CHAPTER 7 & 8

IMAGE COMPRESSION

CHAPTER 7: IMAGE COMPRESSION

- Every day, an enormous amount of digital information is stored, processed and transmitted.
- The entire catalog of the Library of Congress (the world's largest library) is available electronically.
- The storage and communications requirements for information storage are immense!
- Image compression addresses the problem of reducing the amount of data required to represent a digital image.
- Two broad categories of compression techniques:
  - Lossless techniques
    - Useful in image archiving (e.g., storage of legal or medical records)
    - No information is lost.
  - Lossy techniques
    - Useful in applications such as broadcast TV, videoconferencing, and facsimile transmission.
    - A certain amount of information loss is acceptable.

MULTIMEDIA STORAGE REQUIREMENTS (WINDOW OF 640X480 PIXELS)

- **Text**
  - 2 bytes for every 8x8 pixel character
  - # of characters/page = (640x480)/(8x8) = 4,800 bytes
  - Storage/screen page = 4,800x2 = 9,600 bytes = 9.4 KB
- **Vector images**
  - A typical image with 500 lines
  - Each line is defined by its coordinates in *x* & *y* directions, and by an 8-bit attribute field.
  - Coordinates in the *x* direction require 10 bits: $\log_2(640)$.
  - Coordinates in the *y* direction require 9 bits: $\log_2(480)$.
  - Bits/line = 9 + 10 + 9 + 10 + 8 = 46 bits.
  - Storage/screen page = 500x46/8 = 2,875 bytes = 2.8 KB
- **Bit-mapped images**
  - 256 different colors
  - Storage/screen page = 640x480x1 = 307,200 bytes = 300 KB

MULTIMEDIA STORAGE REQUIREMENTS (WINDOW OF 640X480 PIXELS)

- **Speech of telephone quality**
  - Sampled at 8 kHz, and quantized using 8 bits/sample => 64 Kbits/s
  - Storage/second = 64/8 = 64 Kbits = 8 KB
- **Stereo audio of CD quality**
  - Sampled at 44.1 kHz, and quantized using 16 bits/sample.
  - Storage/second = (2x44,100x16)/8 = 176,400 bytes = 172 KB
- **Video**
  - 25 frames/second
  - 1 luminance later, 2 chrominance layers
  - PAL standard: 625 lines x 833 pixels
  - CCIR 601 (studio standard)
    - Luminance (*Y*) sampled at 13.5 MHz
    - Chrominance (*R*-*Y* & *B*-*Y*) sampled at 6.75 MHz
  - Storage/second = 640x480x25x3 = 23,040,000 bytes = 22,500 KB

