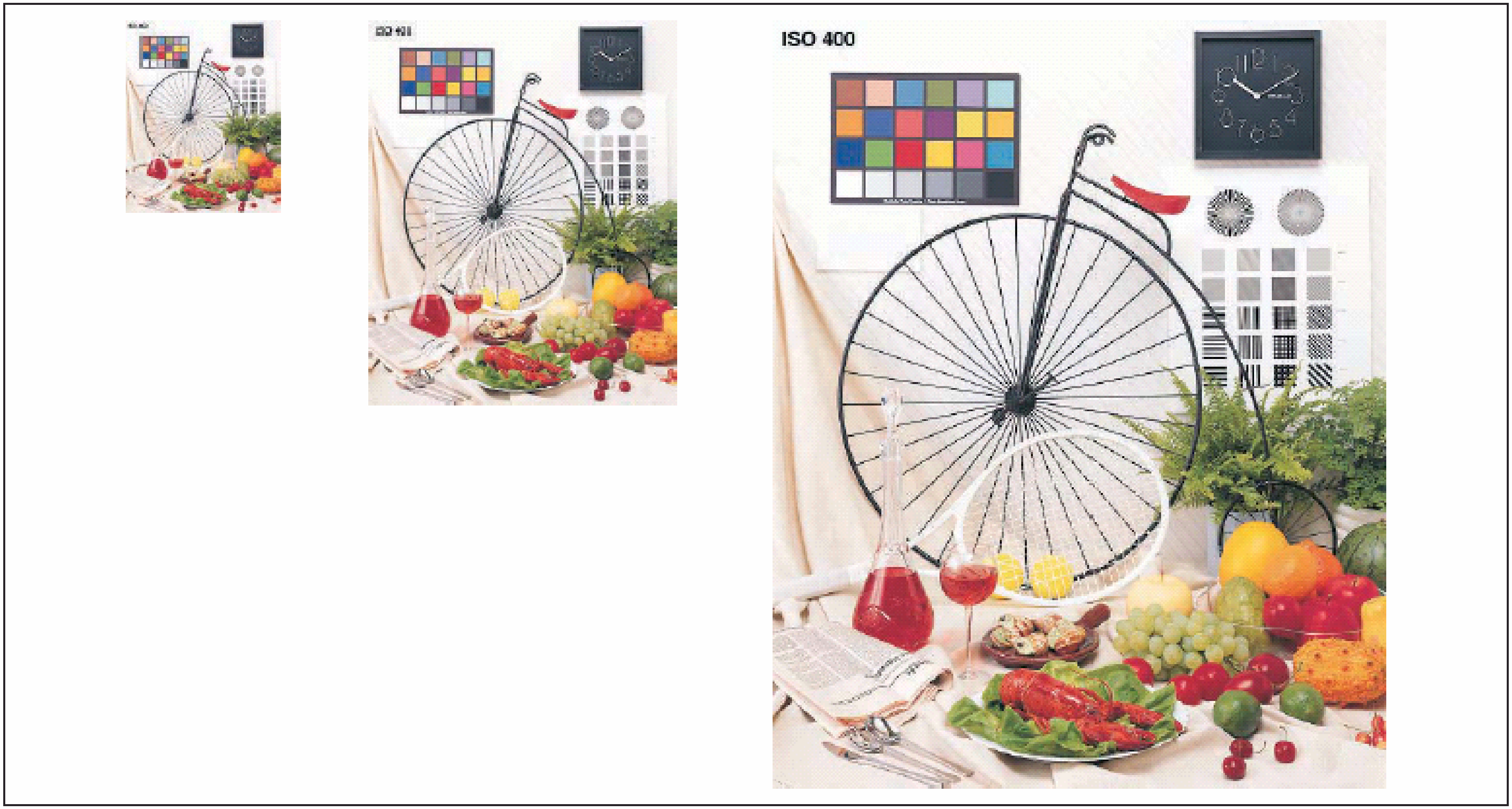
# Quality Scalability of JPEG2000



▲ *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 slides 34-40 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]

