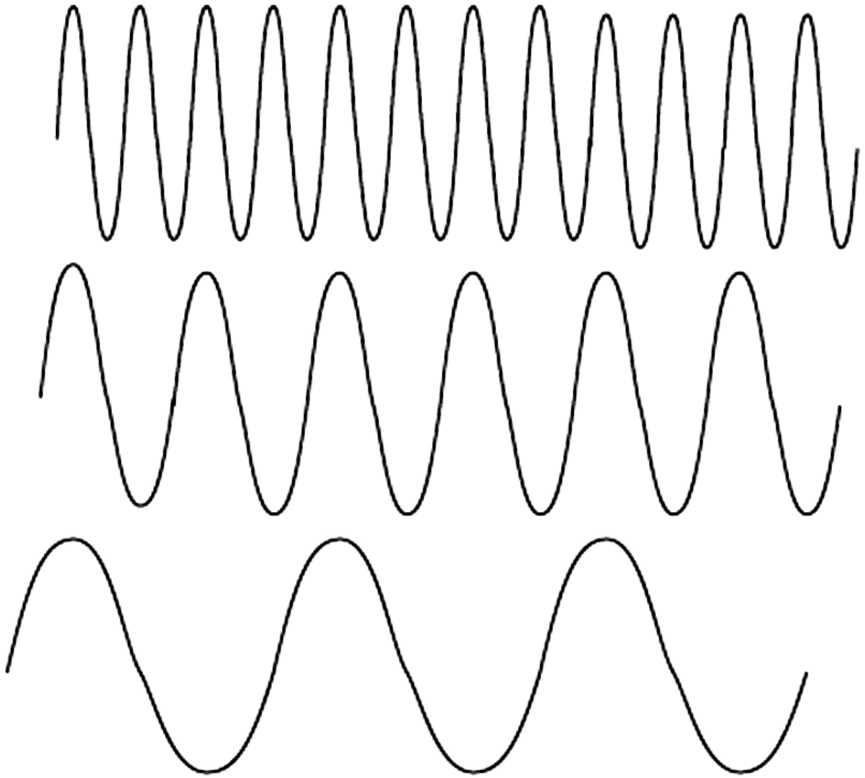
# Frequency Domain

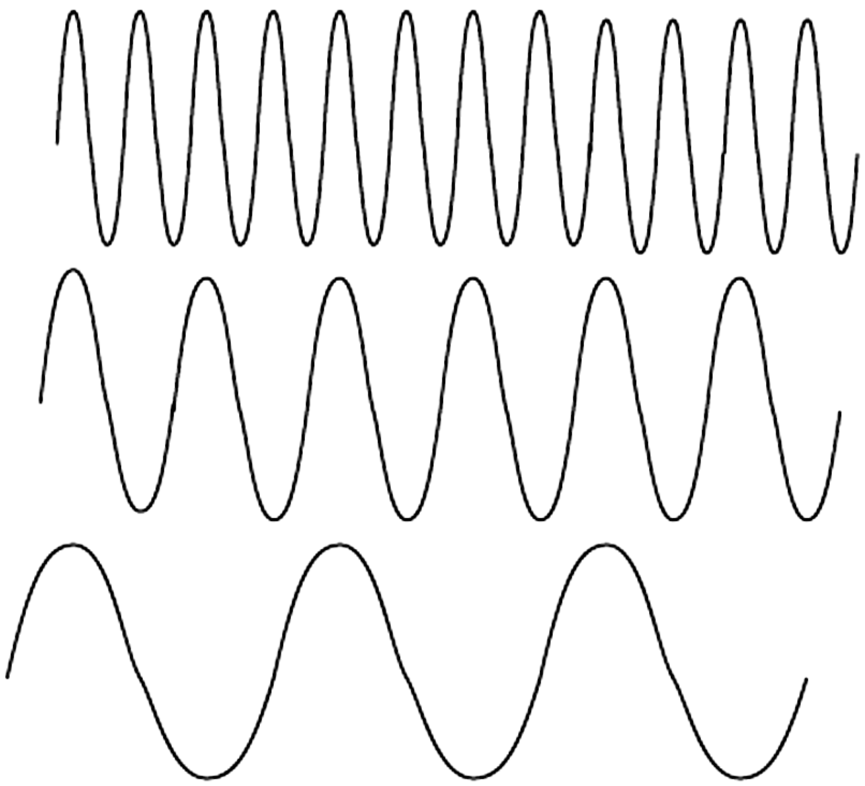


# Fourier Transform

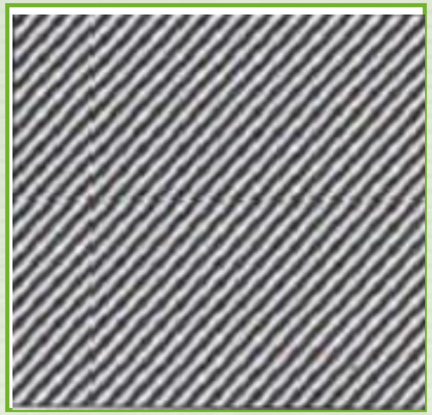
$$F(u) = \int_{-\infty}^{\infty} f(x)e^{-j2\pi ux}\,dx$$

$$f(x) = \int_{-\infty}^{\infty} F(u)e^{j2\pi ux}\,du$$

# Filtering in Frequency Domain



# 2D Frequency Domain

# 1D Discrete Fourier Transform

$$F(u) = \frac{1}{W}\sum_{x=0}^{W-1} f(x)e^{-j2\pi ux/W}, u = 0,1,\dots,W-1$$

$$f(x) = \sum_{u=0}^{W-1} F(u)e^{j2\pi ux/W}, x = 0,1,\dots,W-1$$

# 2D Discrete Fourier Transform

$$F(u,v) = \frac{1}{WH}\sum_{x=0}^{W-1}\sum_{y=0}^{H-1} f(x,y)e^{-j2\pi(ux/W+vy/H)}$$

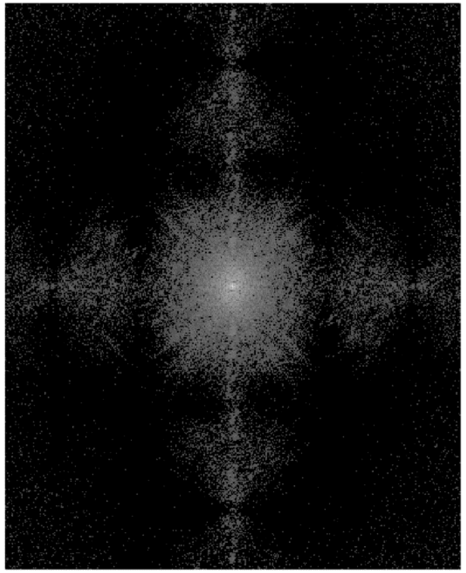
$$f(x,y) = \sum_{u=0}^{W-1}\sum_{v=0}^{H-1} F(u,v)e^{j2\pi(ux/W+vy/H)}$$

