Lecture 7:

Recall:

Discrete Fourier Transform:

Definition:

The 2D DFT of a $M \times N$ image $g = (g(k,l))_{k,l}$, where $0 \le k \le M-1$, $0 \le l \le N-1$ is defined as:

$$\hat{g}(m,n) = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} g(k,l)\, e^{-j2\pi\left(\frac{km}{M} + \frac{ln}{N}\right)}$$

(where $j = \sqrt{-1}$, $e^{j\theta} = \cos\theta + j\sin\theta$)

Remark: The inverse of DFT is given by:

$$g(p,q) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \hat{g}(m,n)\, e^{j2\pi\left(\frac{pm}{M} + \frac{qn}{N}\right)}$$

DFT of g

(no $\frac{1}{Mn}$!)

(no -ve sign)

Why is DFT useful in imaging:

1. DFT of convolution:

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

$\left(g, m \in M_{N \times M}(\mathbb{R}) \right)$

Then, the DFT of $g * w$ $= MN\, DFT(g) \odot DFT(w)$

$\therefore \quad g * w(n,m) = MN\, DFT(g)(n,m)\, DFT(w)(n,m)$

$\therefore$ DFT of convolution can be reduced to simple multiplication!

Remark: Conversely, if $x(n,m) = g(n,m)\,\omega(n,m)$

Then, $$\hat{x}(k,l) = \sum_{p=0}^{N-1} \sum_{q=0}^{M-1} \hat{g}(p,q)\,\hat{\omega}(k-p,\, l-q)$$ (Convolution of g and ω)

Note:

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

$\Big\downarrow$ DFT

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

pixel-wise multiplication

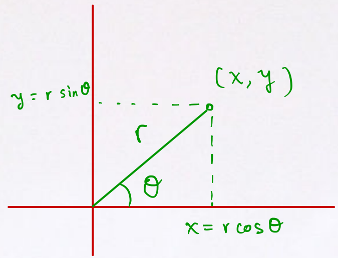
2. Average value of image

$$\text{Average value of } g = \bar{g} = \frac{1}{N^2}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} g(k,l) = \underbrace{\frac{1}{N^2}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} g(k,l)\, e^{-j2\pi(0)}}_{\hat{g}(0,0)}$$

3. DFT of a rotated image

Consider a $N \times N$ image g.

Then: $\hat{g}(m,n) = \frac{1}{N^2}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} g(k,l)\, e^{-j2\pi\left(\frac{km+ln}{N}\right)}$



Write k and l in polar coordinates:

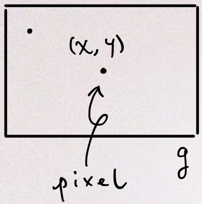
$k \equiv r\cos\theta$; $l = r\sin\theta$

Similarly, write $m \equiv \omega\cos\phi$; $n = \omega\sin\phi$.

Note that: $km + ln = r\omega(\cos\theta\cos\phi + \sin\theta\sin\phi) = r\omega\cos(\theta - \phi)$.

Denote $\mathcal{P}(g) = \{(r,\theta) : (r\cos\theta, r\sin\theta) \text{ is a pixel of } g\}$

(Polar coordinate set of g)



If $\begin{cases} x = r\cos\theta \\ y = r\sin\theta \end{cases}$, then $(r, \theta) \in \mathcal{P}(g)$.