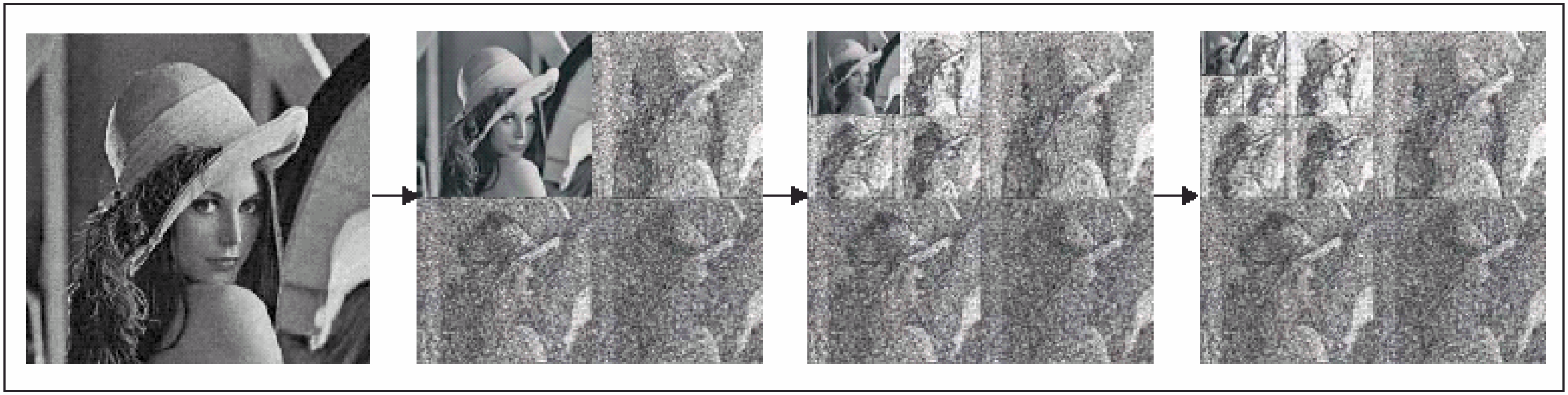
# Why do we want scalability

- 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

# 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 decomposition
  - Wavelet coefficients are coded bit plane by bit plane
  - Spatial scalability can be achieved by reconstructing from only low resolution wavelet coefficients
  - Quality scalability can be achieved by decoding only partial sets of bit planes, starting from the most significant bit plane

# Wavelet Decomposition



▲ 6. Three-level dyadic wavelet decomposition of the image "Lena."

From [skodras01]

