Lecture 10:

Recall:

$$I \xrightarrow{\text{DFT}} \hat{I} \xrightarrow{\text{Process } \hat{I} \text{ (by pixel-wise multiplication)}} \hat{I}^{new} \xrightarrow[\text{(inverse DFT)}]{\text{iDFT}} I^{new}$$

I: image

(Spatial Domain)

$\hat{I}$: Transformed image by DFT

(Frequency domain)

$\left(\hat{I}^{new} = \hat{I} \odot H \right)$

$\odot$: pixel-wise multiplication

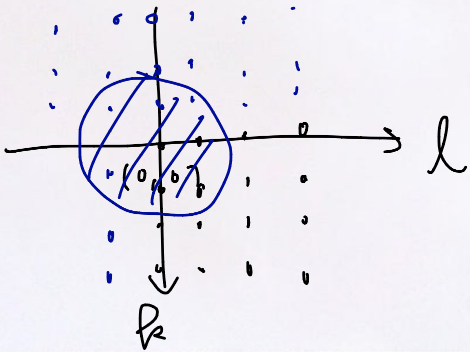
e.g. $\hat{I} \odot H = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \odot \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 2 & 3 \\ 4 & 5 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

Understanding convolution:

Recall: Discrete convolution:

$$\begin{array}{c} v(n,m) \\ \shortparallel \\ g * I(n,m) \end{array} = \underbrace{\sum_{n'=0}^{N-1} \sum_{m'=0}^{N-1} g(n-n', m-m')\, I(n',m')}_{\text{Linear combination of pixel values of } I}$$

In particular, if $g(k,l)$ is only non-zero around $(0,0)$, then, $g * I(n,m)$ is a linear combination of pixel value of I around (n,m)!!



<u>Example:</u> Suppose g looks like the following:

$$g = \begin{array}{cc} \left(\begin{array}{ccccc} & 1 & 2 & 3 & \\ 0 & 4 & 5 & 6 & 0 \\ & 7 & 8 & 9 & \end{array}\right) & \begin{array}{l} \leftarrow k=-1 \\ \leftarrow k=0 \\ \leftarrow k=1 \end{array} \\ \begin{array}{ccccc} & \uparrow & \uparrow & \uparrow & \\ & l=-1 & l=0 & l=1 & \end{array} & \end{array}$$

$$I * g(n,m) = \sum_{n'=0}^{N-1} \sum_{m'=0}^{N-1} I(n-n', m-m')\, g(n',m')$$

$$\text{Linear combination of neighborhood pixel values} \left\{ \begin{array}{l} = 1 \cdot I(n+1, m+1) + 2 \cdot I(n+1, m) + 3 \cdot I(n+1, m-1) \\ + 4 \cdot I(n, m+1) + 5 \cdot I(n, m) + 6 \cdot I(n, m-1) \\ + 7 \cdot I(n-1, m+1) + 8 \cdot I(n-1, m) + 9 \cdot I(n-1, m-1) \end{array} \right.$$

Note:

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

$$\Big\downarrow \text{DFT}$$

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

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.

High/Low frequency components of $\hat{F}$

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

$$\therefore \underset{\substack{\uparrow \\ \text{Fourier coefficients of } F \text{ at } (k,l)}}{\hat{F}(k,l)} = \frac{1}{N^2}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} F(m,n)\, e^{-j2\pi\cdot\left(\frac{mk+nl}{N}\right)}$$

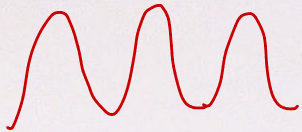
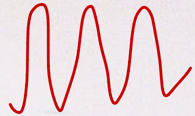
$$e^{-j\pi m}\, e^{-j\pi n}\; e^{-j\frac{2\pi}{N}(mk+nl)}$$

