DIP(Digital Image Processing)
Program Exercise 3

(해당 강의자료의 배포 및 무단 복제를 금함)

http://dms.sejong.ac.kr
http://home.sejong.ac.kr/~yllee/

김남욱, 김명준, 정지연, 김양우, 이영렬

Sejong University, DMS Lab.

Program Exercises

1. Huffman Coding : 5점

- Lena 영상 DPCM
- DPCM된 Lena의 Huffman coding table 생성
- Entropy 계산
- Huffman decoding → decoded image

2. DCT (Discrete Cosine Transform) : 10점

- Use 8x8 DCT already implemented
- Quantization
- (Scanning + Huffman Coding Table) : 이 부분 skip 함
- Inverse Quantization
- 8x8 IDCT
- PSNR 비교

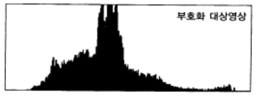
DPCM: removing spatial redundancy of images

232	234	235	232	233	234
233	235	233	233	229	231
234	235	233	233	234	235
232	234	233	232	232	233

Similar values

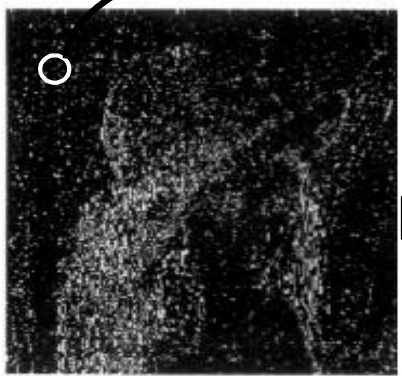
=> Signal Redundancy

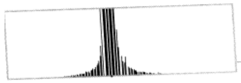
232	2	1	-3	1	1
-1	2	2	0	-4	1
3	1	-2	0	1	1
-2	2	-1	-1	0	1



histogram

DPCM





Quantization + Entropy coding

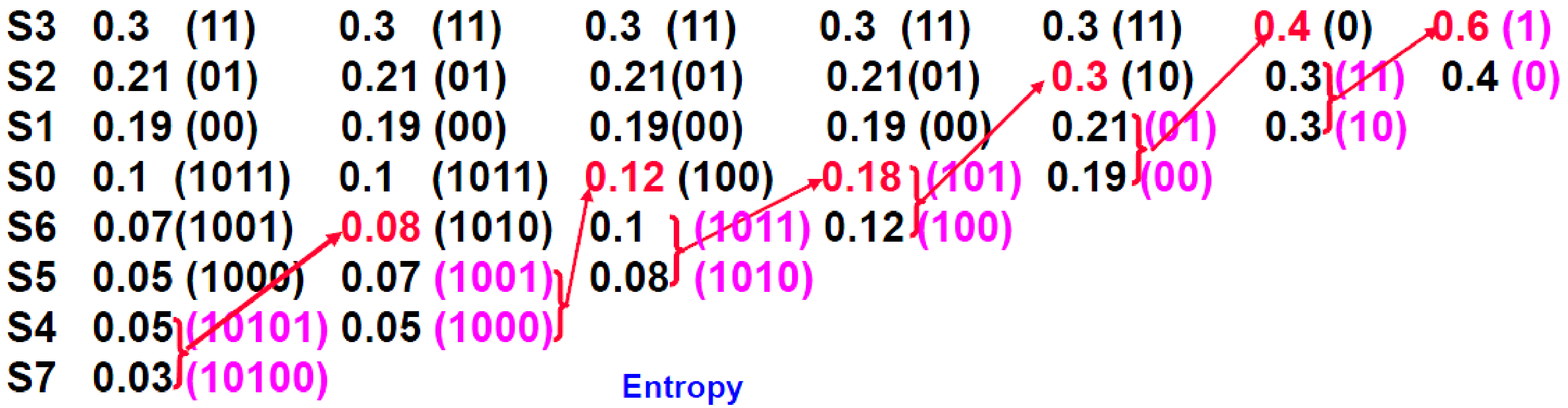
histogram

: Data cluster to 0.

=> Less bits for encoding the data

- DPCM removes redundant data.

1. Huffman Coding



S3	0.3 (11)	0.3 (11)	0.3 (11)	0.3 (11)	0.3 (11)	0.4 (0)	0.6 (1)
S2	0.21 (01)	0.21 (01)	0.21(01)	0.21(01)	0.3 (10)	0.3 (11)	0.4 (0)
S1	0.19 (00)	0.19 (00)	0.19(00)	0.19 (00)	0.21 (01)	0.3 (10)	
S0	0.1 (1011)	0.1 (1011)	0.12 (100)	0.18 (101)	0.19 (00)		
S6	0.07(1001)	0.08 (1010)	0.1 (1011)	0.12 (100)			
S5	0.05 (1000)	0.07 (1001)	0.08 (1010)				
S4	0.05 (10101)	0.05 (1000)					
S7	0.03 (10100)						

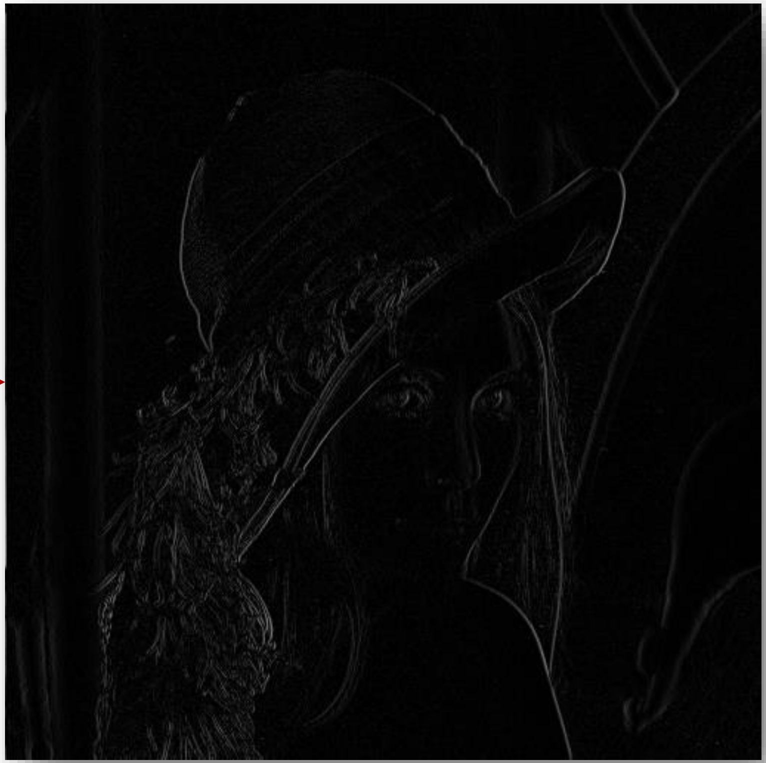
Entropy

Symbol	P(Si)	$-\log_2(P(Si))$	$-\log_2(P(Si)).P(Si)$	Codeword	Code Length	AVG.Length
S3	0.3	1.737	0.521	11	2	0.6
S2	0.21	2.252	0.473	01	2	0.42
S1	0.19	2.396	0.455	00	2	0.38
S0	0.1	3.322	0.332	1011	4	0.4
S6	0.07	3.837	0.269	1001	4	0.28
S5	0.05	4.322	0.216	1000	4	0.2
S4	0.05	4.322	0.216	10101	5	0.25
S7	0.03	5.059	0.152	10100	5	0.15
			2.634			2.68

