Chap. 14 Image Compression

1. Fundamentals

- information theory, fidelity criteria(RMSE, MS-SNR)
- compression standards and formats

2. Basic Compression Methods

- discrete cosine transform (DCT)
- run-length coding
- Huffman coding, arithmetic coding

3. Joint Photographic Experts Group(JPEG)

Image & Video Source Coding

- Source coding: **lossless** (error free) or **lossy**.
- Image/video compression standards(e.g. JPEG, MPEG, H.263) use a combination of lossy and lossless compression algorithms.
- Lossless:
  - Run-Length coding
  - Huffman coding
  - Arithmetic coding
  - Lempel-Ziv-Welch (LZW) coding
- Lossy: DCT, wavelet, fractal, etc
- Standards: JPEG, JPEG-2000, JBIG-1, JBIG-2

Elements of Information Theory

- Notations
  - a set of source symbols $\{a_1, a_2, \cdots, a_J\}$
  - with event occurrence probability $z = \{P(a_j)\}$.
- Self Information : *I*
  - generated by each source symbol,
  - indication of the minimum number of bits required to encode this symbol.

$$\boxed{I(a_j)} = \log_2\left(\frac{1}{P(a_j)}\right) = -\log_2(P(a_j))$$

- high probability => low self information

Self Information: Examples

- Example 1
  - a_0 is the probability of rainy weather
    a_1 is the probability of sunny weather
  - If it is rainy, $a_0 = 1$ and $a_1 = 0$. => $I(a_0) = 0$, $I(a_1) = 1$
  - $a_0 = 0.5$, $a_1 = 0.5$:
    $I(a_0) = I(a_1) = \log_2(1/0.5) = 1$ bit

- Example 2
  - If $a_0=a_1=a_2=a_3=0.25$, $I(a_j) = ?$

log_2(1/0.25) = 2

Elements of Information Theory

- Entropy : *H*
  - uncertainty
  - average number of bits (= average of *I*, or average amount of information) associated with a set of source symbols

$$\boxed{H(z)} = \sum_{j=1}^{J} P(a_j)\boxed{I(a_j)} = -\sum_{j=1}^{J} P(a_j)\log_2(P(a_j))$$

- Example: $P(a_1)=2/3$, $P(a_2)=1/3$
  - self information of a_i ? ($\log_2 1.5 = 0.59$, $\log_2 3 = 1.58$)
  - entropy (*H*) ? 0.59x(2/3)+1.58x(1/3) = 0.92
- Entropy coding : Huffman, Arithmatic, etc

Fidelity Criteria

- simple and convenient way to evaluate information loss

- root mean square error (RMSE) 작아야 좋음
  - root of mean of square of error

편차의 제곱의 평균의 제곱근

$$e_{rms} = \left[\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\hat{f}(x,y) - f(x,y) \right]^2 \right]^{\frac{1}{2}}$$

- mean-square signal-to-noise ratio 큰게 좋음

$$SNR_{ms} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \hat{f}(x,y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\hat{f}(x,y) - f(x,y) \right]^2}$$

Compression Standards

Image Compression Standards, Formats, and Containers

- **Still Image**
  - **Binary**
    - CCITT Group 3
    - CCITT Group 4
    - JBIG (or JBIG1)
    - JBIG2
    - TIFF
  - **Continuous Tone**
    - JPEG
    - JPEG-LS
    - JPEG-2000
    - BMP
    - GIF
    - PDF
    - PNG
    - TIFF
- **Video**
  - DV
  - H.261
  - H.262
  - H.263
  - H.264
  - MPEG-1
  - MPEG-2
  - MPEG-4
  - MPEG-4 AVC
  - AVS
  - HDV
  - M-JPEG
  - QuickTime
  - VC-1 (or WMV9)

요즘 쓰는 것은 거의 다 이것
중요하지 않으니 패스

Compression Standards

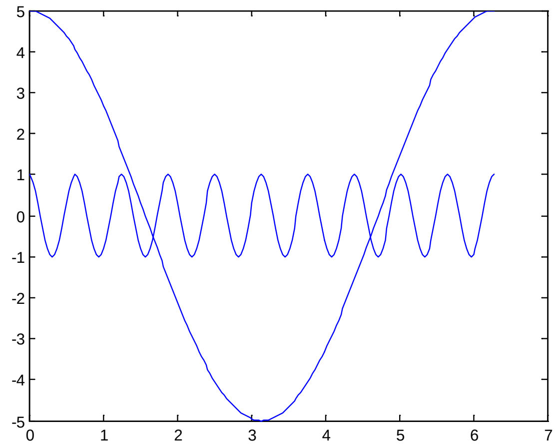
Name	Organization	Description
Bi-Level Still Images		
CCITT Group 3	ITU-T	Designed as a facsimile (FAX) method for transmitting binary documents over telephone lines. Supports 1-D and 2-D run-length [8.2.5] and Huffman [8.2.1] coding.
CCITT Group 4	ITU-T	A simplified and streamlined version of the CCITT Group 3 standard supporting 2-D run-length coding only.
JBIG *or* JBIG1	ISO/IEC/ ITU-T	A *Joint Bi-level Image Experts Group* standard for progressive, lossless compression of bi-level images. Continuous-tone images of up to 6 bits/pixel can be coded on a bit-plane basis [8.2.7]. Context sensitive arithmetic coding [8.2.3] is used and an initial low resolution version of the image can be gradually enhanced with additional compressed data.
JBIG2	ISO/IEC/ ITU-T	A follow-on to JBIG1 for bi-level images in desktop, Internet, and FAX applications. The compression method used is content based, with dictionary based methods [8.2.6] for text and halftone regions, and Huffman [8.2.1] or arithmetic coding [8.2.3] for other image content. It can be lossy or lossless.
Continuous-Tone Still Images		
JPEG	ISO/IEC/ ITU-T	A *Joint Photographic Experts Group* standard for images of photographic quality. Its lossy *baseline coding system* (most commonly implemented) uses quantized discrete cosine transforms (DCT) on 8 × 8 image blocks [8.2.8], Huffman [8.2.1], and run-length [8.2.5] coding. It is one of the most popular methods for compressing images on the Internet.
JPEG-LS	ISO/IEC/ ITU-T	A lossless to near-lossless standard for continuous tone images based on adaptive prediction [8.2.9], context modeling [8.2.3], and Golomb coding [8.2.2].
JPEG-2000	ISO/IEC/ ITU-T	A follow-on to JPEG for increased compression of photographic quality images. Arithmetic coding [8.2.3] and quantized discrete wavelet transforms (DWT) [8.2.10] are used. The compression can be lossy or lossless.

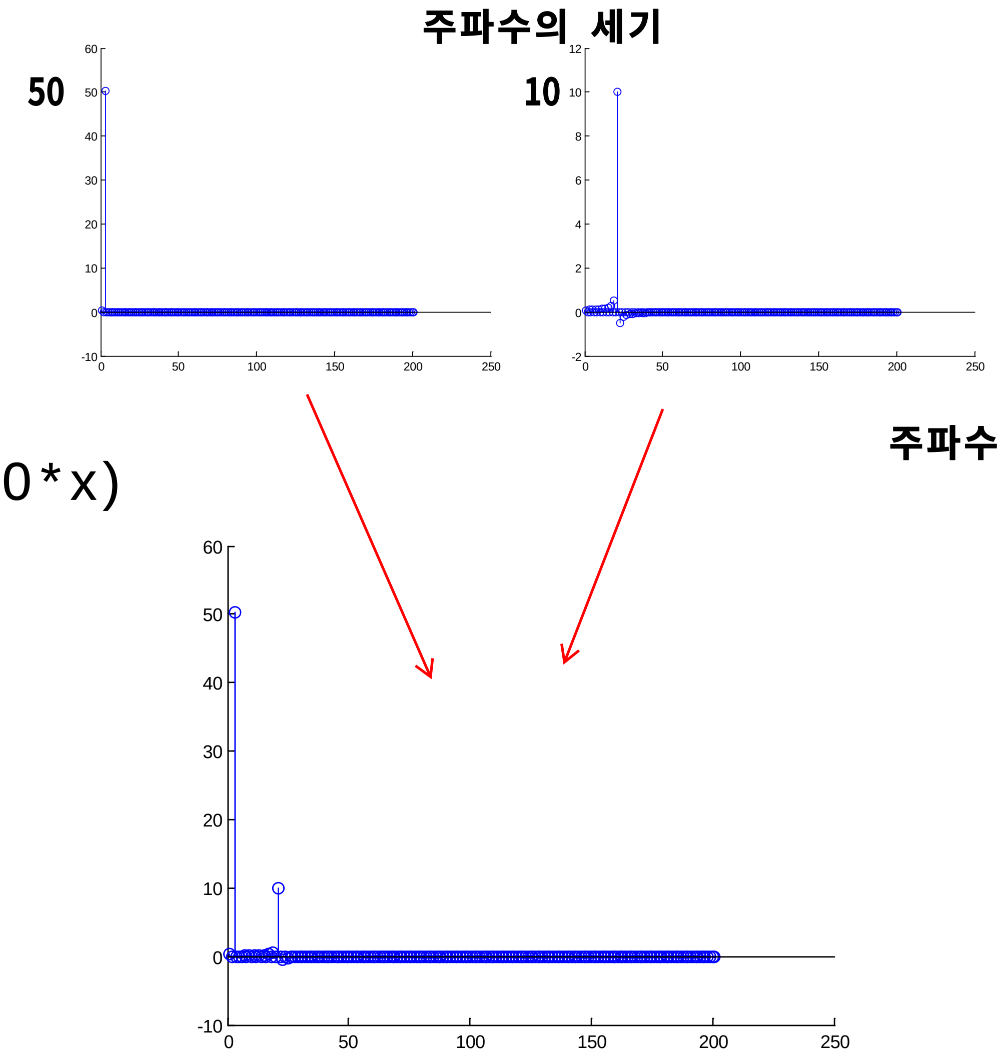
(*Continues*)

