Lecture 8:

Recall:

Note:

(Spatial domain) $I * g$ (Linear filtering: Linear combination of neighborhood pixel values)

$$\downarrow \text{DFT}$$

(Frequency domain) $MN \underbrace{\hat{I} \odot}_{\text{pixel-wise multiplication}} \hat{g}$ (Modifying the Fourier coefficients by multiplication)

Observation:

1. When k and l are close to $N/2$, $\hat{F}\left(\underbrace{\frac{N}{2}+k}_{\approx N}, \underbrace{\frac{N}{2}+l}_{\approx N}\right)$ is associated to $e^{+j\frac{2\pi}{N}\left(\left(\frac{N}{2}+k\right)m+\left(\frac{N}{2}+l\right)n\right)}$

$$\approx e^{j\frac{2\pi}{N}(k'm+l'n)} \quad \text{where } (k',l') \approx (0,0)$$

$$= \cos\left(\frac{2\pi}{N}(k'm+l'n)\right) + i\sin\left(\frac{2\pi}{N}(k'm+l'n)\right)$$

Low-frequency if $(k,l) \approx (0,0)$

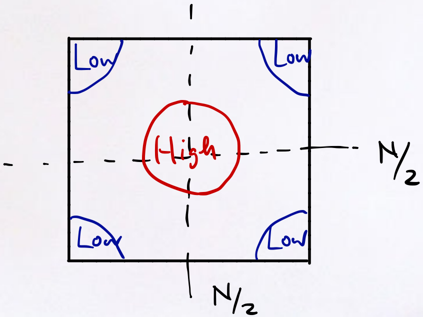
∴ Fourier coefficients at the bottom right are associated to low frequency components!