1. Huffman Coding

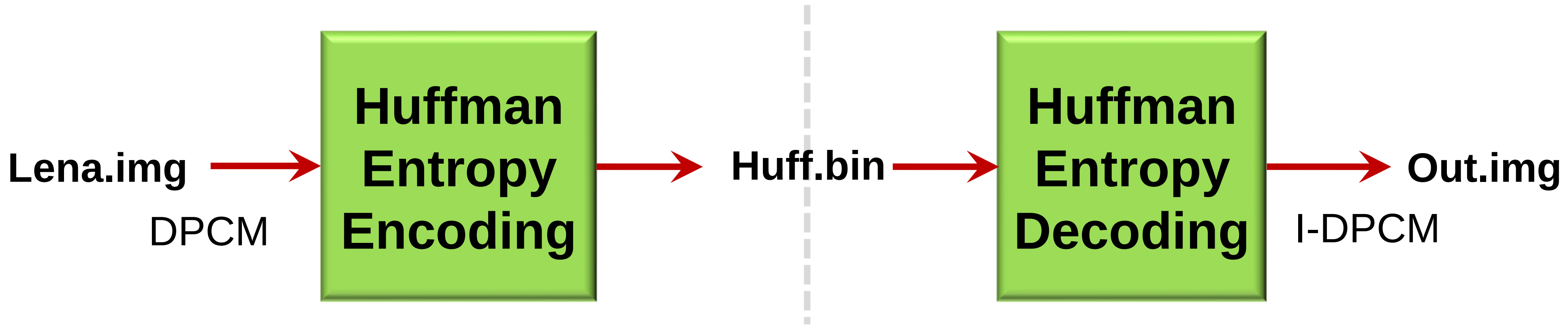
- **Lena의 DPCM**
  - 0에 가까운 값의 분포가 많음

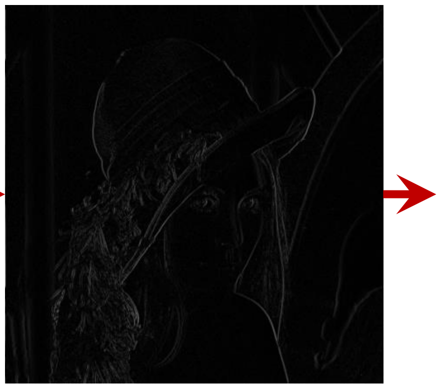
Lena


Lena DPCMed

1. Huffman Coding

- 허프만 코딩을 이용한 Lena.img 압축
  - Lossless(무손실) 압축



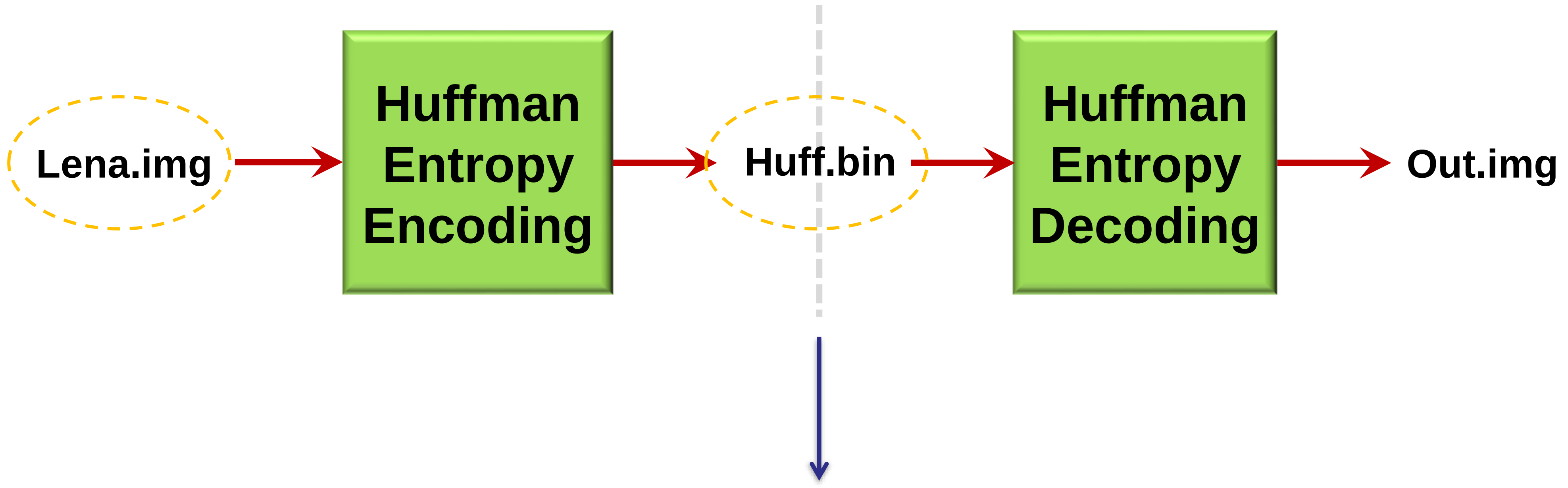


DPCM을 이용
허프만 **Encoding** 비트 생성
및 압축률 비교

허프만 **Decoding**
및 원본 영상과 비교

1. Huffman Coding

- 파일의 크기 비교 및 압축률 비교

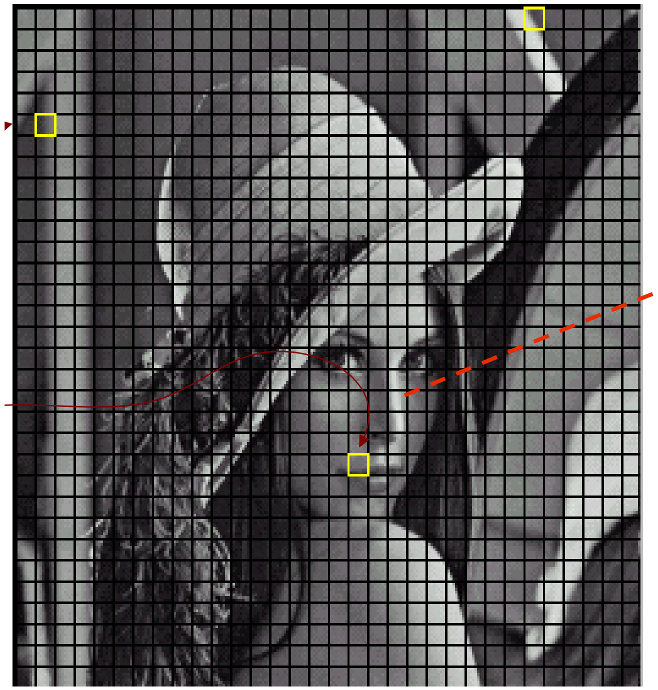


Compression
(Transmission or Save)
Binary file

2. DCT(Discrete Cosine Transform)

- 8x8 DCT를 수행한 영상
- 8x8 IDCT를 통해 원본 영상과 비교
- 8x8 DCT에서 Quantization를 수행한 후 PSNR 비교

