DIP(Digital Image Processing)
Program Exercise

(해당 강의자료의 배포 및 무단 복제를 금함)

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

김남욱, 김명준, 정지연, 김양우, 이영렬

Program Exercises

1. 2D Discrete Fourier Transform & IDFT
   - Subsampling(Zoom-out) -> 256x256
2. DCT & DST
   - Discrete Cosine Transform & IDCT
   - Discrete Sine Transform & IDST

❖ 점수배점 (총 10점)
- 256x256 DFT&IDFT 5점
- 8x8, 16x16 DCT&DST , IDCT, IDST 5점

❖ 채점 및 검사
- 장소 : 대양 AI 센터 817호
- 조교 : 대학원생

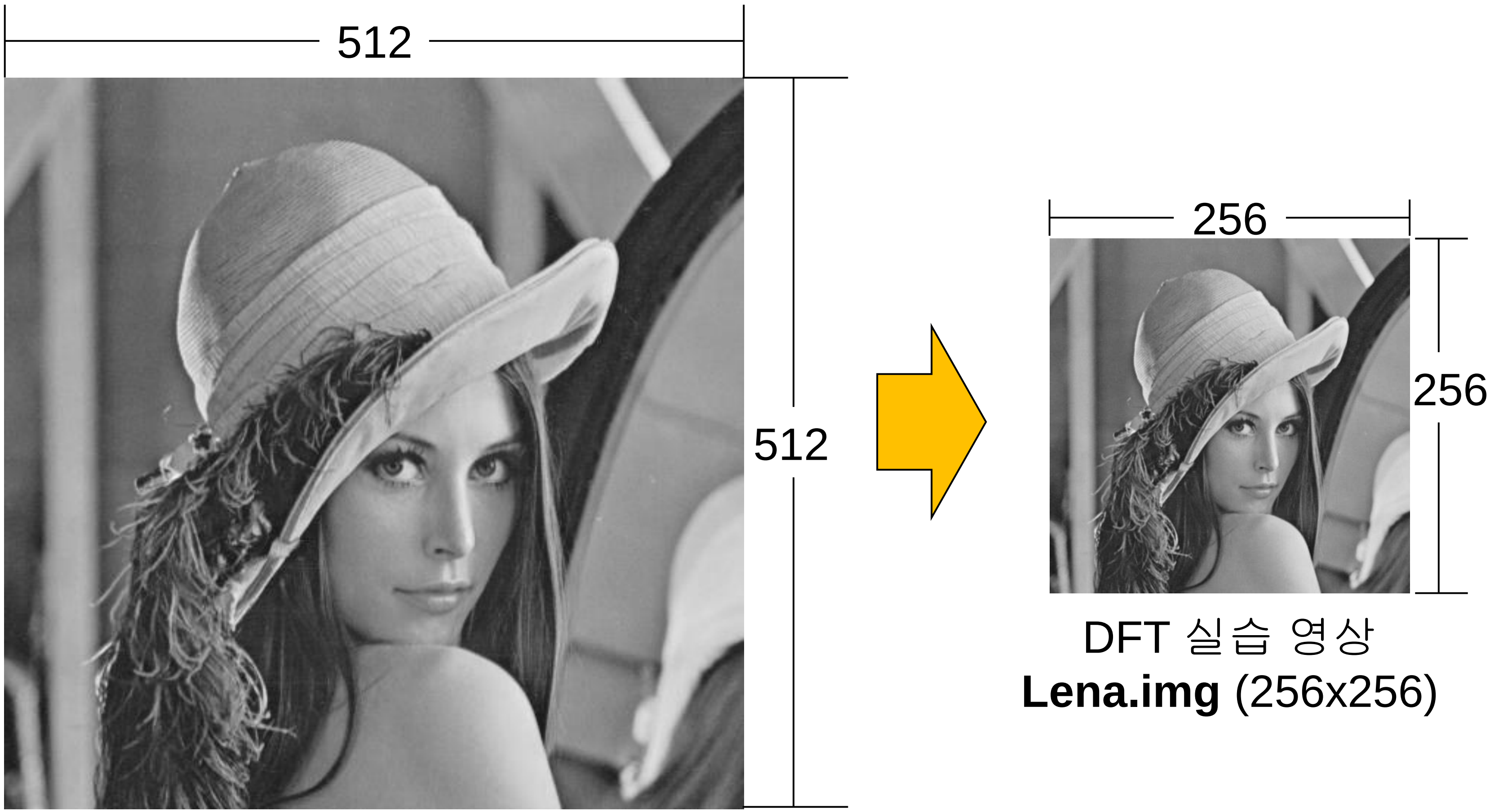
실습이미지 - lena.img

512

512

- 512 x 512 (pixel by pixel size)
- 밝기 값만 가지고 있는 이미지
- 8 bits range

DFT 하기 전 - Lana 이미지 축소(256x256)



DFT 실습 영상
Lena.img (256x256)

1. 2D Discrete Fourier Transform & IDFT

- DFT 2D

$$F(u,v)=\frac{1}{N\times N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m,n)e^{-j\left(\frac{2\pi}{N}\right)um}e^{-j\left(\frac{2\pi}{N}\right)vn}$$

$$=\frac{1}{N\times N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m,n)e^{-j\frac{2\pi}{N}(um+vn)}$$

$$e^{j\theta}=\cos\theta+j\sin\theta$$

- Inverse DFT

$$f(m,n)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} F(u,v)e^{j\left(\frac{2\pi}{N}\right)um}e^{j\left(\frac{2\pi}{N}\right)vn}$$

$$=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} F(u,v)e^{j\frac{2\pi}{N}(um+vn)}$$