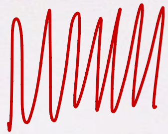
$$\parallel$$

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

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

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

 $= a$  $+ b$ 

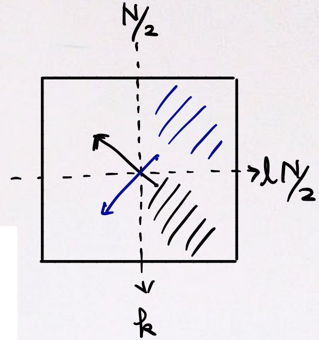
$+ c$ 

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

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

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

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



We have:

$$\begin{aligned}
F(m,n) = & \sum_{0\le k,l\le \frac{N}{2}-1} \left[\hat{F}\left(\frac{N}{2}+k, \frac{N}{2}+l\right) e^{j\frac{2\pi}{N}[\underline{(\frac{N}{2}+k)}m+\underline{(\frac{N}{2}+l)}n]}\right] \\
& + \sum_{1\le k,l\le \frac{N}{2}-1} \left[\overline{\hat{F}\left(\frac{N}{2}+k, \frac{N}{2}+l\right)} e^{j\frac{2\pi}{N}[\underline{(\frac{N}{2}-k)}m+\underline{(\frac{N}{2}-l)}n]}\right] \\
& + \sum_{0\le k,l\le \frac{N}{2}-1} \left[\hat{F}\left(\frac{N}{2}+k, \frac{N}{2}-l\right) e^{j\frac{2\pi}{N}[\underline{(\frac{N}{2}+k)}m+\underline{(\frac{N}{2}-l)}n]}\right] \\
& + \sum_{1\le k,l\le \frac{N}{2}-1} \left[\overline{\hat{F}\left(\frac{N}{2}+k, \frac{N}{2}-l\right)} e^{j\frac{2\pi}{N}[\underline{(\frac{N}{2}-k)}m+\underline{(\frac{N}{2}+l)}n]}\right] \\
& + \sum_{0\le l\le \frac{N}{2}-1} \hat{F}\left(0, \frac{N}{2}+l\right) e^{j\frac{2\pi}{N}[(\frac{N}{2}+l)n]} + \sum_{1\le l\le \frac{N}{2}-1} \overline{\hat{F}\left(0, \frac{N}{2}+l\right)} e^{j\frac{2\pi}{N}[(\frac{N}{2}-l)n]} \\
& + \sum_{0\le k\le \frac{N}{2}-1} \hat{F}\left(\frac{N}{2}+k, 0\right) e^{j\frac{2\pi}{N}[(\frac{N}{2}+k)m]} + \sum_{1\le k\le \frac{N}{2}-1} \overline{\hat{F}\left(\frac{N}{2}+k, 0\right)} e^{j\frac{2\pi}{N}[(\frac{N}{2}-k)m]} + \hat{F}(0,0)
\end{aligned}$$

Observation:

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

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

where $(k', l') \approx (0,0)$

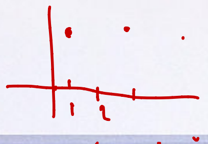
Low-frequency if $(k,l) \approx (0,0)$

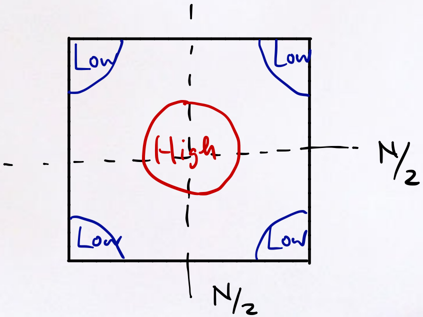
$\therefore$ Fourier coefficients at the bottom right are associated to low frequency components!

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

$$e^{j\frac{2\pi}{N}\left(\frac{N}{2}m+\frac{N}{2}n\right)} = e^{j\pi(m+n)} = (-1)^{m+n}$$





!!

$\therefore$ High-pass filtering $=$ Remove coefficients at 4 corners

Low-pass filtering $=$ Remove coefficients at the center

Centralisation:

Assume periodic conditions on F.