Then: $\underbrace{\hat{g}(m, n) = \hat{g}(\omega, \phi)}_{\text{Identify } \hat{g}(m,n) \text{ with } \hat{g}(\omega,\phi)} = \dfrac{1}{N^2} \displaystyle\sum_{k=0}^{N-1} \sum_{l=0}^{N-1} \underbrace{g(r, \theta)}_{\text{Identify } g(k,l) \text{ with } g(r,\theta)} e^{-j2\pi\left(\frac{r\omega\cos(\theta-\phi)}{N}\right)}$

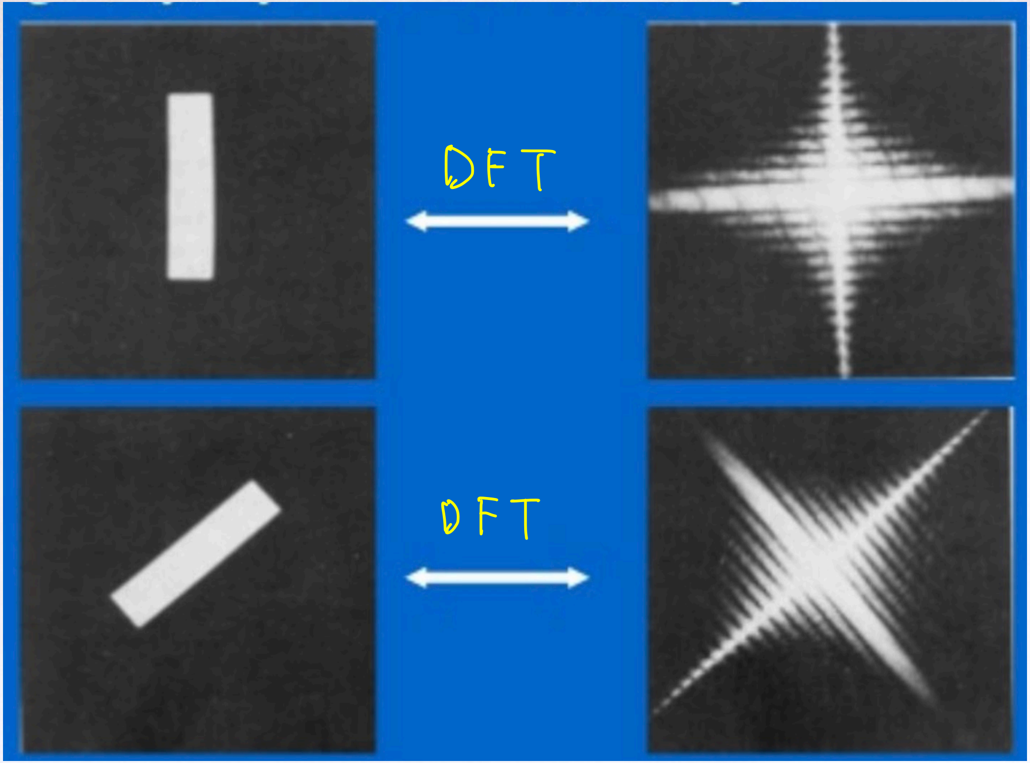
Consider a rotated image $\tilde{g}(r, \theta) = g(r, \theta + \theta_0)$ where θ is defined between $-\theta_0$ to $\pi/2 - \theta_0$.

