Image Transforms

DCT and JPEG

Haar Filter for pattern recognition

ART (Angular Radial Transform)

Lecturer: Sang Hwa Lee

Introduction

- Important transforms for image processing
  - Compression
  - Pattern analysis
- Topics
  - Discrete Cosine Transform (DCT)
  - Haar Transform
  - ART (Angular Radial Transform)

Fourier Transform

- Fourier Analysis for continuous/discrete time signals
  - Use orthogonal basis functions {cos nt, sin mt}

$$x(t) = \sum_{n=0}^{\infty} a_n \cos(nt) + \sum_{m=0}^{\infty} b_m \sin(mt)$$

$$x[n] = \sum_{k=-\infty}^{\infty} X[k] e^{jk\frac{2\pi}{N}n}, \quad \text{with N samples}$$

- Fourier Coefficients
  - Unique representation of a function (signal)
  - Signal identification and reconstruction
    - Pattern recognition and synthesis

Why Transform is Used?

- Frequency domain processing
  - Modulation/demodulation
- Mathematical tool
  - Filtering (convolution-multiplication)
- Pattern identification
  - Fourier coefficients distributions
- Energy compaction
  - Energy is mainly distributed in the lower frequency band.
  - Usually high frequency coefficients are almost zero.

2D DTFS (DFT)

- Application to DTFS to 2D image signal
  - Consider the NxN image block as a periodic discrete signal
  - Orthogonal transform (x, y axes)
- Using the 2D DFT coefficients
  - Image compression: energy compaction
  - Pattern recognition: vector representation and clustering

$$I[x,y]=\sum_{n=0}^{N-1}\sum_{m=0}^{N-1}X[m,n]e^{jm\frac{2\pi}{N}x}e^{jn\frac{2\pi}{N}y},\quad \text{with NxN block}$$

$$X[m,n]=\frac{1}{N}\frac{1}{N}\sum_{y=0}^{N-1}\sum_{x=0}^{N-1}I[x,y]e^{-jm\frac{2\pi}{N}x}e^{-jn\frac{2\pi}{N}y}$$

DCT (Discrete Cosine Transform)

1D DCT (1)

- 1D Discrete Cosine Transform

$$v(k) = \alpha(k) \sum_{m=0}^{N-1} u(m) \cos\left[\frac{\pi(2m+1)k}{2N}\right], \quad 0 \le k \le N-1$$

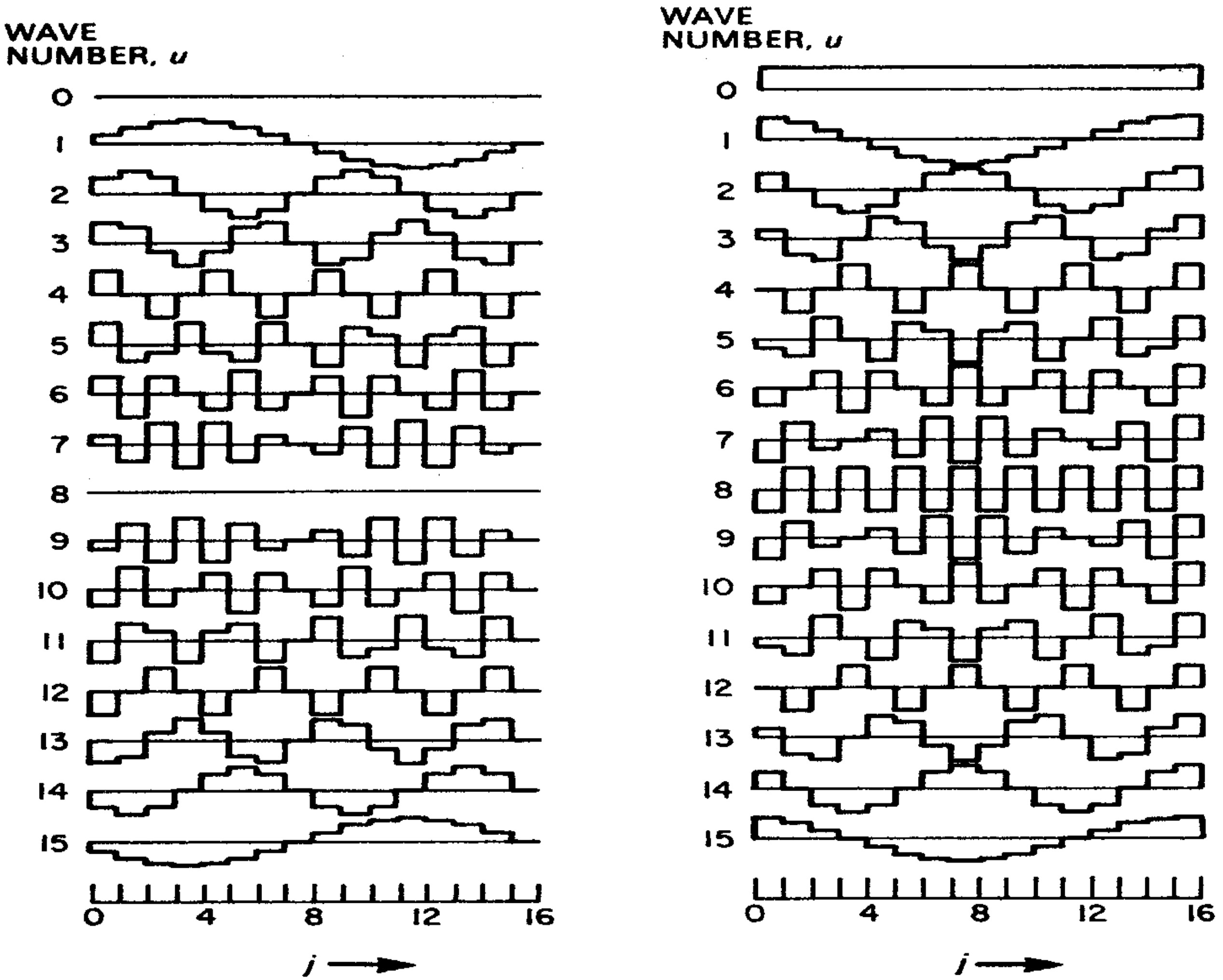
$$u(m) = \sum_{k=0}^{N-1} \alpha(k) v(k) \cos\left[\frac{\pi(2m+1)}{2N} k\right], \quad 0 \le m \le N-1$$

where $\alpha(0) = \frac{1}{\sqrt{N}}$

$$\alpha(k) = \frac{2}{\sqrt{N}}, \quad 1 \le k \le N-1$$

1D DCT (2)

- Basis functions (waveforms) for 1D DCT(N = 16)



(a) Sine component (b) Cosine component

2D DCT (1)

