Review:

1D DFT: $\hat{f}(m) = \frac{1}{N} \sum_{k=0}^{N-1} f(k)\, e^{-j\frac{2\pi mk}{N}}$ $\quad (j = \sqrt{-1},\ e^{j\theta} = \cos\theta + i\sin\theta)$

2D DFT: given an $M \times N$ image g,

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

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

DFT of convolution:

given $M \times N$ images g and w,

$\widehat{g * w}(p,q) = MN\, \hat{g}(p,q)\, \hat{w}(p,q)$,

i.e., $\widehat{g * w} = MN\, \hat{g} \odot \hat{w}$,

where $\odot$ denote pointwise product of two images.

$\big(A \odot B\,(p,q) = A(p,q)\, B(p,q)$ $\quad$ matlab $.\times\big)$

Do we have a converse direction?

The answer is yes.

Prop: Let $g, w \in M_{M\times N}(\mathbb{R})$. Then $\widehat{g \odot w} = \hat{g} * \hat{w}$.

Proof: $$\widehat{g \odot w}(m,n) = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{\ell=0}^{N-1} g(k,\ell)\, w(k,\ell)\, e^{-2\pi j\left(\frac{km}{M} + \frac{\ell n}{N}\right)}$$

$$\hat{g} * \hat{w}(m,n) = \sum_{k=0}^{M-1} \sum_{\ell=0}^{N-1} \hat{g}(k,\ell)\, \hat{w}(m-k, n-\ell)$$

$$= \sum_{k=0}^{M-1}\sum_{l=0}^{N-1}\left(\frac{1}{MN}\sum_{p=0}^{M-1}\sum_{q=0}^{N-1} g(p,q)\, e^{-2\pi j\left(\frac{pk}{M}+\frac{ql}{N}\right)}\right)\left(\frac{1}{MN}\sum_{s=0}^{M-1}\sum_{t=0}^{N-1} w(s,t)\, e^{-2\pi j\left(\frac{s(m-k)}{M}+\frac{t(n-l)}{N}\right)}\right)$$

$$= \frac{1}{(MN)^2}\sum_{p=0}^{M-1}\sum_{q=0}^{N-1} g(p,q)\sum_{s=0}^{M-1}\sum_{t=0}^{N-1} w(s,t)\underbrace{\sum_{k=0}^{M-1} e^{-2\pi j\left(\frac{k(p-s)}{M}+\frac{sm}{M}\right)}}_{(*)}\underbrace{\sum_{l=0}^{N-1} e^{-2\pi j\left(\frac{l(q-t)}{N}+\frac{tn}{N}\right)}}_{(**)}$$

Fix m, n, p, q, s, t, then

$$(*) = \sum_{k=0}^{M-1} e^{-2\pi j\frac{k(p-s)}{M}}\, e^{-2\pi j\frac{sm}{M}} = e^{-2\pi j\frac{sm}{M}}\sum_{k=0}^{M-1} e^{-2\pi j\frac{k(p-s)}{M}}$$

$$= \begin{cases} e^{-2\pi j\frac{sm}{M}}\cdot M & \text{if } p=s \\ e^{-2\pi j\frac{sm}{M}}\dfrac{1-\left(e^{-2\pi j\frac{p-s}{M}}\right)^M}{1-e^{-2\pi j\frac{p-s}{M}}} = 0 & \text{if } p\neq s \end{cases} = M e^{-2\pi j\frac{sm}{M}}\mathbb{1}_{\{0\}}(p-s)$$

Similarly, $(**) = N e^{-2\pi j\frac{tn}{N}}\mathbb{1}_{\{0\}}(q-t)$

$$= \frac{1}{(MN)^2}\sum_{p=0}^{M-1}\sum_{q=0}^{N-1} g(p,q)\sum_{s=0}^{M-1}\sum_{t=0}^{N-1} w(s,t)\, M e^{-2\pi j\frac{sm}{M}}\mathbb{1}_{\{0\}}(p-s)\, N e^{-2\pi j\frac{tn}{N}}\mathbb{1}_{\{0\}}(q-t)$$

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

$$= \widehat{g\odot w}(m,n)$$

Fast Fourier Transform

1D: $\hat{f}(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x)\, e^{-2\pi j \frac{ux}{N}}$

Assume $N = 2^n = 2M$

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

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

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

How about the 2D case?

key idea: DFT is separable.

Let I be a $N_1 \times N_2$ image.

Suppose $N_1 = 2^{n_1}$, $N_2 = 2^{n_2}$, $N = N_1 \times N_2$.

Denote $I = \begin{pmatrix} - & I_1 & - \\ - & I_2 & - \\ & \vdots & \\ - & I_{n_1} & - \end{pmatrix}$

$\hat{I}(m,n) = \frac{1}{N} \sum_{k=0}^{N_1-1} \sum_{l=0}^{N_2-1} I(k,l)\, e^{-2\pi j \left(\frac{km}{N_1} + \frac{ln}{N_2} \right)}$

$= \frac{1}{N_1} \sum_{k=0}^{N_1-1} \left(\frac{1}{N_2} \sum_{l=0}^{N_2-1} I(k,l)\, e^{-2\pi j \frac{ln}{N_2}} \right) e^{-2\pi j \frac{km}{N_1}}$

Fix n and k, this is $\hat{I}_k(n)$.

We first should compute $\hat{I}_k$, for $k = 1, 2, \cdots, N_1$

Cost is $O(N_1 N_2 \log_2 N_2)$

Obtain $\begin{pmatrix} - & \hat{I}_1 & - \\ - & \hat{I}_2 & - \\ & \vdots & \\ - & \hat{I}_{n_1} & - \end{pmatrix}$

Then, $\hat{I}(m, n) = \frac{1}{N_1} \sum_{k=0}^{N_1 - 1} \hat{I}_k(n) \, e^{-2\pi j \frac{km}{N_1}}$

This is m-th DFT of $\begin{array}{c} \begin{array}{ccc} & \phantom{\hat{I}_1} & n \end{array} \\ \begin{pmatrix} - & \hat{I}_1 & - \\ - & \hat{I}_2 & - \\ & \vdots & \\ - & \hat{I}_{n_1} & - \end{pmatrix} \end{array}$

We need to compute another N_2 1D DFT

Cost is $O(N_2 N_1 \log N_1)$.

Total cost $O(N_1 N_2 \log N_2) + O(N_2 N_1 \log N_1)$

$$= O\Big(N(\log N_1 + \log N_2)\Big)$$

$$= O(N \log N)$$

Fast computation of convolution using FFT

Given two $N_1 \times N_2$ images g and w.

2^{n_1} 2^{n_2} $N = N_1 \times N_2$,

Want to compute $g * w$.

Direct computation:

$$g * w\,(m, n) = \sum_{k=0}^{N_1-1} \sum_{\ell=0}^{N_2-1} g(k,\ell)\, w(m-k, n-\ell)$$

Cost: $O(N \cdot N) = O(N^2)$.

Using FFT: $\widehat{g * w} = \hat{g} \odot \hat{w}$

- compute $\hat{g}$ $O(N \log N)$

  and $\hat{w}$ $O(N \log N)$.

- compute $\hat{g} \odot \hat{w}$ $O(N)$

- compute inverse DFT of $\hat{g} \odot \hat{w}$ $O(N \log N)$.

Total cost: $O(N \log N)$

Conversely,

compute $\hat{g} * \hat{w}$ reduce to:

- compute $g \odot w$ $O(N)$
- compute $\widehat{g \odot w}$ $O(N \log N)$

In total, $O(N \log N)$