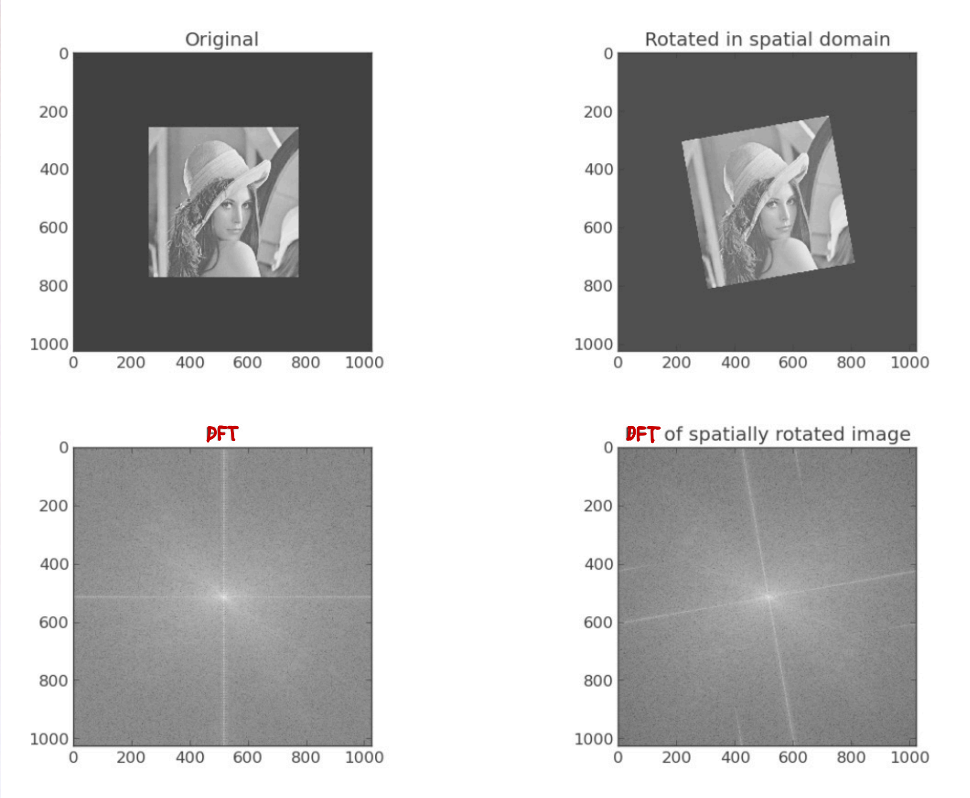
$\therefore$ image g is rotated clockwisely by θ_0.

DFT of $\tilde{g}$ is:

$$\hat{\tilde{g}}(\omega, \phi) = \frac{1}{N^2} \sum_{(r,\theta)\in\mathcal{P}(\tilde{g})} \underbrace{\tilde{g}(r, \theta)}_{= g(r,\, \underbrace{\theta+\theta_0}_{\tilde{\theta}})} e^{-j2\pi\left(\frac{r\omega\cos(\theta-\phi)}{N}\right)} = \frac{1}{N^2} \sum_{(r,\tilde{\theta})\in\mathcal{P}(g)} g(r, \tilde{\theta})\, e^{-j2\pi\left(\frac{r\omega\cos(\tilde{\theta}-\theta_0-\phi)}{N}\right)}$$

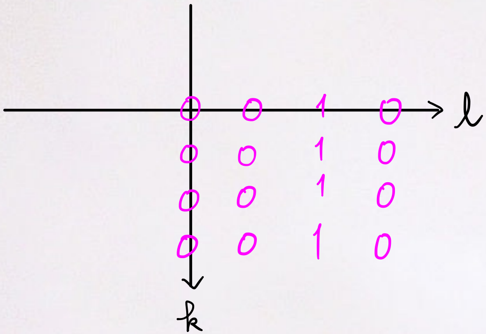
$\therefore\ \hat{\tilde{g}}(\omega, \phi) = \hat{g}(\omega, \phi + \theta_0)$. ($\phi$ is also defined between $-\theta_0$ to $\pi/2 - \theta_0$)


DFT
DFT

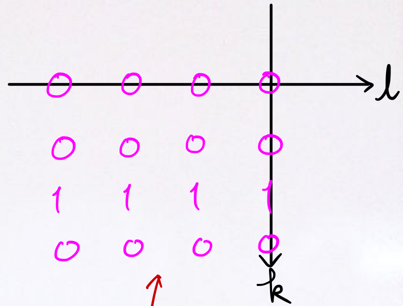

Original
Rotated in spatial domain
DFT
DFT of spatially rotated image
0
200
400
600
800
1000

Example: Let $g = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$. Then: $\hat{g} = \begin{pmatrix} \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$

Note that g in the coordinate system:



Rotated by 90° clockwisely



$\tilde{g}$

Note that indices of $\tilde{g}$ are taken as: $\begin{cases} -3 \leq \ell \leq 0 \\ 0 \leq k \leq 3 \end{cases}$.