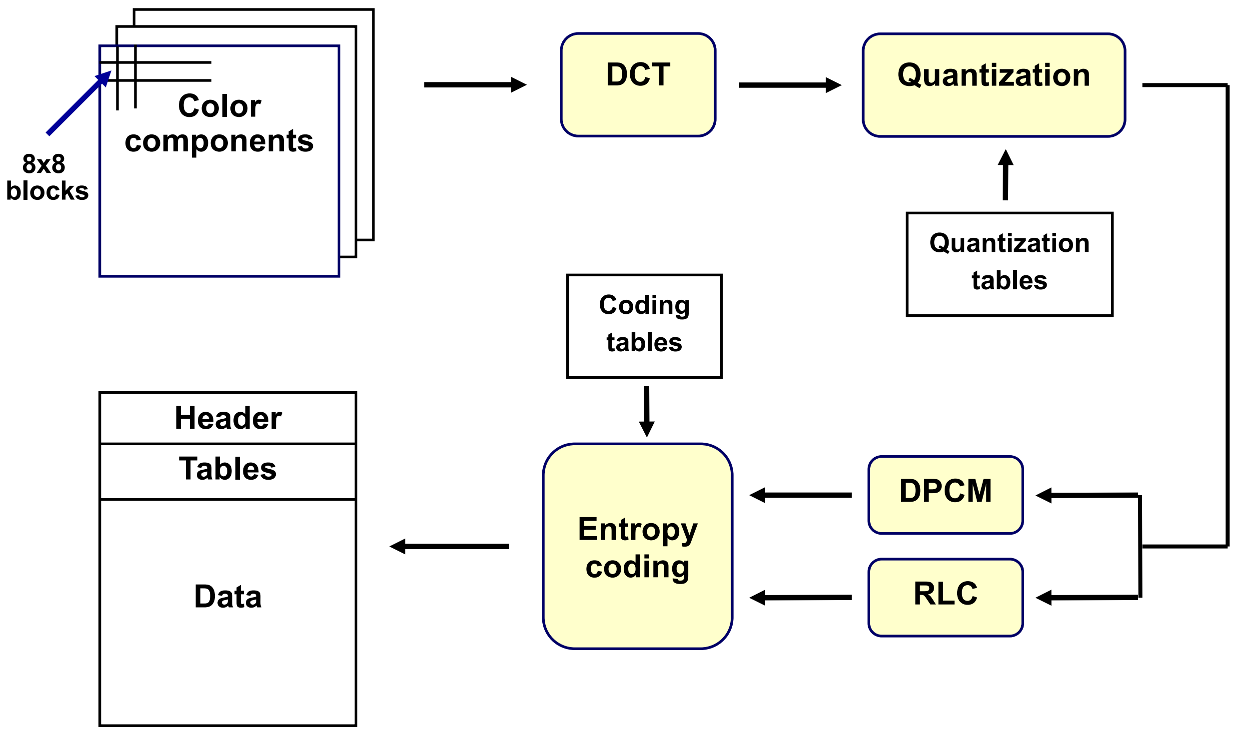
JPEG

- Joint Photographic Experts Group (JPEG)
- A standardized lossy image compression algorithm
- Based on DCT (Discrete Cosine Transform)
- Designed for compressing either full-color or gray-scale images of natural, real-world scenes.
- GIF compression
  - Best for images with flat areas of color (e.g., logos, line drawings, simple cartoons).
  - Lossless
  - Based on LZW (Lempel-Ziv-Welsh)
- Each color layer is handled separately.
- The resolution of individual layers may be different.
- Video color models: YUV, YIQ, YCbCr
- Chroma subsampling: 4:4:4, 4:2:2, 4:2:0

FOUR MODES OF OPERATIONS

- **Sequential baseline**
  - A simple and efficient algorithm.
  - Adequate for most applications.
  - The image is scanned in a raster scan fashion l-to-r/t-to-b.
- **Progressive**
  - The image is encoded in multiple scans at the same spatial resolution.
- **Hierarchical**
  - The image is encoded at multiple spatial resolutions.
  - Lower resolution images may be displayed without having to decompress the image at a higher spatial resolution.
  - Can be implemented using sequential, progressive, or lossless modes.
- **Lossless**
  - The image is encoded to guarantee exact recovery of every sample value.
  - Compression efficiency is inherently lower than those of lossy methods.

SEQUENTIAL BASELINE



DCT & IDCT

DCT

$$F(u,v) = \frac{1}{\sqrt{2N}} C(u)C(v) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) \cos\frac{(2i+1)u\pi}{2N} \cos\frac{(2j+1)v\pi}{2N}$$

$C(x) = \frac{1}{\sqrt{2}}$ if $x = 0$, else 1 if $x > 0$

Inverse DCT

$$\tilde{f}(i,j) = \frac{1}{\sqrt{2N}} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)F(u,v) \cos\frac{(2i+1)u\pi}{2N} \cos\frac{(2j+1)v\pi}{2N}$$

$C(x) = \frac{1}{\sqrt{2}}$ if $x = 0$, else 1 if $x > 0$

A CASE STUDY: TEST IMAGE

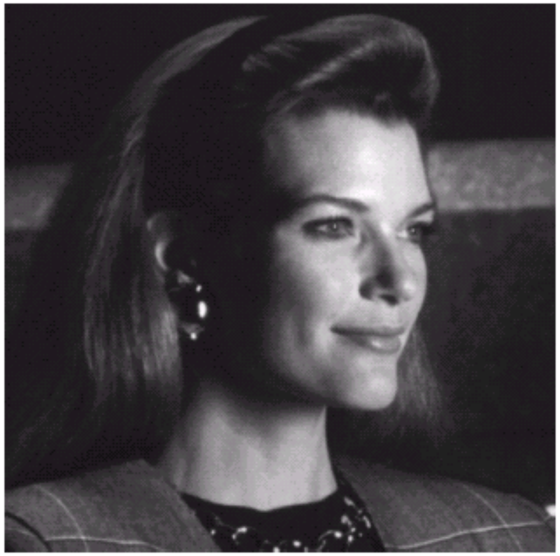


FIGURE 8.23 A 512×512 8-bit monochrome image.