Example of one 8x8 DCT



energy compaction to low freq. components

8x8 block values

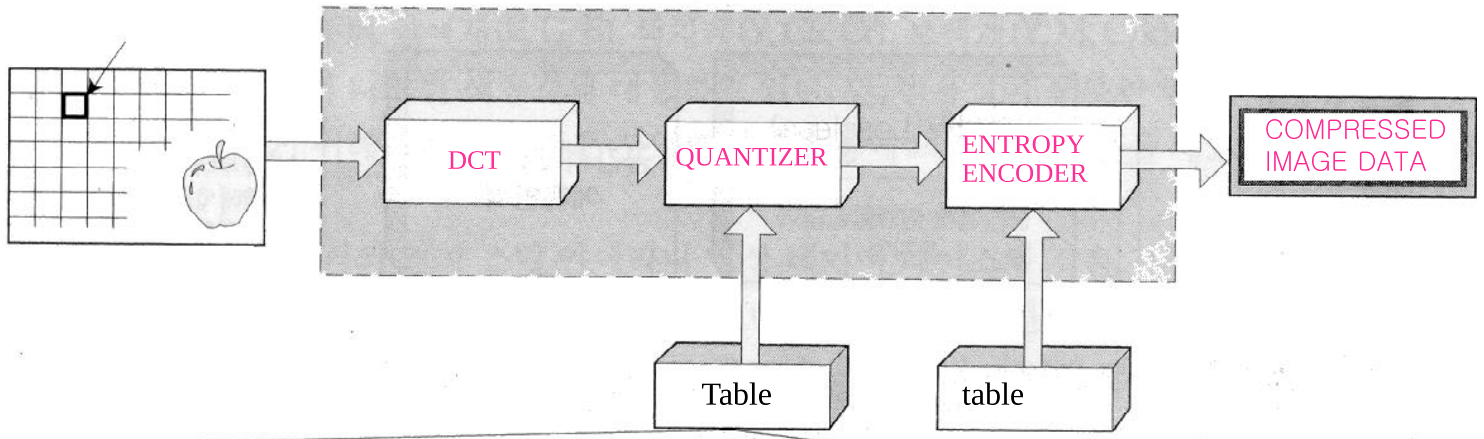
78	74	74	79	83	83	88	88
78	74	74	79	83	83	88	83
76	74	74	79	80	84	87	85
73	71	75	76	79	86	84	87
72	74	75	79	79	80	85	84
67	73	74	78	79	81	86	87
70	68	74	82	78	81	86	86
66	68	74	77	78	80	84	84

$\xrightarrow{\text{DCT}}$

DCT values

742	-16	1.2	-7.2	-1.3	3.9	-9.6	1.7
8.0	-1.4	4.3	-2.0	2.9	1.4	-4.0	0.1
1.9	-3.6	-1.9	-0.1	-0.4	0.8	-2.8	-0.6
-0.8	0.5	-0.9	3.1	-0.8	1.7	-0.6	2.4
2.1	0.9	-0.4	-0.1	-1.6	2.3	1.1	-1.1
-3.0	1.2	1.8	-1.3	1.9	-3.2	0.4	-3.1
1.7	0.2	1.6	-0.3	-0.7	-0.4	-0.7	2.9
-2.6	-2.4	-3.4	1.6	1.2	-0.1	2.0	0.6

JPEG block diagram



Luma quantization table

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	66
14	13	16	24	40	57	69	57
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	36	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Chroma quantization 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

Quantization of DCT values

- Quantized DCT value : $P(i,j) = Round\left(\dfrac{DCT\_Val(i,j)}{Q \text{ or } q(i,j)}\right)$
- Use quantization table q(i,j) or quantizer scale Q. (Qp is a quality factor.)

DCT value

1034.9	381.6	97.6	-17.6	-1.4	-8.9	2.7	-0.6
-144.4	-66.0	120.9	48.7	-15.4	-6.4	0.6	-2.9
29.1	-17.2	-43.4	22.3	20.5	-9.9	-1.1	-0.5
-3.4	17.1	2.6	-20.9	-0.9	9.9	-3.4	-1.9
12.1	0.4	-3.7	2.7	-6.6	-2.4	7.1	-1.6
-3.1	4.1	4.6	-5.1	0.7	0.7	-3.9	3.4
-1.9	-2.2	0.1	3.8	0.3	-0.6	0.2	-0.9
-0.7	-3.4	1.2	1.5	0.1	-0.9	0.2	1.3

Quantized value

Q=20

52	19	5	-1	0	0	0	0
-7	-3	6	2	-1	0	0	0
1	-1	-2	1	1	0	0	0
0	1	0	-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

Q=40

26	10	2	0	0	0	0	0
-4	-2	3	1	0	0	0	0
1	0	-1	1	1	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