- Basis functions for 2D DCT
  - Separable for *x* and *y* directions

2D DCT (2)

- 8x8 block 2D DCT (64 coefficients)
  - Efficient energy compaction
  - Blocking artifact at low bit rates
  - Fast FDCT/IDCT algorithms exist

$$F_{vu} = \frac{1}{4} C_v C_u \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} S_{yx} cos\left(v\pi \frac{2y+1}{2N}\right) cos\left(u\pi \frac{2x+1}{2N}\right)$$

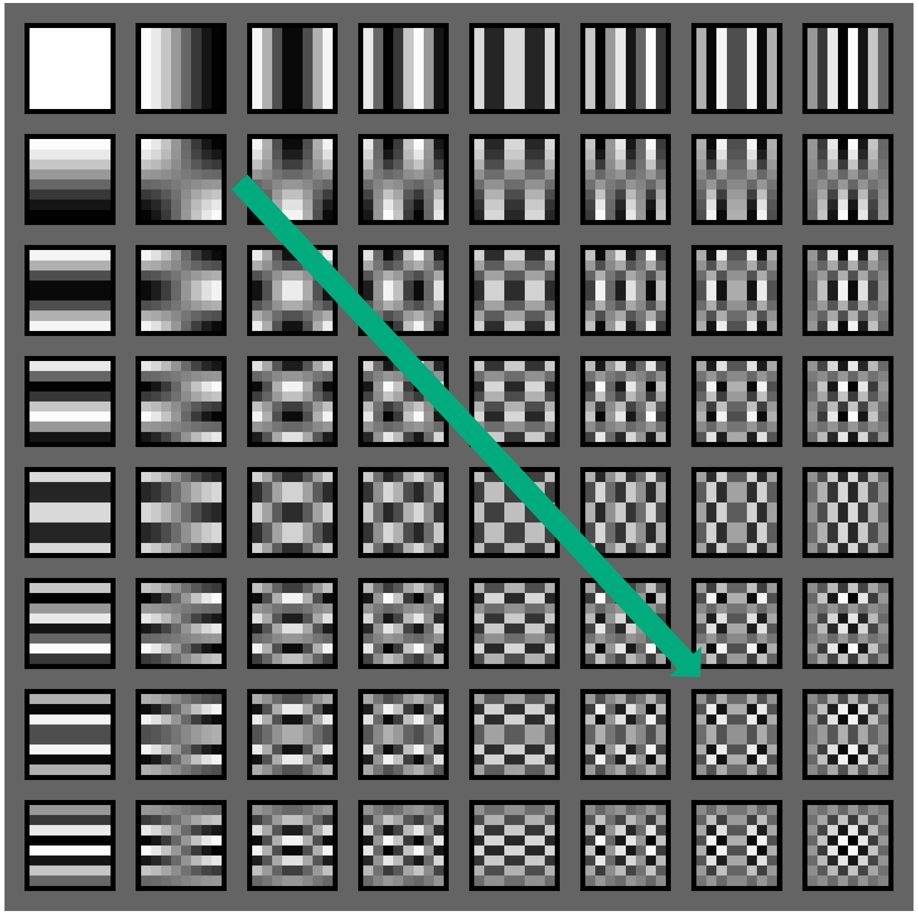
$$S_{yx} = \frac{1}{4} \sum_{v=0}^{N-1} \sum_{u=0}^{N-1} C_v C_u F_{vu} cos\left(v\pi \frac{2y+1}{2N}\right) cos\left(u\pi \frac{2x+1}{2N}\right)$$

$$C_u = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } u = 0 \\ 1 & \text{else} \end{cases} \qquad C_v = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } v = 0 \\ 1 & \text{else} \end{cases}$$

2D DCT (3)

2-D DCT Waveforms (8x8: total 64 coefficients)

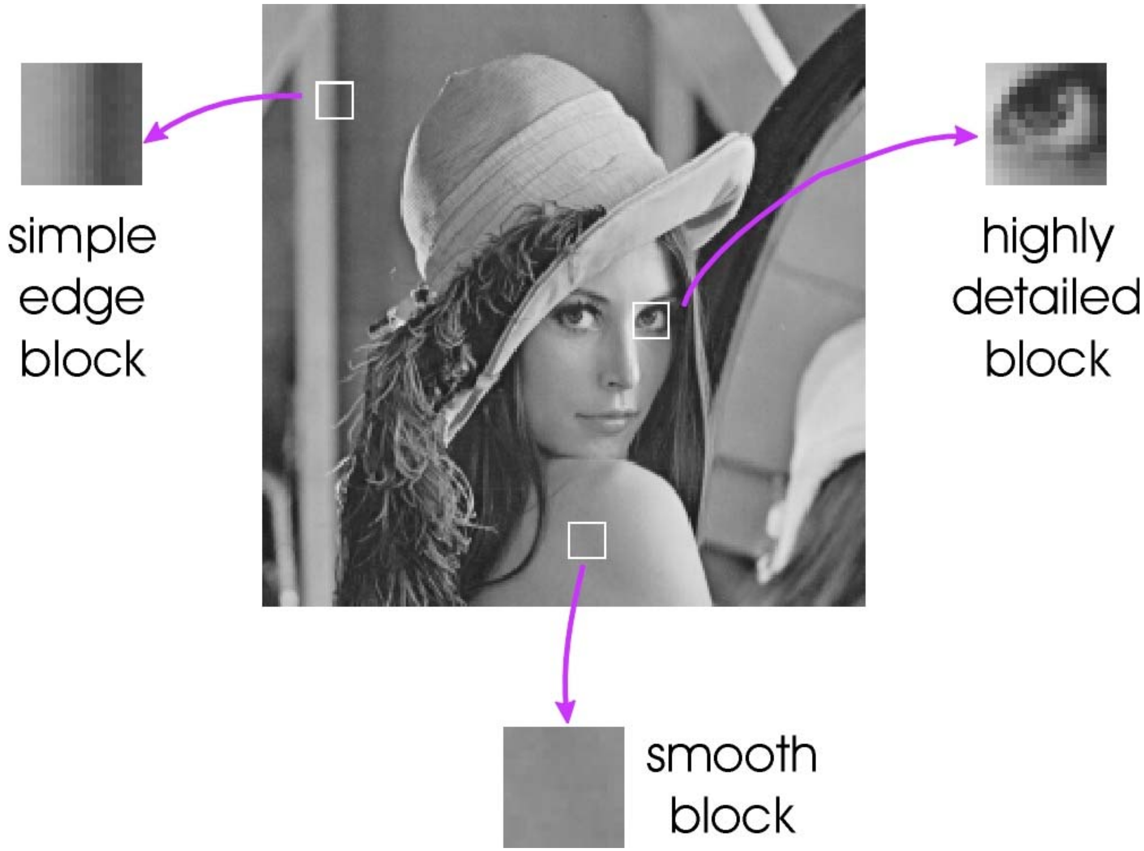
Higher freq.
components
in the *x* direction