# 2D Discrete Fourier Transform

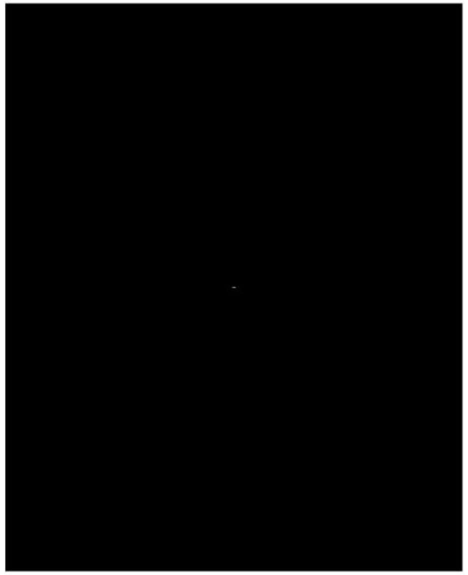
$$F(u,v) = R(u,v) + jI(u,v)$$

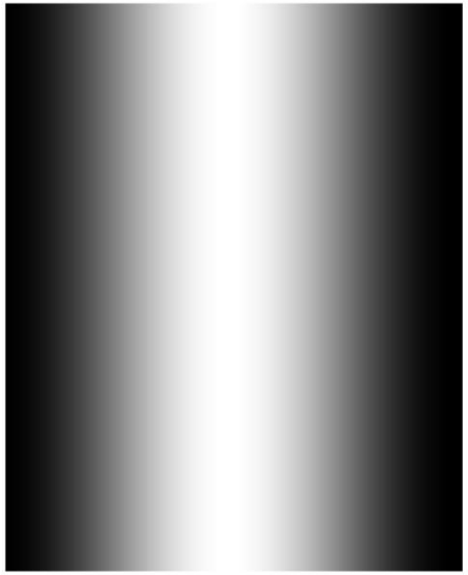
$$|F(u,v)| = [R^2(u,v) + I^2(u,v)]^{1/2}$$