Default quan. Table
for luminance

{q(i,j)}=

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

* Qp

Default quan. Table
for chrominance

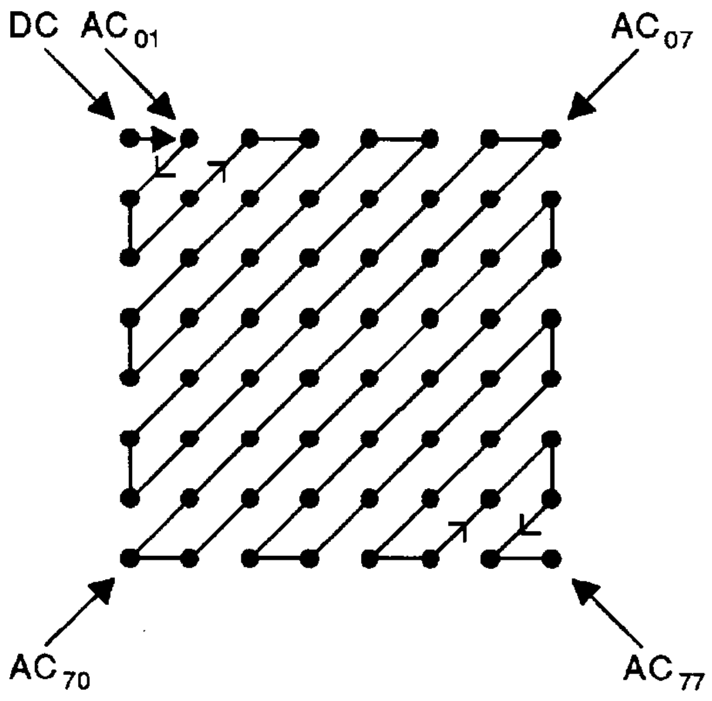
{q(i,j)}=

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

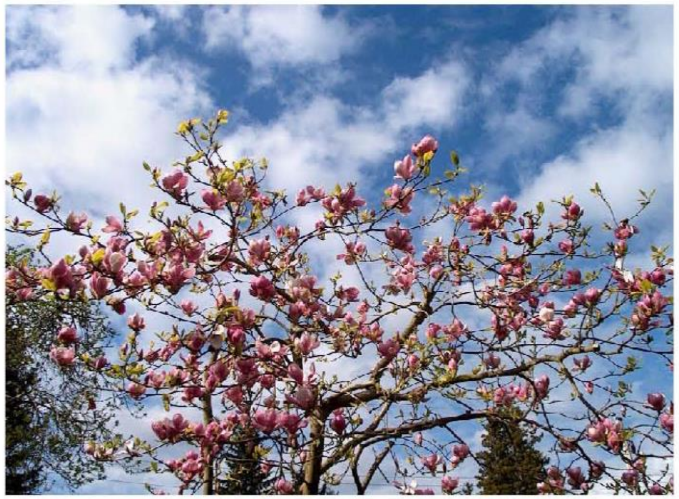
* Qp

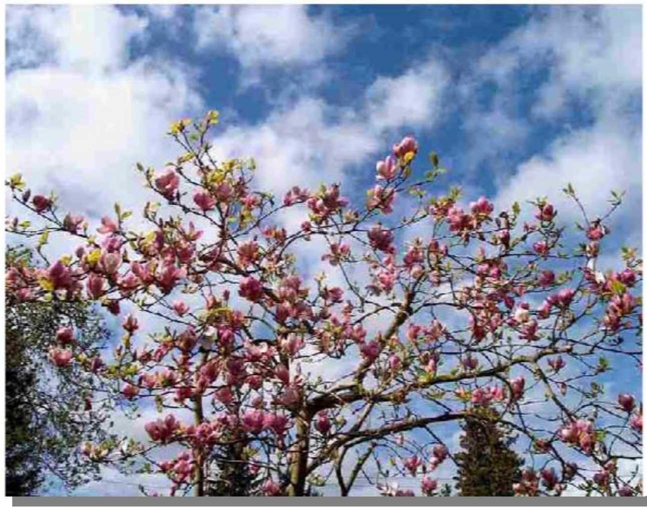
Scanning of Quantized values (reference)

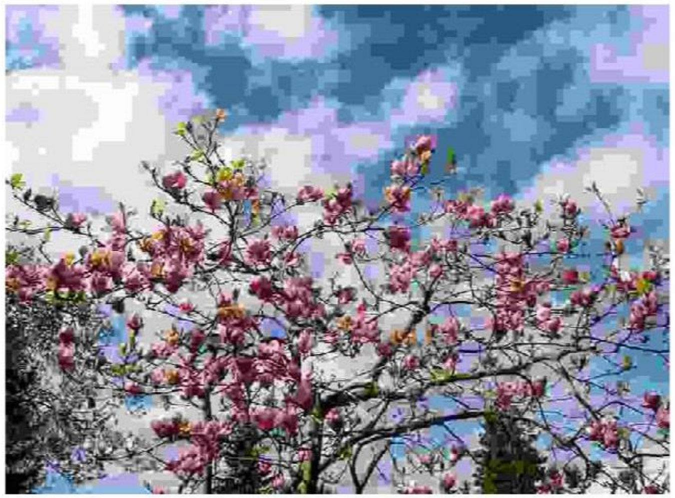
- Zig-zag scan and Huffman coding (실험에서는 안 함)
  - DC, AC coefficients quantization, independently
  - Separate quantization tables
    - ✓ Zig-zag scan from low frequency data to high frequency data



JPEG pictures


(a) Original Image (24bits/pixel)


(b) 0.75 bits/pixel


(c) 0.25 bits/pixel

Use Type-2 DCT(Discrete Cosine Transform)

$$F(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\frac{\pi(2x+1)u}{2N}\cos\frac{\pi(2y+1)v}{2N}$$

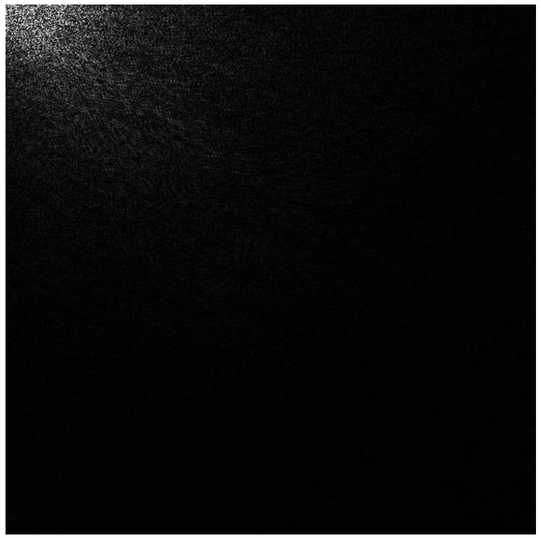
$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \alpha(u)\alpha(v)F(u,v)\cos\frac{\pi(2x+1)u}{2N}\cos\frac{\pi(2y+1)v}{2N}$$

$$\text{where } \alpha(0) = \frac{1}{\sqrt{N}},\ \alpha(k) = \sqrt{\frac{2}{N}}, k = 1,\dots,N-1$$