# 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

# Temporal-Frequency Domain Partition

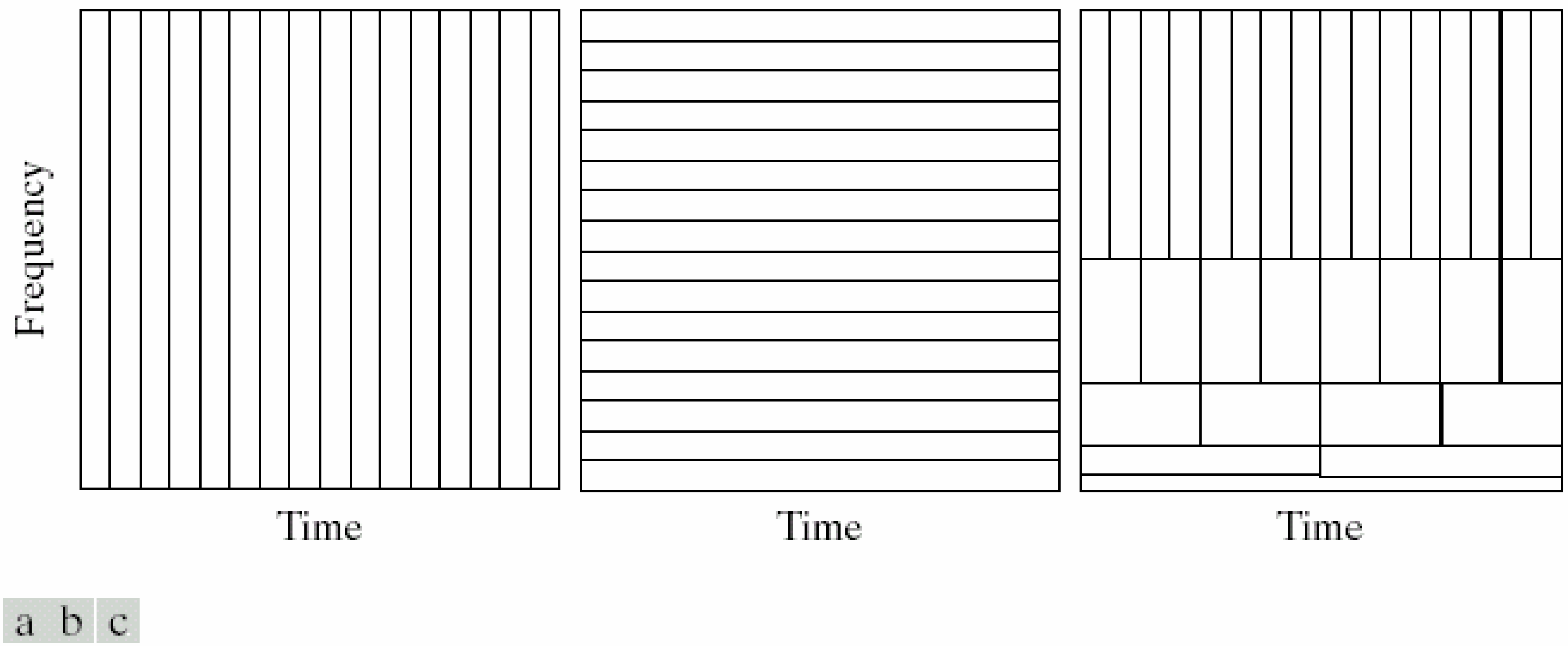
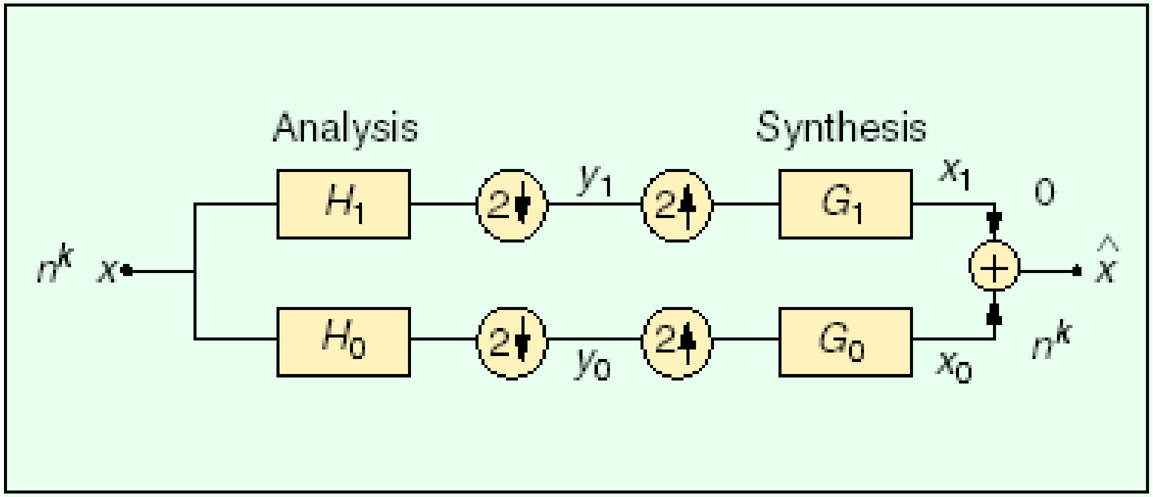


