Lecture 8:

Image enhancement in the frequency domain:

Goal: 1. Remove $\underbrace{\text{high-frequency components}}_{\text{noise}}$ (low-pass filter) for image denoising.

2. Remove $\underbrace{\text{low-frequency components}}_{\text{non-edge}}$ (high-pass filter) for the extraction of image details.

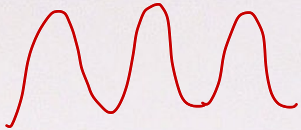
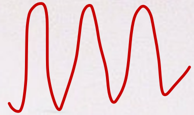
Let $\hat{F}$ be the DFT of an $N \times N$ image F. (indices taken from 0 to $N-1$)

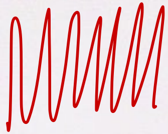
Then: for all $0 \leq m, n \leq N-1$,

$$F(m,n) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} \hat{F}(k,l)\, e^{j\frac{2\pi}{N}(km + ln)}$$

$\therefore$ $\hat{F}(k,l)$ is associated to the complex function $g(m,n) = e^{j\frac{2\pi}{N}(km+ln)}$

Goal: Remove "jumpy" components by setting suitable $\hat{F}(k,l)$ to zero.

 $= a$  $+ b$ 

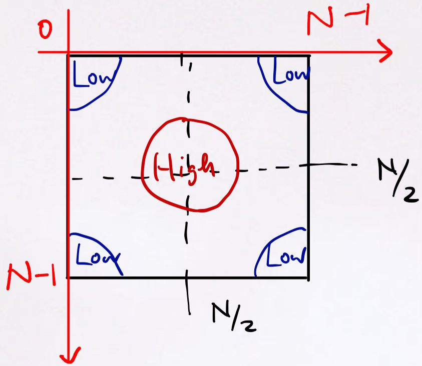
$+ c$ 

To remove noise, truncate c (let $c = 0$)

Observation:

If the image I takes indices between 0 to N-1,
then the DFT of I takes indices between 0 to N-1.

We have:



Properties of Fourier coefficients $\hat{F}$

Let F be a $N \times N$ image, N = even. Let $\hat{F}$ = DFT of F.

$$\therefore \quad \hat{F}(k, l) = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} F(m,n)\, e^{-j 2\pi \cdot \left(\frac{mk + nl}{N}\right)}$$

Fourier coefficients of F at (k, l)

Observe that: for $0 \le k, l \le \frac{N}{2} - 1$

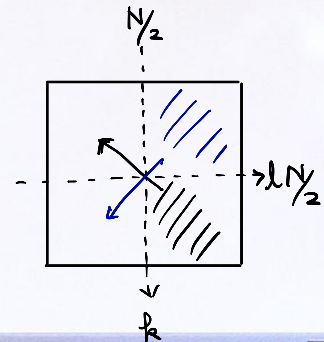
$$\hat{F}\left(\frac{N}{2} + k, \frac{N}{2} + l\right) = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} F(m,n)\, e^{-j\frac{2\pi}{N}\left(m\left(\frac{N}{2} + k\right) + n\left(\frac{N}{2} + l\right)\right)}$$

$$= \overline{\frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} F(m,n)\,(-1)^{m+n}\, e^{-j\frac{2\pi}{N}\left(m(-k) + n(-l)\right)}}$$

$$= \overline{\frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} F(m,n)\, e^{-j\frac{2\pi}{N}\left(m\left(\frac{N}{2} - k\right) + n\left(\frac{N}{2} - l\right)\right)}}$$

$$= \overline{\hat{F}\left(\frac{N}{2} - k, \frac{N}{2} - l\right)}$$

$\therefore$ Computing part of $\hat{F}$ can determine the rest !!

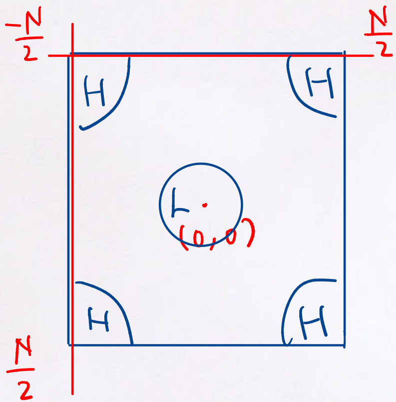


Centralisation:

Let F be an image whose indices are taken between $-\frac{N}{2}$ to $\frac{N}{2}$

Then, $DFT(F)$ is a matrix whose indices are also taken between $-\frac{N}{2}$ to $\frac{N}{2}$.