2. DCT examples

- **DCT**이후의 값은 **DC**와 **AC**로 분포

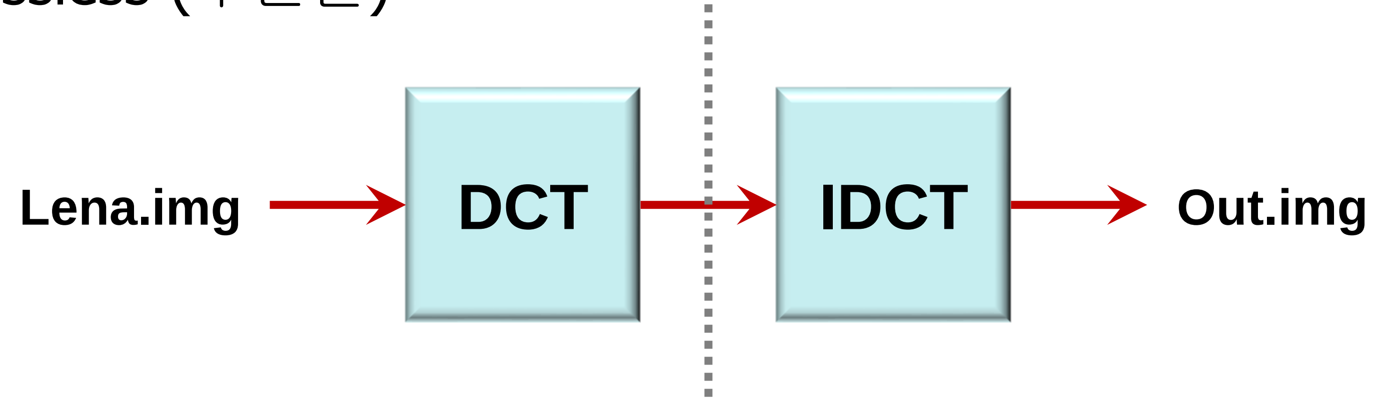
Lena



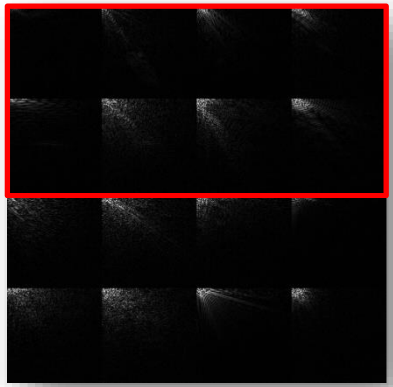
512x512 DCT of Lena

2. DCT(Discrete Cosine Transform)

- Lossless (무손실)

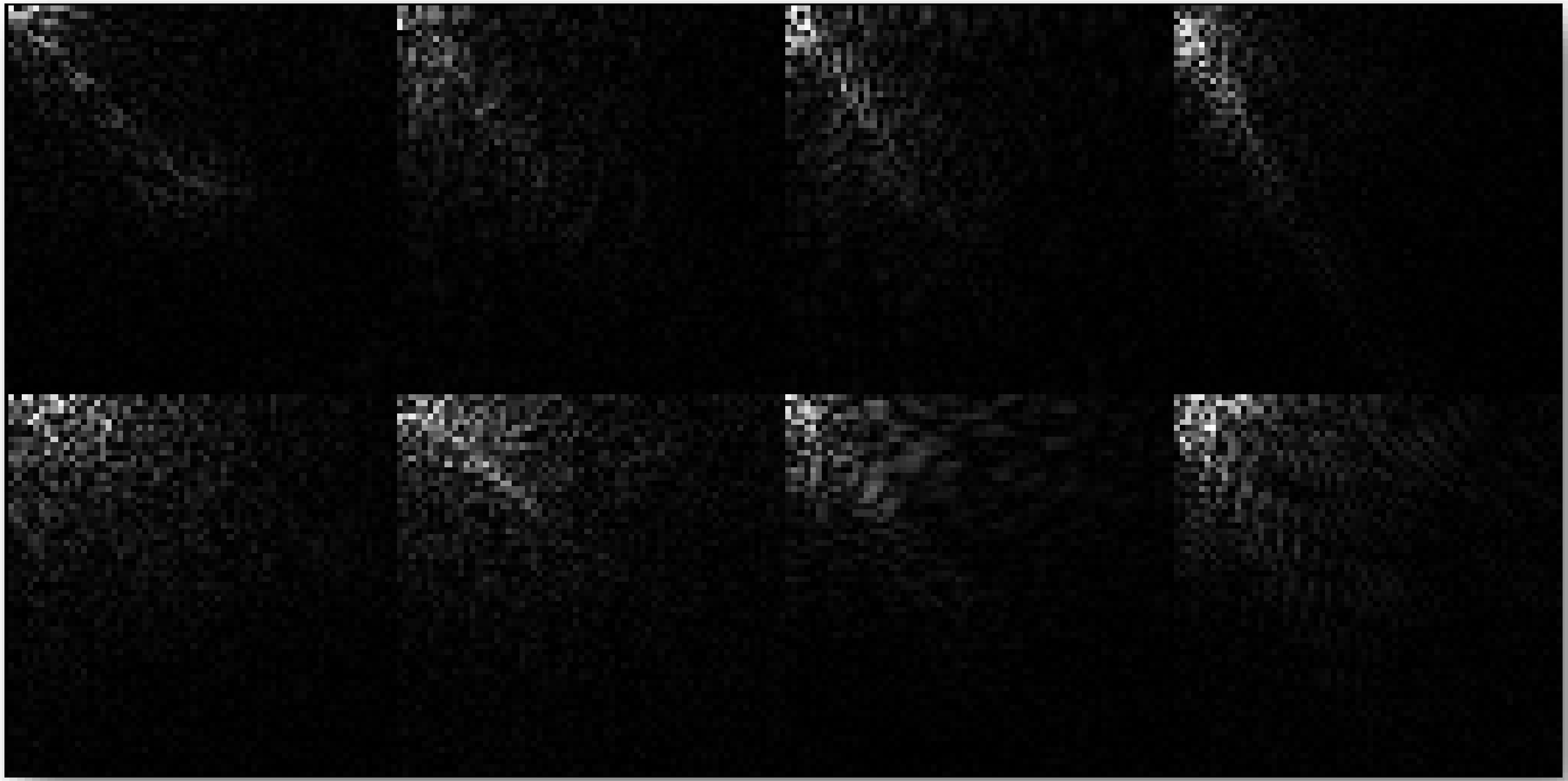


ex) Sixteen 128x128 DCT



2. DCT(Discrete Cosine Transform)

- DCT Domain의 값의 분포



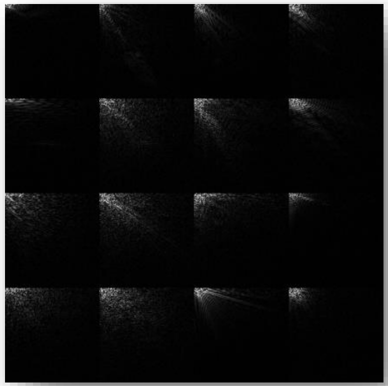
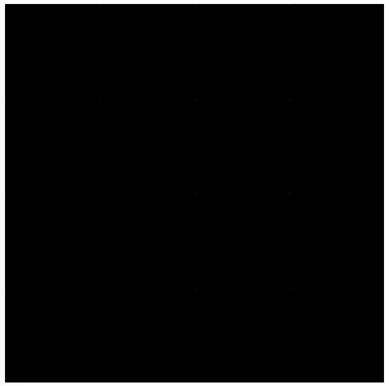
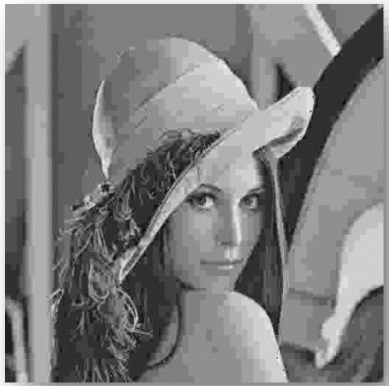
2. Quantization(양자화)