A CASE STUDY: AN 8X8 BLOCK

52	55	61	66	70	61	64	73
63	59	66	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

A CASE STUDY: LEVEL SHIFTING

The quantity 2^{n-1} is subtracted from each pixel value.

$n=8 \Rightarrow 2^{n-1}= 128$

-76	-73	-67	-62	-58	-67	-64	-55
-65	-69	-62	-38	-19	-43	-59	-56
-66	-69	-60	-15	16	-24	-62	-55
-65	-70	-57	-6	26	-22	-58	-59
-61	-67	-60	-24	-2	-40	-60	-58
-49	-63	-68	-58	-51	-65	-70	-53
-43	-57	-64	-69	-73	-67	-63	-45
-41	-49	-59	-60	-63	-52	-50	-34

A CASE STUDY: APPLICATION OF DCT

-415	-29	-62	25	55	-20	-1	3
7	-21	-62	9	11	-7	-6	6
-46	8	77	-25	-30	10	7	-5
-50	13	35	-15	-9	6	0	3
11	-8	-13	-2	-1	1	-4	1
-10	1	3	-3	-1	0	2	-1
-4	-1	2	-1	2	-3	1	-2
-1	-1	-1	-2	-1	-1	0	-1

A CASE STUDY: NORMALIZATION MATRIX

$Z(u,v)$

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

A CASE STUDY: QUANTIZATION

-26	-3	-6	2	2	0	0	0
1	-2	-4	0	0	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	0	0	0	0
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	0

DCT coefficients are quantized using the formula

$$\hat{T}(u,v) = round\left[\frac{T(u,v)}{Z(u,v)}\right]$$

38 consecutive zeros!

A CASE STUDY: ZIGZAG REORDERING

[-26 -3 1 -3 -2 -6 2 -4 1 -4 1 1 5 0 2 0 0 -1 2 0 0 0 0 0 -1 -1 EOB]

a special EOB Huffman code word indicates that the remainder of the coefficients are zeros.

A CASE STUDY: PREPARATION FOR ENTROPY CODING

- Entropy coding is **lossless**.
- **DC** and **AC** coefficients are treated <u>differently</u>.
- Differential Pulse Code Modulation (DPCM) on DC coefficients
  - Each DPCM-coded DC coefficient is represented by a pair of symbols: (**CATEGORY**, **AMPLITUDE**)
  - CATEGORY: indicates the # of bits needed to represent the coefficient.
  - AMPLITUDE: contains the actual bits.
  - The 1's complement notation is used for negative numbers.
- Run-length coding (RLC) on AC coefficients
  - RLC replaces each AC coefficient by a pair (**RUNLENGTH**, **VALUE**)
  - RUNLENGTH: indicates the # of zeros in the run.
  - VALUE: the next nonzero coefficient.
  - The special pair **(0,0)** indicates the EOB after the last nonzero AC coefficient.

A CASE STUDY: DPCM ON DC COEFFICIENTS

- Each 8x8 block has only one DC coefficient
- The DC coefficient is unlikely to change drastically within a short distance.
- DPCM is an ideal scheme for coding the DC coefficients
- $d_0 = DC_0$, $d_i = DC_{i+1} - DC_i$
- Coding of the first difference
  - -26 (-17) = -9
  - From Table 8.17, -9 lies in category 4
  - For category *K*, additional *K* bits are needed.
  - Positive difference: *K* LSBs
  - Negative difference: *K* LSBs – 1 => 0111 – 1 = 0110
  - DPCM coded DC code word = $\underbrace{101}_{\text{category}}\underbrace{0110}_{\text{amplitude}}$

A CASE STUDY: RLC ON AC COEFFICIENTS

- Each 8x8 block has 63 AC coefficients.
- Most image blocks tend to have small high frequency coefficients.
- These small coefficients are zeroed out by quantization.
- Coding of the first AC coefficient
  - (RUNLENGTH, VALUE) = (0,-3)
  - From Table 8.17, -3 lies in category 2
  - From Table 8.19, (0/2) maps to 01
  - For category *K*, additional *K* bits are needed.
  - Positive difference: *K* LSBs
  - Negative difference: *K* LSBs – 1 => 01-1 = 00
  - RLC coded AC code word = 0100