Basic steps for filtering in frequency domain

- Follow the following procedures:

1. Multiply input image by **$(-1)^{m+n}$** to center the transform
2. Compute F(u, v), the DFT of the image from the previous step
(3. Multiply F(u, v) by a filter function H(u, v) studied in Image Enhancement)
4. Compute inverse DFT of the result
5. Obtain the real part of the result in the step 4.
6. Multiply the result by **$(-1)^{m+n}$**

using:

$$f(m,n)\exp\left[j\frac{2\pi}{N}u_0 m\right]\exp\left[j\frac{2\pi}{N}v_0 n\right] \Leftrightarrow F(u-u_0, v-v_0)$$

Shift in Frequency domain

$$when\ u_0 = v_0 = 128\ (N = 256)$$

$$f(m,n)(-1)^{m+n} \Leftrightarrow F(u-128, v-128)$$

2D Discrete Fourier Transform

1. Multiply input image by $(-1)^{m+n}$ to center the transform
2. Compute F(u, v), the DFT of the image from the previous step

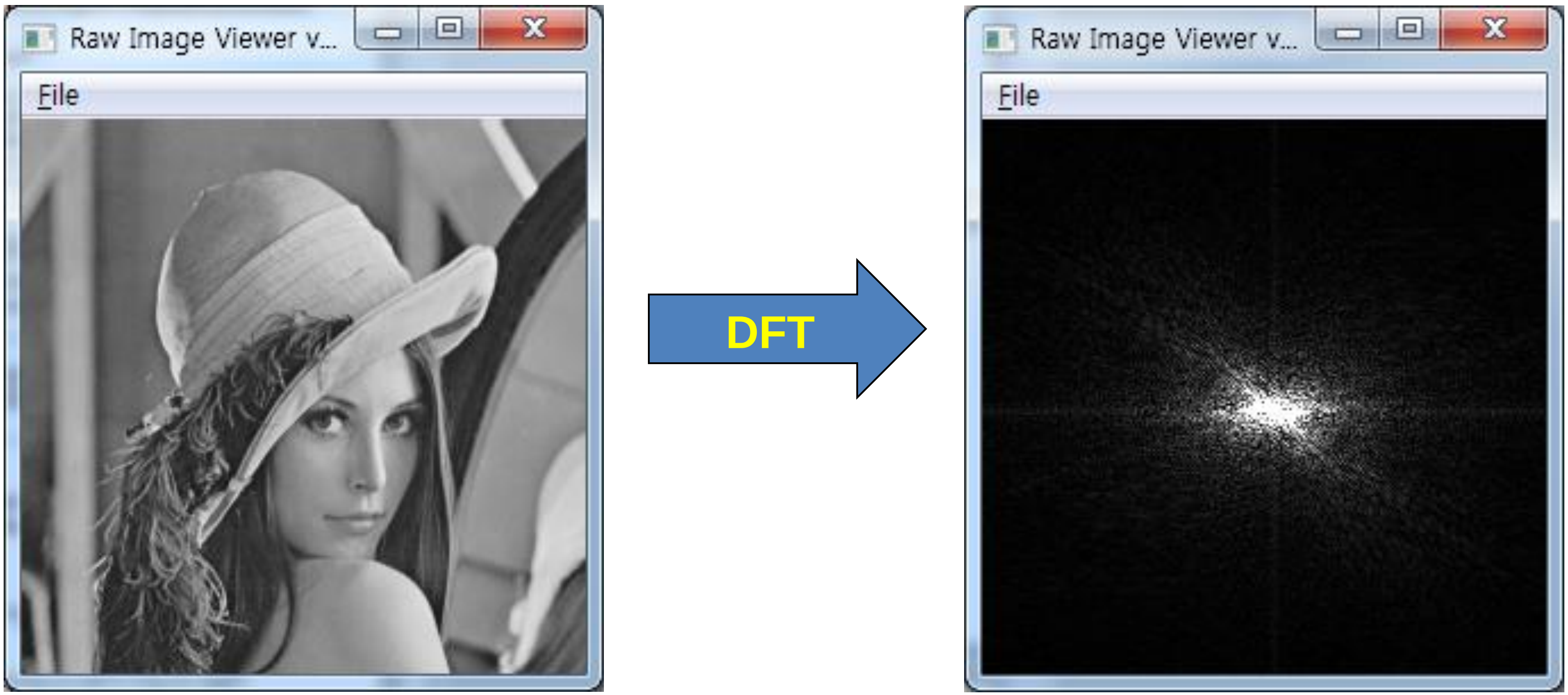
$$F(u-128, v-128) = \frac{1}{NXN} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} (-1)^{m+n} f(m,n) e^{-j2\pi(\frac{um}{N}+\frac{vn}{N})}$$

$$|\mathrm{F(u,v)}| = Coef(u,v) = sqrt(Re\{F(u,v)\}^2 + Im\{F(u,v)\}^2)$$

Only display purpose of DFT for the Magnitude, |F(u,v)|,

$$ByteCoeff(u,v) = 255.0 \times \log_{10}(coef(u,v) + 1.0) / \log_{10}(DCvalue + 1.0)$$