Name	Organization	Description
Continuous-Tone Still Images		
BMP	Microsoft	*Windows Bitmap*. A file format used mainly for simple uncompressed images.
GIF	CompuServe	*Graphic Interchange Format*. A file format that uses lossless LZW coding [8.2.4] for 1- through 8-bit images. It is frequently used to make small animations and short low resolution films for the World Wide Web.
PDF	Adobe Systems	*Portable Document Format*. A format for representing 2-D documents in a device and resolution independent way. It can function as a container for JPEG, JPEG 2000, CCITT, and other compressed images. Some PDF versions have become ISO standards.
PNG	*World Wide Web Consortium* (W3C)	*Portable Network Graphics*. A file format that losslessly compresses full color images with transparency (up to 48 bits/pixel) by coding the difference between each pixel's value and a predicted value based on past pixels [8.2.9].
TIFF	Aldus	*Tagged Image File Format*. A flexible file format supporting a variety of image compression standards, including JPEG, JPEG-LS, JPEG-2000, JBIG2, and others.

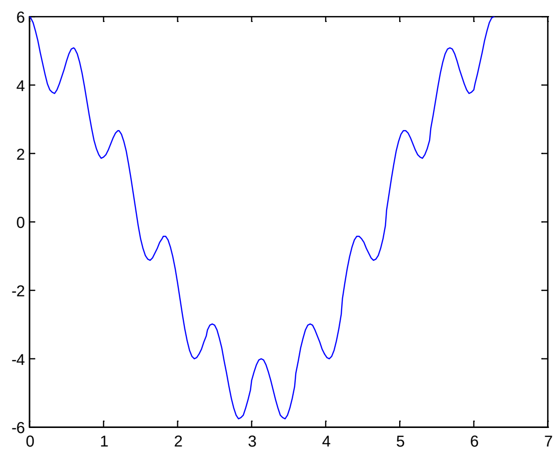
단순영상에서는 lossless한 방법이면서도 효율이 매우 좋아서 살아남음

1D Discrete Cosine Transform (DCT)





y1 = 5*cos(x), y2 = cos(10*x)



y1+y2

1D Discrete Cosine Transform

$$T(u)=\sum_{x=0}^{n-1} g(x)r(x,u)=\sum_{x=0}^{n-1} g(x)\left[e^{-j\frac{2\pi ux}{n}}\right] \qquad g(x)=\sum_{u=0}^{n-1} T(u)s(x,u)=\sum_{u=0}^{n-1} T(u)\left[\frac{1}{n^2}e^{j\frac{2\pi ux}{n}}\right]$$

$$T(u)=\sum_{x=0}^{n-1} g(x)r(x,u)=\sum_{x=0}^{n-1} g(x)\alpha(u)\cos\left(\frac{(2x+1)u\pi}{2n}\right)$$

$$g(x)=\sum_{u=0}^{n-1} T(u)s(x,u)=\sum_{u=0}^{n-1} T(u)\alpha(u)\cos\left(\frac{(2x+1)u\pi}{2n}\right)$$

$$\alpha(0)=\sqrt{\frac{1}{n}}, \quad \alpha(u)=\sqrt{\frac{2}{n}}, \quad u=1,2,\cdots,n-1$$

- u : index of DCT coefficients
- x : index of discrete data (e.g. image signal)
- n : number of discrete data

1D DCT Example

$$T(u)=\sum_{x=0}^{n-1} g(x)r(x,u)=\sum_{x=0}^{n-1} g(x)\alpha(u)\cos\left(\frac{(2x+1)u\pi}{2n}\right)$$

$$\alpha(0)=\sqrt{\frac{1}{n}},\quad \alpha(u)=\sqrt{\frac{2}{n}},\quad u=1,2,\cdots,n-1$$

$g(x) = \{ 1, 1, 1, 1 \}$ => $n = 4$

$T(0) = 0.500000 * (g(0) * \cos((0 * \pi)/8) + g(1) * \cos((0 * \pi)/8)$
$+ g(2) * \cos((0 * \pi)/8) + g(3) * \cos((0 * \pi)/8))$

=> DC coefficient

$T(1) = 0.707107 * (g(0) * \cos((1 * \pi)/8) + g(1) * \cos((3 * \pi)/8)$
$+ g(2) * \cos((5 * \pi)/8) + g(3) * \cos((7 * \pi)/8))$

$T(2) = 0.707107 * (g(0) * \cos((2 * \pi)/8) + g(1) * \cos((6 * \pi)/8)$
$+ g(2) * \cos((10 * \pi)/8) + g(3) * \cos((14 * \pi)/8))$

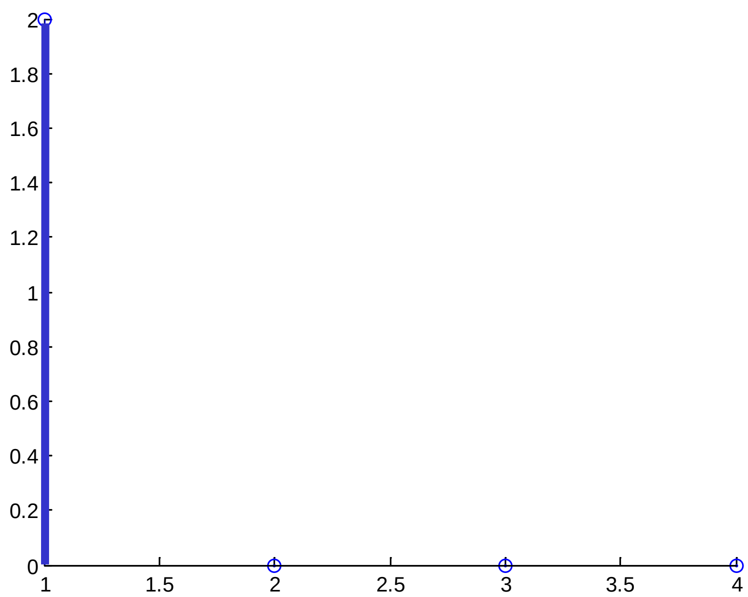
$T(3) = 0.707107 * (g(0) * \cos((3 * \pi)/8) + g(1) * \cos((9 * \pi)/8)$
$+ g(2) * \cos((15 * \pi)/8) + g(3) * \cos((21 * \pi)/8))$

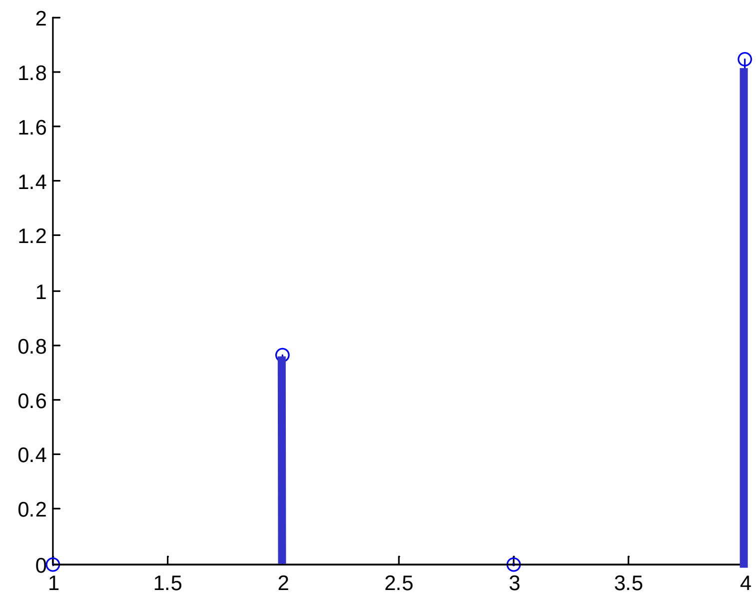
$T(u) = \{ 2, 0, 0, 0 \}$

1D DCT Example

$$g(x) = \{\,1,\ 1,\ 1,\ 1\,\} <=> T(u) = \{2, 0, 0, 0\}$$

$$g(x) = \{\,1, -1, 1, -1\,\} <=> T(u) = \{0, 0.7654, 0, 1.8478\}$$





1D DCT Example

$g(x) = \{ 1, 1, 1, 1, 1, 1, 1, 1 \}$ ==> $N = 8$

$T(0) = 0.353553 * (g(0) * \cos((0 * \pi)/16 + g(1) * \cos((0 * \pi)/16 + g(2) * \cos((0 * \pi)/16 + g(3) * \cos((0 * \pi)/16$
$+ g(4) * \cos((0 * \pi)/16 + g(5) * \cos((0 * \pi)/16 + g(6) * \cos((0 * \pi)/16 + g(7) * \cos((0 * \pi)/16)$

$T(1) = 0.500000 * (g(0) * \cos((1 * \pi)/16 + g(1) * \cos((3 * \pi)/16 + g(2) * \cos((5 * \pi)/16 + g(3) * \cos((7 * \pi)/16$
$+ g(4) * \cos((9 * \pi)/16 + g(5) * \cos((11 * \pi)/16 + g(6) * \cos((13 * \pi)/16 + g(7) * \cos((15 * \pi)/16)$

