Lecture 8:

Recall: Discrete Fourier Transform:

Definition: The 1D discrete Fourier Transform (DFT) of a function $f(k)$, defined at discrete points $k = 0, 1, 2, \ldots, N-1$ is defined as:

$$\hat{f}(m) = \frac{1}{N} \sum_{k=0}^{N-1} f(k)\, e^{-j \frac{2\pi m k}{N}}$$

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

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:

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

Remark: The inverse of DFT is given by:

$$g(p,q) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \hat{g}(m,n)\, e^{j 2\pi \left(\frac{pm}{M} + \frac{qn}{N} \right)}$$

($\hat{g}$: DFT of g)

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

(no -ve sign)

Why is DFT useful in imaging:

1. DFT of convolution:

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

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

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

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

Recall: Shift-invariant image transformation = 2D convolution.

$\therefore$ Easy computation/manipulation of shift-invariant transf. after DFT!!

2. Average value of image

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 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:

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

4. 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:

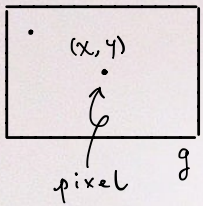
$k \equiv r\cos\theta$; $l \equiv r\sin\theta$

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

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

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

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:

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

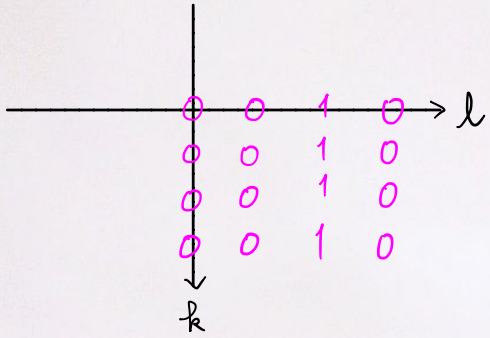
$\tilde{\theta}$

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

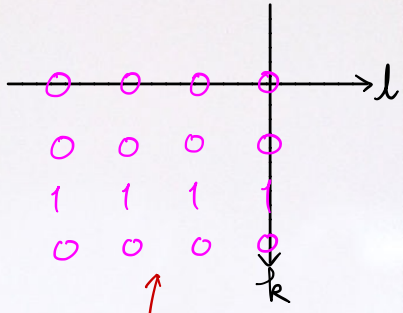
DFT of an image rotated by θ_0 = DFT of the original image and then rotated by θ_0.

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: $\begin{cases} -3 \le l \le 0 \\ 0 \le k \le 3 \end{cases}$.

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

$$= \begin{pmatrix} 0 & 0 & 0 & 1/4 \\ 0 & 0 & 0 & -1/4 \\ 0 & 0 & 0 & 1/4 \\ 0 & 0 & 0 & -1/4 \end{pmatrix} \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} \quad k$$

l: -3 -2 -1 0

$\therefore \quad 0 \le k \le 3$

$-3 \le l \le 0$

How to compute DFT fast?

Goal: Convert image I to $\hat{I}$ (Fast?) $\xrightarrow{\text{DFT}}$ Manipulate / adjust $\hat{I}$ (Fourier coefficients) to get a new $\hat{I}^{new}$

$\downarrow$

Convert $\hat{I}^{new}$ into the spatial domain (Fast?)

Fast Fourier Transform

Recall: DFT is separable $\Rightarrow$ 2D DFT = Two 1D DFT!

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

1D DFT

1D DFT

Suffices to consider how to compute 1D DFT fast!!

1D DFT is: $\hat{f}(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)\, \underbrace{e^{-j2\pi(\frac{ux}{N})}}_{\omega_N^{ux}}$ where $\omega_N = e^{-j\frac{2\pi}{N}}$

Assume $N = 2^n = 2M$ $(\therefore M = 2^{n-1})$.