We have:



Procedures for image processing by modifying Fourier coefficients

Given an image $I = (I_{ij})_{-\frac{N}{2} \le i,j \le \frac{N}{2}}$.

Compute DFT of I (Denote $\hat{I} = \mathrm{DFT}(I)$)

Then: obtain a new DFT matrix, $\hat{I}^{new}$, by:

$$\hat{I}^{new} = H \odot \hat{I}$$

(Here $\underset{\substack{\uparrow \\ \text{pixel-wise multiplication}}}{H \odot \hat{I}}(u,v) = H(u,v)\hat{I}(u,v)$)

H is a suitable filter.

Finally, obtain an improved image by inverse DFT:

$$I^{new} = \underbrace{i\mathrm{DFT}}_{\text{inverse DFT}}(\hat{I}^{new})$$

Note: Let $h = \underbrace{iDFT}_{\text{inverse DFT}}(H) \iff H = DFT(h)$

$$I^{new}(x,y) = C\, h * I(x,y) = \sum_u \sum_v h(x-u, y-v)\, I(u,v)$$

$$\underbrace{H \odot \hat{I}}_{= \hat{I}^{new}} \xrightarrow{\text{inverse DFT}} C \; \underbrace{\boxed{h * I}}_{\ne}$$

C: normalizing constant

Let $H = DFT(h)$.

$$\hat{I}^{new} = H \odot \hat{I}$$

Then:

$$I^{new} = iDFT(\hat{I}^{new})$$

$$\boxed{I^{new} = Ch * I}$$

$$DFT(h * I) = K \underbrace{\boxed{\underbrace{DFT(h)}_{H} \odot \underbrace{DFT(I)}_{\hat{I}}}}_{\hat{I}^{new}}$$

$$DFT(h * I) = K \hat{I}^{new}$$

$$h * I = iDFT(K \hat{I}^{new})$$

$$h * I = K \, iDFT(\hat{I}^{new})$$

$$\Rightarrow \underbrace{iDFT(\hat{I}^{new})}_{I^{new}} = \frac{1}{K} h * I$$

$$I^{new} = \underbrace{\frac{1}{K}}_{C} h * I$$

Example of Low-pass filters for image denoising

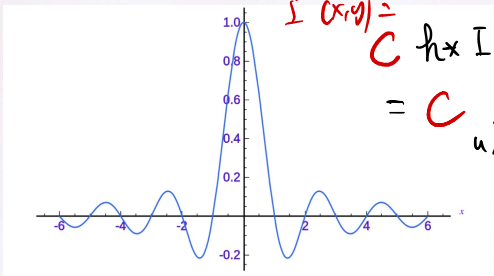
Assume that we work on the centered spectrum!

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, $iDFT(H(u,v))$ looks like:



$$I^{new}(x,y) = C\, h * I(x,y)$$

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

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

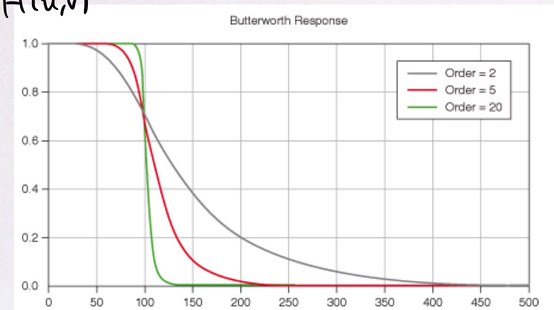
Good: Simple

Bad : Produce ringing effect!