A CASE STUDY: CODING CATEGORIES

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
$-7, \dots, -4, 4, \dots, 7$	3	3
$-15, \dots, -8, 8, \dots, 15$	4	4
$-31, \dots, -16, 16, \dots, 31$	5	5
$-63, \dots, -32, 32, \dots, 63$	6	6
$-127, \dots, -64, 64, \dots, 127$	7	7
$-255, \dots, -128, 128, \dots, 255$	8	8
$-511, \dots, -256, 256, \dots, 511$	9	9
$-1023, \dots, -512, 512, \dots, 1023$	A	A
$-2047, \dots, -1024, 1024, \dots, 2047$	B	B
$-4095, \dots, -2048, 2048, \dots, 4095$	C	C
$-8191, \dots, -4096, 4096, \dots, 8191$	D	D
$-16383, \dots, -8192, 8192, \dots, 16383$	E	E
$-32767, \dots, -16384, 16384, \dots, 32767$	F	N/A

A CASE STUDY: DC BASE CODES

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

A CASE STUDY: AC BASE CODES

Run/ Category	Base Code	Length	Run/ 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)*.

A CASE STUDY: AC BASE CODES

2/8	1111111110001101	24	A/8	1111111111001110	24
2/9	1111111110001110	25	A/9	1111111111001111	25
2/A	1111111110001111	26	A/A	1111111111010000	26
3/1	111010	7	B/1	111111010	10
3/2	111110111	11	B/2	1111111111010001	18
3/3	11111110111	14	B/3	1111111111010010	19
3/4	1111111110010000	20	B/4	1111111111010011	20
3/5	1111111110010001	21	B/5	1111111111010100	21
3/6	1111111110010010	22	B/6	1111111111010101	22
3/7	1111111110010011	23	B/7	1111111111010110	23
3/8	1111111110010100	24	B/8	1111111111010111	24
3/9	1111111110010101	25	B/9	1111111111011000	25
3/A	1111111110010110	26	B/A	1111111111011001	26
4/1	111011	7	C/1	1111111010	11
4/2	1111111000	12	C/2	1111111111011010	18
4/3	1111111110010111	19	C/3	1111111111011011	19
4/4	1111111110011000	20	C/4	1111111111011100	20
4/5	1111111110011001	21	C/5	1111111111011101	21
4/6	1111111110011010	22	C/6	1111111111011110	22
4/7	1111111110011011	23	C/7	1111111111011111	23
4/8	1111111110011100	24	C/8	1111111111100000	24
4/9	1111111110011101	25	C/9	1111111111100001	25
4/A	1111111110011110	26	C/A	1111111111100010	26

A CASE STUDY: COMPLETELEY CODED ARRAY

1010110 0100 001 0100 0101 100001 0110 100011 001 100011 001

[-26 -3 1 -3 -2 -6 2 -4 1 -4 1 1 5 0 2 0 0 -1 2 0 0 0 0 0 -1 -1 EOB]

001 100101 11100110 110110 0110 11110100 000 1010

# of bits needed to store the 8x8 block = 64x8 = 512

# of bits after JPEG compression = 92

Compression ratio = 512/92 => 5.6:1

A CASE STUDY: DECOMPRESSION BEGINS

-26	-3	-6	2	2	0	0	0
1	-2	-4	0	0	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	0	0	0	0
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	0

A CASE STUDY: DENORMALIZATION

$$\dot{T}(u,v) = \hat{T}(u,v)Z(u,v)$$

-416	-33	-60	32	48	0	0	0
12	-24	-56	0	0	0	0	0
-42	13	80	-24	-40	0	0	0
-56	17	44	-29	0	0	0	0
18	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	0

A CASE STUDY: INVERSE DCT

-70	-64	-61	-64	-69	-66	-58	-50
-72	-73	-61	-39	-30	-40	-54	-59
-68	-78	-58	-9	13	-12	-48	-64
-59	-77	-57	0	22	-13	-51	-60
-54	-75	-64	-23	-13	-44	-63	-56
-52	-71	-72	-54	-54	-71	-71	-54
-45	-59	-70	-68	-67	-67	-61	-50
-35	-47	-61	-66	-60	-48	-44	-44