Then: $\hat{f}(u) = \frac{1}{2M}\sum_{x=0}^{2M-1} f(x)\, \omega_{2M}^{ux}$

Separate the summation into odd and even parts:

$$\hat{f}(u) = \frac{1}{2}\left\{\frac{1}{M}\sum_{y=0}^{M-1} f(2y)\, \underbrace{\omega_{2M}^{u(2y)}}_{\omega_M^{uy}} + \frac{1}{M}\sum_{y=0}^{M-1} f(2y+1)\, \underbrace{\omega_{2M}^{u(2y+1)}}_{\omega_{2M}^{u(2y)}\,\omega_{2M}^{u}\ (\omega_M^{uy})}\right\}$$

Let $f_{even} = (f(0), f(2), \ldots, f(2M-2))^T$ — even part of f

$f_{odd} = (f(1), f(3), \ldots, f(2M-1))^T$ — odd part of f

Then: $\hat{f}(u) = \frac{1}{2}\left\{\hat{f}_{even}(u) + \hat{f}_{odd}(u)\, \omega_{2M}^{u}\right\}$ for $u = 0, 1, 2, \ldots, M-1$

($\hat{f}_{even}(u)$, $\hat{f}_{odd}(u)$: only defined for $u = 0, 1, 2, \ldots, M-1$)

For $u \geq M$, consider: $\hat{f}(u+M) = \frac{1}{2}\left\{\frac{1}{M}\sum_{y=0}^{x-1} f(2y)\, \omega_M^{uy+My} + \frac{1}{M}\sum_{y=0}^{M-1} f(2y+1)\, \omega_M^{uy+My}\, \omega_{2M}^{u+M}\right\}$

for $u = 0, 1, 2, \ldots, M-1$

$\omega_M^{uy+My} = \omega_M^{uy}$; $\omega_{2M}^{u+M} = -\omega_{2M}^{u}$

$\therefore \quad \hat{f}(u+M) = \frac{1}{2}\left\{ \hat{f}_{even}(u) - \hat{f}_{odd}(u)\, \omega_{2M}^{u} \right\}$ for $u = 0, 1, 2, \ldots, M-1$

FFT algorithm: Let $N = 2^n$ and $f \in \mathbb{R}^N$

Step 1: Split f into: $f_{even} = [f(0), f(2), \ldots, f(2M-2)]^T$

$f_{odd} = [f(1), f(3), \ldots, f(2M-1)]^T$.

Step 2: Compute $\hat{f}_{even} = F_M f_{even}$ and $\hat{f}_{odd} = F_M f_{odd}$

$M = 2^{n-1}$; F_M DFT matrix

$F_M = \left(\omega_M^{ux} \right)_{0 \le u, x \le M-1}$ = $M \times M$ matrix!

Step 3: For $u = 0, 1, 2 \ldots, M-1$, compute

$$\hat{f}(u) = \frac{1}{2}\left[\hat{f}_{even}(u) + \hat{f}_{odd}(u)\, \omega_{2M}^{u} \right]$$

$$\hat{f}(u+M) = \frac{1}{2}\left[\hat{f}_{even}(u) - \hat{f}_{odd}(u)\, \omega_{2M}^{u} \right]$$

$\therefore$ Reduce the matrix multiplication by $\frac{1}{2}$!

For step 2, we can apply the splitting idea again to compute $\hat{f}_{even}$ and $\hat{f}_{odd}$!

Computational cost of FFT:

Let C_m be the computational cost of $F_M \vec{x}$. Then: $C_1 = 1$!!

Clearly, $C_N = 2C_M + 3M$ (2 matrix multiplication by F_M, M multiplication, M additions and M subtractions)

$\therefore \quad C_{2^n} = 2C_{2^{n-1}} + 3\overset{\shortparallel}{M}^{2^{n-1}} \Rightarrow 2^{-n} C_{2^n} = 2^{-(n-1)} C_{2^{n-1}} + \frac{3}{2}$.

$= 2^{-(n-2)} C_{2^{n-2}} + 2\left(\frac{3}{2}\right)$

$= \vdots$

$= C_1 + n\left(\frac{3}{2}\right)$

$\because C_1 = 1$

$\therefore C_{2^n} = 2^n + n2^n\left(\frac{3}{2}\right)$

We conclude that the computational cost C_N is bounded by $KN(\log_2 N)$ (or $\mathcal{O}(N \log_2 N)$)

e.g. If $N = 2^{10}$, then $N^2 = 2^{20}$ (Computational cost for conventional matrix multiplication)

For FFT, $N \log_2 N = 2^{10} \cdot 10 < 2^{14} \Rightarrow 2^6$ times faster !!