- Lossy (손실)

$$P(i,j) = Round\left(\frac{DCT\_Val(i,j)}{Q \text{ or } q(i,j)}\right)$$

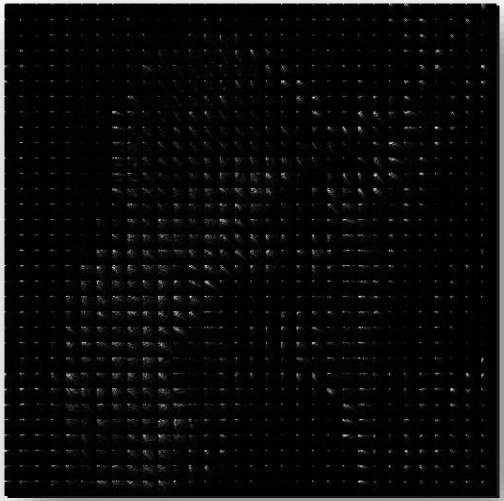
$$\begin{tikzcd}
\text{Lena.img} \arrow[r] & \boxed{\text{DCT}} \arrow[r] & \boxed{\text{Q}} \arrow[r] & \boxed{\text{IQ}} \arrow[r] & \boxed{\text{IDCT}} \arrow[r] & \text{Out.img}
\end{tikzcd}$$

