Fast Fourier Transform (FFT).

(You can visit MATH3310 lecture notes for more details)

$n = 2m$ is an even integer.

$$\vec{x} = (x_0, x_1, \dots, x_{n-1}) \in \mathbb{C}^n$$

Recall : (1D DFT and 1D iDFT)

write $w_n = e^{2\pi j/n}$

$$\hat{x} = DFT(\vec{x}) = \frac{1}{n^2} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \bar{w}_n & \bar{w}_n^2 & \cdots & \bar{w}_n^{n-1} \\ \vdots & \bar{w}_n^2 & \bar{w}_n^{2\cdot 2} & \cdots & \bar{w}_n^{2(n-1)} \\ & \vdots & \vdots & & \vdots \\ 1 & \bar{w}_n^{n-1} & \bar{w}^{2\cdot(n-1)} & \cdots & \bar{w}_n^{(n-1)^2} \end{bmatrix} \vec{x}$$

$$\vec{x} = iDFT(\hat{x}) = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w_n & w_n^2 & \cdots & w_n^{n-1} \\ \vdots & w_n^2 & w_n^{2\cdot 2} & \cdots & w_n^{2(n-1)} \\ & \vdots & \vdots & & \vdots \\ 1 & w_n^{n-1} & w^{2\cdot(n-1)} & \cdots & w_n^{(n-1)^2} \end{bmatrix} \vec{x}$$

Goal : Perform Fourier Transform faster than
(Also Inverse Fourier Transform)
matrix multiplication.

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{m-1} \\ y_m \\ \vdots \\ y_{2m-1} \end{bmatrix} = \vec{y} = \overline{F}_n \vec{x} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \overline{\omega}_n & \overline{\omega}_n^{2} & \cdots & \overline{\omega}_n^{n-1} \\ 1 & \overline{\omega}_n^{2} & \overline{\omega}_n^{2\cdot 2} & \cdots & \overline{\omega}_n^{2(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \overline{\omega}_n^{n-1} & \overline{\omega}^{2\cdot(n-1)} & \cdots & \overline{\omega}_n^{(n-1)^2} \end{bmatrix} \vec{x}$$

For $0 \le h \le m-1$,

$$y_h = \sum_{k=0}^{n-1} (\overline{\omega}_n)^{kh} x_k$$

$$= \sum_{\substack{k=0 \\ k \text{ even}}}^{n-1} (\overline{\omega}_n)^{kh} x_k + \sum_{\substack{k=0 \\ k \text{ odd}}}^{n-1} (\overline{\omega}_n)^{kh} x_k$$

$$= \sum_{k'=0}^{m-1} (\overline{\omega}_{2m})^{2k'h} x_{2k'} + \sum_{k''=0}^{m-1} (\overline{\omega}_{2m})^{(2k''+1)h} x_{2k''+1}$$

$$= \sum_{k'=0}^{m-1} (\overline{\omega}_{2m})^{2k'h} x_{2k'} + \overline{\omega}_{2m}^{h} \sum_{k''=0}^{m-1} (\overline{\omega}_{2m})^{2k''h} x_{2k''+1}$$

Note $(\overline{\omega}_{2m})^{2k'h} = e^{-2\pi j \frac{2k'h}{2m}}$

$$= e^{-2\pi j \frac{k'h}{m}} = (\overline{\omega}_m)^{k'h}$$

write $\vec{x}' = (x_0, x_2, \cdots, x_{2(m-1)}) \in \mathbb{C}^m$

$\vec{x}'' = (x_1, x_3, \cdots, x_{2m-1}) \in \mathbb{C}^m$

$$= \sum_{k'=0}^{m-1} (\overline{\omega}_m)^{k'h} x'_{k'} + \overline{\omega}_{2m}^{h} \sum_{k''=0}^{m-1} (\overline{\omega}_m)^{k''h} x''_{k''}$$

$$= (\overline{F}_m \vec{x}')_h + \overline{\omega}_{2m}^{h} (\overline{F}_m \vec{x}'')_h$$