$T(2) = 0.500000 * (g(0) * \cos((2 * \pi)/16 + g(1) * \cos((6 * \pi)/16 + g(2) * \cos((10 * \pi)/16 + g(3) * \cos((14 * \pi)/16$
$+ g(4) * \cos((18 * \pi)/16 + g(5) * \cos((22 * \pi)/16 + g(6) * \cos((26 * \pi)/16 + g(7) * \cos((30 * \pi)/16)$

$T(3) = 0.500000 * (g(0) * \cos((3 * \pi)/16 + g(1) * \cos((9 * \pi)/16 + g(2) * \cos((15 * \pi)/16 + g(3) * \cos((21 * \pi)/16$
$+ g(4) * \cos((27 * \pi)/16 + g(5) * \cos((33 * \pi)/16 + g(6) * \cos((39 * \pi)/16 + g(7) * \cos((45 * \pi)/16)$

$T(4) = 0.500000 * (g(0) * \cos((4 * \pi)/16 + g(1) * \cos((12 * \pi)/16 + g(2) * \cos((20 * \pi)/16 + g(3) * \cos((28 * \pi)/16$
$+ g(4) * \cos((36 * \pi)/16 + g(5) * \cos((44 * \pi)/16 + g(6) * \cos((52 * \pi)/16 + g(7) * \cos((60 * \pi)/16)$

$T(5) = 0.500000 * (g(0) * \cos((5 * \pi)/16 + g(1) * \cos((15 * \pi)/16 + g(2) * \cos((25 * \pi)/16 + g(3) * \cos((35 * \pi)/16$
$+ g(4) * \cos((45 * \pi)/16 + g(5) * \cos((55 * \pi)/16 + g(6) * \cos((65 * \pi)/16 + g(7) * \cos((75 * \pi)/16)$

$T(6) = 0.500000 * (g(0) * \cos((6 * \pi)/16 + g(1) * \cos((18 * \pi)/16 + g(2) * \cos((30 * \pi)/16 + g(3) * \cos((42 * \pi)/16$
$+ g(4) * \cos((54 * \pi)/16 + g(5) * \cos((66 * \pi)/16 + g(6) * \cos((78 * \pi)/16 + g(7) * \cos((90 * \pi)/16)$

$T(7) = 0.500000 * (g(0) * \cos((7 * \pi)/16 + g(1) * \cos((21 * \pi)/16 + g(2) * \cos((35 * \pi)/16 + g(3) * \cos((49 * \pi)/16$
$+ g(4) * \cos((63 * \pi)/16 + g(5) * \cos((77 * \pi)/16 + g(6) * \cos((91 * \pi)/16 + g(7) * \cos((105 * \pi)/16)$

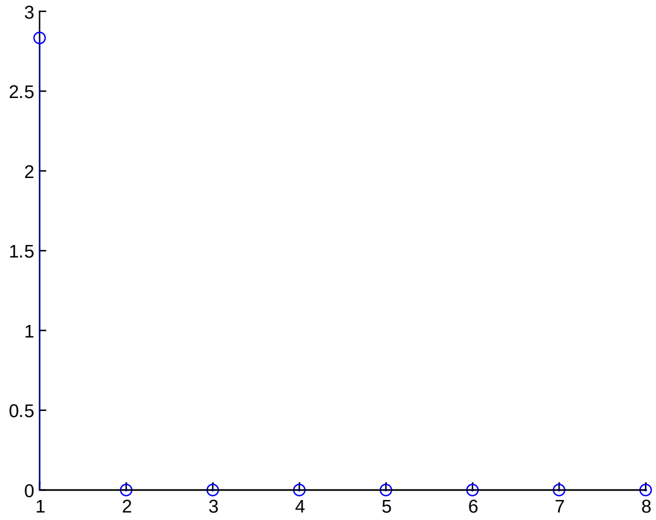
$$T(u) = \{ 2.8284, 0, 0, 0, 0, 0, 0, 0 \}$$