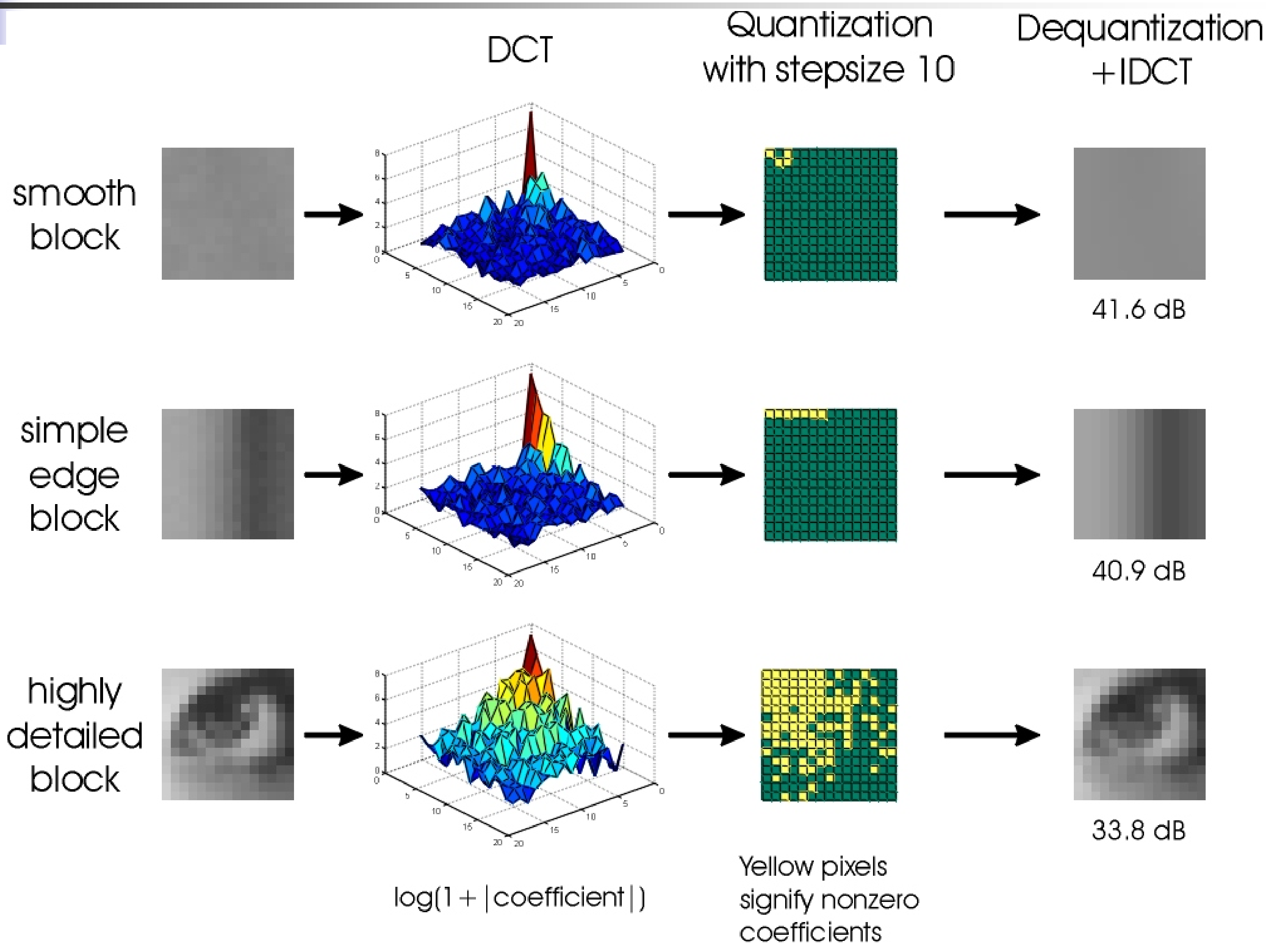
Higher freq.
components
in the *x*-*y* directions

Higher freq. components
in the *y* direction

DCT Energy Compaction (1)



DCT Energy Compaction (2)



DCT Energy Compaction (3)