$$\phi[u,v] = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$$

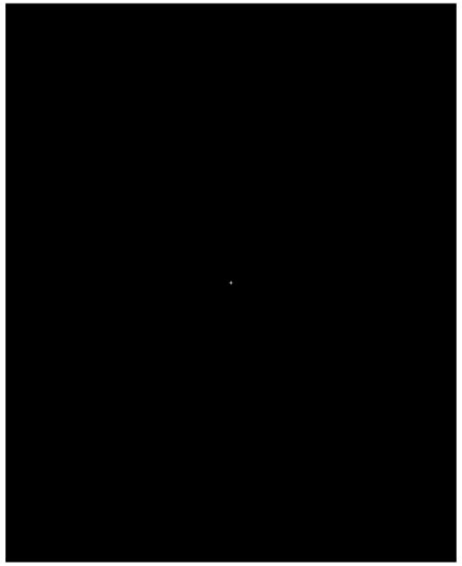


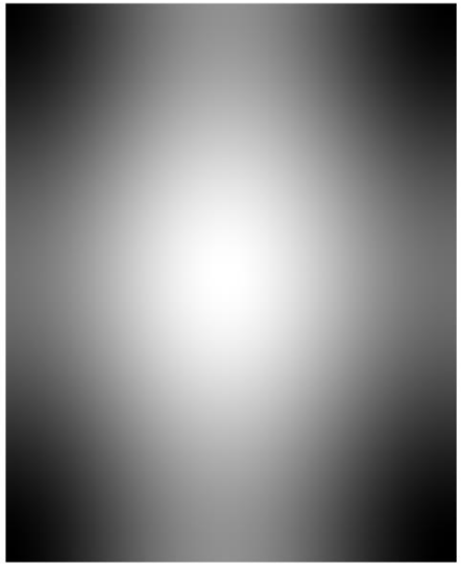
# 2D Frequency Domain



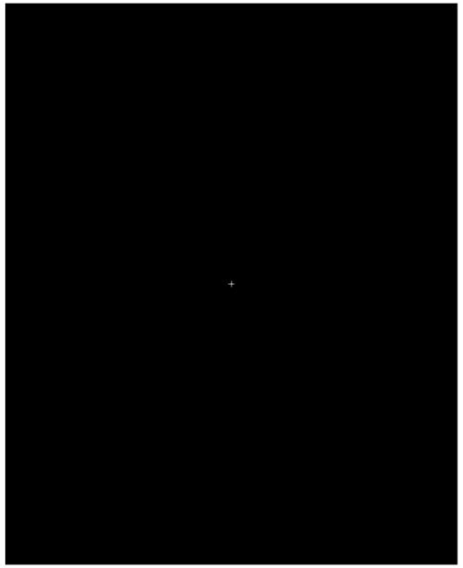


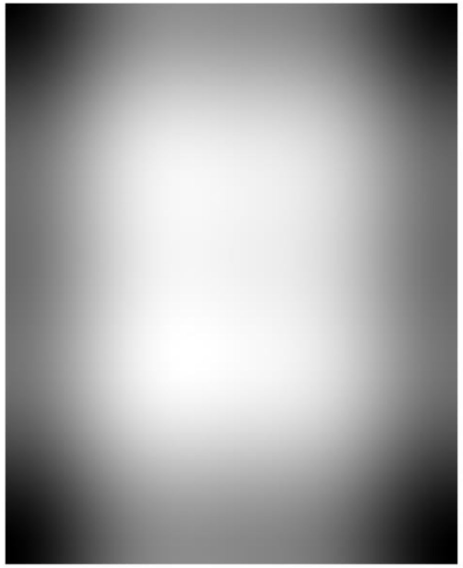
# 2D Frequency Domain



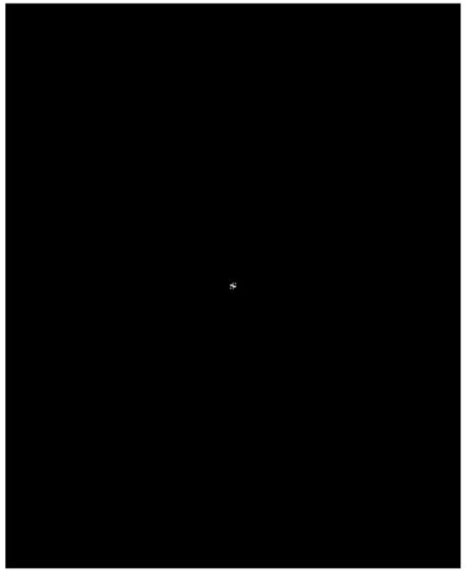


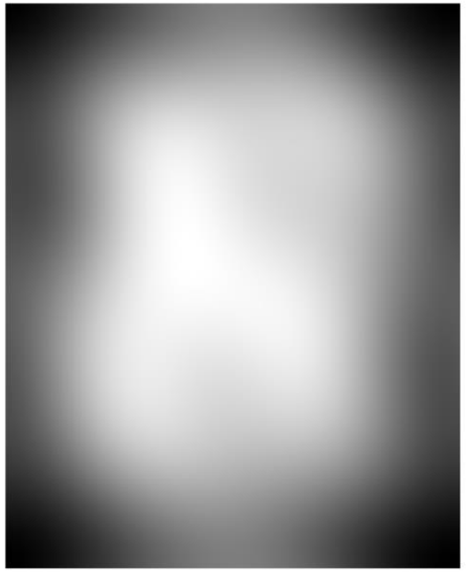
# 2D Frequency Domain



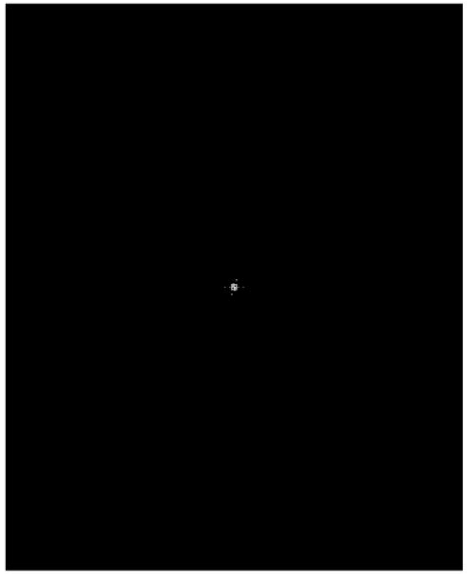


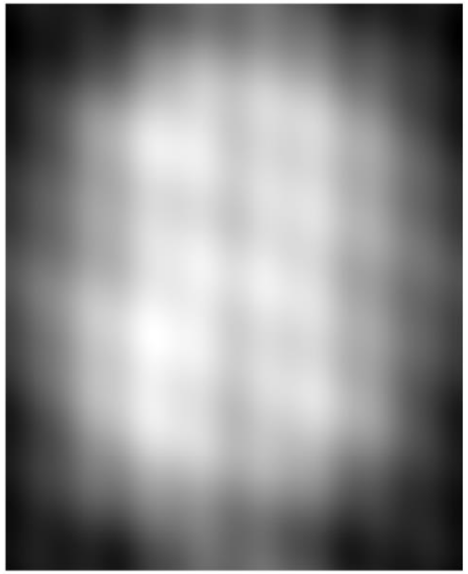
# 2D Frequency Domain



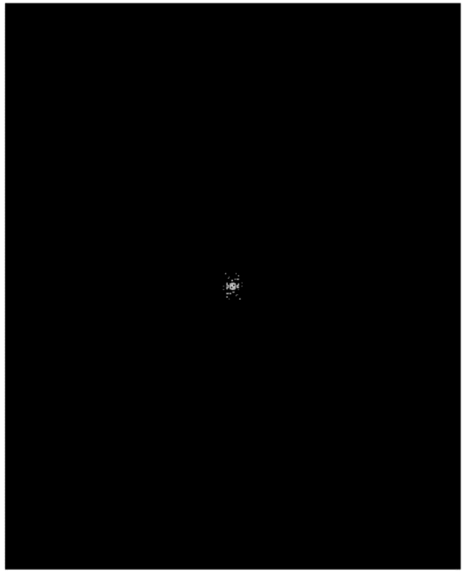


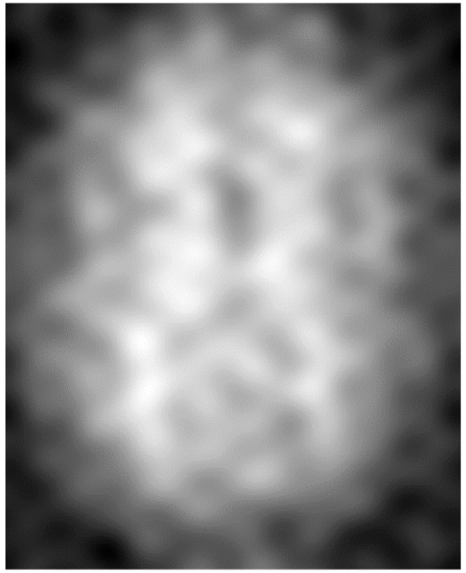
# 2D Frequency Domain



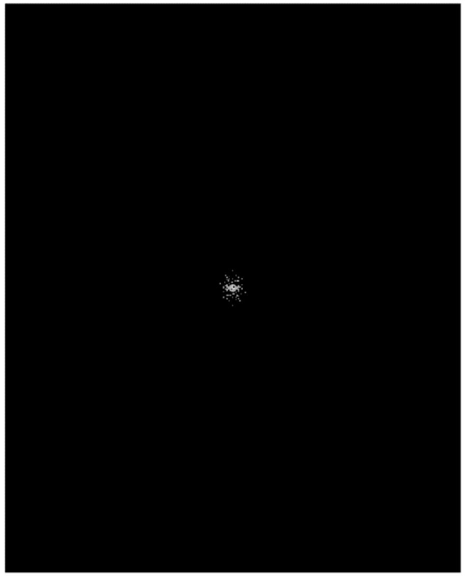


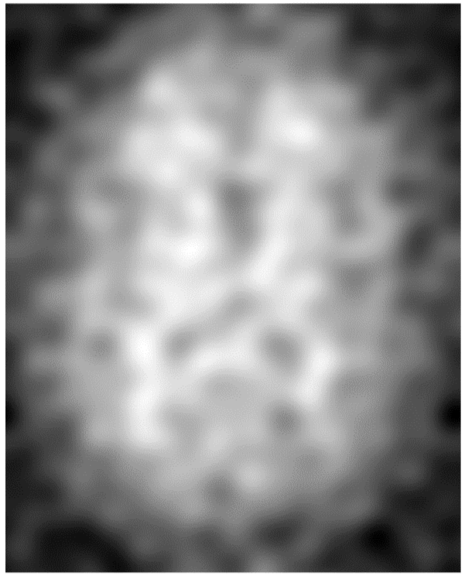
# 2D Frequency Domain



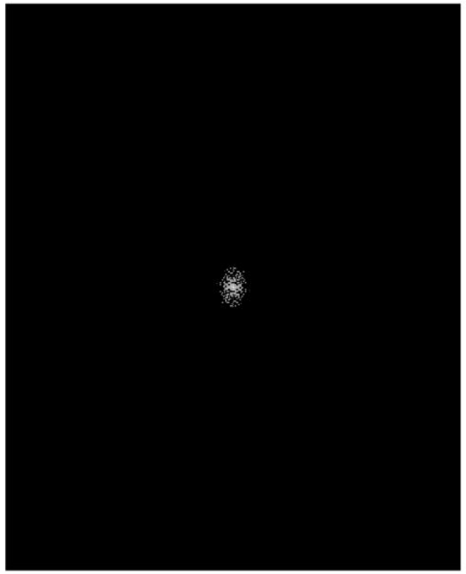