Now, DFT of $\tilde{g} = \hat{\tilde{g}}$ (given by: $\displaystyle\sum_{k=0}^{3}\sum_{l=-3}^{0} \tilde{g}(k,l)\, e^{-j2\pi\left(\frac{km+ln}{4}\right)}$)

$$
= \begin{array}{cc}
\begin{pmatrix} 0 & 0 & 0 & 1/4 \\ 0 & 0 & 0 & -1/4 \\ 0 & 0 & 0 & 1/4 \\ 0 & 0 & 0 & -1/4 \end{pmatrix} & \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} \downarrow k \\
l \rightarrow \begin{array}{cccc} -3 & -2 & -1 & 0 \end{array} &
\end{array}
$$

$\therefore$ $0 \le k \le 3$
$-3 \le l \le 0$

4. DFT of a shifted image

Let $g = (g(k', l'))$ be a $N \times N$ image, where the indices are taken as:

$-k_0 \le k' \le N-1-k_0$ and $-l_0 \le l' \le N-1-l_0$

Let $\tilde{g}$ be shifted image of g defined as:

$\tilde{g}(k,l) = g(k-k_0, l-l_0)$ where $0 \le k \le N-1$
$0 \le l \le N-1$

Then: $$\hat{\tilde{g}}(m,n) = \frac{1}{N^2}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} g(k-k_0, l-l_0)\, e^{-j2\pi\left(\frac{km+ln}{N}\right)}$$

$$= \underbrace{\frac{1}{N^2}\sum_{k'=-k_0}^{N-1-k_0}\sum_{l'=-l_0}^{N-1-l_0} g(k', l')\, e^{-j2\pi\left(\frac{k'm+l'n}{N}\right)}}_{\hat{g}(m,n)}\, e^{-j2\pi\left(\frac{k_0m+l_0n}{N}\right)}$$

$$\therefore \quad \hat{\hat{g}}(m,n) = \hat{g}(m,n)\, e^{-j2\pi\left(\frac{k_0 m + l_0 n}{N}\right)}$$

<u>Remark:</u> $\hat{g}(m-m_0, n-n_0) = \mathrm{DFT}(\tilde{g})$ with carefully chosen indices!

where $\tilde{g}(k,l) = g(k,l) \times e^{j2\pi\left(\frac{m_0 k + n_0 l}{N}\right)}$

Note:

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

$\big\downarrow$ 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.

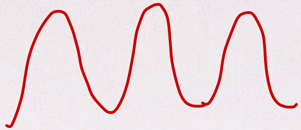
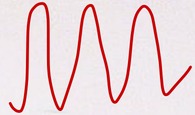
Let $\hat{F}$ be the DFT of an $N \times N$ image F. (indices taken from 0 to $N-1$)

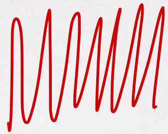
Then: for all $0 \leq m, n \leq N-1$,

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

$\therefore$ $\hat{F}(k,l)$ 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:

1. When k and l are close to 0, $\hat{F}(k,l)$ is associated to $g(m,n) = e^{j\frac{2\pi}{N}(km+ln)}$

$$\approx e^{j\frac{2\pi}{N}(0m+0n)}$$

$$\approx 1 \text{ (constant)}$$

∴ Fourier coefficients at the bottom left are associated to low frequency components! (Not "jumpy")

2. When k and l are close to N, $\hat{F}(k,l)$ is associated to

$$g(m,n) = e^{j\frac{2\pi}{N}(km+ln)} \approx e^{j\frac{2\pi}{N}(Nm+Nn)} = e^{j2\pi(m+n)} \approx 1 \quad \text{(Not "jumpy")}$$

∴ Fourier coefficients at the bottom riget 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:

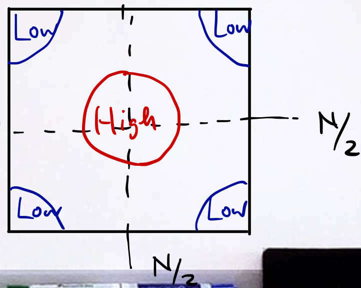
When k and l are close to $N/2$

$\hat{F}(k,l)$ is associated to:

$$g(m,n) = e^{j\frac{2\pi}{N}(km+ln)} \approx e^{j\frac{2\pi}{N}(\frac{N}{2}m+\frac{N}{2}n)}$$

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

(most "jumpy")



∴ High-pass filtering = Remove coefficients at 4 corners

Low-pass filtering = Remove coefficients at the center

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 $\frac{N}{2}$.

In this case, Fourier coefficients located at 4 corners of DFT(F) are associated to high-frequency components (jumpy)

Fourier coefficients located in the middle of DFT(F) are associated to low-frequency components (less jumpy)

Procedures for image processing by modifying Fourier coefficients

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

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

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

$$\hat{I}^{new} = H \odot \hat{I} \quad \left(\text{Here } H \odot \hat{I}(u,v) = H(u,v)\hat{I}(u,v) \right)$$

$\odot$: pixel-wise multiplication

H is a suitable filter.

Finally, obtain an improved image by inverse DFT:

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

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

$$H \odot \hat{I} \xrightarrow{\text{inverse DFT}} C \, h * I$$

normalizing constant

Example of Low-pass filters for image denoising

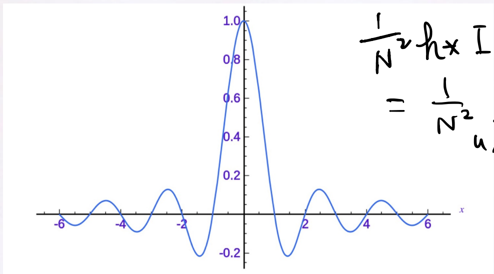
Assume that we work on the centered spectrum!

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:



$$\frac{1}{N^2} h * I(x,y) = \frac{1}{N^2} \sum_{u,v} h(x-u, y-v)\, I(u,v)$$

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

Good: Simple

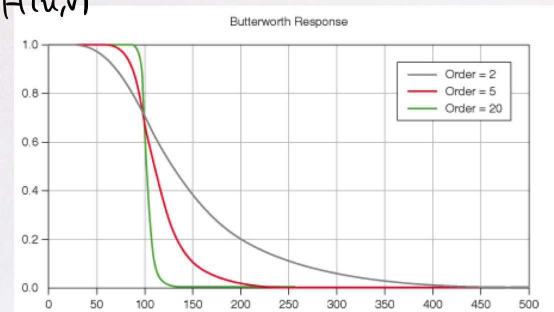
Bad : Produce ringing effect!

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

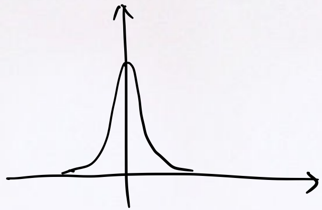
$$H(u,v) = \frac{1}{1 + \left(D(u,v)/D_0\right)^n}$$

$$D(u,v) = u^2 + v^2$$

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



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



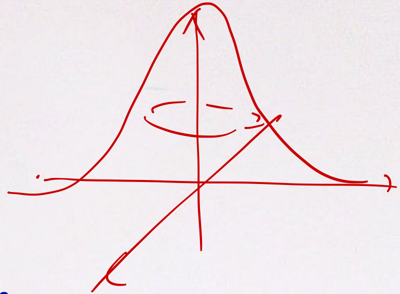
Good: Produce less / no visible ringing effect if n is carefully chosen!!

3. Gaussian low-pass filter

$u^2 + v^2$

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

σ = spread of the Gaussian function



F.T. of Gaussian is also Gaussian!!

Good: No visible ringing effect!!