2. Similarly, we can check that Fourier coefficients at the 4 corners are associated to low frequency components.

3. Fourier coefficients in the middle are associated to high-frequency components

$$e^{j\frac{2\pi}{N}\left(\frac{N}{2}m+\frac{N}{2}n\right)}$$

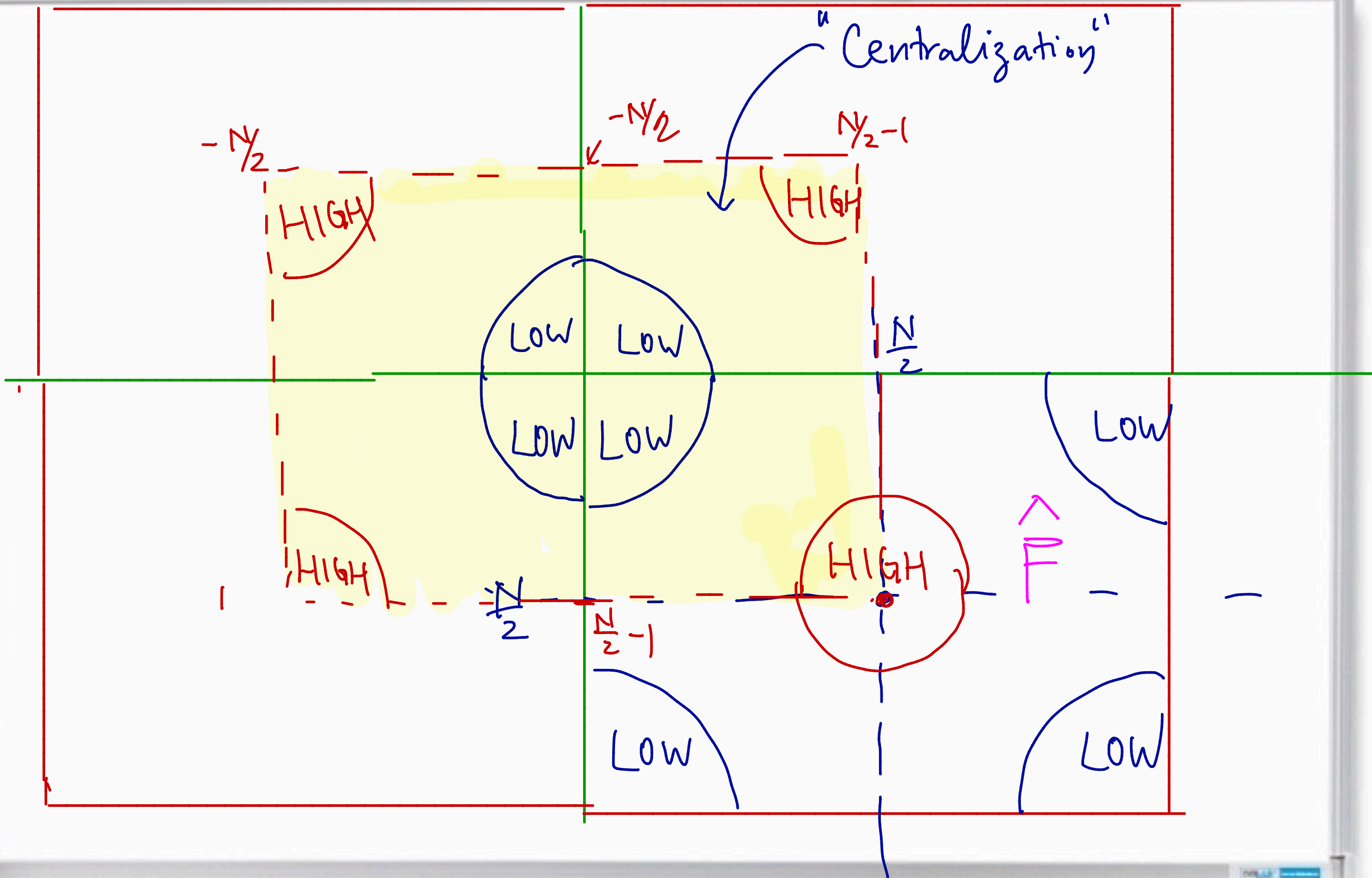
$$= e^{j\pi(m+n)} = (-1)^{m+n}$$



!!

∴ High-pass filtering
= Remove coefficients at 4 corners

Low-pass filtering
= Remove coefficients at the center


"Centralization"
-N/2
-N/2
N/2-1
HIGH
HIGH
LOW
LOW
LOW
LOW
N/2
HIGH
HIGH
N/2
N/2-1
LOW
LOW
LOW
LOW
F̂

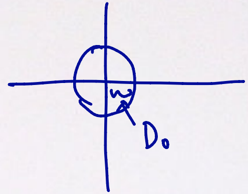
Example of Low-pass filters for image denoising

Assume that we work on the centered spectrum!

That is, consider $\hat{F}(u,v)$ where $-N/2 \le u \le N/2 - 1$, $-N/2 \le v \le N/2 - 1$.

1 Ideal low pass filter (ILPF):

$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) := u^2 + v^2 \le D_0^2 \\ 0 & \text{if } D(u,v) > D_0^2 \end{cases}$$

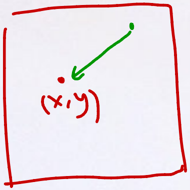


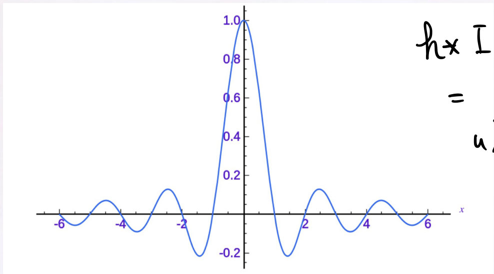
In 1-dim cross-section, $\boxed{\mathcal{F}^{-1}(\underset{h}{H(u,v)})}$ looks like:

$I \to \hat{I} \to \hat{I} \odot H$

$\downarrow$ iDFT

$\tilde{I}$





$$h * I(x,y) = \sum_{u,v} h(x-u, y-v)\, I(u,v)$$

every pixel values of I has an effect on $h * I(x,y)$!!

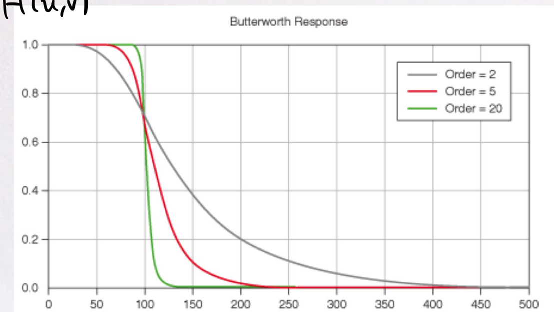
Good: Simple

Bad : Produce ringing effect!

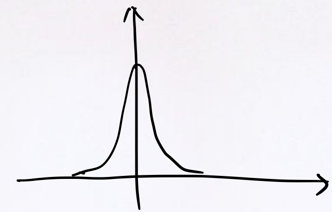
2. Butterworth low-pass filter (BLPF) of order n ($n \geq 1$ integer) :

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

$H(u,v)$ in 1-dim



$\mathcal{F}^{-1}(H(u,v))$ in 1-dim

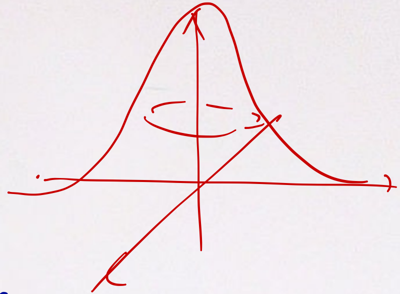


Good: Produce less / no visible ringing effect if n is carefully chosen!!