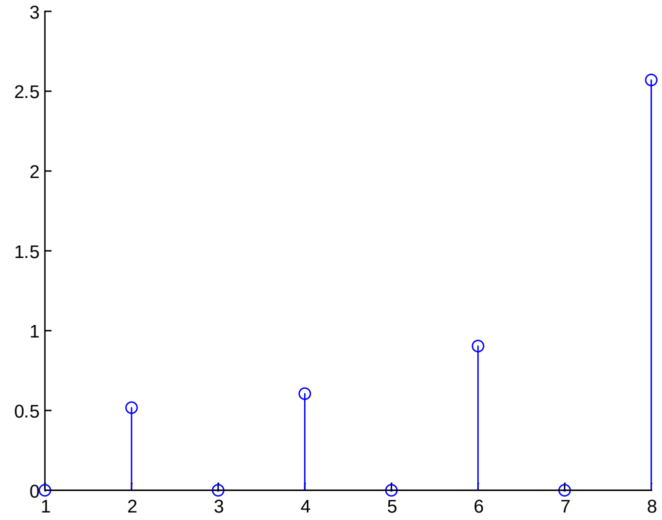
1D DCT Example

$$g(x) = \{1,1,1,1,1,1,1,1\} <=> T(u) = \{2.8284,0,0,0,0,0,0,0\}$$

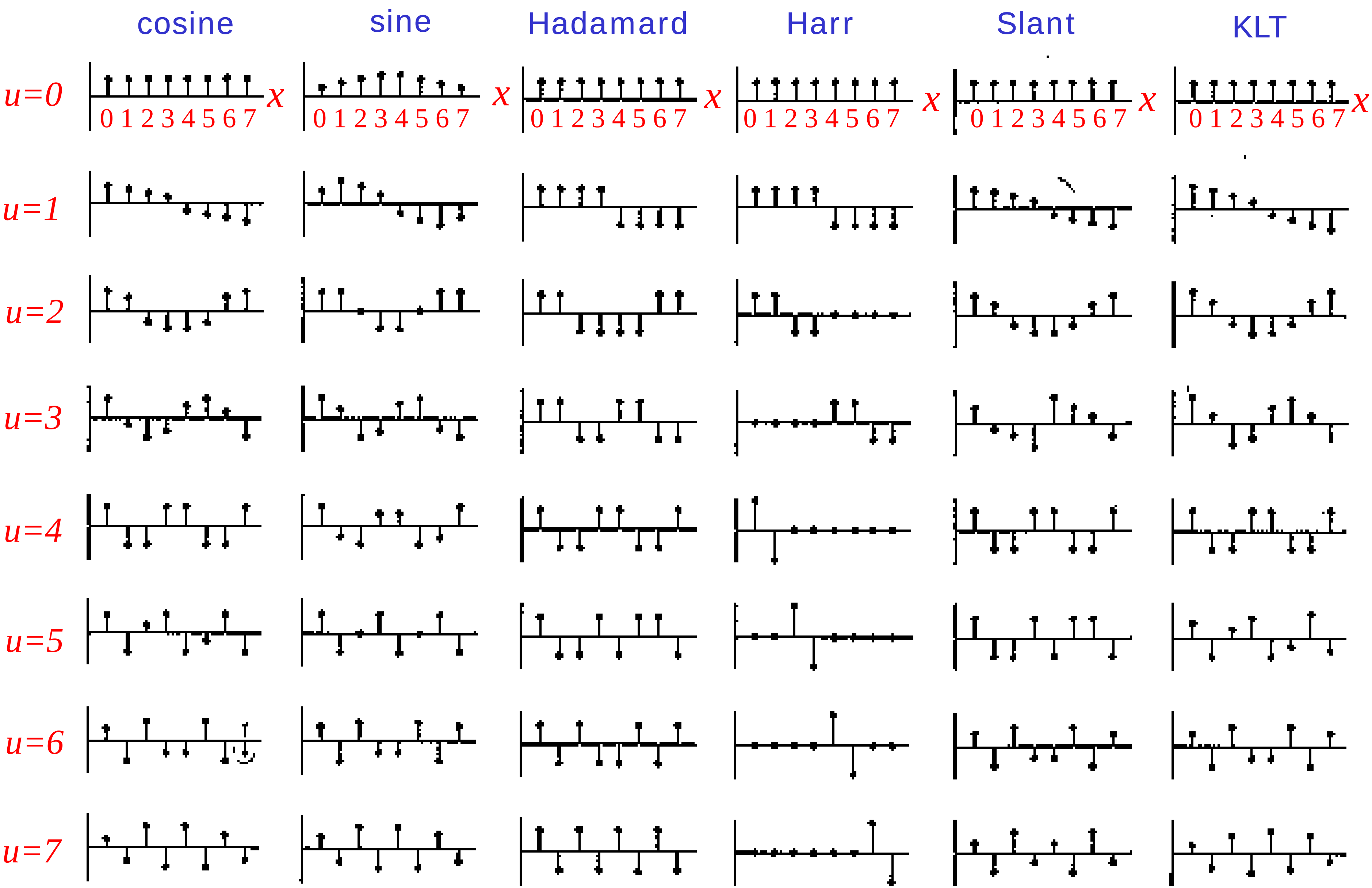
$$g(x) = \{1,-1,1,-1, 1,-1,1,-1\}$$

$$<=> T(u) = \{0, 0.5098, 0, 0.6013, 0, 0.9, 0, 2.5629\}$$

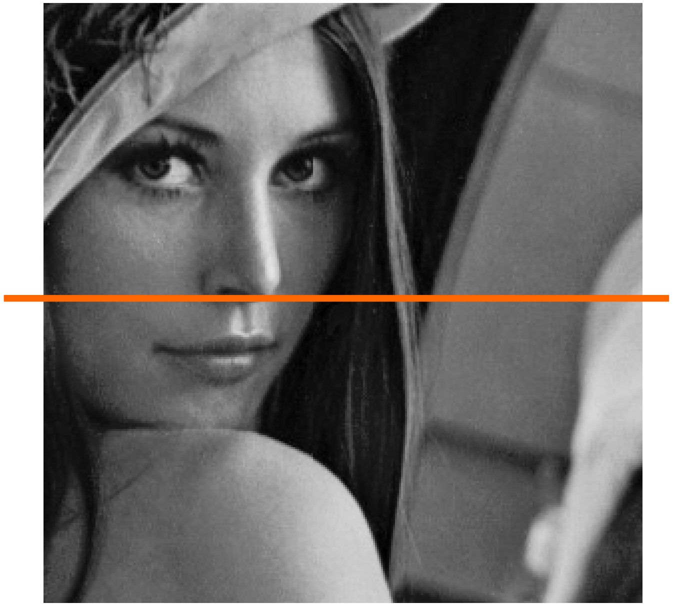


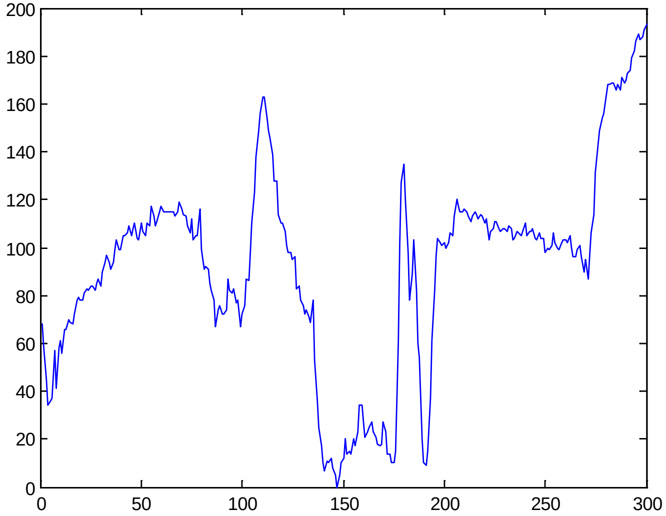


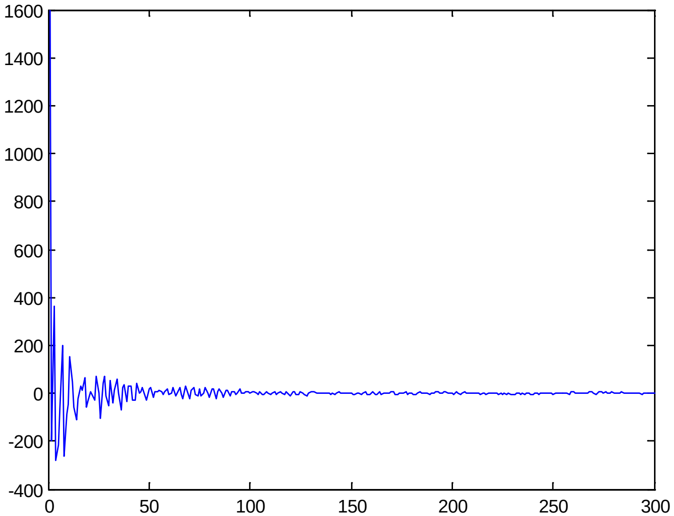
Examples of 1-D $r(x,u)$ or $s(x,u)$



Examples of 1-D *g(x,u)*







- Mostly strong low freq.

Why DCT ?

DFT보다도 훨씬 간단하면서 주파수 영역에 대한 정보를 그대로 가지고 있다.

- When interpreting frequency:
  - DFT (complex) contains too much computation,
  - DWT/DHT (integer) contains too coarse frequency bases,
  - KLT is optimal but is signal dependent.

- DCT maintains the good interpretation of "frequency", while makes it "real" (non-complex).

2D Discrete Cosine Transform (DCT)

$$T(u,v)=\sum_{x=0}^{n-1}\sum_{y=0}^{n-1} g(x,y)r(x,y,u,v) \qquad g(x,y)=\sum_{u=0}^{n-1}\sum_{v=0}^{n-1} T(u,v)s(x,y,u,v)$$

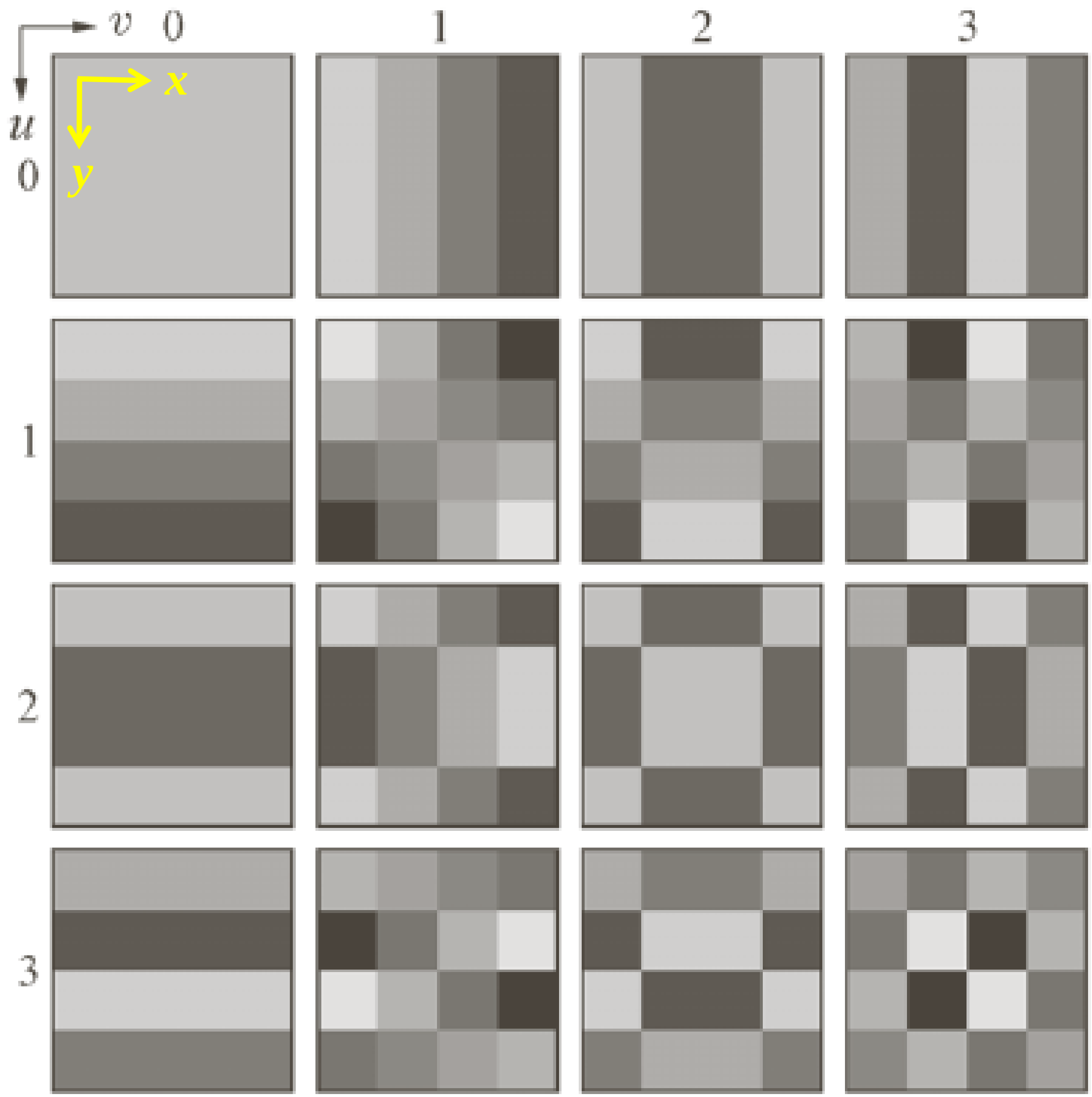
$$r(x,y,u,v)=s(x,y,u,v)=\alpha(u)\alpha(v)\cos\left[\frac{(2x+1)u\pi}{2n}\right]\cos\left[\frac{(2y+1)v\pi}{2n}\right]$$

$$\alpha(u)=\sqrt{\frac{1}{n}} \quad for\ \ u=0, \quad \alpha(u)=\sqrt{\frac{2}{n}} \quad for\ \ u=1,2,\ldots,n-1$$