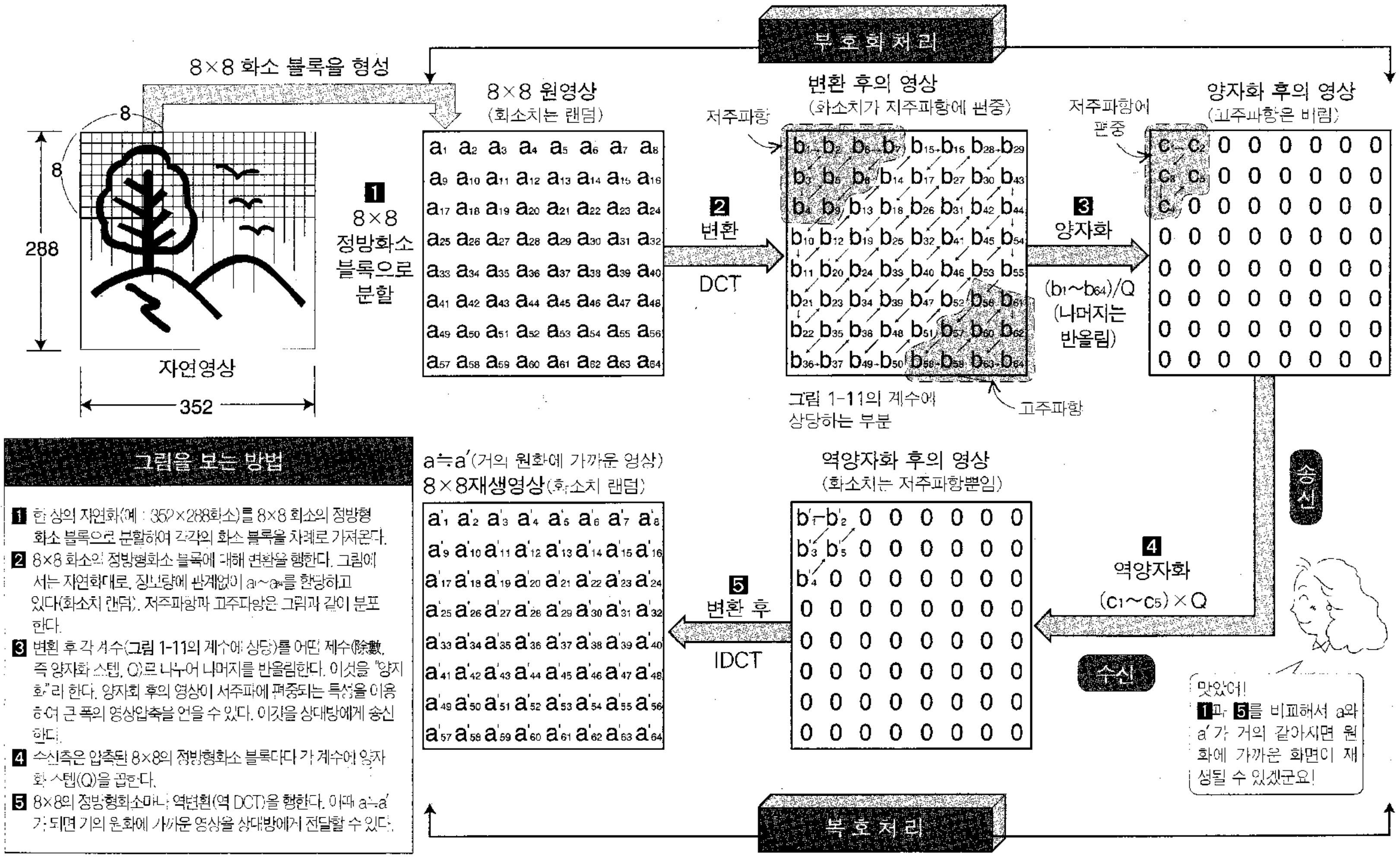


그림 1-12 화면내(공간적) 상관관계에 따른 정보압축의 방법

DCT Energy Compaction (4)

DCT encoding example

$f\,(x,y)$

$$\begin{pmatrix} 79 & 75 & 79 & 82 & 82 & 86 & 94 & 94 \\ 76 & 78 & 76 & 82 & 83 & 86 & 85 & 94 \\ 72 & 75 & 67 & 78 & 80 & 78 & 74 & 82 \\ 74 & 76 & 75 & 75 & 86 & 80 & 81 & 79 \\ 73 & 70 & 75 & 67 & 78 & 78 & 79 & 85 \\ 69 & 63 & 68 & 69 & 75 & 78 & 82 & 80 \\ 76 & 76 & 71 & 71 & 67 & 79 & 80 & 83 \\ 72 & 77 & 78 & 69 & 75 & 75 & 78 & 78 \end{pmatrix}$$

FDCT →

$F\,(u,v)$

$$\begin{pmatrix} 619 & -29 & 8 & 2 & 1 & -3 & 0 & 1 \\ 22 & -6 & -4 & 0 & 7 & 0 & -2 & -3 \\ 11 & 0 & 5 & -4 & -3 & 4 & 0 & -3 \\ 2 & -10 & 5 & 0 & 0 & 7 & 3 & 2 \\ 6 & 2 & -1 & -1 & -3 & 0 & 0 & 8 \\ 1 & 2 & 1 & 2 & 0 & 2 & -2 & -2 \\ -8 & -2 & -4 & 1 & 2 & 1 & -1 & 1 \\ -3 & 1 & 5 & -2 & 1 & -1 & 1 & -3 \end{pmatrix}$$

Quant. →

$F^Q\,(u,v)$

$$\begin{pmatrix} 39 & -3 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Inverse Q & IDCT →

$r\,(x,y)$

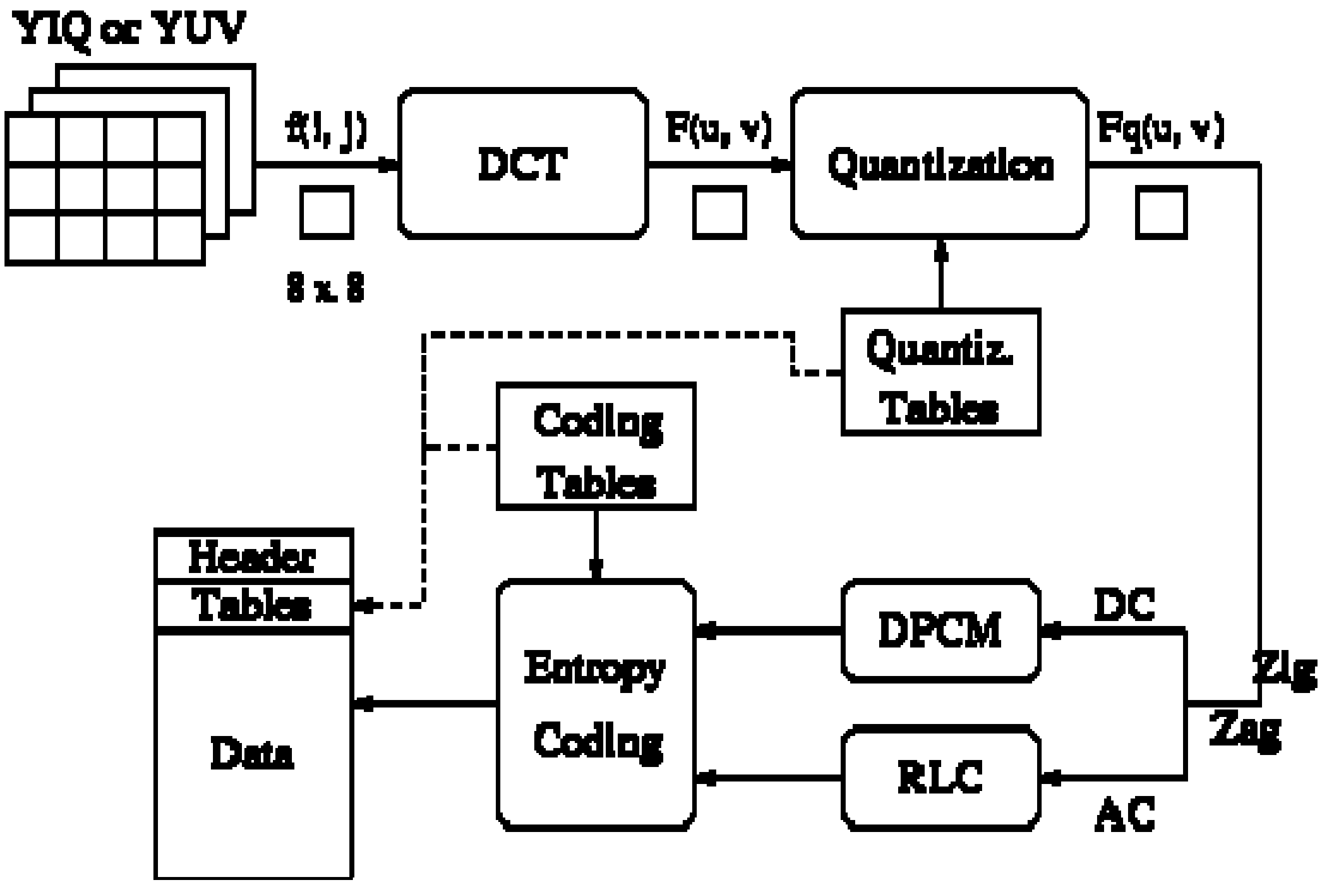
$$\begin{pmatrix} 74 & 75 & 77 & 80 & 85 & 91 & 95 & 98 \\ 77 & 77 & 78 & 79 & 82 & 86 & 89 & 91 \\ 78 & 77 & 77 & 77 & 78 & 81 & 83 & 84 \\ 74 & 74 & 74 & 74 & 76 & 78 & 81 & 82 \\ 69 & 69 & 70 & 72 & 75 & 78 & 82 & 84 \\ 68 & 68 & 69 & 71 & 75 & 79 & 82 & 85 \\ 73 & 73 & 72 & 73 & 75 & 77 & 80 & 81 \\ 78 & 77 & 76 & 75 & 74 & 75 & 76 & 77 \end{pmatrix}$$