**FIGURE 7.21** Time-frequency tilings for (a) sampled data, (b) FFT, and (c) FWT basis functions.

# Two Band Subband Decomposition

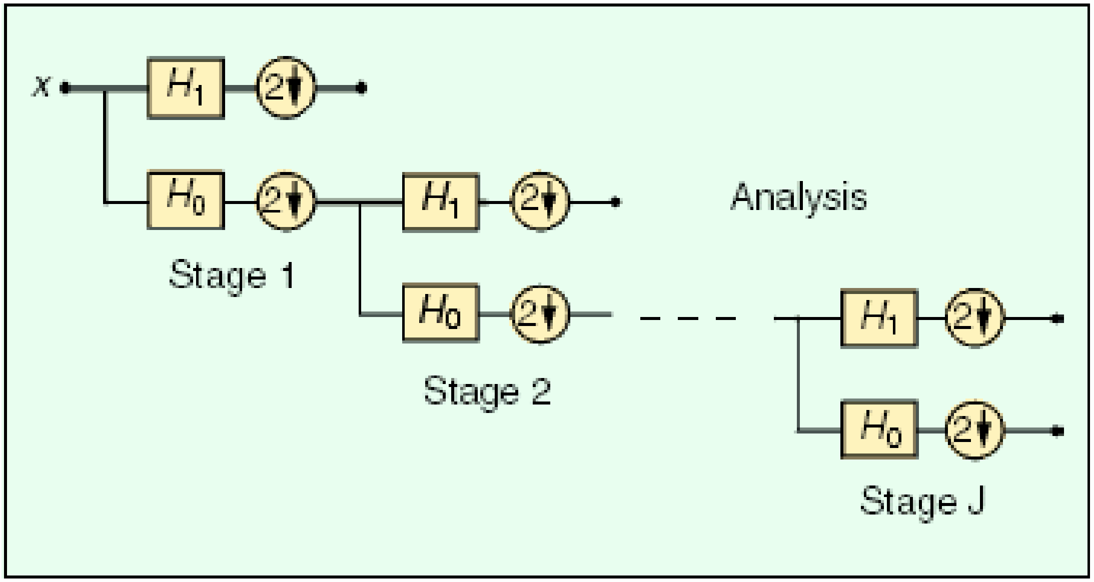


▲ *1. Multirate filter bank. Two channel analysis followed by synthesis.*

From [Vetterli01]

H0: Lowpass filter,  y0: a blurred version of x
H1: Highpass filter, y1: edges in x
Example:  haar wavelet: averaging and difference