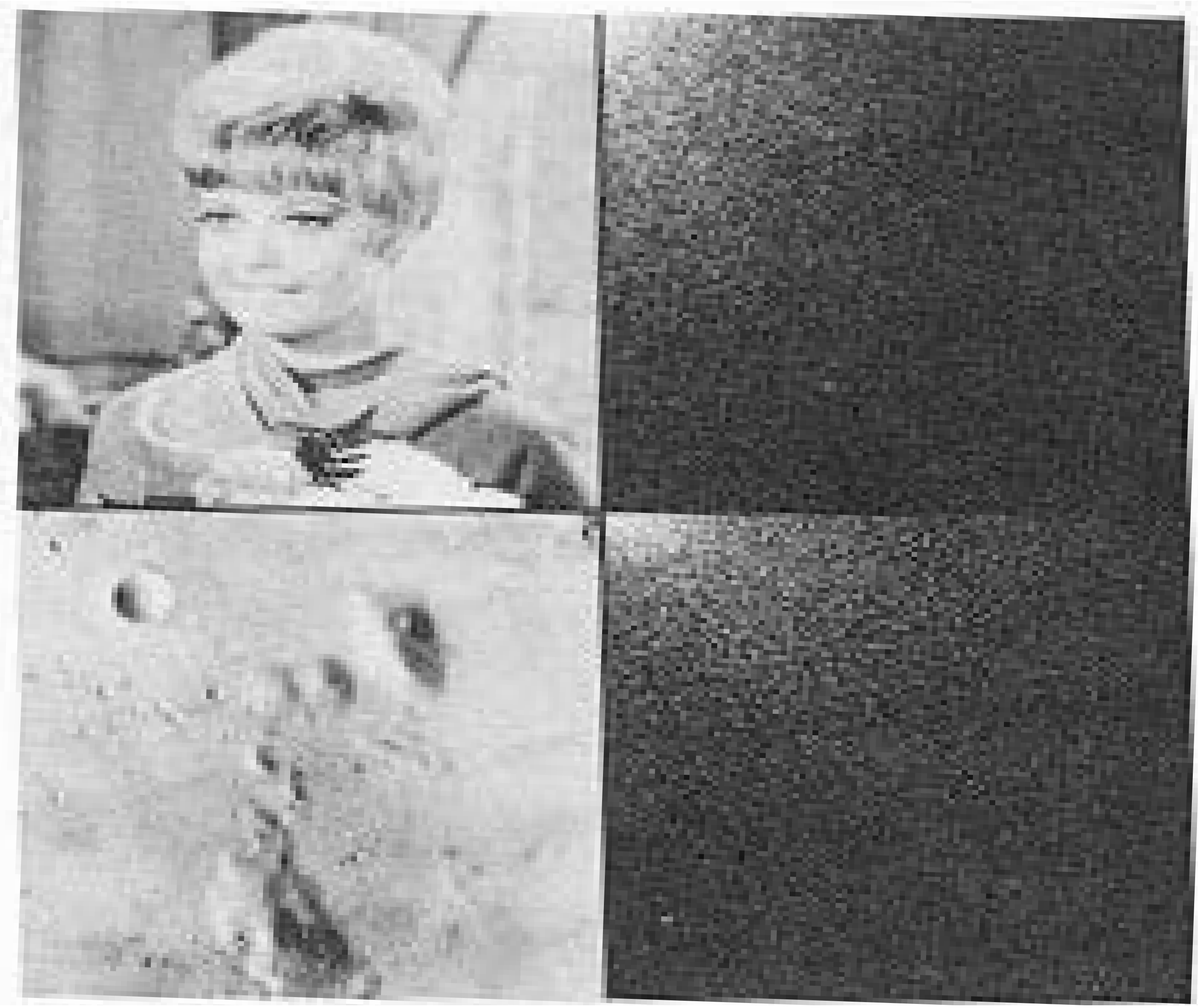
$$\alpha(v)=\sqrt{\frac{1}{n}} \quad for\ \ v=0, \quad \alpha(v)=\sqrt{\frac{2}{n}} \quad for\ \ v=1,2,\ldots,n-1$$

- A transform coding for de-correlation
- In most image compression standards, DCT is performed based on a sub-block which consists of 8 x 8 pixels <- empirically determined : tradeoff between compression ratio and complexity

2D DCT (4x4 pixels): $r(x,y, u,v)$ or $s(x,y, u,v)$



Two Examples of 2-D DCT



DFT vs. DCT

subblocks (8x8) → DCT → 50 % of coeff. → IDCT → merge

diff 영상

rmse가 제일 낮다
제일 성능이 좋음
compression이 잘됨

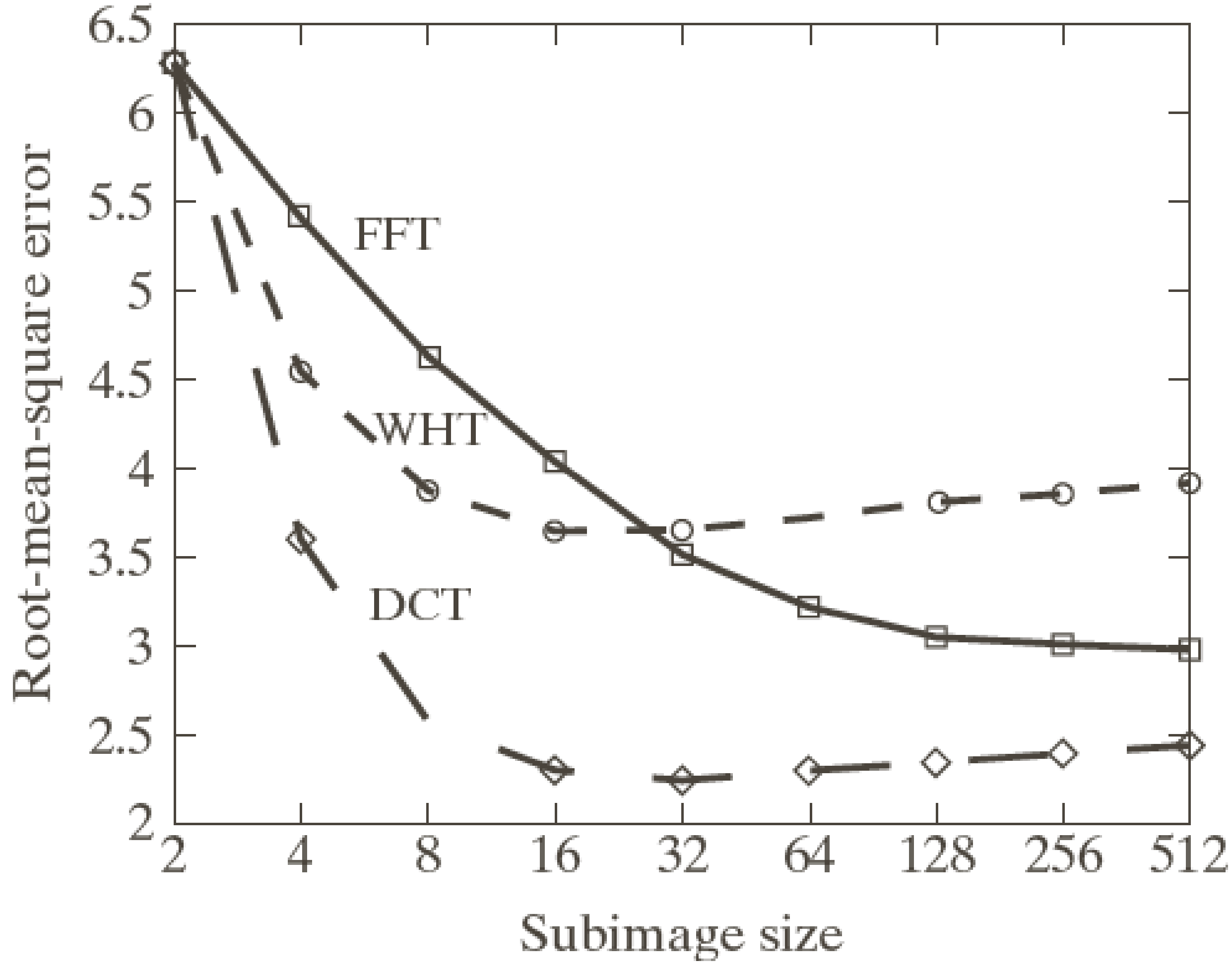
DFT	Hadamard	DCT
rms=2.32	rms=1.78	rms=1.13

Energy Compaction of 2-D DCT

- For most image coding standards, DCT is performed on *8x8* blocks, to maximize the data correlation.

- Simulations have shown that DCT outperforms Hadamard and DFT in the energy compaction, and is close to the Hotelling (or called Karhunen Loeve transform) optimal decomposition.