A CASE STUDY: LEVEL SHIFTING

The quantity 2^{n-1} is added from each pixel value.

n=8 => 2^{n-1}= 128

58	64	67	64	59	62	70	78
56	55	67	89	98	88	74	69
60	50	70	119	141	116	80	64
69	51	71	128	149	115	77	68
74	53	64	105	115	84	65	72
76	57	56	74	75	57	57	74
83	69	59	60	61	61	67	78
93	81	67	62	69	80	84	84

A CASE STUDY: DIFFERENCE

52	55	61	66	70	61	64	73
63	59	66	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

−

58	64	67	64	59	62	70	78
56	55	67	89	98	88	74	69
60	50	70	119	141	116	80	64
69	51	71	128	149	115	77	68
74	53	64	105	115	84	65	72
76	57	56	74	75	57	57	74
83	69	59	60	61	61	67	78
93	81	67	62	69	80	84	84

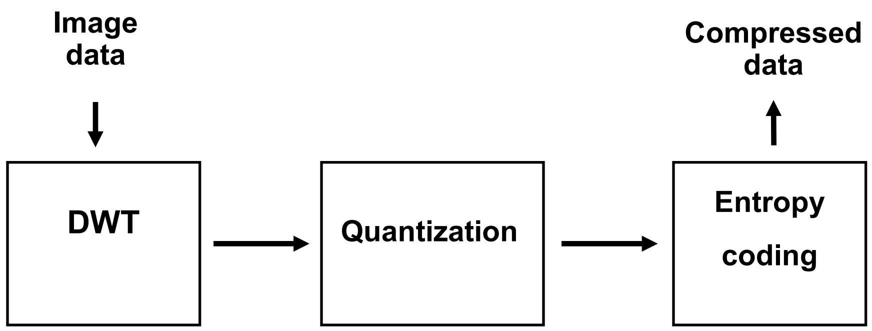
=

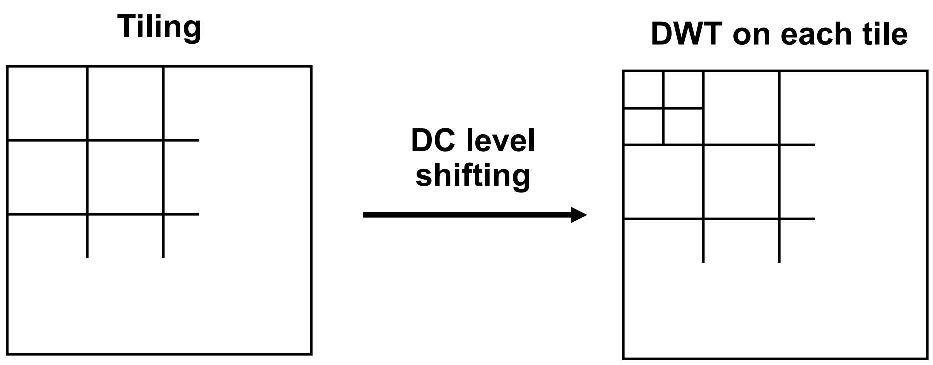
-6	-9	-6	2	11	-1	-6	-5
7	4	-1	1	11	-3	-5	3
2	9	-2	-6	-3	-12	-14	9
-6	7	0	-4	-5	-9	-7	1
-7	8	4	-1	11	4	3	-2
3	8	4	-4	2	11	1	1
2	2	5	-1	-6	0	-2	5
-6	-2	2	6	-4	-4	-6	10

JPEG 2000

- Major limitation of JPEG: coding performance degrades rapidly at low bit rates (especially below 0.25 bpp).
- The latest series of standards from the JPEG committee.
- Designed to address a variety of applications including color facsimile, printing, scanning, digital photography, remote sensing, and medical imagery.
- Based on the Discrete Wavelet Transform (DWT).
- Addresses a number of problems including:
  - Low bit rate compression
  - Lossless and lossy compression
  - Computer-generated imagery
  - Error resilience for transmission in noisy environments
- Core compression algorithm: EBCOT (Embedded Block Coding with Optimized Truncation)