3. Gaussian low-pass filter

$$H(u,v) = \exp\left(-\frac{\overset{u^2+v^2}{D(u,v)}}{2\sigma^2}\right)$$

σ = spread of the Gaussian function



F.T. of Gaussian is also Gaussian!!

Good: No visible ringing effect!!

Examples for high-pass filtering for feature extraction

1. Ideal high-pass filter: (IHPF)

$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0^2 \\ 1 & \text{if } D(u,v) > D_0^2 \end{cases}$$

Bad: Produce ringing

2. Butterworth high-pass filter:

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

$(H(u,v) = 0 \text{ if } D(u,v) = 0)$

Choose the right n

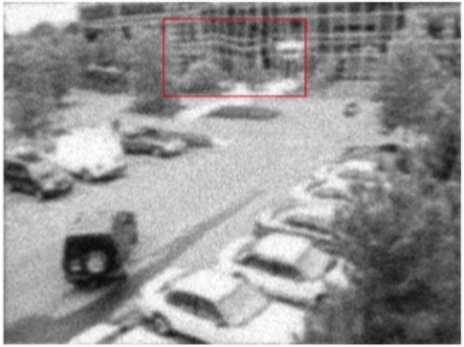
Good: Less ringing

3. Gaussian high-pass filter

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

Good: No visible ringing!

Image deblurring



Atmospheric turbulence

Motion Blur

Speeding problem

Image deblurring in the frequency domain:

Mathematical formulation of image blurring

Let g be the observed (blurry) image.

Let f be the original (good) image.

Model g as: $g = H(f) + n$

where H is the degradation function/operator and n is the additive noise.

Assumption on H:

1. H is position invariant:

   Let $g(x,y) = H(f)(x,y)$ and let $\tilde{f}(x,y) := f(x-\alpha, y-\beta)$.

   Then: $H(\tilde{f})(x,y) = g(x-\alpha, y-\beta)$

2. Linear: $H(f_1 + f_2) = H(f_1) + H(f_2)$

   $H(\alpha f) = \alpha H(f)$ where α is a scalar multiplication.

3. Linearity can be extended to integral:

$$H\left(\iint \alpha(u,v) f(x-u, y-v)\, du\, dv\right) = \iint \alpha(u,v) H(f)(x-u, y-v)\, du\, dv$$

With the above assumption, consider an impluse signal:

$$\delta(x,y) = \begin{cases} 1 & \text{if } (x,y) = (0,0) \\ 0 & \text{if } (x,y) \neq (0,0) \end{cases}$$

Then: $f(x,y) = f * \delta(x,y) = \displaystyle\sum_{\alpha=-M/2}^{M/2-1} \sum_{\beta=-N/2}^{N/2-1} f(\alpha,\beta)\,\delta(x-\alpha, y-\beta)$

$$\therefore\quad g(x,y) = H(f)(x,y)$$

$$= \sum_{\alpha=-M/2}^{M/2-1} \sum_{\beta=-N/2}^{N/2-1} f(\alpha,\beta)\, H(\delta)(x-\alpha, y-\beta) \quad \text{(by linearity and position-invariant)}$$

$$= \sum_{\alpha=-M/2}^{M/2-1} \sum_{\beta=-N/2}^{N/2-1} f(\alpha,\beta)\, h(x-\alpha, y-\beta) \quad \text{where} \quad h(x,y) = H(\delta)(x,y)$$

$$= f * h(x,y)$$

$\therefore$ With the above assumption,

Degradation / Blur = Convolution

Remark:

1. h is called the point spread function

2. $\therefore g(x,y) = h * f(x,y) + n(x,y)$

In the frequency domain,

$$G(u,v) = \underset{\substack{\uparrow \\ \text{constant}}}{c}\, H(u,v)\, F(u,v) + N(u,v)$$

$\therefore$ Deblurring can be done by:

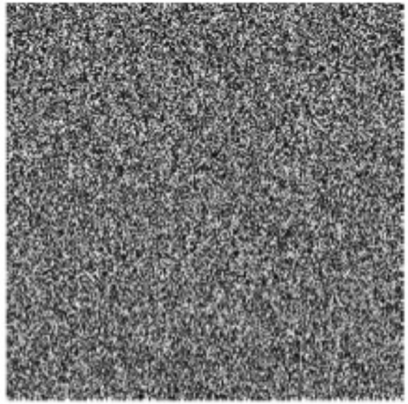
Compute: $F(u,v) \approx \dfrac{G(u,v)}{cH(u,v)}$ — from observed image / — from known degradation

Obtain: $f(x,y) = DFT^{-1}(F(u,v))$

(Does NOT work very well due to noise!)

Original

Blurred image



Direct inverse filtering