2D DFT as 1D DFT

For $\vec{f} = [f_0, f_1, \cdots, f_{N-1}]^T \in \mathbb{C}^N$,

The 1D DFT $\hat{f} \in \mathbb{C}^N$ is defined as:

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

Also, can be written into matrix form:

$$\hat{f} = U \vec{f}$$ (U is the same matrix as in 2D DFT)

2D DFT ($\hat{F} = UFU$)

is a separable transform,

which can be separated into 2 1D DFT:

(i) 1D DFT on the columns

and (ii) 1D DFT on the rows

One can apply (i) first and then (ii),

or equivalently apply (ii) first and then (i).

apply 1D DFT on the columns of $\bar{F}$:

$$\tilde{F} = U\bar{F} = \left[U\vec{f}_0 \,\middle|\, U\vec{f}_1 \,\middle|\, \cdots \,\middle|\, U\vec{f}_{N-1} \right]$$

$$\tilde{F}(m, \beta) = \frac{1}{N} \sum_{\alpha=0}^{N-1} \bar{F}(\alpha, \beta) e^{-j\frac{2\pi}{N}(\alpha m)}$$

apply again 1D DFT on the rows of $\tilde{F}$:

$$\hat{F} = \tilde{F}U = \begin{bmatrix} - & \tilde{F}_0 U & - \\ - & \tilde{F}_1 U & - \\ & \vdots & \\ - & \tilde{F}_{N-1} U & - \end{bmatrix} = (U\tilde{F}^T)^T$$

$$= U\bar{F}U$$

$$\hat{F}(m, n) = \frac{1}{N} \sum_{\beta=0}^{N-1} \tilde{F}(m, \beta) e^{-j\frac{2\pi}{N}(\beta n)}$$

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

Computation cost of DFT:

To compute a 1D DFT on N-vector,
there are N numbers to compute,
each need N multiplication and $N-1$ summations

computation cost $= O(n^2)$

To compute a 1D DFT on N-vector,
there are N^2 numbers to compute,
each need $2N$ multiplication and $2(N-1)$ summations

computation cost $= O(n^3)$

For 1D DFT, we have a faster algorithm, called Fast Fourier Transform (FFT), which only needs $O(n \log n)$ to compute.

For 2D DFT, we can use FFT to compute DFT on columns, totally $n \cdot O(n \log n) = O(n^2 \log n)$

and then again FFT on the rows, $O(n^2 \log n)$ again.

So totally only $O(n^2 \log n)$ is required for 2D DFT.

2D DFT of Rotated Image

$$g \in \mathbb{C}^{N \times N}$$

rewrite : $k = r\cos\theta$, $l = r\sin\theta$

$m = w\cos\phi$, $n = w\sin\phi$

DFT of g in terms of w, ϕ : $\hat{g}(w, \phi)$

Rotate the image by θ_0 degree

to get $\tilde{g}(r, \theta) = g(r, \theta + \theta_0)$

Then :

$$\hat{\tilde{g}}(w, \phi) = \hat{g}(w, \phi + \theta_0)$$

i.e. rotate an image by θ_0,

then rotate the DFT also by θ_0.

e.g.

$$g = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$U = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & w & w^2 & w^3 \\ 1 & w^2 & w^4 & w^6 \\ 1 & w^3 & w^6 & w^9 \end{bmatrix}$$

$$\hat{S} = U S U$$

$$= \begin{bmatrix} 1/4 & -1/4 & 1/4 & -1/4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