JPEG 2000: BLOCK DIAGRAM





1-DIM DISCRETE WAVELET TRANSFORM (DWT)

$$W_\varphi(j_0,k)=\frac{1}{\sqrt{M}}\sum_x f(x)\varphi_{j_0,k}(x)$$: **approximation coefficients**

$$W_\psi(j,k)=\frac{1}{\sqrt{M}}\sum_x f(x)\psi_{j,k}(x)$$: **detail coefficients**

$$f(x)=\frac{1}{\sqrt{M}}\sum_k W_\varphi(j_0,k)\varphi_{j_0,k}(x)+\frac{1}{\sqrt{M}}\sum_{j=j_0}^{\infty}\sum_k W_\psi(j,k)\psi_{j,k}(x)$$

$f(x),\varphi_{j_0,k}(x)$, and $\psi_{j,k}(x)$ are functions of the discrete variable $x=0,1,2,\ldots,M-1$.

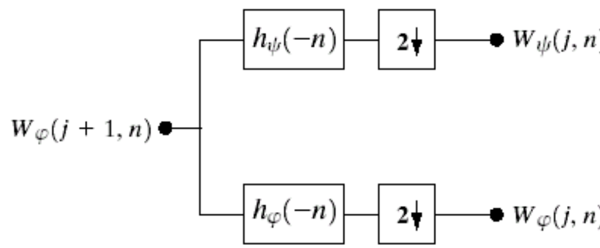
Normally, $j_0=0$, and M is a power of 2.

Summations are performed over $x=0,1,2,\ldots,M-1$, $j=0,1,2,\ldots,J-1$, $k=0,1,\ldots,2^j-1$.

1-DIM FAST WAVELET TRANSFORM (FWT)

Computationally efficient implementation of DWT.

FIGURE 7.15 An FWT analysis bank.



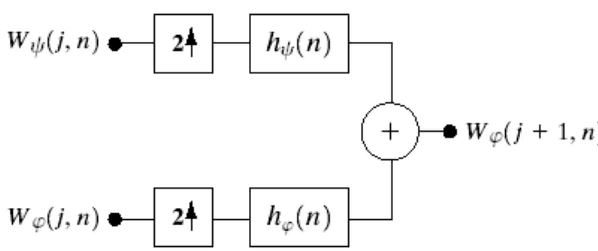


FIGURE 7.18 The FWT^{-1} synthesis filter bank.

2-DIM DISCRETE WAVELET TRANSFORM (DWT)

$\varphi(x, y)$: 2 - dim scaling function

$\psi^H(x, y), \psi^V(x, y), \psi^D(x, y)$: 2 - dim wavelets

$\varphi(x, y) = \varphi(x)\varphi(y)$: **separable scaling function**

$\psi^H(x, y) = \psi(x)\varphi(y)$: **measures variations along columns**

$\psi^V(x, y) = \varphi(x)\psi(y)$: **measures variations along rows**

$\psi^D(x, y) = \psi(x)\psi(y)$: **measures variations along diagonals**

2-DIM DISCRETE WAVELET TRANSFORM (DWT)

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

$$i = \{H, V, D\}$$

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

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

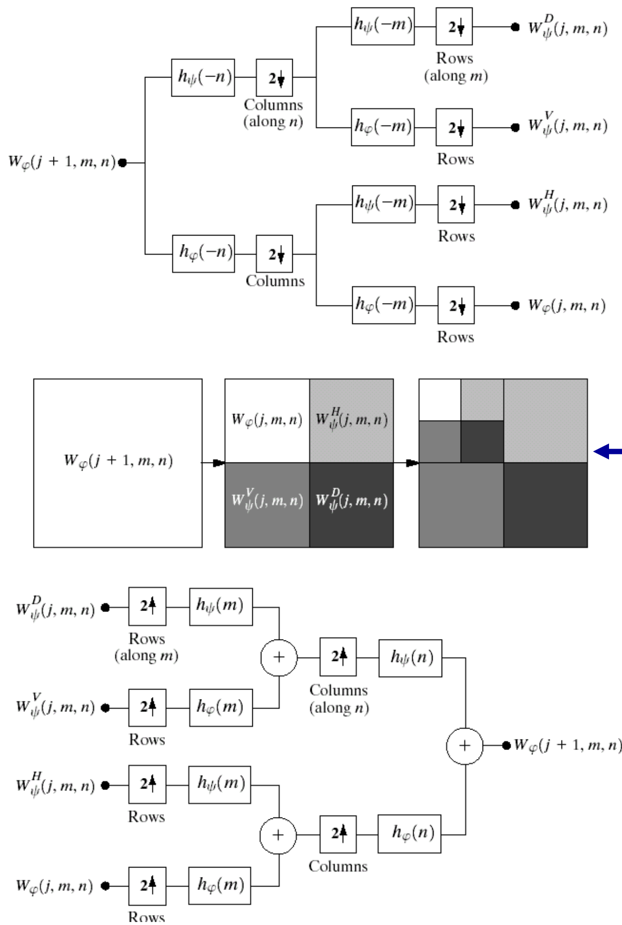
$$W_\psi^i(j, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\psi_{j^i_{j,m,n},m,n}(x,y)$$