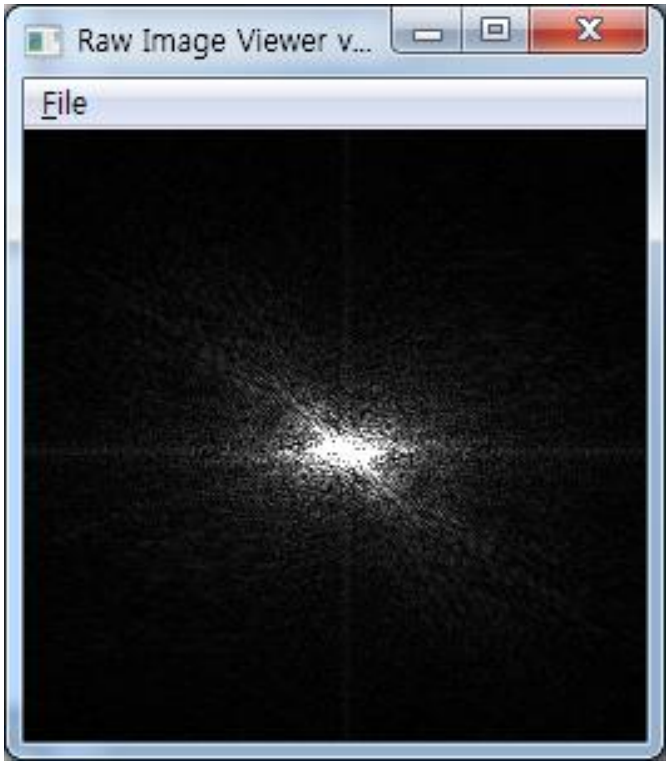
2D Discrete Fourier Transform 결과



Original
Lena.img
(256x256)

DFT.img

2D Inverse Discrete Fourier Transform 결과

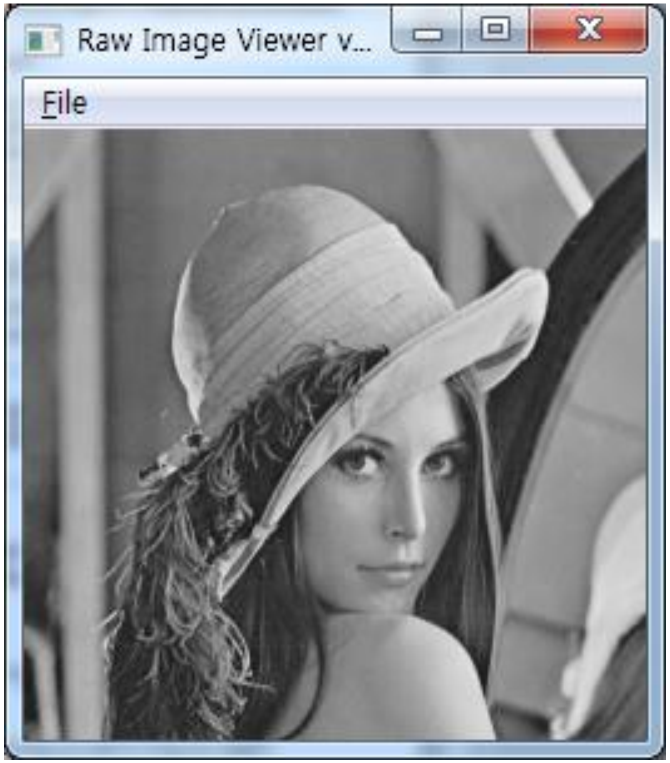


IDFT

DFT.img

Inverse DFT
Out.img

원본 영상과 IDFT영상 비교



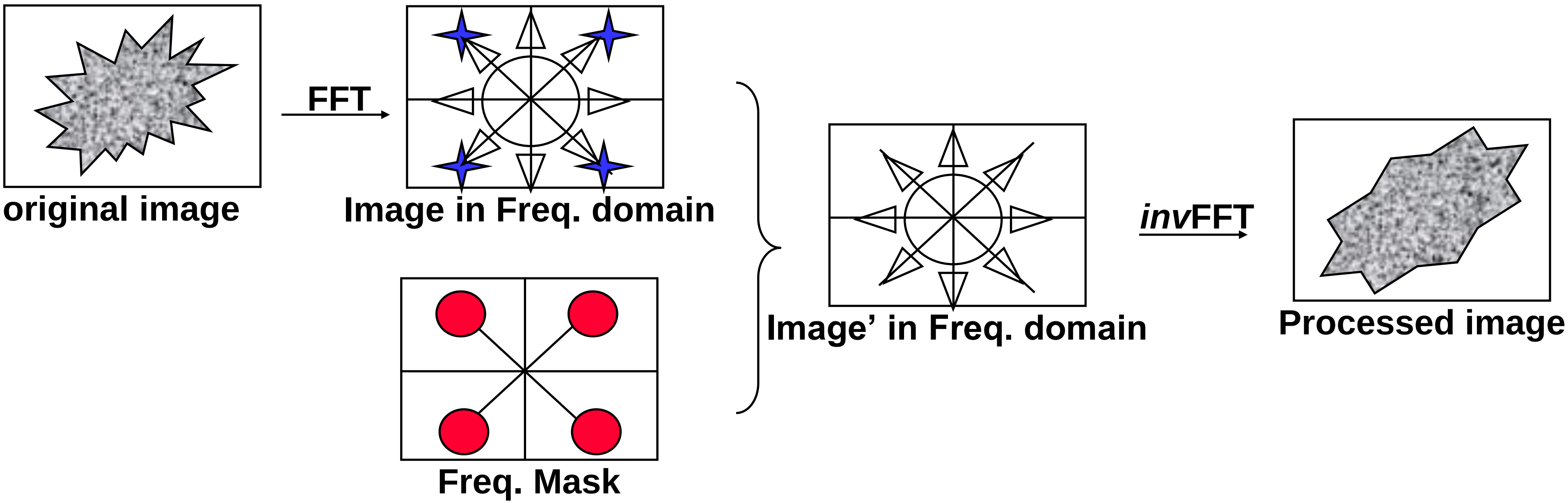
Original
Lena.img
(256x256)