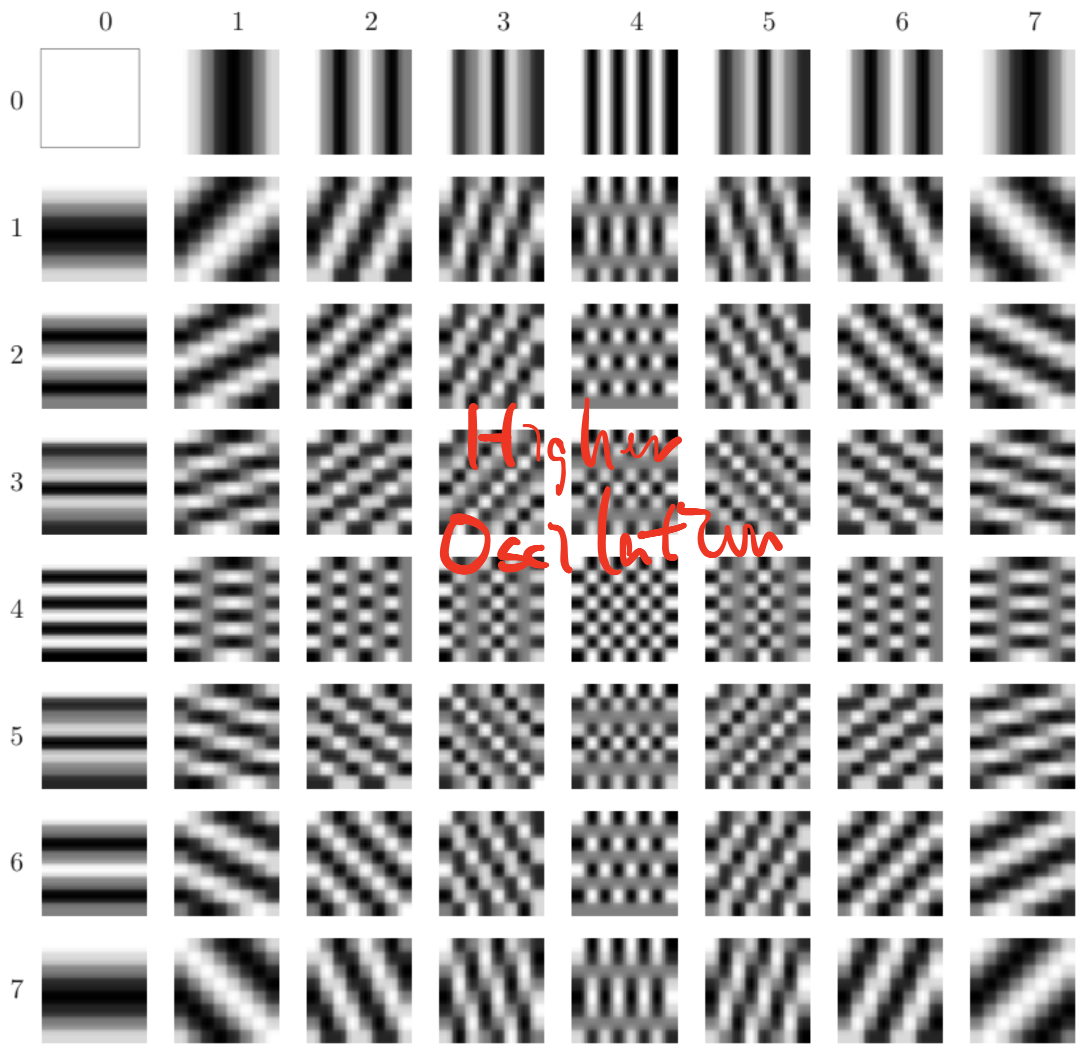
$$S_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\hat{S}_1 = \begin{bmatrix} 1/4 & 0 & 0 & 0 \\ -1/4 & 0 & 0 & 0 \\ 1/4 & 0 & 0 & 0 \\ -1/4 & 0 & 0 & 0 \end{bmatrix}$$