$$f(x,y) = \frac{1}{\sqrt{MN}} \sum_m \sum_n W_\varphi(j_0, m, n)\varphi_{j_0,m,n(x,y)} + \frac{1}{\sqrt{MN}} \sum_{i=H,V,D} \sum_{j=j_0} \sum_m \sum_n W_\psi^i(j,m,n)\psi^i_{j,m,n}(x,y)$$

Normally, $j_0 = 0$, and $M = N = 2^j$.

Summations are performed over $x = 0,1,2,\ldots, M - 1$, $j = 0,1,2,\ldots, J - 1$, $m,n = 0,1,\ldots,2^j - 1$.

2-DIM FAST WAVELET TRANSFORM (FWT)



1st and 2nd level decomposition

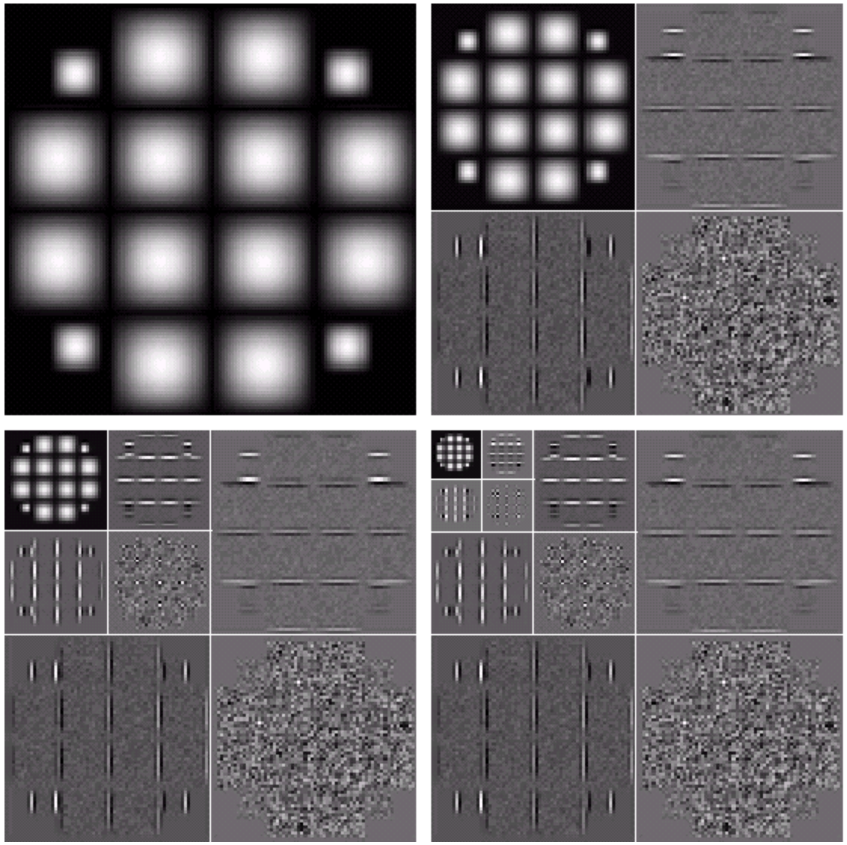
A FWT EXAMPLE

128x128 computer-generated image

1st level decomposition

2nd level decomposition

3rd level decomposition



a b
c d

FIGURE 7.23 A three-scale FWT.