=
Same Image

Inverse DFT
Out.img

Spatial Filtering in Frequency Domain

- Frequency mask



2. DCT&DST

- **DCT & DST??**
  - 저주파/고주파 성분을 구분
  - 정보 손실이 없는 무손실 변환
    - ✓ 변환전은 영상의 성분이 화면에 고루 분포
    - ✓ 변화후는 성분이 저주파 영역으로 집중

ex)

  - 저주파/고주파 성분을 분리, 고주파 성분 버림
  - 사람 눈은 일반적으로 저주파보다 고주파 성분에 둔감
    - ✓ 일반적으로 영상들은 저주파 성분이 많이 분포하는 경우가 많음

2. DCT & DST

- DCT(type 2) – 8x8, 16x16 implementation

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

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

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

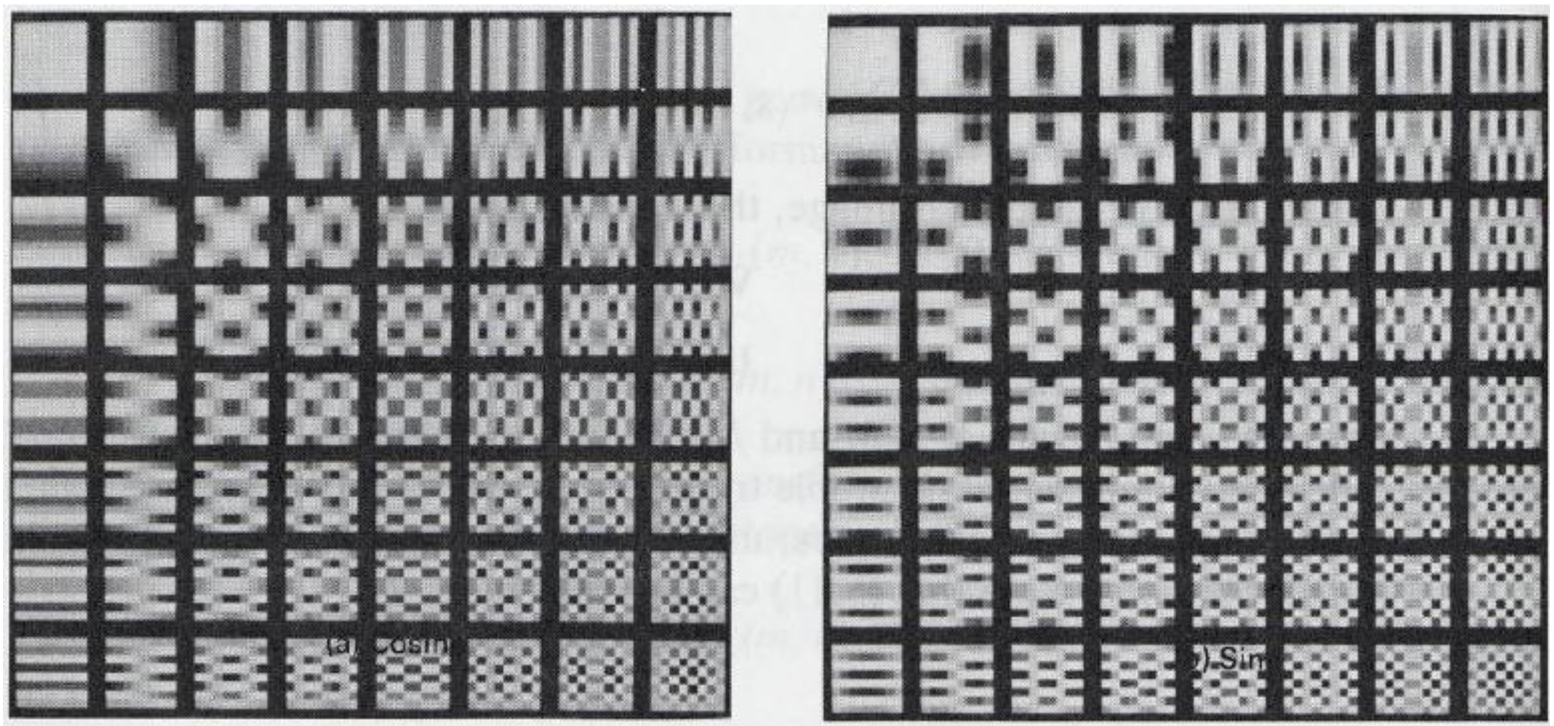
- DST – 8x8, 16x16 implementation