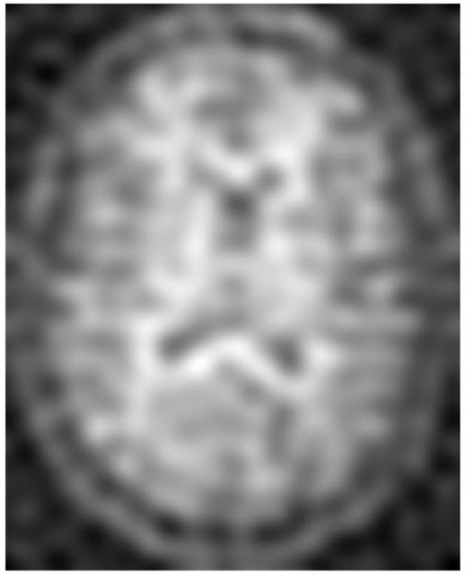
# 2D Frequency Domain



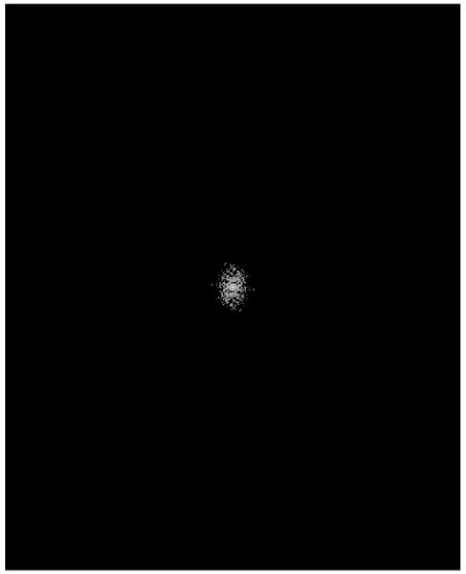


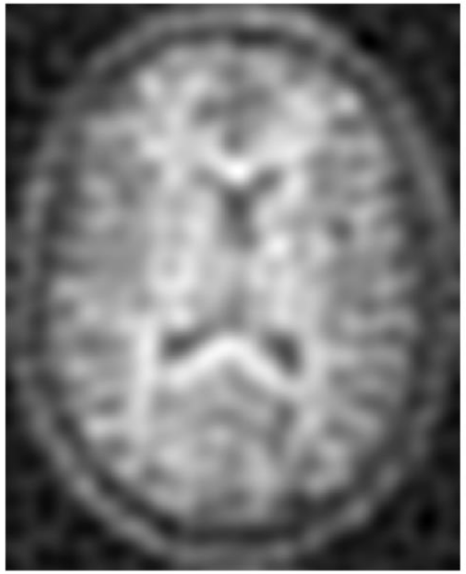
# 2D Frequency Domain



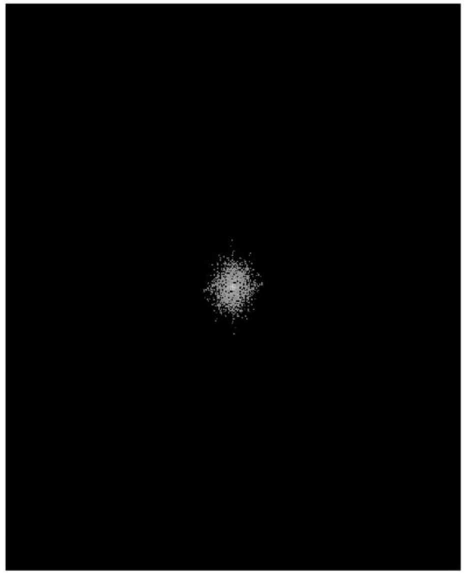


# 2D Frequency Domain

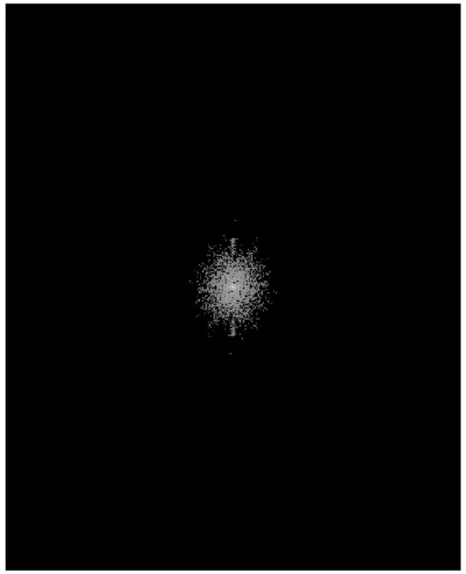


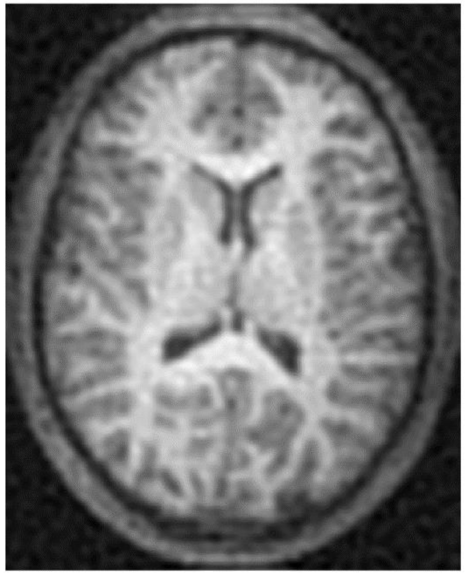


# 2D Frequency Domain

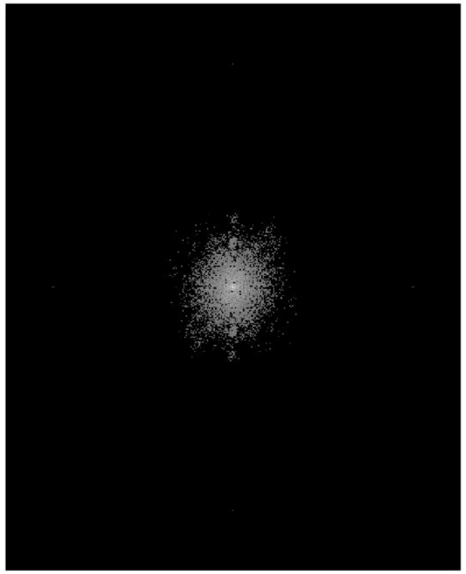


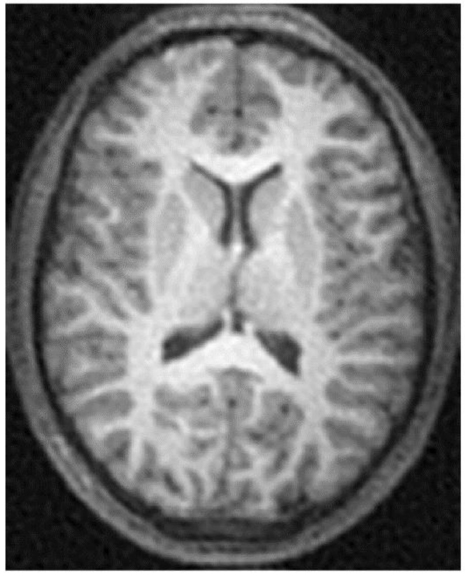
# 2D Frequency Domain



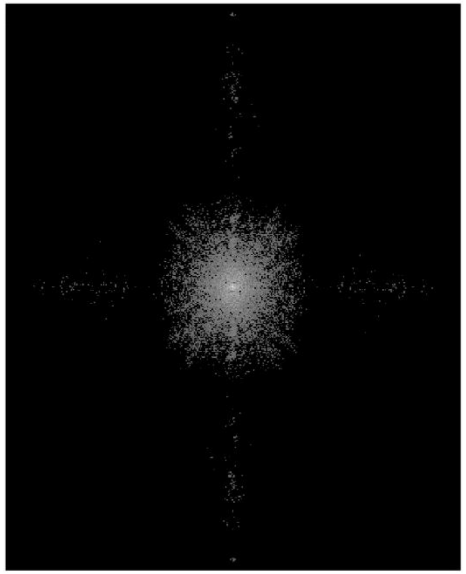


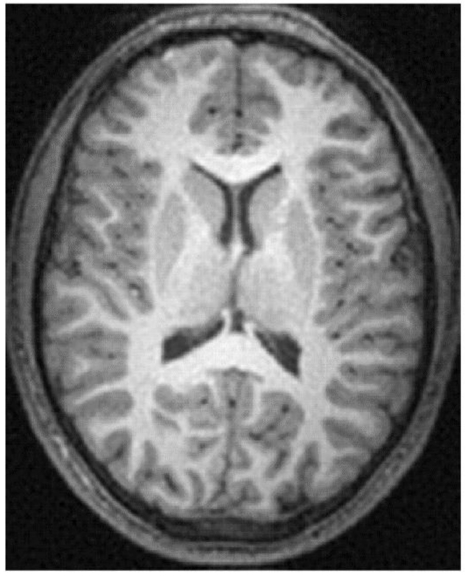
# 2D Frequency Domain



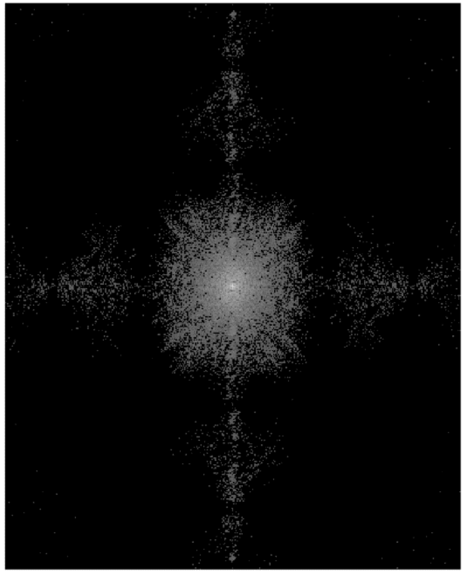


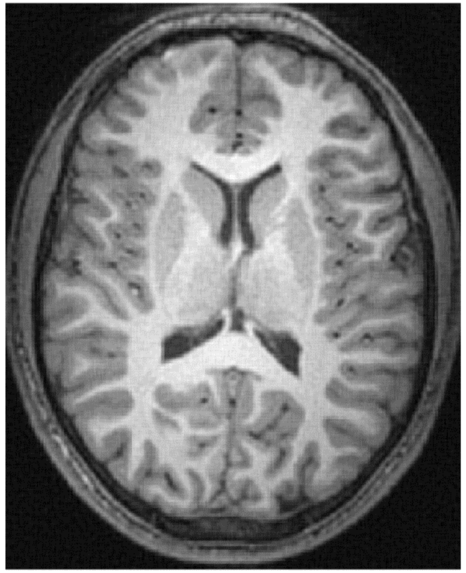
# 2D Frequency Domain



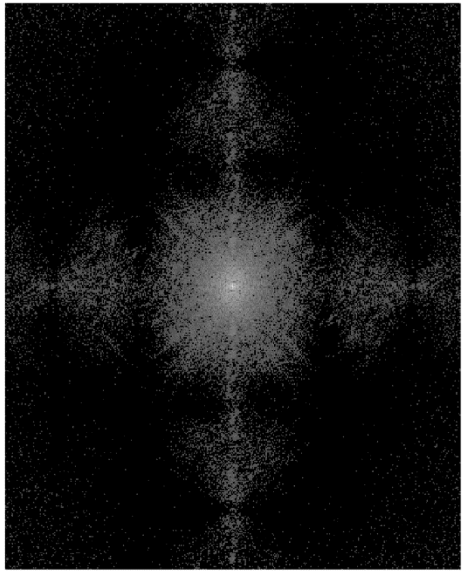


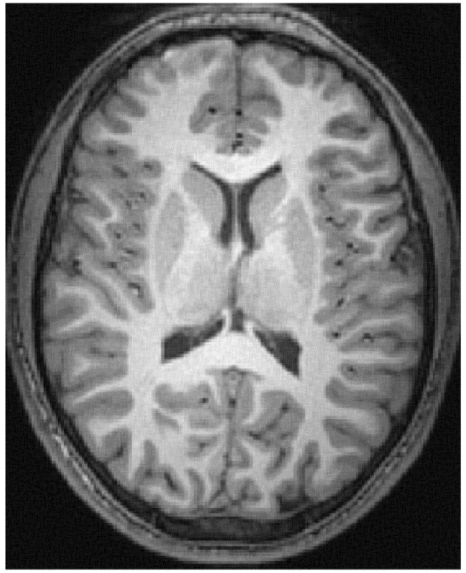