→

$e\,(x,y) = f\,(x,y) - r\,(x,y)$

$$\begin{pmatrix} 5 & 0 & 2 & 2 & -3 & -5 & -1 & -4 \\ -1 & 1 & -2 & 3 & 1 & 0 & -4 & 1 \\ -6 & -2 & -10 & 1 & 2 & -3 & -9 & -2 \\ 0 & 2 & 1 & 1 & 10 & 2 & 0 & -3 \\ 4 & 1 & 5 & -5 & 3 & 0 & -3 & 1 \\ 1 & -5 & -1 & -2 & 0 & -1 & 0 & -5 \\ 3 & 3 & -1 & -2 & -8 & 2 & 0 & 2 \\ -6 & 0 & 2 & -6 & 1 & 0 & 2 & 1 \end{pmatrix}$$

JPEG Encoding Structure



Quantization

- **Quantization table**
  - No default values for quantization tables
  - Application may specify the tables
  - Q(u, v) : quantization table
    integer value from 1 to 255

$$Quantization \quad : F^{Q}(u,v) = round\left(\frac{F(u,v)}{Q(u,v)}\right)$$

$$Dequantization: \quad R(u,v) = F^{Q}(u,v) \times Q(u,v)$$

Table 8.1 Luminance quantization matrix Q_{uv} (example only) [367]

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

Source: © 1993 ITU-T.

Table 8.2 Chrominance quantization matrix Q_{uv} (example only) [367]

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

Source: © 1993 ITU-T.

Zigzag Scan

- Zigzag scan of DCT coefficients

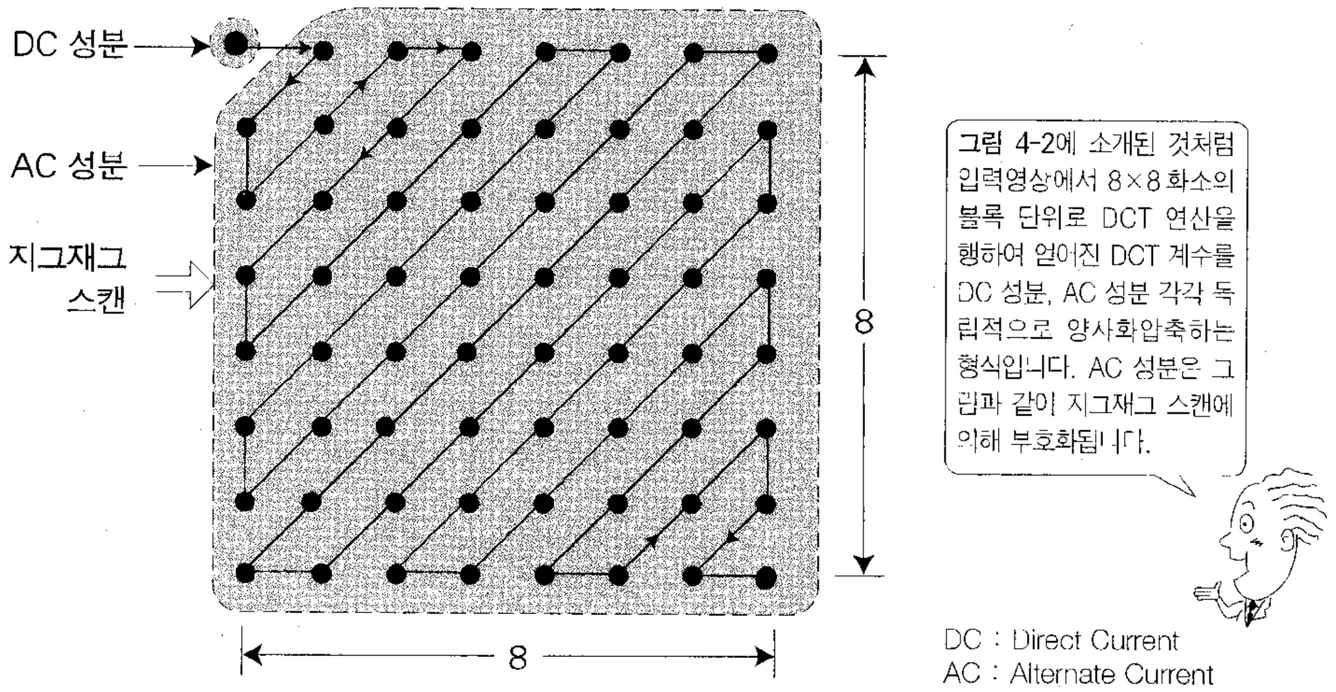
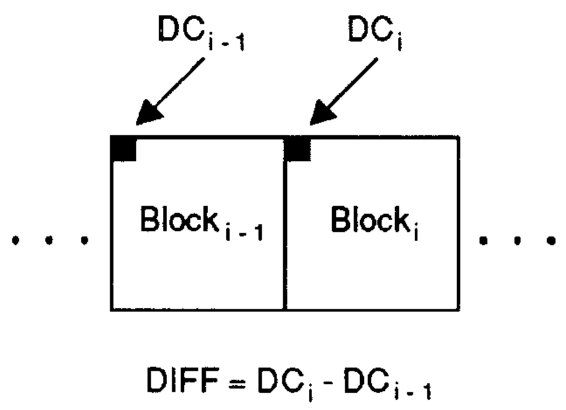


그림 4-3 DCT 부호화방식에 있어서 DCT 계수의 DC 성분과 AC 성분

RLE on AC components

- 8x8 DCT block has many zeros for AC components.
  - Number of zeros are encoded
- RLE Construction
  - *Skip*: number of zeros
  - *Value*: next non-zero component
  - (0,0) end-of-block
- DC coefficients
  - DPCM coding



Entropy Coding

- Huffman coding
  - Variable length coding (VLC)
  - Huffman tables (2 AC and DC tables for baseline)

Table 8.3 Difference categories for DC coding [367]

SSSS	DIFF values
0	0
1	$-1, 1$
2	$-3, -2, 2, 3$
3	$-7 \cdots -4, 4 \cdots 7$
4	$-15 \cdots -8, 8 \cdots 15$
5	$-31 \cdots -16, 16 \cdots 31$
6	$-63 \cdots -32, 32 \cdots 63$
7	$-127 \cdots -64, 64 \cdots 127$
8	$-255 \cdots -128, 128 \cdots 255$
9	$-511 \cdots -256, 256 \cdots 511$
10	$-1023 \cdots -512, 512 \cdots 1023$
11	$-2047 \cdots -1024, 1024 \cdots 2047$

Source: © 1993 ITU-T.