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

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

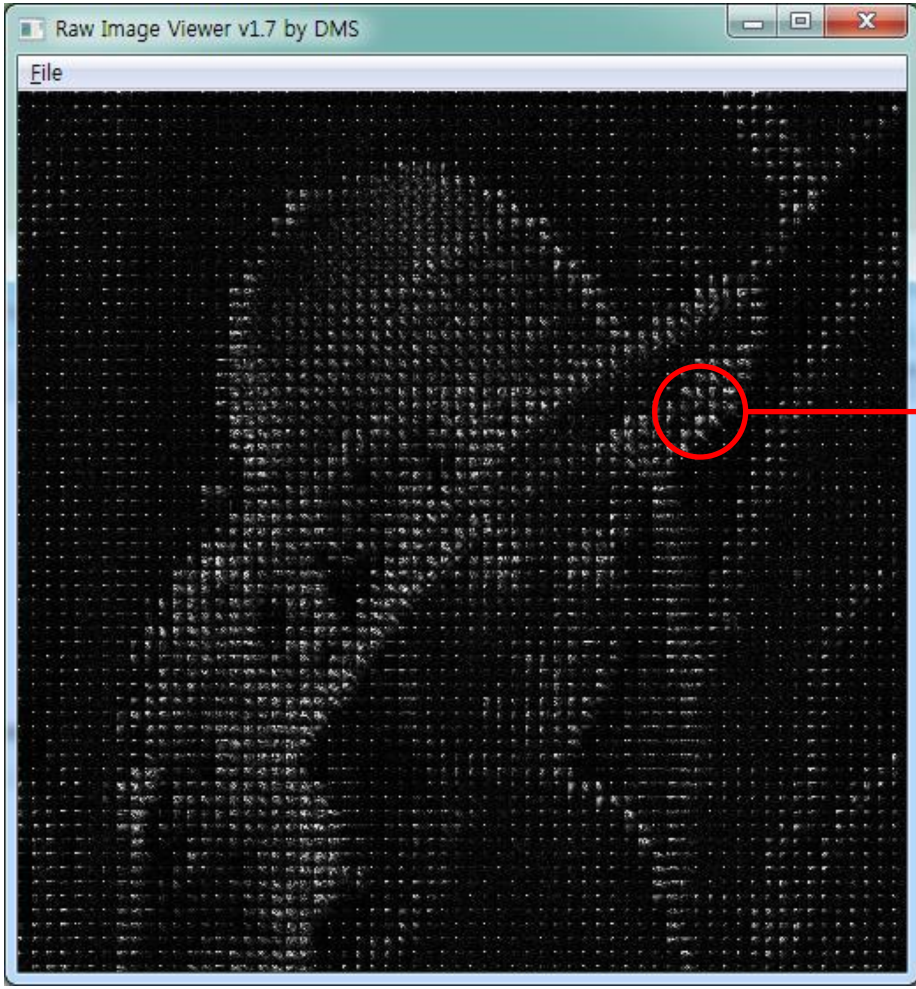
8×8 2D transform

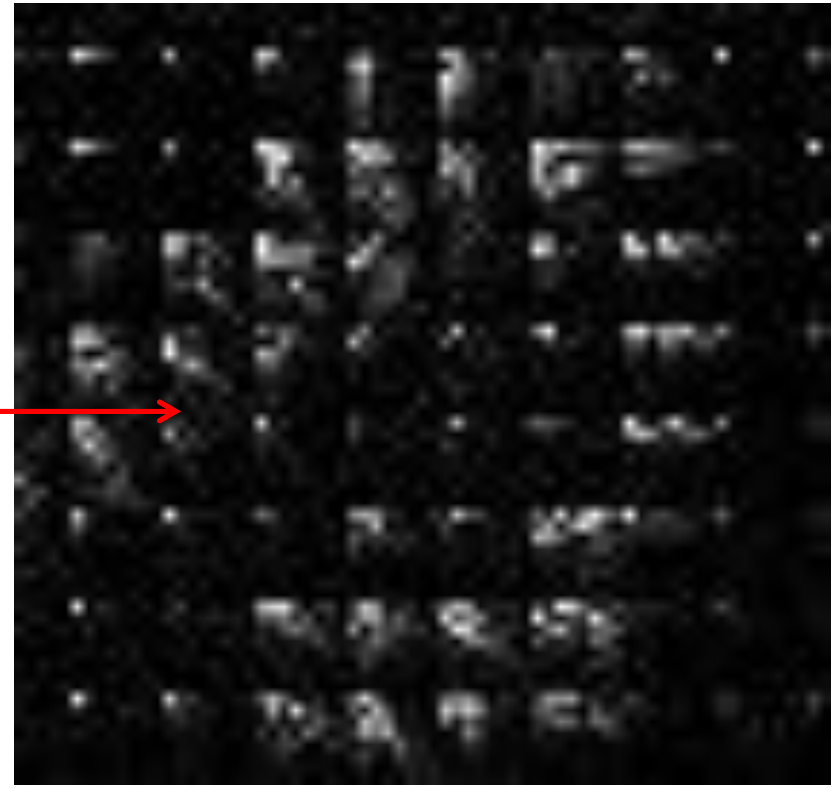
- Basis Kernels



Cosine Transform **Sine Transform**

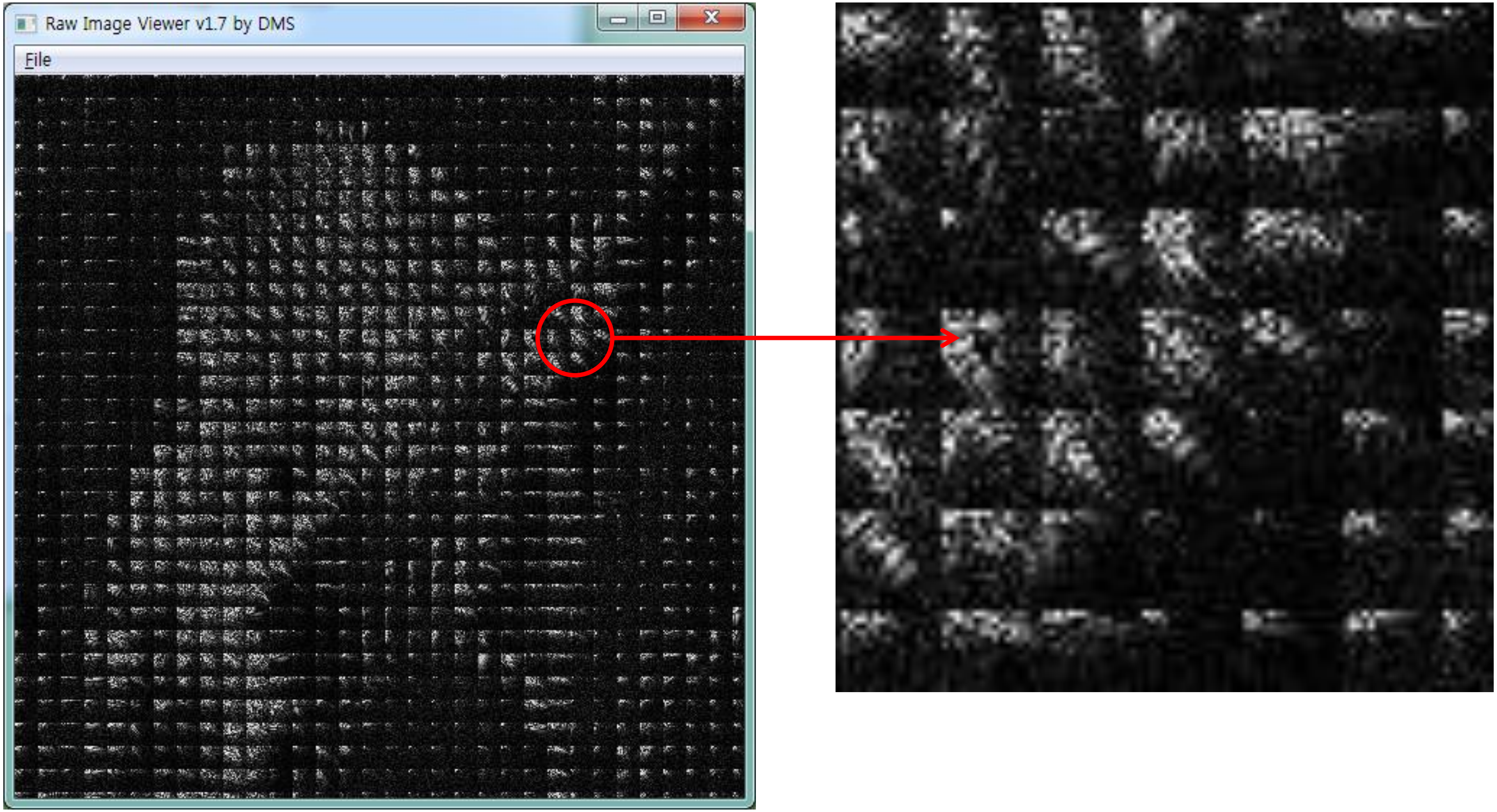
(1) 8x8 DCT(Discrete Cosine Transform)