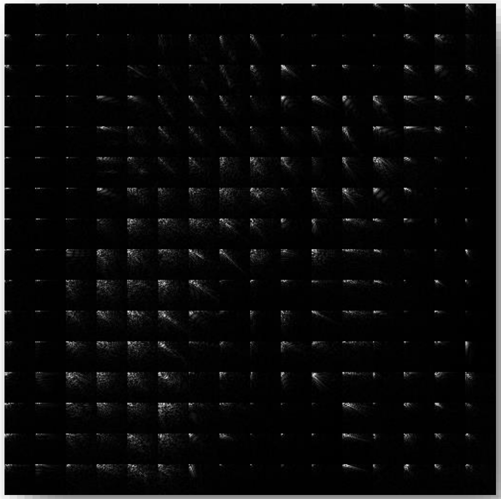
ex) 128x128 DCT

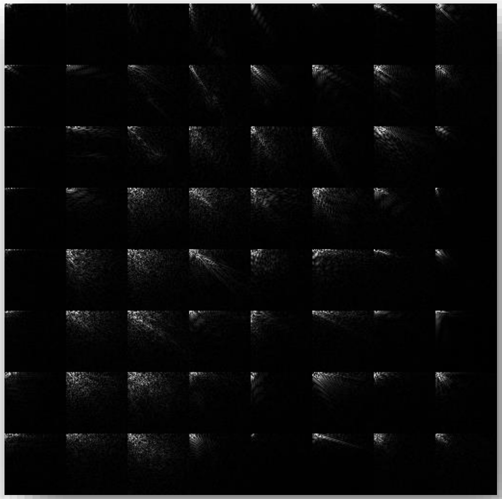




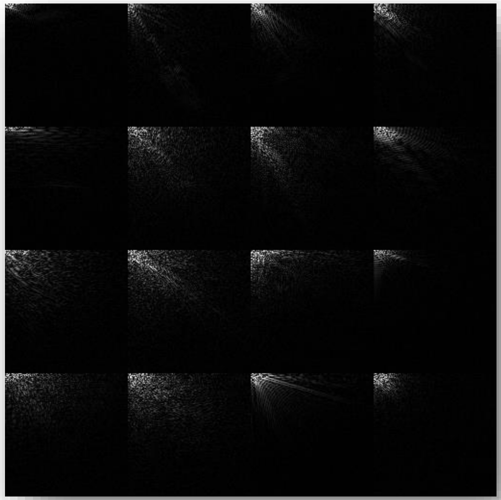
4 X 4

8 X 8

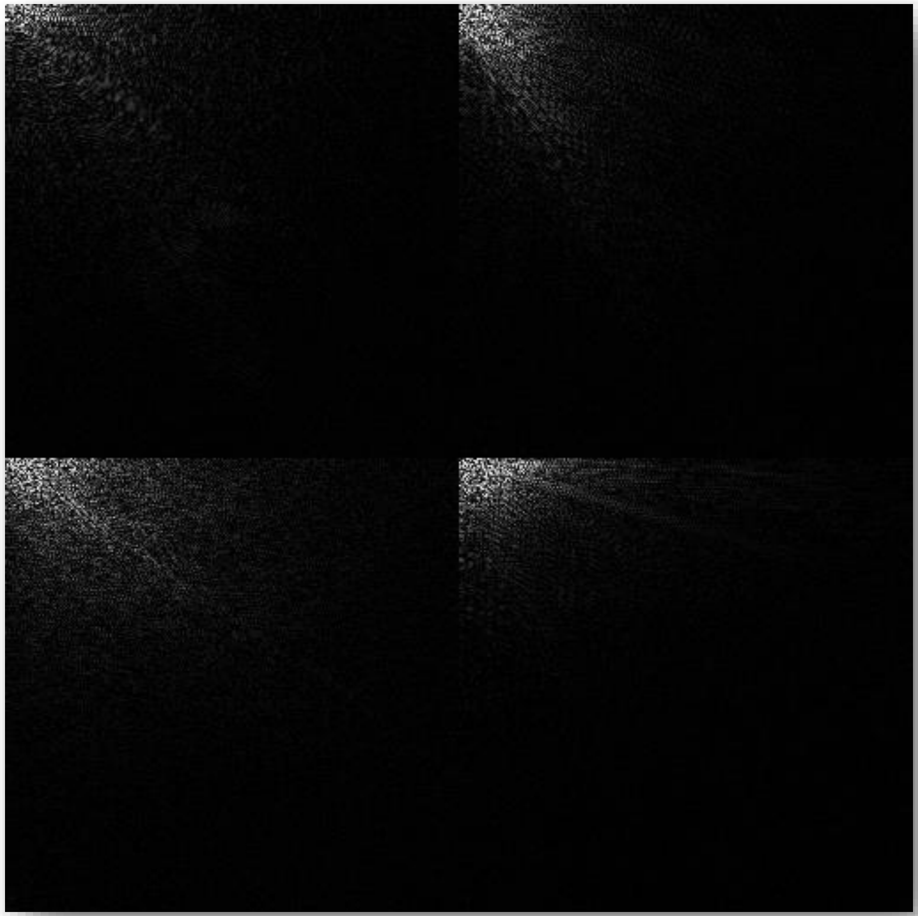

16 X 16

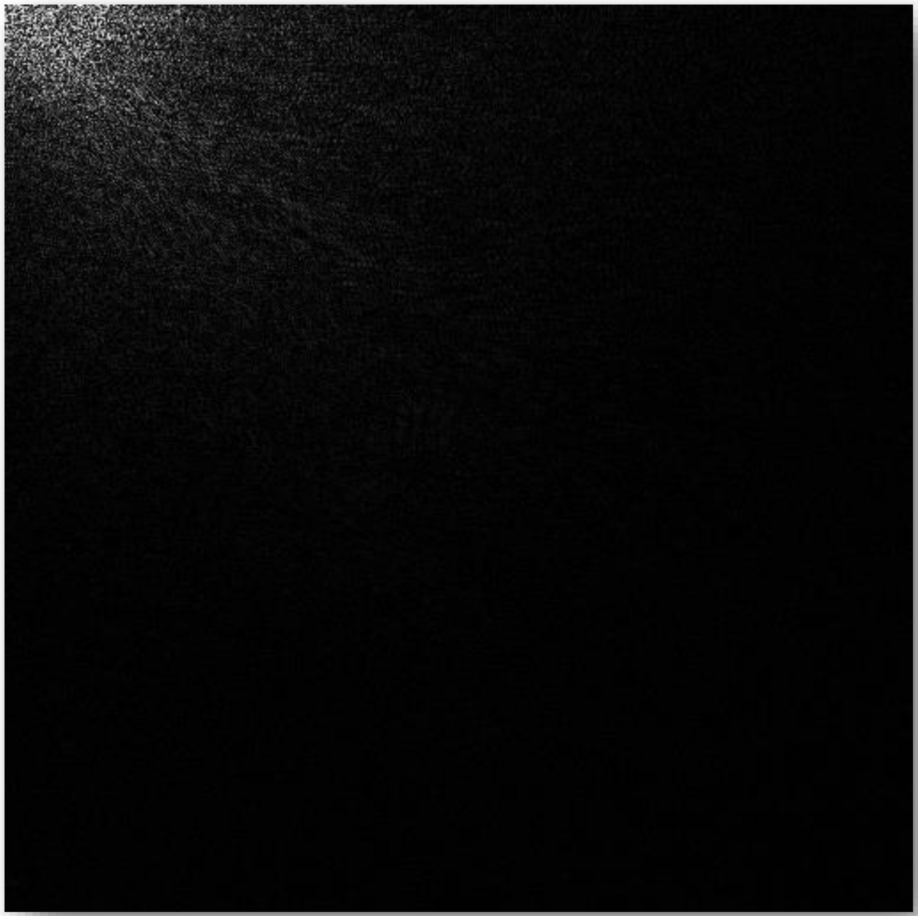

32 X 32


64 X 64

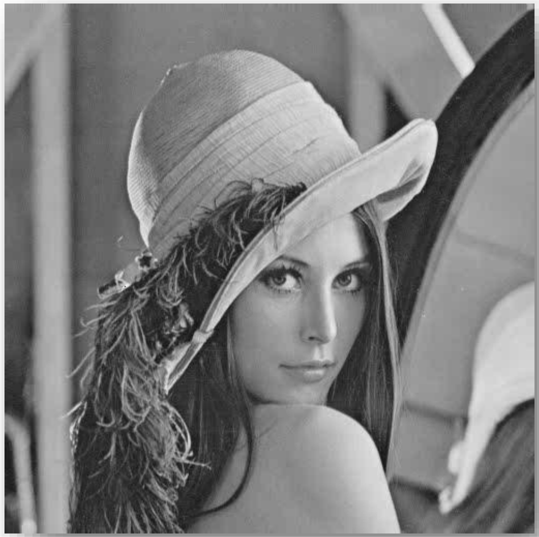

128 X 128

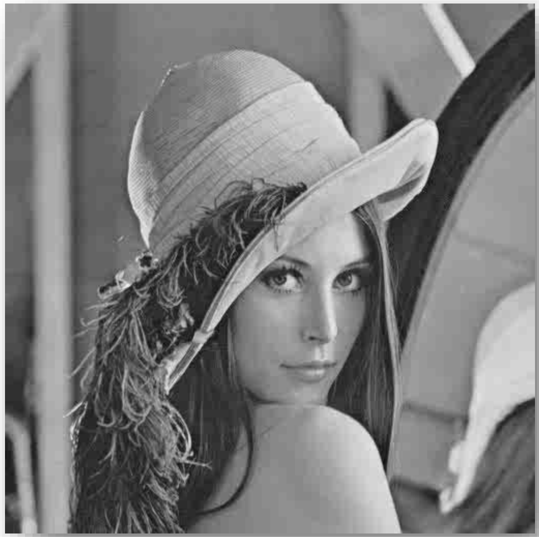
2. DCT(Discrete Cosine Transform)