# 2D Frequency Domain



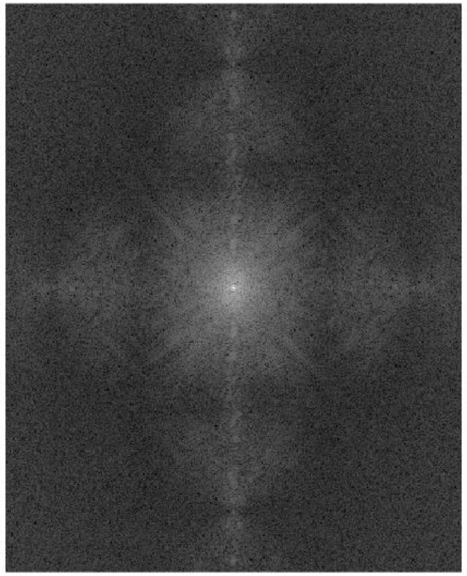


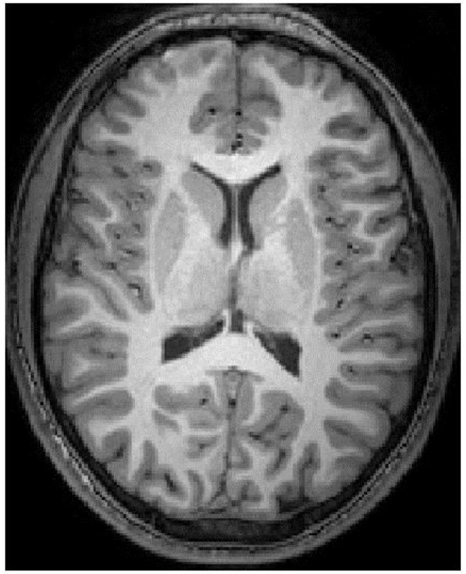
# 2D Frequency Domain



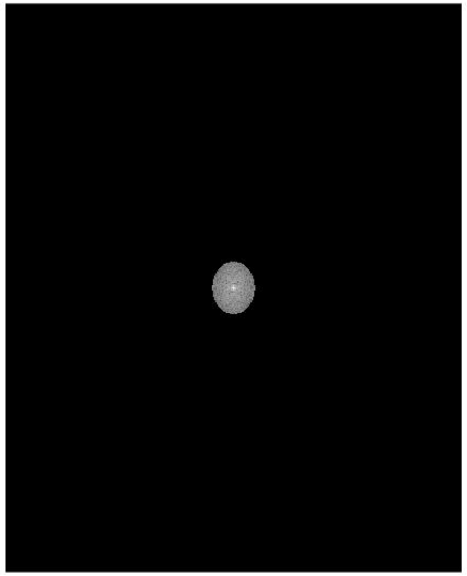


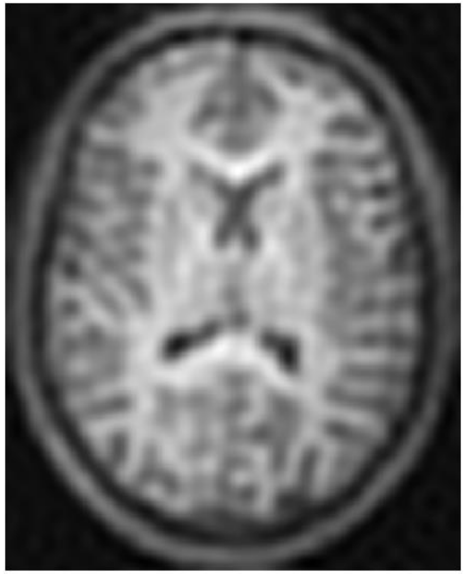
# 2D Discrete Fourier Transform



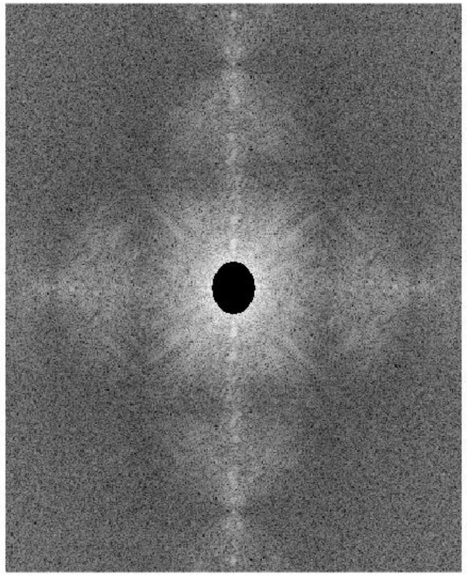


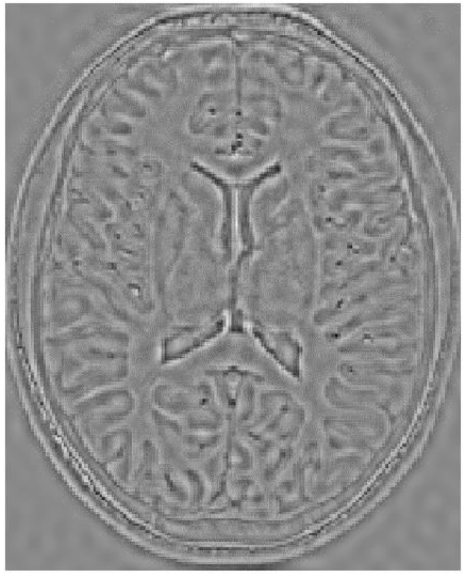
# Filtering in Frequency Domain



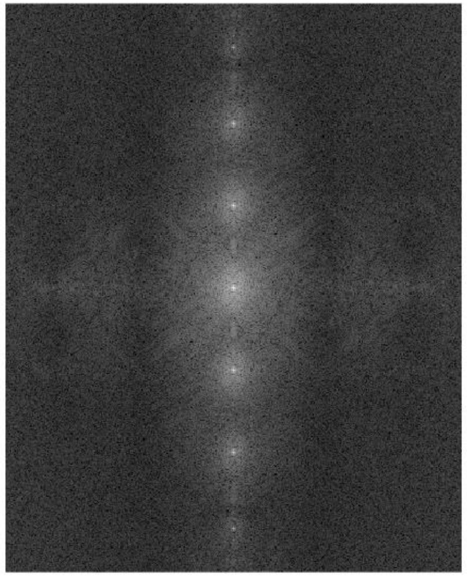


# Filtering in Frequency Domain

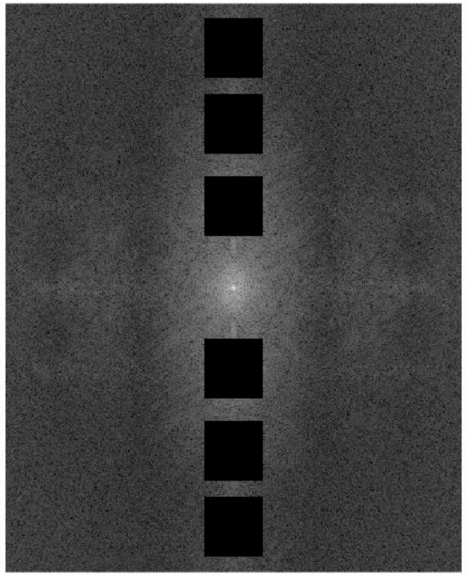




# Filtering in Frequency Domain



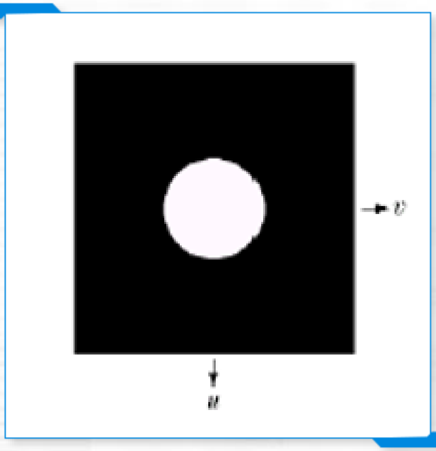
# Filtering in Frequency Domain



# Filtering in Frequency Domain

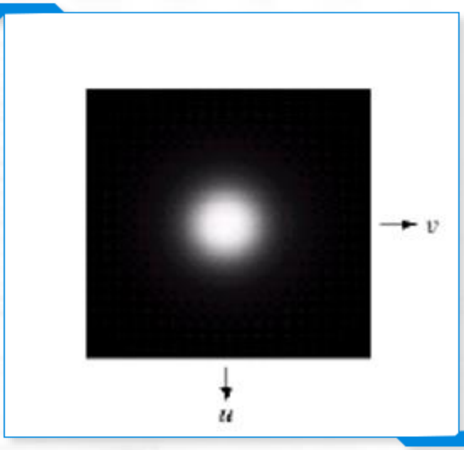
$$H(u,v) = \begin{cases} 1, \text{if } D(u,v) \leq D_0 \\ 0, \text{if } D(u,v) > D_0 \end{cases}$$