Real Part of Element Images of DFT:

lower oscilation

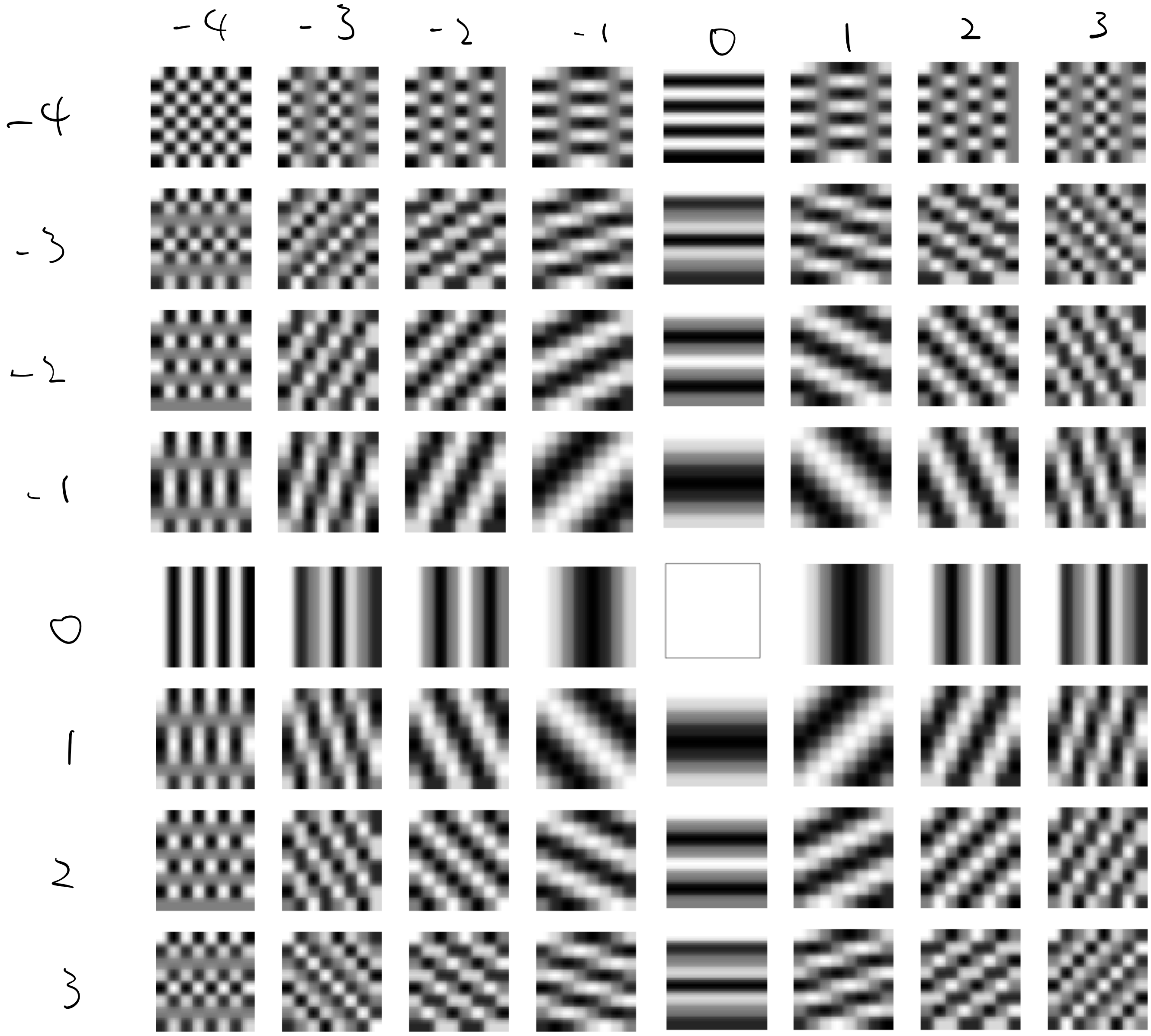
lower oscilation