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{m-1} \\ y_m \\ \vdots \\ y_{2m-1} \end{bmatrix} : \vec{y} = F_n \vec{x} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \bar{w}_n & \bar{w}_n^{2} & \cdots & \bar{w}_n^{n-1} \\ \vdots & \bar{w}_n^{2} & \bar{w}_n^{2\cdot 2} & \cdots & \bar{w}_n^{2(n-1)} \\ & \vdots & \vdots & & \vdots \\ 1 & \bar{w}_n^{n-1} & \bar{w}^{2\cdot(n-1)} & \cdots & \bar{w}_n^{(n-1)^2} \end{bmatrix} \vec{x}$$

$$0 \le h \le m-1$$

$$y_{m+h} : \sum_{k=0}^{n-1} (\bar{w}_n)^{k(h+m)} x_k$$

$$= \sum_{\substack{k=0 \\ k \text{ even}}}^{n-1} (\bar{w}_n)^{k(h+m)} x_k + \sum_{\substack{k=0 \\ k \text{ odd}}}^{n-1} (\bar{w}_n)^{k(h+m)} x_k$$

$$= \sum_{k'=0}^{m-1} (\bar{w}_{2m})^{2k'(h+m)} x_{2k'} + \sum_{k''=0}^{m-1} (\bar{w}_{2m})^{(2k''+1)(h+m)} x_{2k''+1}$$

$$= \sum_{k'=0}^{m-1} (\bar{w}_{2m})^{2k'(h+m)} x_{2k'} + \sum_{k''=0}^{m-1} (\bar{w}_{2m})^{(2k''+1)(h+m)} x_{2k''+1}$$

$$= \sum_{k'=0}^{m-1} (\bar{w}_{2m})^{2k'(h+m)} x_{2k'} + \bar{w}_{2m}^{h+m} \sum_{k''=0}^{m-1} (\bar{w}_{2m})^{2k''(h+m)} x_{2k''+1}$$

$$= \sum_{k'=0}^{m-1} (\bar{w}_m)^{k'(h+m)} x'_{k'} + \bar{w}_{2m}^{h+m} \sum_{k''=0}^{m-1} (\bar{w}_m)^{k''(h+m)} x''_{k''}$$

Note $\bar{w}_m^{km} = e^{-2\pi j \frac{km}{m}} = e^{-2\pi j k} = 1$,

$\bar{w}_{2m}^{m} = e^{-2\pi j \frac{m}{2m}} = e^{-\pi j} = -1$.

$$= (\bar{F}_m \vec{x}')_h - \bar{w}_{2m}^{h} (\bar{F}_m \vec{x}'')_h$$

In vector Form:

$$\begin{bmatrix} y_0 \\ \vdots \\ y_{m-1} \end{bmatrix} = \bar{F}_m \vec{x}' + \begin{bmatrix} \bar{\omega}_{2m}^{0} \\ \bar{\omega}_{2m}^{1} \\ \vdots \\ \bar{\omega}_{2m}^{m-1} \end{bmatrix} \odot \bar{F}_m \vec{x}''$$

$$\begin{bmatrix} y_m \\ \vdots \\ y_{2m-1} \end{bmatrix} = \bar{F}_m \vec{x}' - \begin{bmatrix} \bar{\omega}_{2m}^{0} \\ \bar{\omega}_{2m}^{1} \\ \vdots \\ \bar{\omega}_{2m}^{m-1} \end{bmatrix} \odot \bar{F}_m \vec{x}''$$

Originally, we need to compute $\underset{\underset{\mathbb{C}^{n\times n}}{\uparrow}}{\bar{F}_n} \; \underset{\underset{\mathbb{C}^{n}}{\uparrow}}{\vec{x}}$,

now, we only need to compute $\underset{\underset{\mathbb{C}^{m\times m}}{\uparrow}}{\bar{F}_m} \; \underset{\underset{\mathbb{C}^{m}}{\uparrow}}{\vec{x}'}$

and $\underset{\underset{\mathbb{C}^{m\times m}}{\uparrow}}{\bar{F}_m} \; \underset{\underset{\mathbb{C}^{m}}{\uparrow}}{\vec{x}''}$,

reduce a large matrix multiplication

to 2 smaller matrix multiplication.

If $m = n/2$ is also even, we can further reduce

the 2 smaller matrix multiplication

to 4 much smaller matrix multiplication

If $n = 2^l$,

$$n \to \frac{n}{2} \to \frac{n}{4} \to \cdots \to 1.$$

Image Blurring

g : observation, blurry image

f : true, clean image.

Write $g = D(f) + n$

D : Degradation operator, n : noise

Suppose :

1. D is position invariant :

$$\tilde{f}(x,y) = f(x-\alpha, y-\beta)$$

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

2. D is linear :

$$D(af_1 + bf_2) = aD(f_1) + bD(f_2).$$

$\forall f_1, f_2$ image
a, b scalar.