Energy Compaction Comparison



8*8이나 16*16으로 나눠서 처리할 때 제일 가성비가 좋다

Run-Length Coding

- Run length: length of zero run -> number of zeros

- **Example 1 :**

  Input sequence:

  0,0,-3,5,1,0,-2,0,0,0,0,2,-4,3,-2,0,0,0,1,0,0,-2

  Run-length sequence

  #2,-3,5,1,#1,-2,#4,2,-4,3,-2,#3,1,#2,-2

  (2,-3)(0,5)(0,1)(1,-2)(4,2)(0,-4)(0,3)(0,-2)(3,1)(2,-2)

  (0의 갯수, 0이 아닌 원소) 로 pairing

- **Example 2 :**

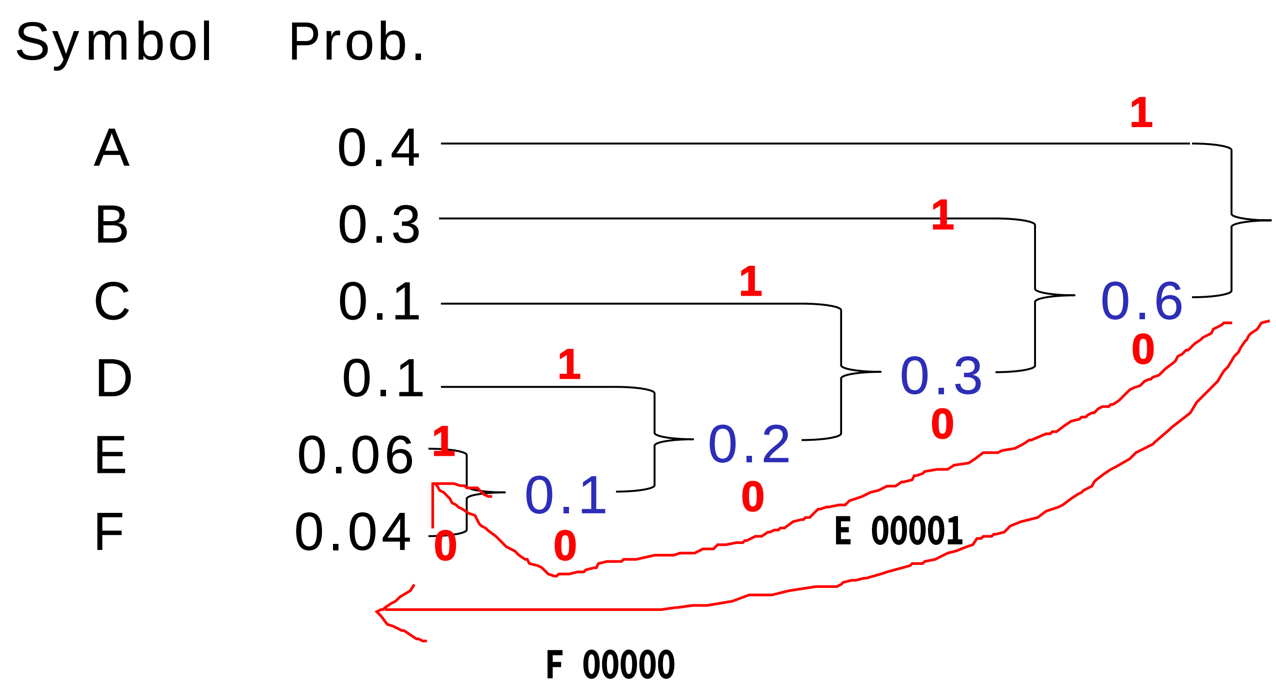
  Input sequence:

  1, 0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,1,0,0,-2

  Run-length sequence:

  (0,1)(9,2)(6,1)(2,-2)

Huffman Coding



- JPEG
- CCITT, JBIG2 확률이 낮은 코드에는 긴 비트를 할당
- MPEG-1,2,4
- H.261-4

Codeword

A	1
B	01
C	001
D	0001
E	00001
F	00000

Huffman Coding

Symbol	Prob.	Codeword
A	0.4	1
B	0.3	01
C	0.1	001
D	0.1	0001
E	0.06	00001
F	0.04	00000

FEBC = 000000000101001

101111001110001 = 1/01/1/1/1/001/1/1/0001 = ABAAACAAD

Entropy : $H = 0.4 \times \log_2(1/0.4) + 0.3 \times \log_2(1/0.3) \ldots$

$= 2.14$ bits/symbol

Average length per symbol of Huffman coding :

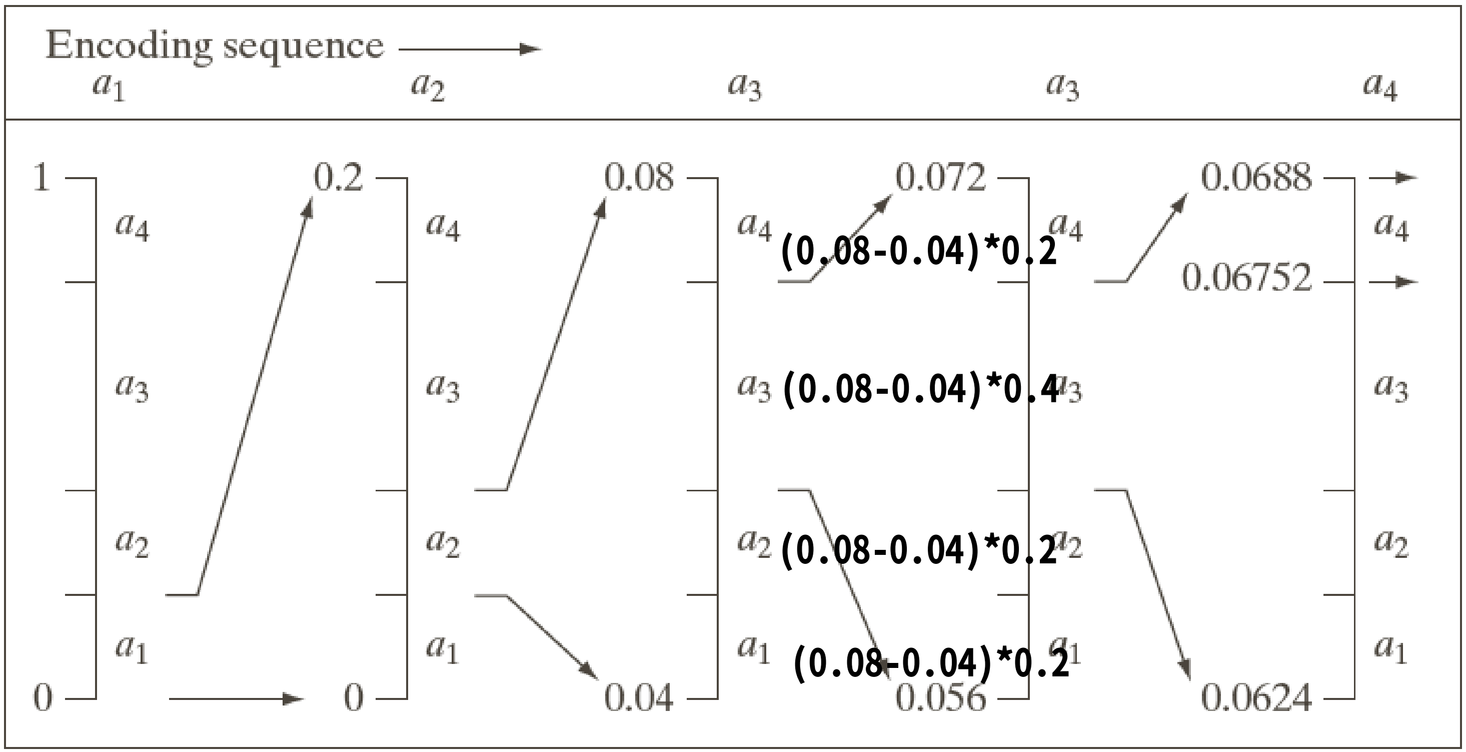
$L = 0.4 \times 1 + 0.3 \times 2 + 0.1 \times 3 + 0.1 \times 4 + 0.06 \times 5 + 0.04 \times 5$

$= 2.2$ bits/symbol **양자화를 위하여 어쩔수 없이 약간의 손실이 있음 하지만 엔트로피와 거의 근사함**

Arithmetic Coding

- JBIG1,2
- JPEG-2000
- H.264
- MPEG4 AVC

Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)

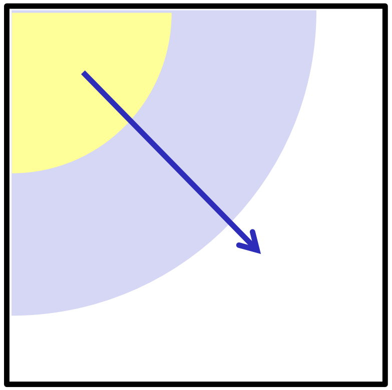


- $a_1a_2a_3a_3a_4$: 0.06752 <- any value in [0.06752, 0.0688)

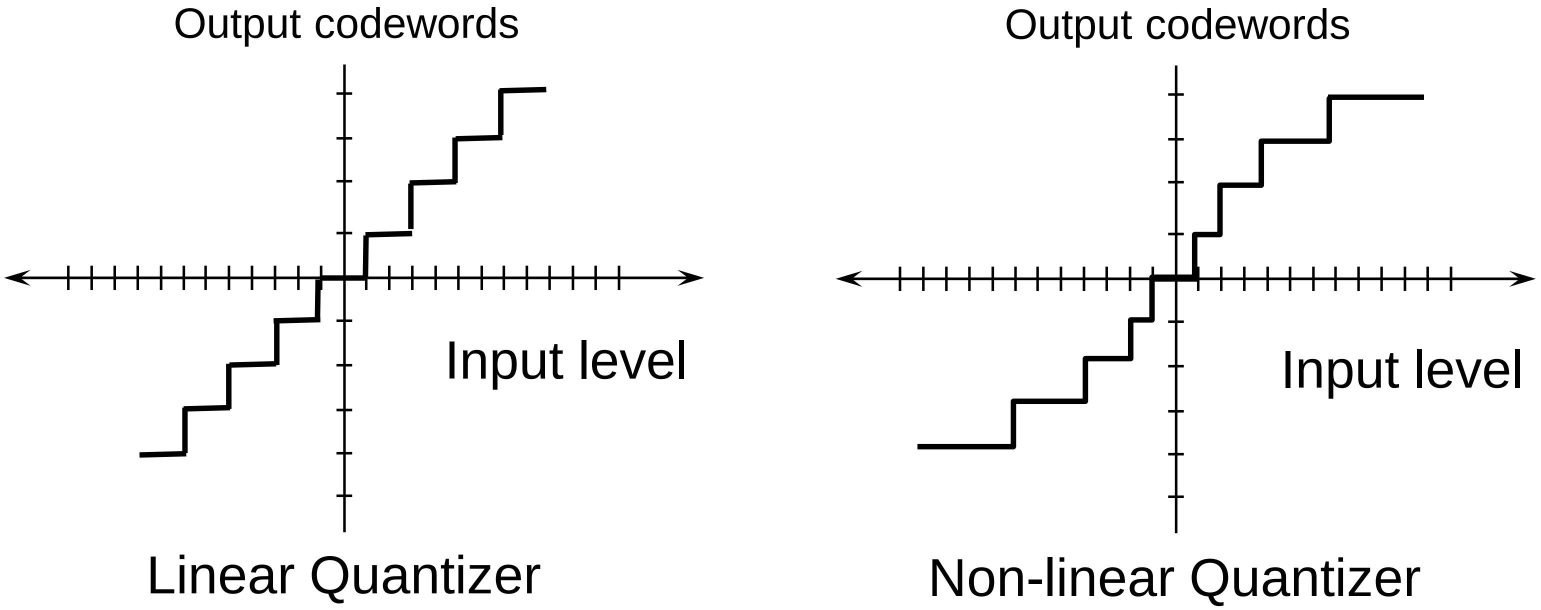
Joint Photographic Experts Group (JPEG)