$$D(u,v) = \sqrt{(u - W/2)^2 + (v - H/2)^2}$$



# Filtering in Frequency Domain

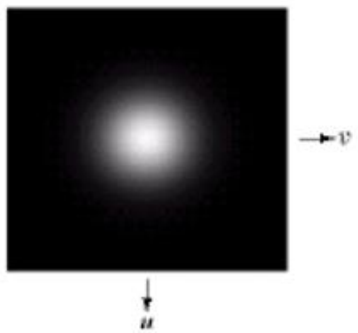
$$H(u,v) = \frac{1}{1 + \left[\frac{D(u,v)}{D_0}\right]^{2n}} = \frac{1}{1 + \left[\frac{(u - W/2)^2 + (v - H/2)^2}{D_0}\right]^{n}}$$



# Filtering in Frequency Domain

$$H(u,v) = e^{-\frac{D^2(u,v)}{2D_0^2}}$$

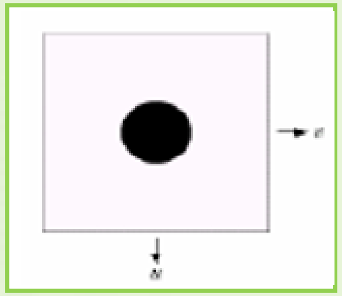
$$D(u,v) = \sqrt{(u - W/2)^2 + (v - H/2)^2}$$



# High Frequency Filters

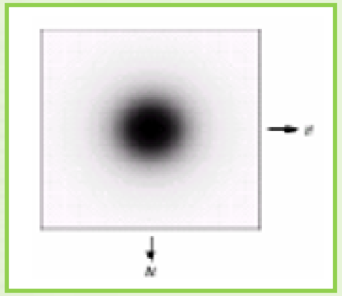
$$H_{lp}(u,v) = \begin{cases} 1, \text{if } D(u,v) \le D_0 \\ 0, \text{if } D(u,v) > D_0 \end{cases}$$

$$H_{hp}(u,v) = 1 - H_{lp}(u,v)$$



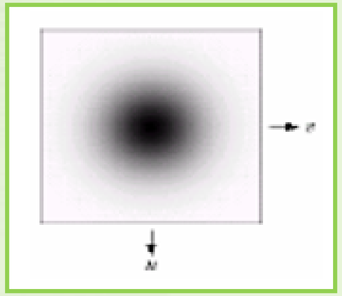
$$H(u,v) = \frac{1}{1 + \left[\frac{D(u,v)}{D_0}\right]^{2n}}$$