Then : $\exists h$

s.t. $D(f) = f * h$.

Then $\hat{g} = c\,\hat{h} \odot \hat{f} + \hat{n}$

Remove the noise and inverse the $\hat{h}$ can get back the clean image.

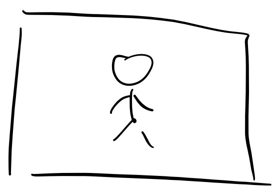
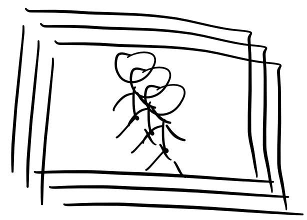
Example.

Assuming periodicity.

Suppose $g \in \mathbb{R}^{N\times N}$, $0 \le m, n \le N-1$, $c > 0$

$g = D(f)$ given by:

$$g(m,n) = \sum_{j=0}^{7} f(m - cj, n - cj)$$

$f =$  $g =$ 

$$\tilde{f}(x,y) = f(x-\alpha, y-\beta)$$

Then
$$D(\tilde{f})(m,n) = \sum_{j=0}^{7} \tilde{f}(m - cj, n - cj)$$

$$= \sum_{j=0}^{7} f(m - cj - \alpha, n - cj - \beta)$$

$$= \sum_{j=0}^{7} f((m-\alpha) - cj, (n-\beta) - cj)$$

$$= g(m-\alpha, n-\beta)$$

$\therefore$ Position Invariant.

$$D(af_1 + bf_2)$$

$$= \sum_{j=0}^{7} (af_1 + bf_2)(m - cj, n - cj)$$

$$= \sum_{j=0}^{T} a f_1(m-cj, n-cj) + b f_2(m-cj, n-cj)$$

$$= a \sum_{j=0}^{T} f_1(m-cj, n-cj) + b \sum_{j=0}^{T} f_2(m-cj, n-cj)$$

$$= a D(f_1) + b D(f_2).$$

$\therefore$ linear.

$h = D(\delta)$, where

$$\delta(m,n) = \begin{cases} 1, & \text{if } m=n=0 \\ 0, & \text{else.} \end{cases}$$

$$h(m,n) = \sum_{j=0}^{T} \delta(m-cj, n-cj)$$

$$\therefore \quad h(m,n) = \begin{cases} 1, & \text{if } m=n=cj,\ 0 \le j \le T. \\ 0, & \text{else} \end{cases}$$

$$\hat{g} = c\,\hat{h} \odot \hat{f} + \hat{n}$$

$$\hat{g}(k,l) = c\,\hat{h}(k,l) \cdot \hat{f}(k,l) + \hat{n}(k,l)$$

Construct $\tilde{f}(k,l) = T(k,l)\,\hat{g}(k,l)$
by a suitable $T(k,l)$.

Direct Inverse Filtering:

$$T(k,l) = \frac{1}{c\,\hat{h}(k,l) + \varepsilon\,\mathrm{sign}(\hat{h}(k,l))}$$

$$\tilde{f}(k,l) = \frac{\hat{g}(k,l)}{c\,\hat{h}(k,l) + \varepsilon\,\mathrm{sign}(\hat{h}(k,l))}$$

$$= \frac{c\,\hat{h}(k,l)\,\hat{f}(k,l)}{c\,\hat{h}(k,l) + \varepsilon\,\mathrm{sign}(\hat{h}(k,l))} + \frac{\underline{\underline{\hat{n}(k,l)}}}{\underline{\underline{c\,\hat{h}(k,l)}} + \varepsilon\,\mathrm{sign}(\hat{h}(k,l))}$$

usually large in high frequency

usually small in high frequency

$\rightarrow$ Increase noise.

Modified Inverse filtering:

Butterworth LPF

$$T(k,l) = \frac{B(k,l)}{c\hat{h}(k,l) + \varepsilon\,\mathrm{sign}(\hat{h}(k,l))}$$

Then

close to zero in high frequency

large in high frequency

$$\tilde{f}(k,l) = T(k,l)\,\hat{g}(k,l)$$

$$= T(k,l)\,\hat{h}(k,l)\,\hat{f}(k,l) + \frac{\hat{n}(k,l)\,B(k,l)}{c\hat{h}(k,l) + \varepsilon\,\mathrm{sign}(\hat{h}(k,l))}$$

$\longrightarrow$ Reduce Noise.