Table 8.4 Huffman code table for luminance and chrominance DC difference [367]

	Luminance DC		Chrominance DC	
SSSS	Code length	Codeword	Code length	Codeword
0	2	00	2	00
1	3	010	2	01
2	3	011	2	10
3	3	100	3	110
4	3	101	4	1110
5	3	110	5	11110
6	4	1110	6	111110
7	5	11110	7	1111110
8	6	111110	8	11111110
9	7	1111110	9	111111110
10	8	11111110	10	1111111110
11	9	111111110	11	11111111110

Source: © 1993 ITU-T.

JPEG Modes

- Sequential mode
  - Left-to-right, top-to-bottom scan
  - Baseline sequential mode use Huffman coding
- Lossless mode
  - Spatial prediction + Huffman coding
- Progressive mode
  - Spectral selection:
    - Send DC and few AC coeff's first and gradually some more ACs
  - Successive approximation:
    - send DCT coeff's MSB to LSB

Compression Results (1)

Original

image

(24bpp)

JPEG
Compressed
image

(8:1 -- 3bpp)

JPEG
Compressed
image

(32:1 --

0.75bpp)

JPEG
Compressed
image

(128:1 --

0.1875bpp)

Compression Results (2)

QF=5: 9438 bytes

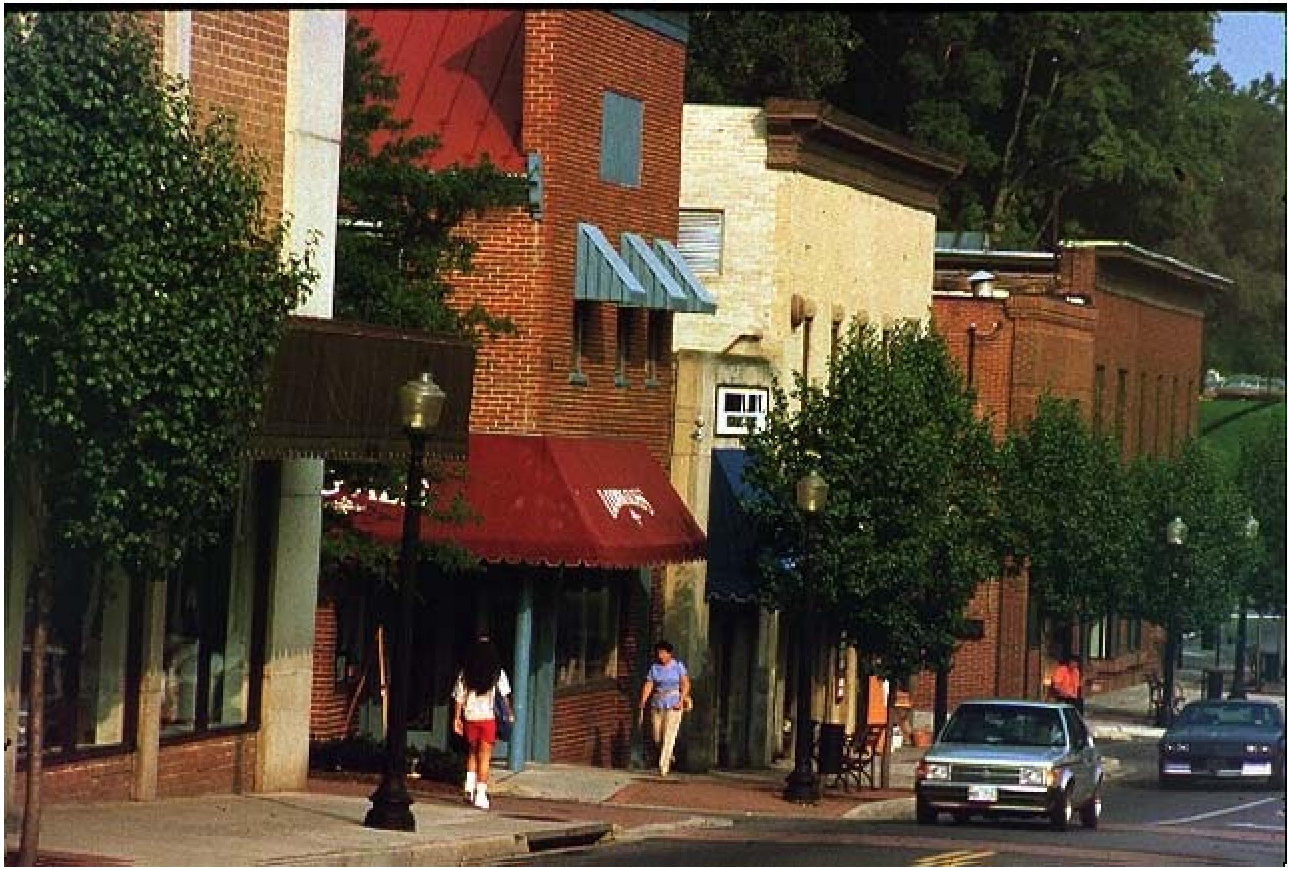
QF=10: 15325 bytes

QF=25: 29360 bytes

QF=50: 46295 bytes

QF=75: 70586 bytes

QF=100: 326321 bytes



No compression: 781300 bytes

Haar Transform (Filter)

Haar transform (1)

- Haar function
  - non-sinusoidal periodic, orthonormal, complete

$$haar(0,0,t) = \frac{1}{\sqrt{N}}, \qquad t \in (0,1)$$

$$haar(r,m,t) = \begin{cases} 2^{\frac{r}{2}} \Big/ \sqrt{N} & \frac{m-1}{2^r} \le t < \frac{m-\frac{1}{2}}{2^r} \\ 2^{-\frac{r}{2}} \Big/ \sqrt{N} & \frac{m-\frac{1}{2}}{2^r} \le t < \frac{m}{2^r} \\ 0 & \text{elsewhere for } t \in (0,1) \end{cases}$$

$$\Rightarrow \text{Haar transform is obtained by letting } t = \frac{m}{N}, \quad m = 0,1,\cdots,N-1$$