$$H(u,v) = \frac{1}{1 + \left[\frac{D_0}{D(u,v)}\right]^{2n}}$$



$$H(u,v) = e^{-\frac{D^2(u,v)}{2D_0^2}}$$

$$H(u,v) = 1 - e^{-\frac{D^2(u,v)}{2D_0^2}}$$



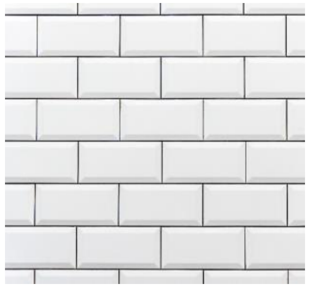
# Spatial Domain vs Frequency Domain

# Spatial Domain vs Frequency Domain

$$f = g \otimes h$$
$$F[x,y] = DFT(g \otimes h)$$
$$F[x,y] = \sum_{u=0}^{W-1}\sum_{v=0}^{H-1}\sum_{k,l} g[u-k, v-l]h[k,l]\,e^{-j\pi\left(\frac{ux}{W}+\frac{vy}{H}\right)}$$
$$= \sum_{u=0}^{W-1}\sum_{v=0}^{H-1}\sum_{k,l} g[u-k, v-l]\,e^{-j\pi\left(\frac{ux}{W}+\frac{vy}{H}\right)}h[k,l]$$
$$= \sum_{\mu=-k}^{W-k-1}\sum_{v=-i}^{H-l-1}\sum_{k,l} g[\mu,v]\,e^{-j\pi\left(\frac{(k+\mu)x}{W}+\frac{(l+v)y}{H}\right)}h[k,l]$$
$$= \sum_{k,l} G[x,y]e^{-j\pi\left(\frac{kx}{W}+\frac{ly}{H}\right)}h[k,l]$$
$$= G[x,y]\mathrm{H}[x,y]$$

# Non-local Mean Denoising

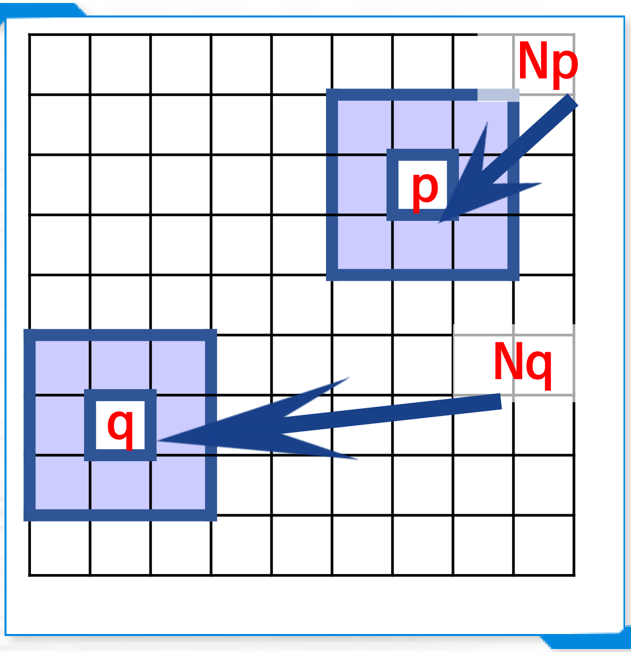
## Image self-similarity



출처 A. Buades, B. Coll, J.M. Morel “A non local algorithm for image denoising”, CVPR 2005

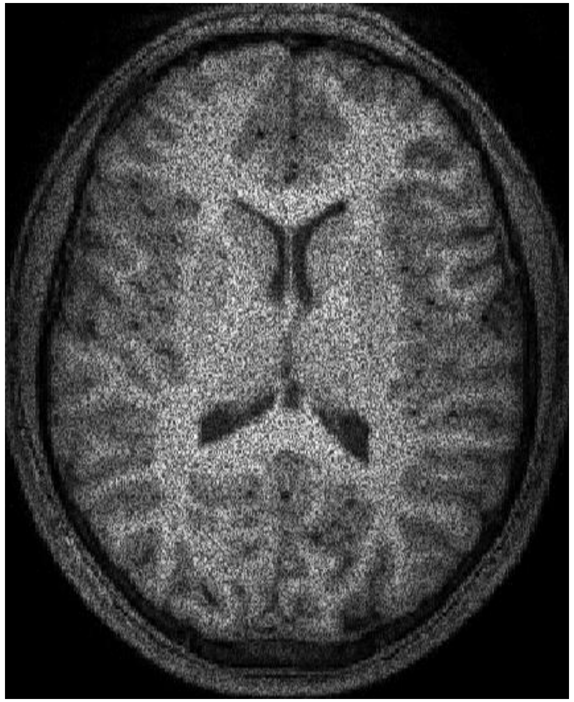
# Non-local Mean Denoising

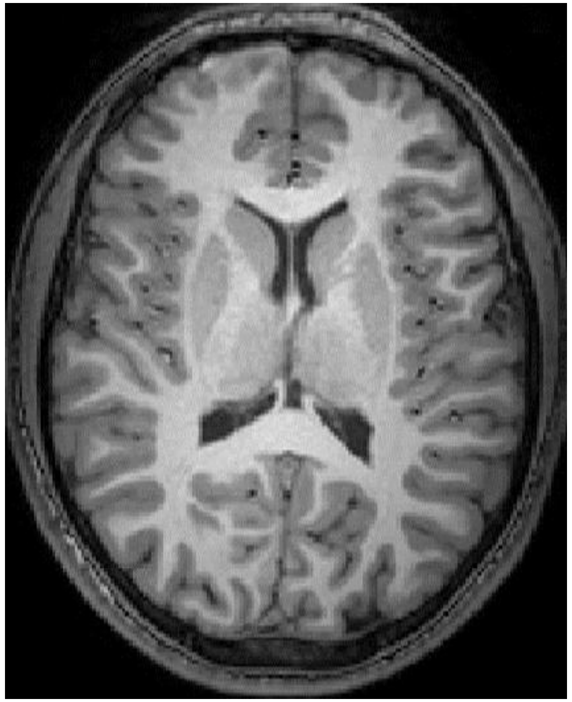
$$I_n(p) = \frac{1}{W} \sum_{q \in \emptyset} e^{-\frac{d(N_p, N_q)}{h^2}} I(\mathrm{q})$$