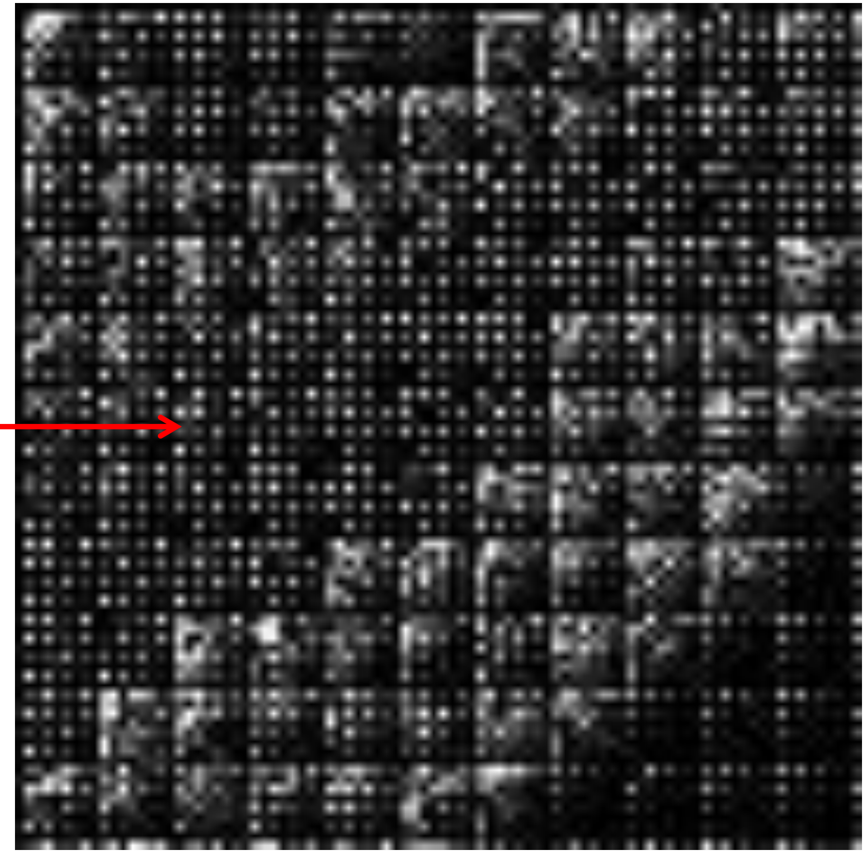


DCT **Lena.img** (512x512)

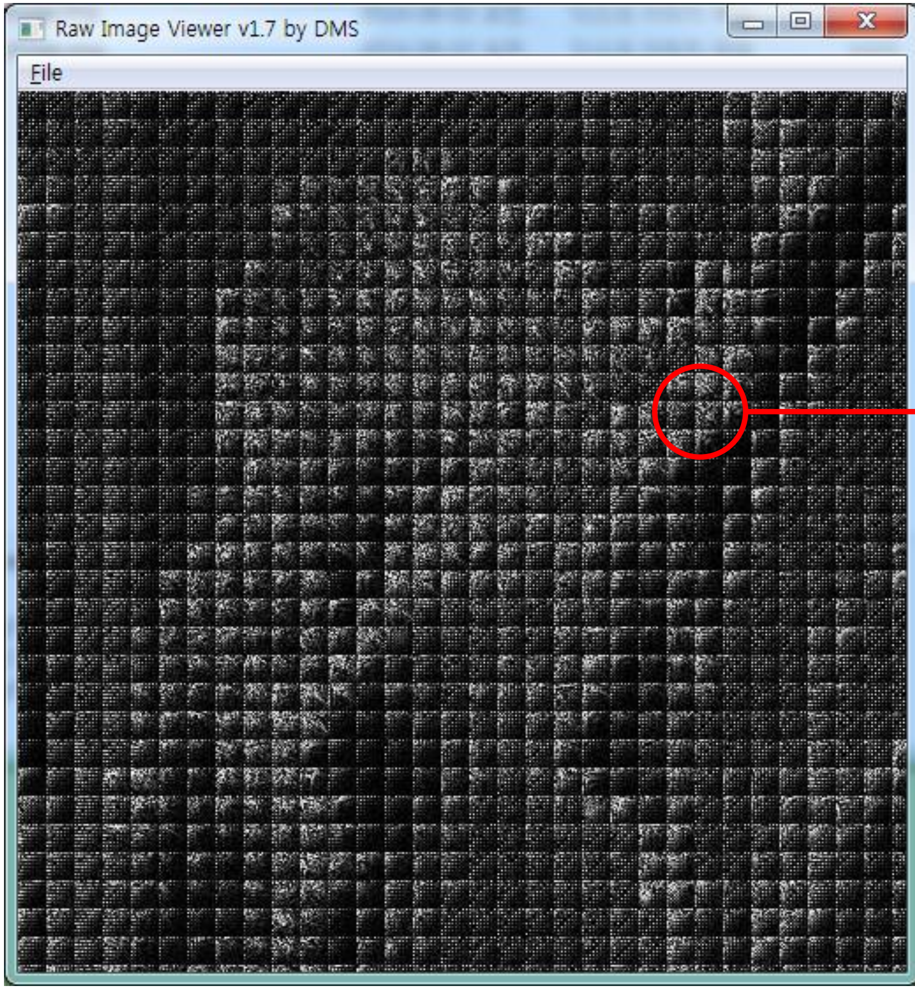
(2) 16x16 DCT(Discrete Cosine Transform)