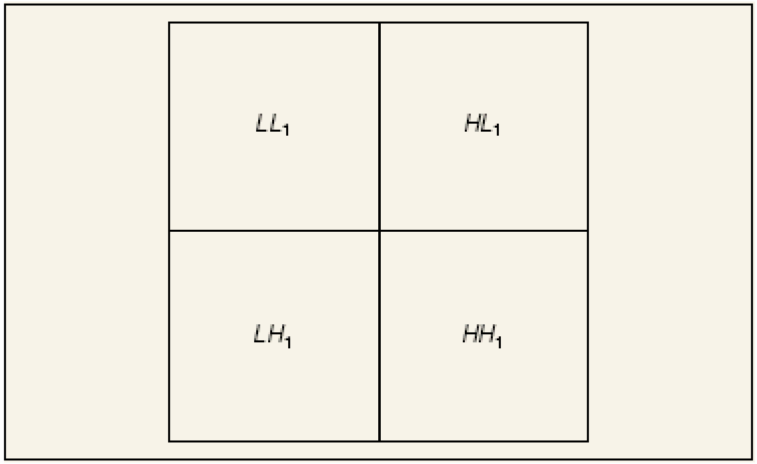
# Wavelet Transform = Subband Tree



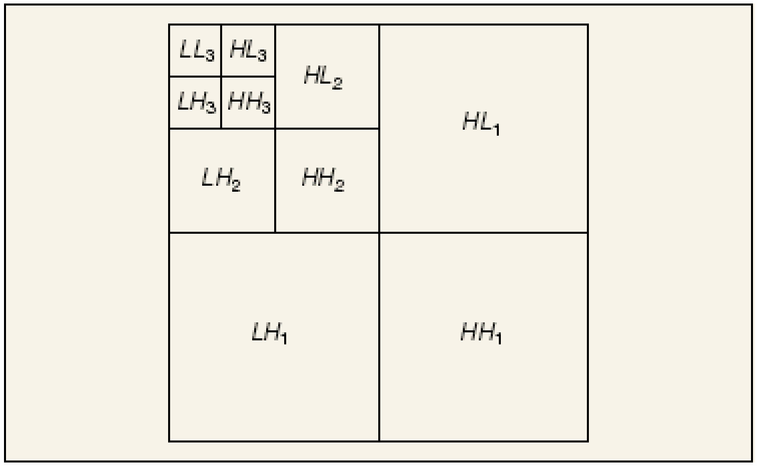
▲ *3. Iterated filter bank. The lowpass branch gets split repeatedly to get a discrete-time wavelet transform.*

From [Vetterli01]

# Wavelet Transform for Images



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

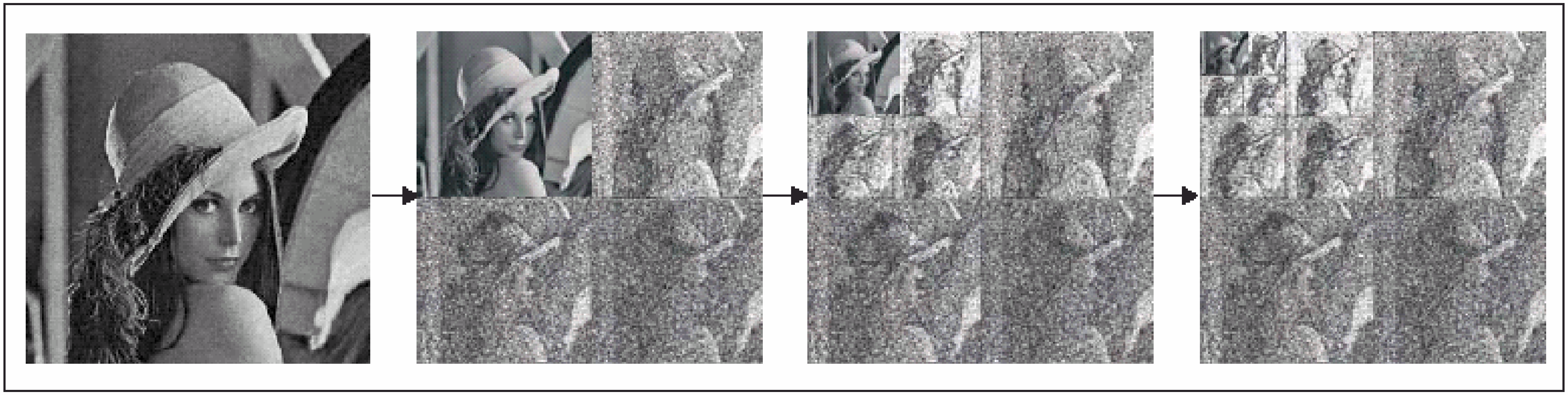


▲ 6. The subband labeling scheme for a three-level, 2-D wavelet transform.

From [Usevitch01]

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

# Wavelet Decomposition



▲ 6. Three-level dyadic wavelet decomposition of the image "Lena."

From [skodras01]