# Denoising via Dictionary

# Denoising via Dictionary

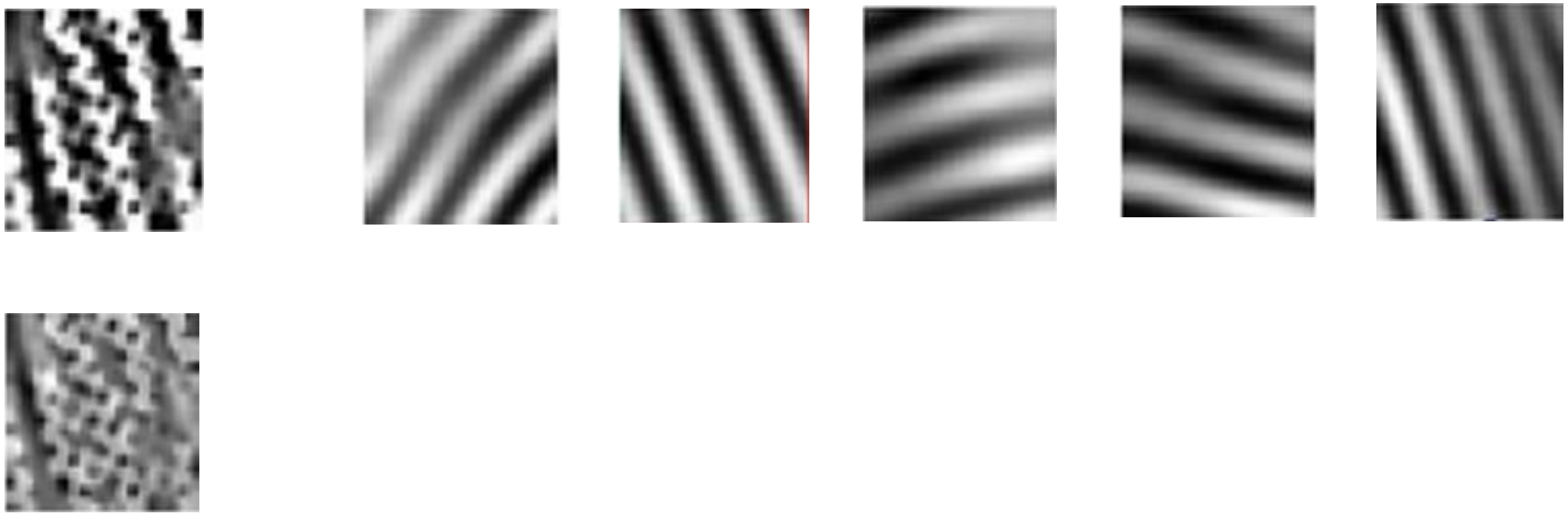




# Denoising via Dictionary

$$\min_{\alpha \in \mathbb{R}^m} \underbrace{\| y - D\alpha \|_2^2}_{\text{Data fitting term}} + \underbrace{\lambda\varphi(\alpha)}_{\text{Regularization term}}$$

# Denoising via Dictionary



# Dictionary Learning

$$\min_{\alpha \in \mathbb{R}^m} \| y - D\alpha \|_2^2 + \lambda \varphi(\alpha)$$

$$\min_{\alpha, D} \| y - D\alpha \|_2^2 + \lambda \varphi(\alpha)$$

# Dictionary Learning

- Initialize Dictionary
- Repeat

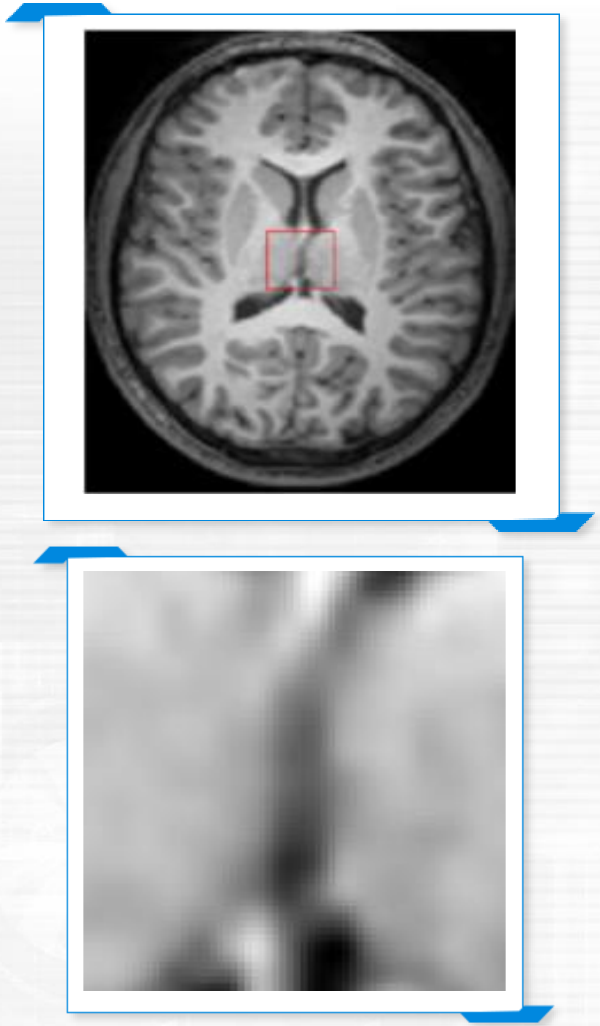
  Sparse Coding

  Codebook Update using K-SVD

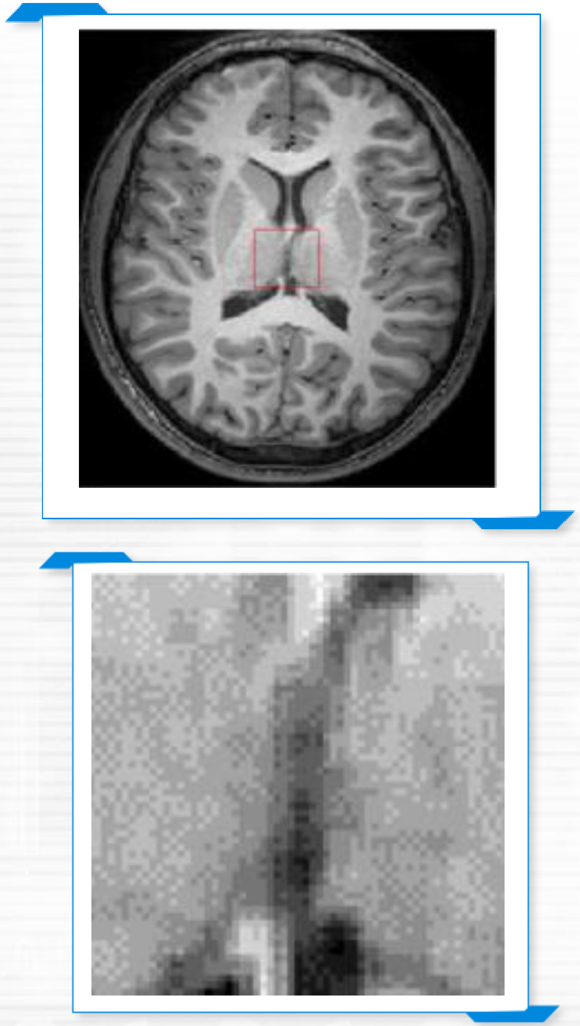
# K-SVD

# K-SVD

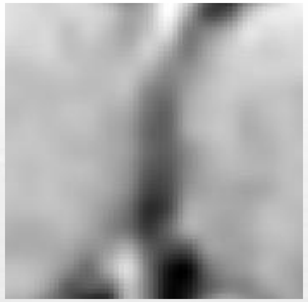
# Super-Resolution

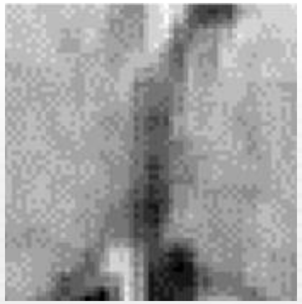


VS



# Super-Resolution





출처 "Image super-resolution via sparse representation", IEEE Transactions on Image Processing 2010

# Dictionary Training

$$D_h = \arg\min_{\{D_h, Z\}} \| X^h - D_h Z \|_2^2 + \lambda \| Z \|_1$$

$$D_l = \arg\min_{\{D_l, Z\}} \| Y^l - D_l Z \|_2^2 + \lambda \| Z \|_1$$

$$\min_{\{D_h, D_l, Z\}} \frac{1}{N} \| X^h - D_h Z \|_2^2 + \frac{1}{M} \| Y^l - D_l Z \|_2^2 + \lambda \left( \frac{1}{N} + \frac{1}{M} \right) \| Z \|_1$$

# Dictionary Training

$$\min_{\{D_h, D_l, Z\}} \| X_c - D_c Z \|_2^2 + \hat{\lambda} \| Z \|_1$$

$$X_c = \begin{bmatrix} \frac{1}{\sqrt{N}} X^h \\ \frac{1}{\sqrt{M}} Y^l \end{bmatrix}, \qquad D_c = \begin{bmatrix} \frac{1}{\sqrt{N}} D^h \\ \frac{1}{\sqrt{M}} D^l \end{bmatrix}$$