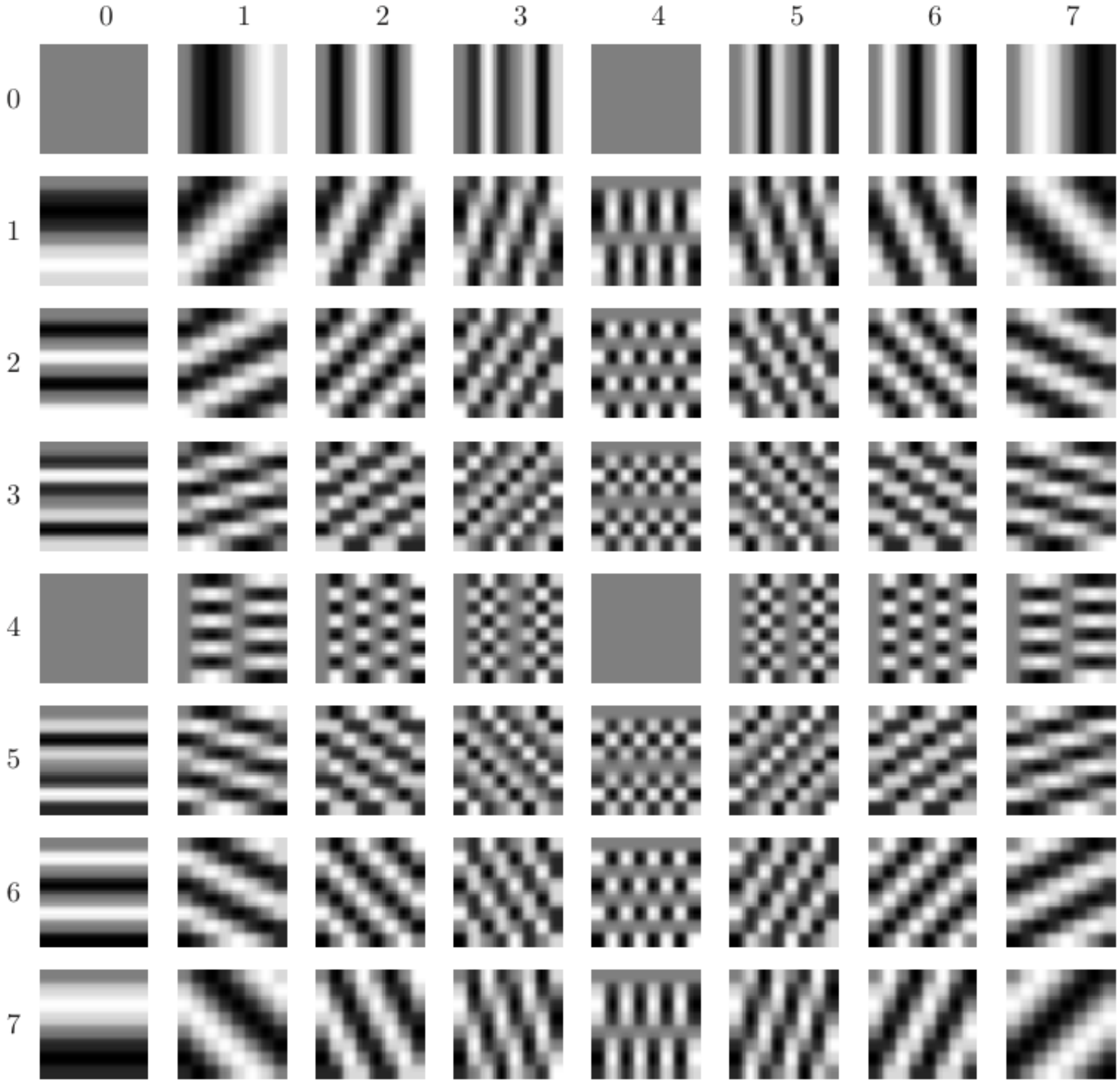
lower oscilation

lower oscilation

Centralization:



Imaginary Part :



Centralization :