We can let $\tilde{F}(u,v) = \hat{F}(u-\frac{N}{2}, v-\frac{N}{2})$ where $0 \le u \le N-1$, $0 \le v \le N-1$

Then, High-frequency components are located at 4 corners of $\tilde{F}(u,v)$

Low-frequency components are located at center of $\tilde{F}(u,v)$

Mathematically,

Consider the discrete Fourier transform of $(-1)^{x+y}F(x,y)$:

$$e^{j2\pi\left(\frac{\frac{N}{2}x+\frac{N}{2}y}{N}\right)}$$

$$DFT(f(x,y)(-1)^{x+y})(u,v)$$

$$=\frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)e^{j\pi(x+y)}exp\left(-j2\pi\left(\frac{ux}{N}+\frac{vy}{N}\right)\right)$$

$$=\frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)exp\left(-j2\pi\left(\frac{(u-N/2)x}{\cancel{M}\ N}+\frac{(v-N/2)y}{N}\right)\right)$$

$$=\hat{F}\left(u-\frac{N}{2}, v-\frac{N}{2}\right) = \tilde{F}(u\cdot\ ,v\ .)$$

Therefore, to compute $\tilde{F}(u,v)$, we can compute DFT of $(-1)^{x+y}f(x,y)$.

Definition 3.5: A **low-pass filter (LPF)** (LPF) leaves low frequencies unchanged, while attenuating the high frequencies.

A **high-pass filter (HPF)** leaves high frequencies unchanged, while attenuating the low frequencies.

Basic steps of filtering in the frequency domain

1. Multiply $f(x,y)$ by $(-1)^{x+y}$.
2. Compute $\tilde{F}(u,v) = DFT(f(x,y)(-1)^{x+y})(u,v)$.
3. Multiply $\tilde{F}$ by a real "filter" function $H(u,v)$ to get

$$G(u,v) = H(u,v)\tilde{F}(u,v)$$

   (point-wise multiplication, but not matrix multiplication)

4. Compute inverse DFT of $G(u,v)$.
5. Take real part of the result in Step 4.
6. Multiply the result in Step 5 by $(-1)^{x+y}$.

Remark: 1. H is taken either to remove high-frequency coefficients/low-frequency coefficients.

2. $\mathcal{F}^{-1}(G(u,v)) = g(x,y) = \frac{1}{N^2}\cdot\mathcal{F}^{-1}(H(u,v)) * \mathcal{F}^{-1}(\tilde{F}(u,v)) = \frac{1}{N^2} h * \tilde{f}(x,y)$

Filtering in the frequency domain = Linear filtering in the spatial domain!

Example of Low-pass filters for image denoising

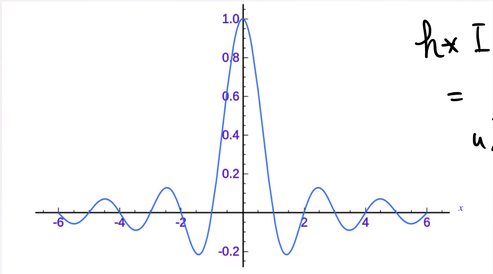
Assume that we work on the centered spectrum!

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, $\mathcal{F}^{-1}(H(u,v))$ looks like:



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

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

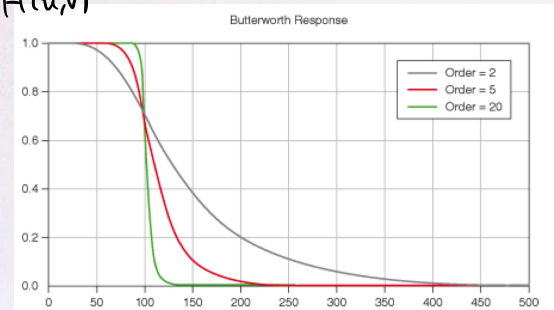
Good: Simple

Bad : Produce ringing effect!

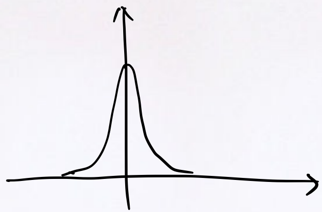
2. Butterworth low-pass filter (BLPF) of order n ($n \geq 1$ integer):

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

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



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

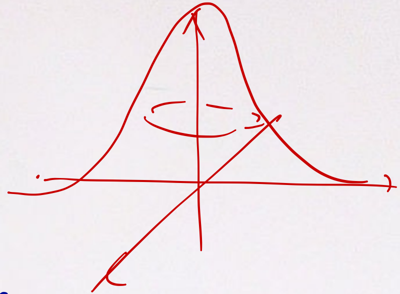


Good: Produce less / no visible ringing effect if n is carefully chosen!!

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!