- Digital compression and coding of continuous-tone still images
- Most widely used compression standard
- Also used to compress video frame-by-frame (motion JPEG)
- ISO/IEC 10918-1/2/3 (now ITU-T Recommendation T.81, 83, 84): Requirements & Guidelines, Compliance Testing, and Extensions.
- Lossless and lossy modes

JPEG: How It Works

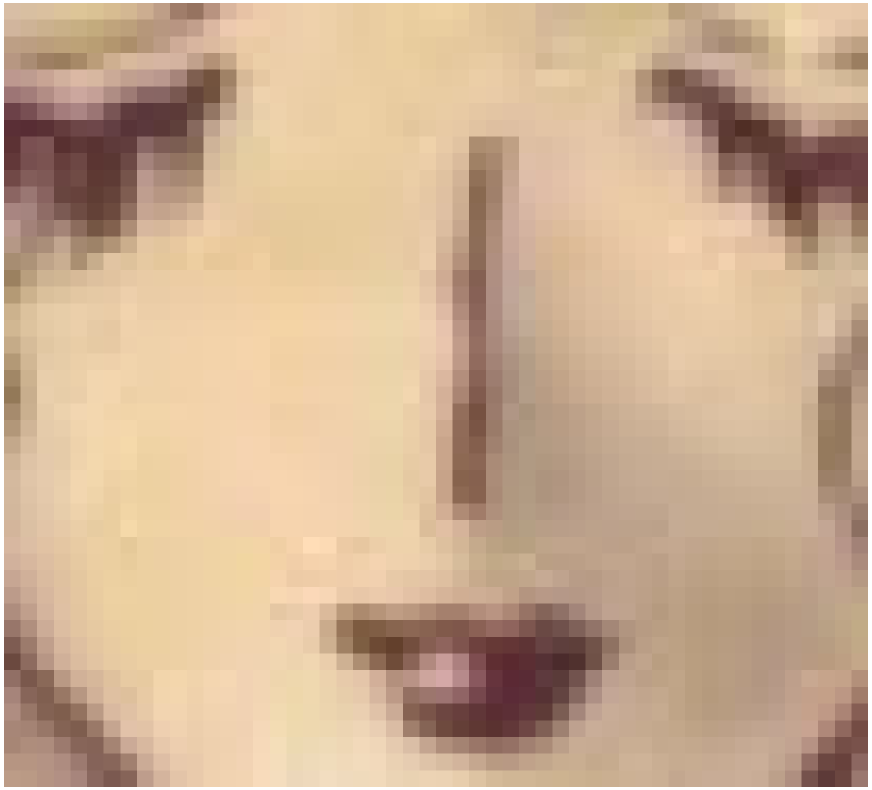


- DCT
  - Divide the image data into two parts
  - visually more important and less important parts
- Quantization
  - Allocate more space for visually important data

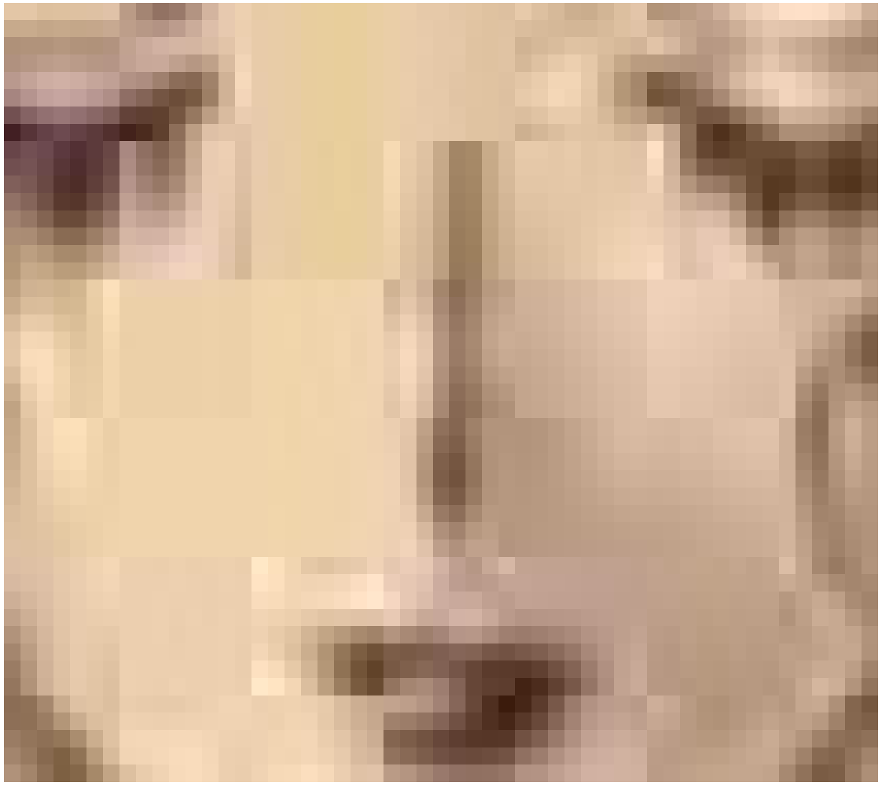


327KB(335,684바이트)

44.9KB(46,024바이트)



원 영상 확대

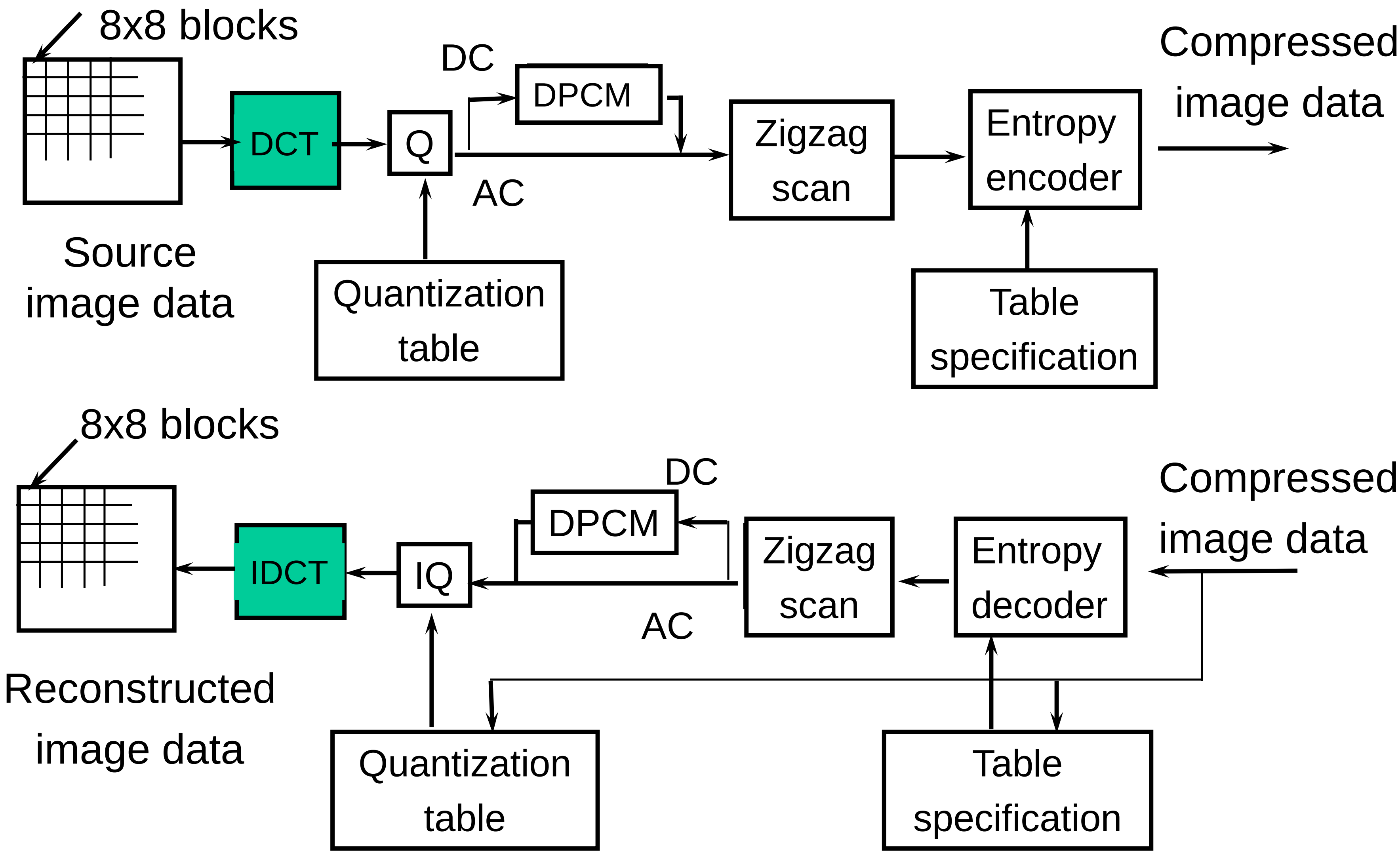


최대압축 확대

JPEG Performance

Bits/pixel	Quality	Compression Ratio
≥ 2	Indistinguishable	8-to-1
1.5	Excellent	10.7-to-1
0.75	Very Good	21.4-to-1
0.50	Good	32-to-1
0.25	Fair	64-to-1