Haar transform (2)

- Basis Functions (N = 16)
  - One cycle of harmonic waveforms
  - How much energy for the specific frequency exists at a local position

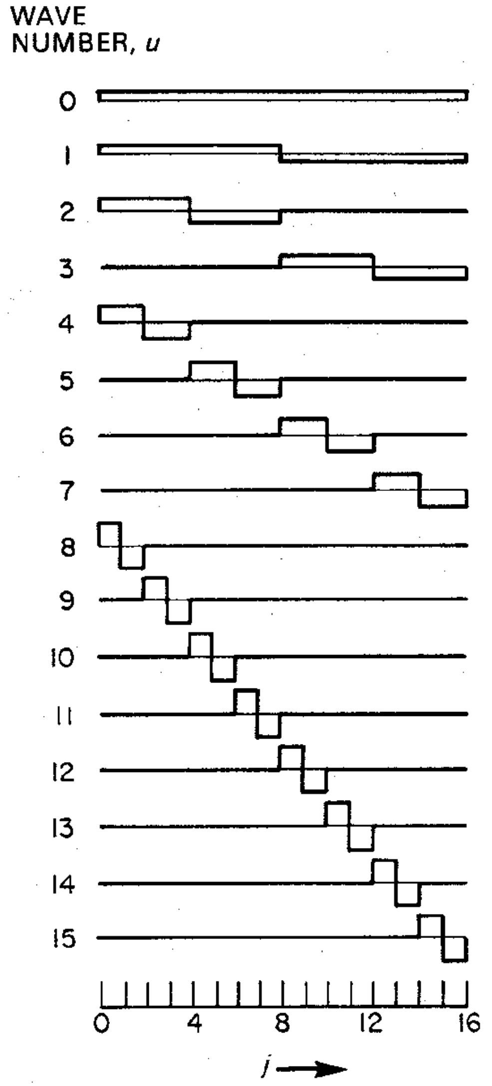


FIGURE 8.4-4. Haar transform basis functions, $N = 16$.

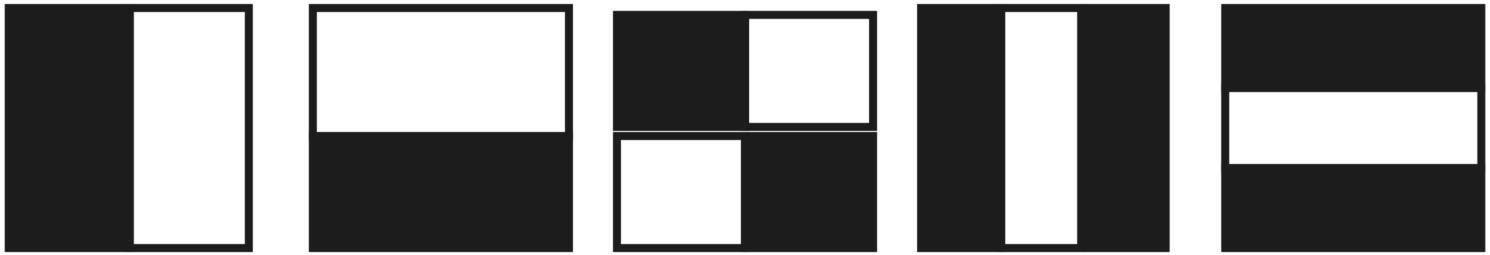
Haar transform (3)

$$\mathbf{H}_4 = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

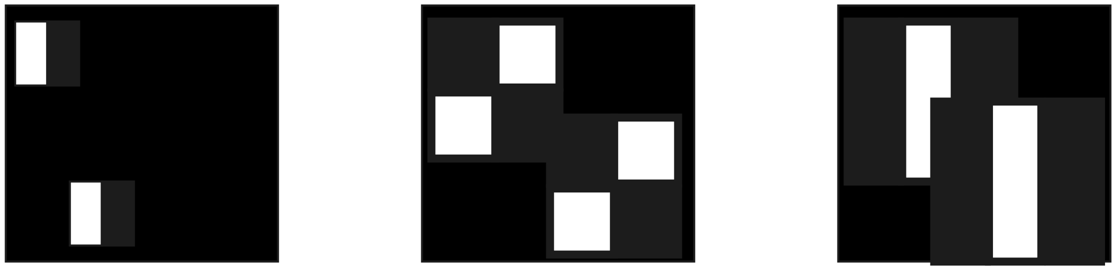
$$\mathbf{H}_8 = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$

2-D Haar-like transform (1)

- 2-D Haar-like Feature Bank (Location & Size)
  - Various filters with respect to locations & sizes
  - For pattern analysis

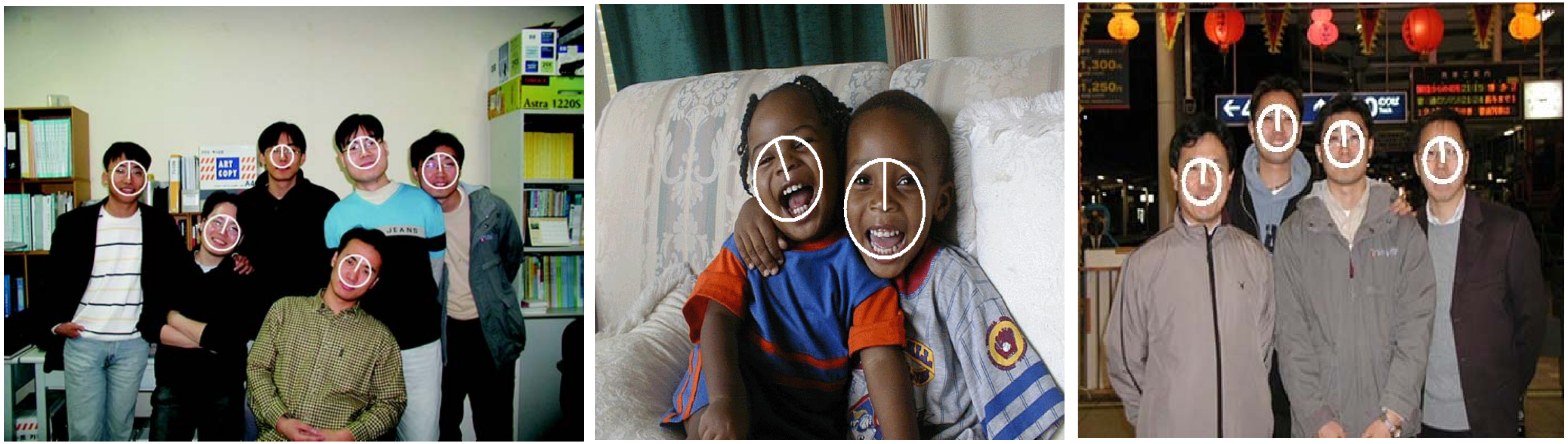


EX:



2-D Haar-like transform (2)

- Linear combination 2-D Haar-like filters for face
  - openCV: face detection



Haar 필터 – 패턴 인식 (얼굴)

ART (Angular Radial Transform)

ART (1)

- MPEG-7: Region-based shape descriptor
- suitable for shape analysis

Basis function of ART in the polar coordinates

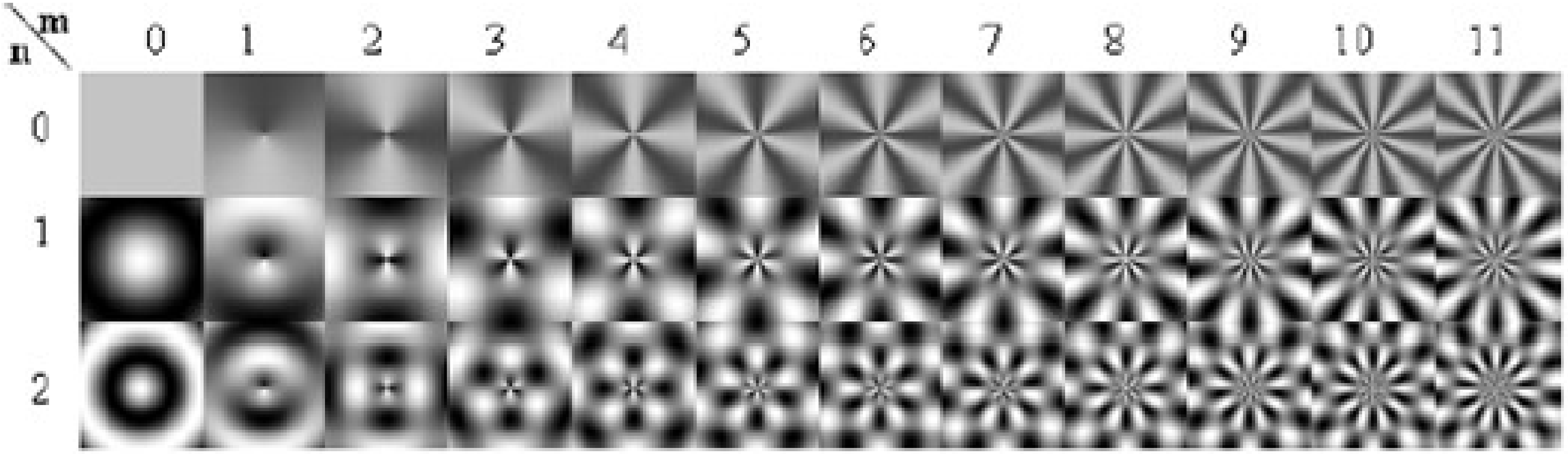
$$ART_{nm} = \int_0^{2\pi}\int_0^1 A_m(\theta)R_n(\rho)I(\rho,\theta)\rho d\rho d\theta$$

$$A_m(\theta) = \frac{1}{2\pi}\exp(jm\theta),$$

$$R_n(\rho) = \begin{cases} 1, & n = 0, \\ 2\cos(\pi n\rho), & n \neq 0. \end{cases}$$

ART (1)

- Usually use 36 patterns (3 radial, 12 angular frequencies)
- 36-D vector descriptor of magnitude



3 radial, 12 angular ART Basis patterns

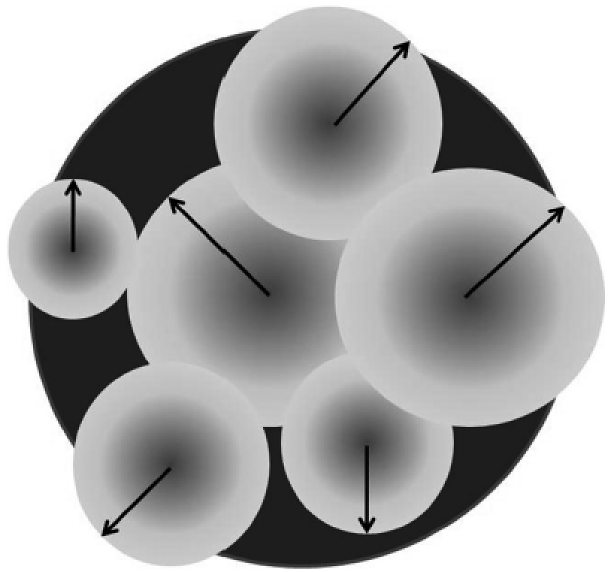
ART Application (1)

- Human shape recognition and classification
  - Using 2000 human shapes DB and 36-D ART vectors
- Design the classifier based on
  - 36-D ART mean vectors
  - Distance variance of each cluster

$$m_d = E\left[\left\|\mathbf{M}_{ART} - \mathbf{DB}_{ART}\right\|\right],$$

$$\sigma_d^2 = Var\left[\left\|\mathbf{M}_{ART} - \mathbf{DB}_{ART}\right\|\right].$$

$$T_h = m_d + \alpha\sigma_d.$$



Clustered 36D ART vector space

ART Application (2)

- Example: Shape (human) recognition

True-positive recognition

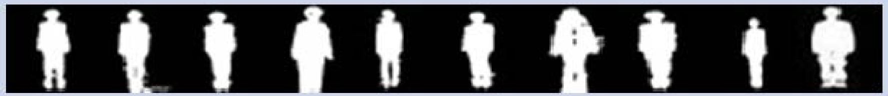
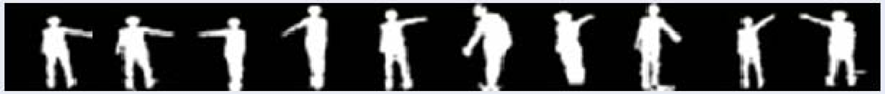
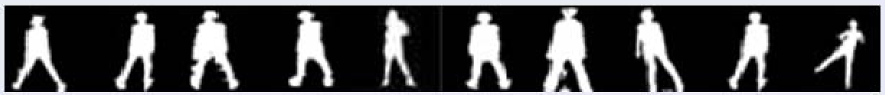
True-negative recognition

False-negative errors

False-positive errors

ART Application (3)

- Vector clustering of ART vectors
  - Multi-modes modeling of ART vector space
  - Human pose classification

class	Samples of DB	# of class	α
1		445	0.4
2		321	0.3
3		714	0.5
4		398	0.4
5		546	0.4
6		255	0.3