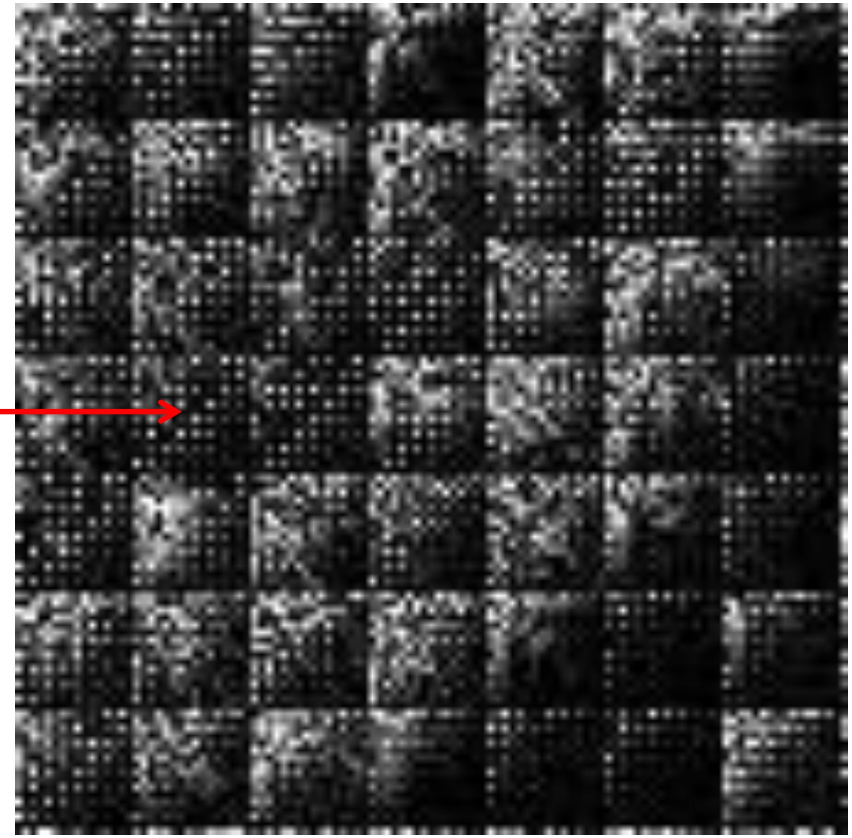


(3) 8x8 DST(Discrete Sine Transform)



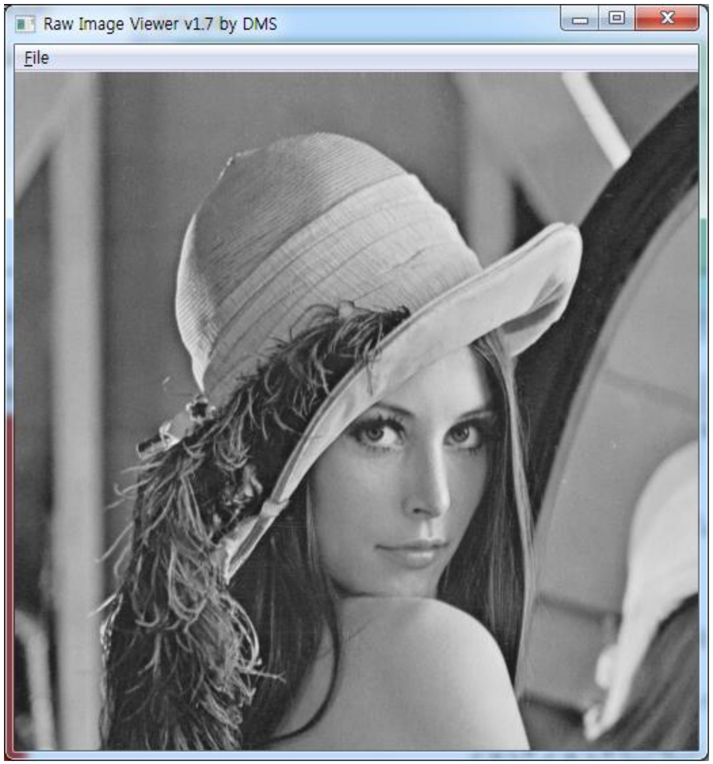
(4) 16x16 DST(Discrete Sine Transform)



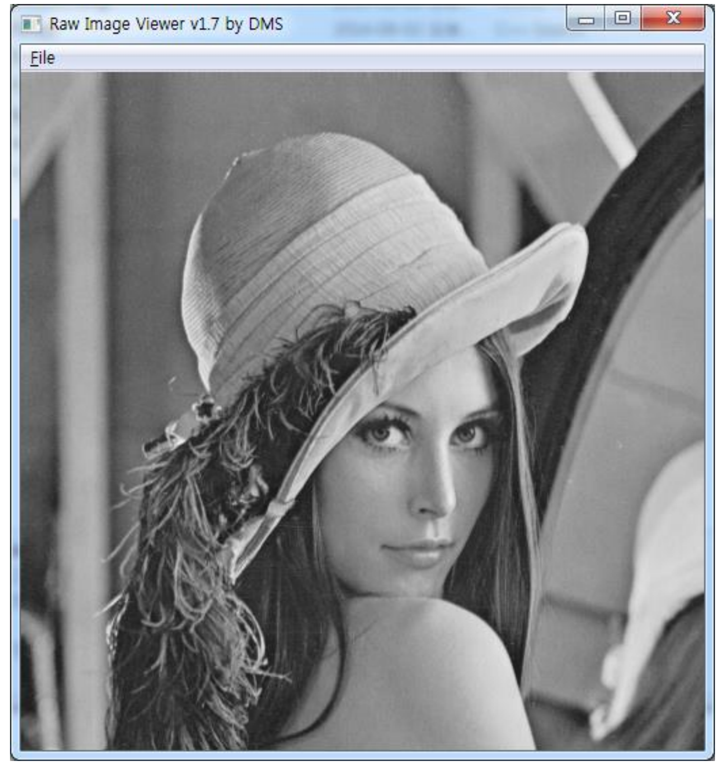


Inverse Discrete Cosine Transform(IDCT Result)

- DCT -> IDCT 영상은 같음(DST도 동일한 결과)



Original **Lena.img** (512x512)



IDCT **Lena.img** (512x512)

Reference

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

```
/* 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;
}
```