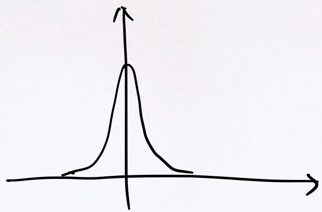
2. Butterworth low-pass filter (BLPF) of order n ($n \geq 1$ integer) :

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

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



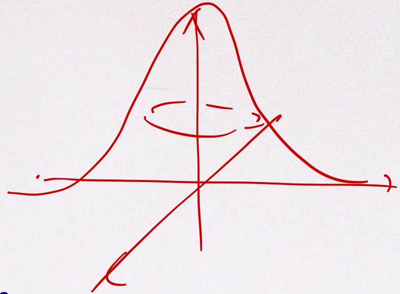
$\mathcal{F}^{-1}(H(u,v))$ in 1-dim



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:

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

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

Choose the right n

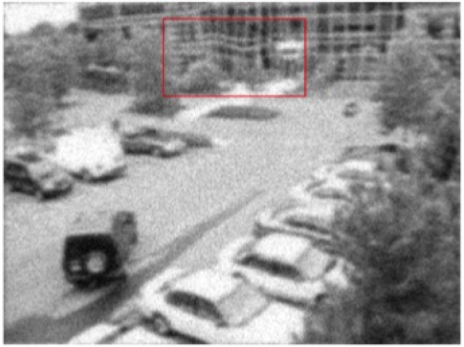
Good: Less ringing

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

$$f \xrightarrow{D} g$$

(not a matrix, just a transformation)

clean, original image

blurry image of f

Let g be the observed (blurry) image.

Let f be the original (good) image.

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

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

Assumption on D:

1. D is position invariant:

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

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

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

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

With the above assumption, consider an impluse image $\delta \in M_{(N+1)\times(N+1)}$ (indices taken between $-\frac{N}{2}$ to $\frac{N}{2}$)

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

Let $\tilde{\delta}_{\alpha,\beta}$ be the translated image of δ by (α,β):

$$\tilde{\delta}_{\alpha,\beta}(x,y) = \delta(x-\alpha, y-\beta) \quad \text{for} \quad -\frac{N}{2} \leq x,y \leq \frac{N}{2}$$

Note: $f(x,y) = f * \delta(x,y) = \sum_{\alpha=-\frac{N}{2}}^{N/2-1} \sum_{\beta=-N/2}^{N/2-1} f(\alpha,\beta)\,\delta(x-\alpha, y-\beta) = \sum_{\alpha=-\frac{N}{2}}^{\frac{N}{2}} \sum_{\beta=-\frac{N}{2}}^{\frac{N}{2}} f(\alpha,\beta)\,\tilde{\delta}_{\alpha,\beta}(x,y)$

for all $-\frac{N}{2} \leq x,y \leq \frac{N}{2}$.

$$\therefore \quad f = \sum_{\alpha=-\frac{N}{2}}^{\frac{N}{2}} \sum_{\beta=-\frac{N}{2}}^{\frac{N}{2}} \underset{\cap\ \mathbb{R}}{f(\alpha,\beta)}\ \underbrace{\tilde{\delta}_{\alpha,\beta}}_{M_{(N+1)\times(N+1)}}$$

Let g be the blurry image of f. That is, $g = D(f)$

$$g = D(f) = D\left(\sum_{\alpha=-\frac{N}{2}}^{\frac{N}{2}} \sum_{\beta=-\frac{N}{2}}^{\frac{N}{2}} f(\alpha,\beta)\,\tilde{\delta}_{\alpha,\beta}\right)$$

$$= \sum_{\alpha=-\frac{N}{2}}^{\frac{N}{2}} \sum_{\beta=-\frac{N}{2}}^{\frac{N}{2}} f(\alpha,\beta)\, D(\tilde{\delta}_{\alpha,\beta}) \quad \text{(linearity of } D\text{)}$$

$$\therefore\ g(x,y) = \sum_{\alpha=-\frac{N}{2}}^{\frac{N}{2}} \sum_{\beta=-\frac{N}{2}}^{\frac{N}{2}} f(\alpha,\beta)\, D(\tilde{\delta}_{\alpha,\beta})(x,y)$$

$$= \sum_{\alpha=-\frac{N}{2}}^{\frac{N}{2}} \sum_{\beta=-\frac{N}{2}}^{\frac{N}{2}} f(\alpha,\beta)\, D(\delta)(x-\alpha, y-\beta) \quad \text{(position-invariant)}$$

$$= f * h(x,y) \quad \text{where} \quad h = D(\delta)$$

$$\therefore\ g = f * h$$

$\therefore$ With the above assumption,

Degradation/Blur = Convolution