# Comparison of Different Filters

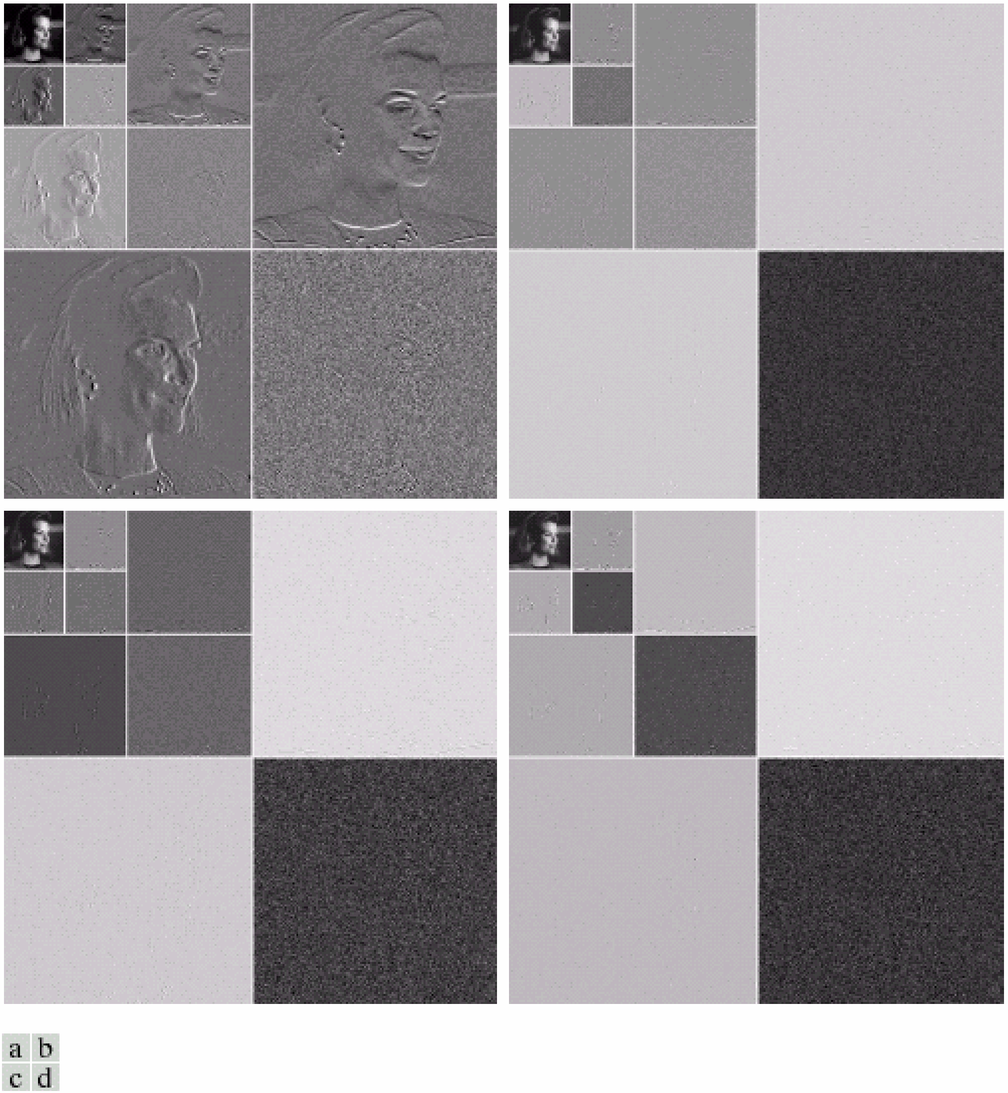
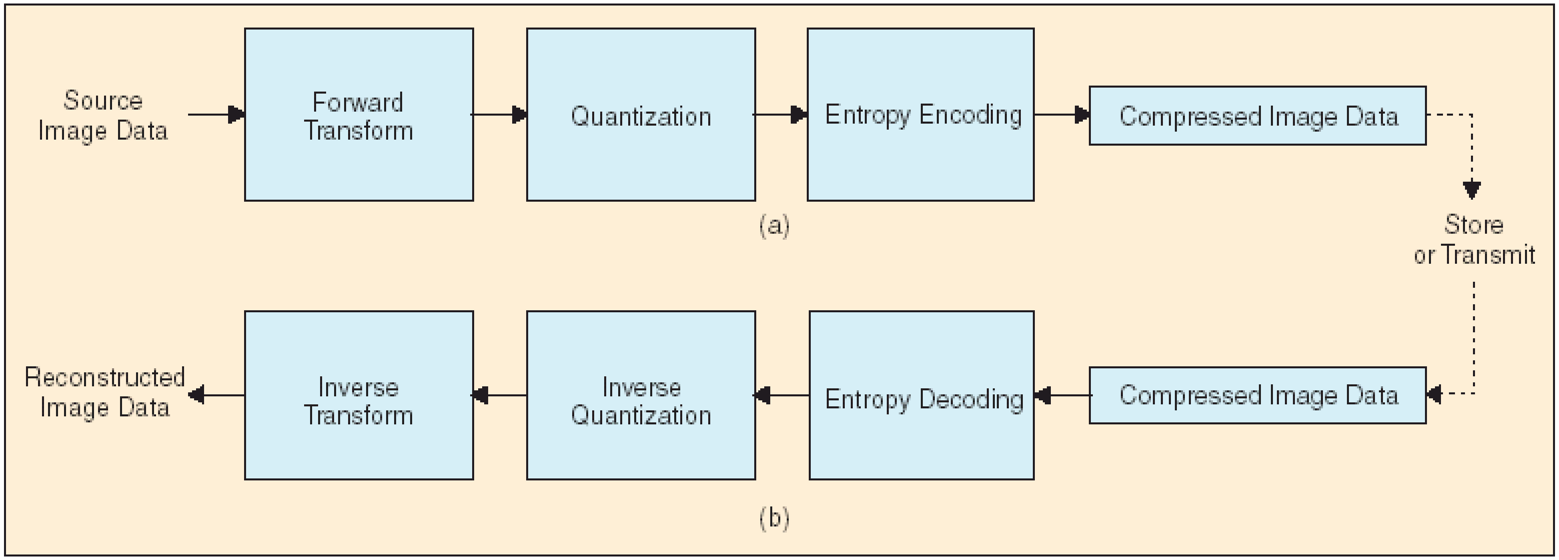