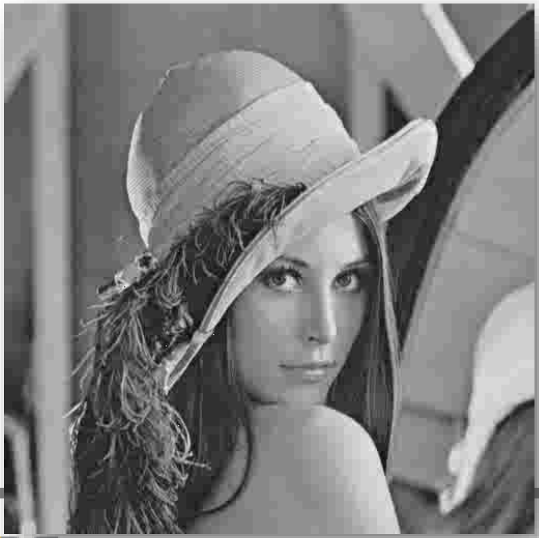

256 X 256

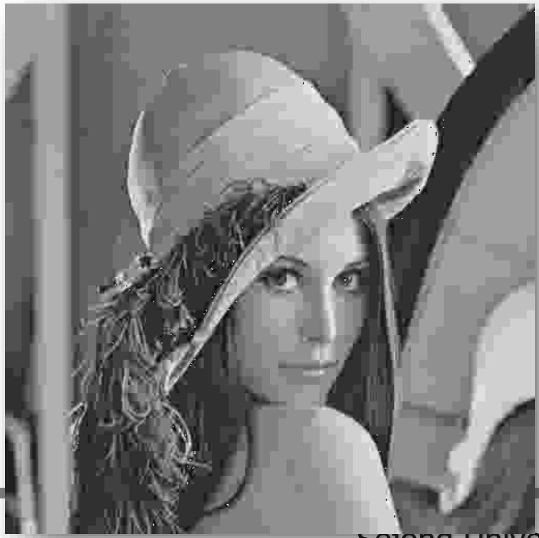

512 X 512

2. After Quantization

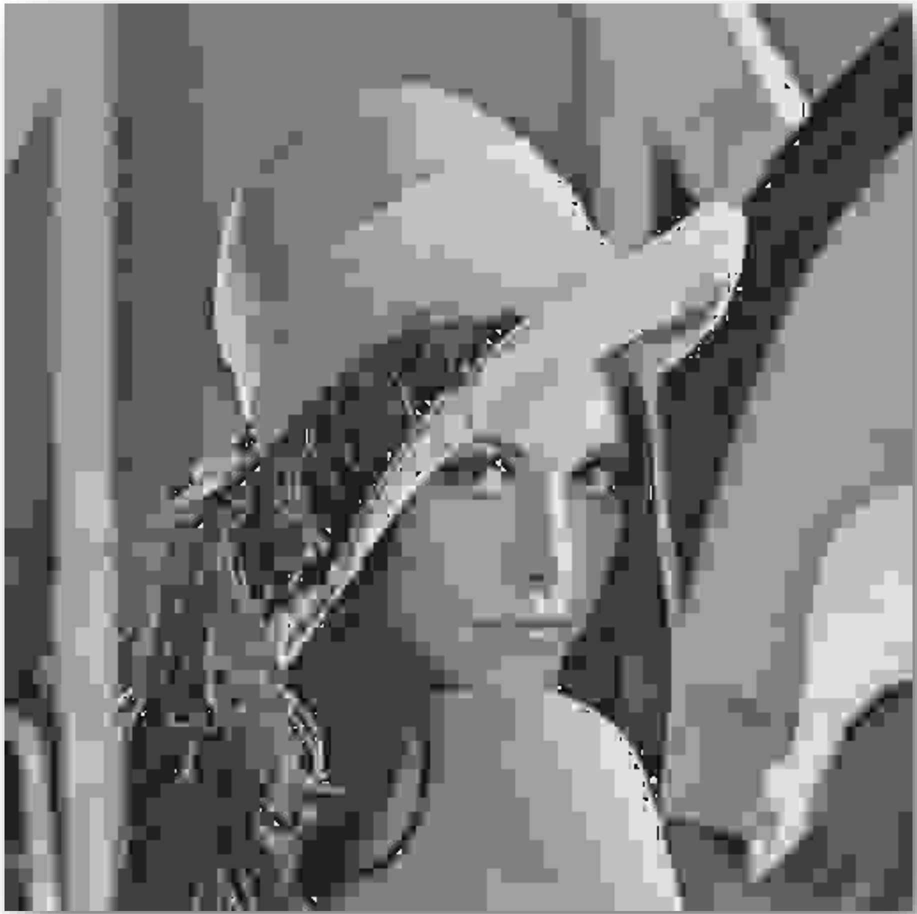

(37.60 PSNR)


(34.45 PSNR)


(31.11 PSNR)


(27.35 PSNR)

Cont.


23.54 PSNR

PSNR

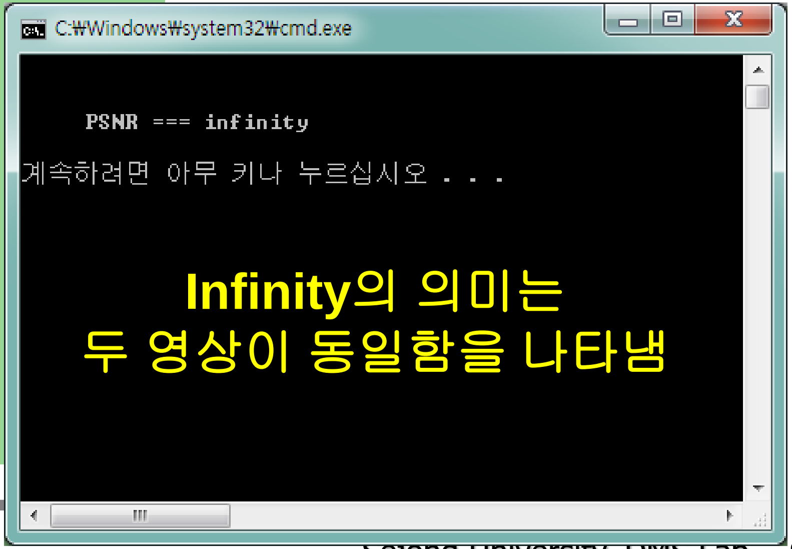
- 최대 신호 대 잡음비(Peak Signal-to-noise ratio, PSNR)
  - 주로 영상 또는 동영상 손실 압축에서 화질 손실 정보를 평가 할때 사용
  - 최대 신호 대 잡음비는 신호의 Power에 대한 고려 없이 평균 제곱 오차를 이용해서 계산

$$
\begin{aligned}
PSNR &= 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \\
&= 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right)
\end{aligned}
$$

Reference

- PSNR(Peak signal-to-noise ratio)
  - 두 영상을 비교 - 화질 측정
  - 아래 코드를 그대로 사용(size = 영상의 크기)

```c
/* PSNR구하는 함수 */
void PSNR( unsigned char **a, unsigned char **b )
{
    double M, psnr;

    M = MSE( a, b );                                // MSE(function) call

    if( M == 0 )
    {
        printf("\n\n     PSNR === infinity \n\n");  // 모든 MSE 값이 0과 같으면 무한대를 표시
    }else
    {
        psnr = 10*log10((255*255)/M);               // Calculation of PSNR
        printf("\n\n     PSNR  ===  %fl\n\n",psnr);
    }

}

double MSE( unsigned char **a, unsigned char **b )
{
    int i,j;
    double result,sum=0;

    for(i=0;i<size;i++)
        for(j=0;j<size;j++)
            sum+=(a[i][j]-b[i][j])*(a[i][j]-b[i][j]);

    result = (double)sum/(double)(size*size);   // Calculation of MSE
    return result;
}
```



```
     PSNR === infinity

계속하려면 아무 키나 누르십시오 . . .
```

Infinity의 의미는
두 영상이 동일함을 나타냄

LPF in frequency domain

- Design ideal LPF: $H_{lp}(u,v)$
- $F(u,v)\ H_{lp}(u,v)$
- $Y(u,v)=\text{InvDFT}\{F(u,v)\ H_{lp}(u,v)\}$ while changing cutoff frequency
- $f_{lp}(x,y)$

- Design Gaussian LPF: $H_{glp}(u,v)$
- $F(u,v)\ H_{glp}(u,v)$
- $Y(u,v)=\text{InvDFT}\{F(u,v)\ H_{glp}(u,v)\}$ while changing cutoff frequency
- $f_{lp}(x,y)$

- 4 point

HPF in frequency domain

- Design ideal HPF: $H_{hp}(u,v)$
- $F(u,v)\ H_{hp}(u,v)$
- $Y(u,v)=InvDFT\{F(u,v)\ H_{hp}(u,v)\}$ while changing cutoff frequency
- $f_{hp}(x,y)$

- Design Gaussian HPF: $H_{ghp}(u,v)$
- $F(u,v)\ H_{ghp}(u,v)$
- $Y(u,v)=InvDFT\{F(u,v)\ H_{ghp}(u,v)\}$ while changing cutoff frequency
- $f_{hp}(x,y)$

- 4 point