JPEG Encoder and Decoder
Sequential DCT-Based Systems



JPEG: Sequential DCT-Based Systems

- Baseline System
  - minimum capability that must be present in all JPEG systems
  - source sample precision limited to 8 bits
  - only Huffman coding for entropy coding
  - up to 2 sets of entropy coding tables

- Extended Sequential Systems
  - capabilities beyond the baseline requirement
  - source sample precision of 8 or 12 bits
  - can use Arithmetic coding
  - up to 4 sets of entropy coding tables

An Example Of JPEG Baseline System : Encoding

Original:

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

Step1: Level shift by -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

An Example Of JPEG Baseline System : Encoding

Step2: 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

Step3: Quantization

- Divide the DCT coefficients by Q table in the next page. Q table was acquired by experiments

-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

Default Quantization Tables Used in Step 3

Luminance Table:

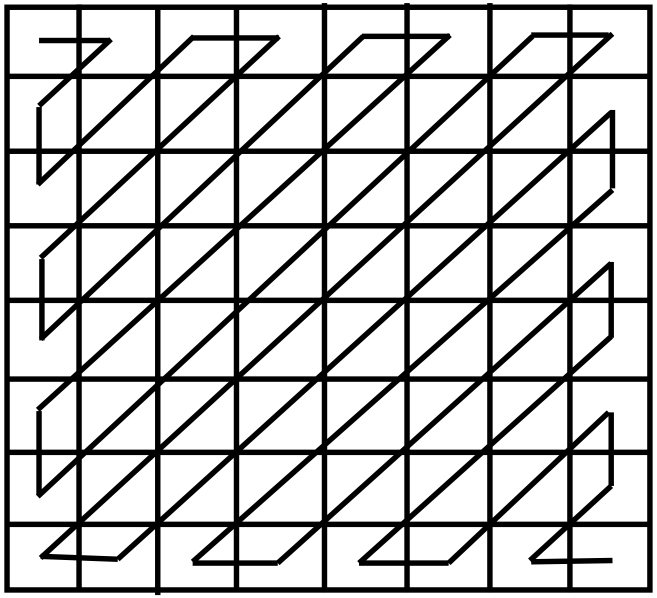
16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Chrominance Table:

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

Step 4: Zigzag Scan

-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



- Rearrange the coefficients by a zigzag path to maximize the performance of run-length coding

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

An Example Of JPEG Baseline System : Encoding

Step 4: zigzag scan:

can be switched

[- 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]

Step 5: DPCM of DC coefficient (assuming previous DC value of -17):

[- 9 - 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]

Step 6: Run Length Coding of AC terms: 64 → 37 bytes

-9 (0,-3) (0,1) (0,-3) (0,-2) (0,-6) (0,2) (0,-4) (0,1) (0,-4) (0,1)
(0,1) (0,5) (1,2) (2,-1) (0,2) (5,-1) (0,-1) (0,-1) EOB

Step 7: Huffman coding

- Use Huffman table for binary conversion in the following slides
- After entropy(Huffman) coding: 64 -> 37 -> 92 bits (11.5 bytes)

001 100011 001 001 100101 11100110 110110
0110 11110100 000 1010

Step 7-1: Huffman Coding for DC Value

- Use Table A.3 and A.4
- What is the binary codes for DC value of -9 in this example?

1. From Table A.3, category for -9 is 4, which means 4 bits to represent the actual value of -9
2. From Table A.4, binary codeword for category 4 = 101
3. From Table A.3 again, for binary value of -9(negative value) using 4 bits(DC category 4), try one's complement -> 9 : 1001, -9 : 0110
4. combine codeword and code value => 1010110

Step 7-2: Huffman Coding for AC Value for (0, -3)

- Use Table A.3 and A.5
- What is the binary codes for AC value of (0, -3) in this example?

1. From Table A.3, category for -3 is 2, which means 2 bits to represent binary value of (0, -3)
2. From Table A.5, binary codeword for run/category pair 0/2 = 01
3. From Table A.3 again, for binary value of -3(negative value) using 2 bits (AC category 2), try one's complement -> 3 : 11, -3 : 00
4. Combine codeword and code value => 0100

Step 7-2: Huffman Coding for AC Value for (1, 2)

- Use Table A.3 and A.5
- What is the binary codes for AC value of (1, 2) in this example?

1. From Table A.3, category for 2 is 2, which means 2 bits to represent binary value of (1, 2)
2. From Table A.5, binary codeword for run/category pair 1/2 = 111001
3. From Table A.3 again, for binary value of 2 using 2 bits (AC category 2), just convert its value into binary -> 2 : 10
4. Combine codeword and code value => 11100110

An Example of JPEG Baseline System : Decoding

Decoder input:

1010110 0100 001 0100 0101 100001 0110 100011
001 100011 001 001 100101 11100110 110110
0110 11110100 000 1010

Step 1: Huffman decoding

-9 (0,-3) (0,1) (0,-3) (0,-2) (0,-6) (0,2) (0,-4) (0,1) (0,-4) (0,1)
(0,1) (0,5) (1,2) (2,-1) (0,2) (5,-1) (0,-1) (0,-1) EOB

Step 2-4: RLD, Inverse DPCM (-9 + (-17)), and Inverse zigzag scan

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

An Example of JPEG Baseline System : Decoding

Step 5: inverse quantization

-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

Step 6: IDCT

-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

An Example of JPEG Baseline System : Decoding

Step7: level shift by +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

Error:

- 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

Example of JPEG Coding

(a)

(b)

Table A.3 JPEG Coefficient Coding Categories

TABLE A.3 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 A.4 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

TABLE A.4 JPEG default DC code (luminance).

Table A.5 JPEG Default AC Code (luminance) : part 1

TABLE A.5 JPEG default AC code (luminance).

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
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

(Continued)

Table A.5 JPEG Default AC Code (luminance) : part 2

TABLE A.5
(Continued)

Run/ Category	Base Code	Length	Run/ Category	Base Code	Length
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
5/1	1111010	8	D/1	11111111010	12
5/2	1111111001	12	D/2	1111111111100011	18
5/3	1111111110011111	19	D/3	1111111111100100	19
5/4	1111111110100000	20	D/4	1111111111100101	20
5/5	1111111110100001	21	D/5	1111111111100110	21
5/6	1111111110100010	22	D/6	1111111111100111	22
5/7	1111111110100011	23	D/7	1111111111101000	23
5/8	1111111110100100	24	D/8	1111111111101001	24
5/9	1111111110100101	25	D/9	1111111111101010	25
5/A	1111111110100110	26	D/A	1111111111101011	26
6/1	1111011	8	E/1	111111110110	13
6/2	11111111000	13	E/2	1111111111101100	18
6/3	1111111110100111	19	E/3	1111111111101101	19
6/4	1111111110101000	20	E/4	1111111111101110	20
6/5	1111111110101001	21	E/5	1111111111101111	21
6/6	1111111110101010	22	E/6	1111111111110000	22
6/7	1111111110101011	23	E/7	1111111111110001	23
6/8	1111111110101100	24	E/8	1111111111110010	24
6/9	1111111110101101	25	E/9	1111111111110011	25
6/A	1111111110101110	26	E/A	1111111111110100	26
7/1	11111001	9	**F/0**	**111111110111**	**12**
7/2	11111111001	13	F/1	1111111111110101	17
7/3	1111111110101111	19	F/2	1111111111110110	18
7/4	1111111110110000	20	F/3	1111111111110111	19
7/5	1111111110110001	21	F/4	1111111111111000	20
7/6	1111111110110010	22	F/5	1111111111111001	21
7/7	1111111110110011	23	F/6	1111111111111010	22
7/8	1111111110110100	24	F/7	1111111111111011	23
7/9	1111111110110101	25	F/8	1111111111111100	24
7/A	1111111110110110	26	F/9	1111111111111101	25
			F/A	1111111111111110	26

Summary

- Lossless:
  - Run-Length coding
  - Huffman coding
  - Arithmetic coding
  - Lempel-Ziv-Welch (LZW) coding
- Lossy: DCT, wavelet, fractal, etc
- Standards: JPEG, JPEG-2000, JBIG-1, JBIG-2
- Self information, entropy, RMSE