**FIGURE 8.42** Wavelet transforms of Fig. 8.23 with respect to (a) Haar wavelets, (b) Daubechies wavelets, (c) symlets, and (d) Cohen-Daubechies-Feauveau biorthogonal wavelets.

# 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.
- 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.

# What Should You Know

- JPEG Processing for a gray scale image:
  - Divide the image into data units of 8x8 blocks
  - Each 8x8 block go through 3 steps:
    - Transform (DCT), quantization, run-length+Huffman coding
    - Know details of each step
  - Quality vs. bit rate trade-off can be controlled by setting a quality factor, which scales a predefined quantization matrix.
- How does JPEG handle color images
  - The RGB coordinate is converted to YCbCr
    - How to do the conversion?
  - The Cb and Cr components are typically down sampled by a factor of 2 in both horizontal and vertical directions
    - Why should we down-sample chrominance components?
  - Each component is divided into 8x8 blocks and coded as above
- Performance of JPEG
  - Very good quality at 1bpp for 24 bpp color images, acceptable quality at 0.5 bpp, poor quality at 0.25 bpp or below.

# What Should You Know (Cnt'd)

- JPEG2000
  - What does quality and spatial scalability mean
    - Know the difference between wavetlet transform and DFT/DCT
    - Understand the concept of multi-resolution representation
  - How to achieve spatial scalability
    - By using wavelet transform in encoding, and ordering the subband in increasing order of resolution
    - Decoder can choose to decode only a partial set of subbands
  - How to achieve quality scalability
    - By using bit plane coding on wavelet coefficients

# References

- Official JPEG site: www.jpeg.org
- Gonzalez and Woods, Digital image processing, 2nd edition, Prentice Hall, 2002. Sec.8.5.3.
- G. K. Wallace: The JPEG still picture compression standard, *Communications of ACM*, April 1991. (available at the course website)
- A. Skodras, C. Christopoulos, T. Ebrahimi, The JPEG2000 Still Image Compression Standard, IEEE Signal Processing Magazine, Sept. 2001. (available at the course website)
- This site contains an easy-to-follow tutorial on JPEG
  - http://web.usxchange.net/elmo/jpeg.htm
- A site containing relevant links on JPEG
  - http://datacompression.info/JPEG.shtml