2-D DISCRETE FOURIER TRANSFORM

DEFINITION

$$F(k,l) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)\cdot e^{-j2\pi\left(\frac{mk}{M}+\frac{nl}{N}\right)}$$ **forward DFT**

$$f(m,n) = \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} F(k,l)\cdot e^{+j2\pi\left(\frac{mk}{M}+\frac{nl}{N}\right)}$$ **inverse DFT**

- ***The DFT is a transform of a discrete, complex 2-D array of size M x N into another discrete, complex 2-D array of size M x N***

Approximates the Continuous Fourier Transform (CFT) under certain conditions

Both f(m,n) and F(k,l) are 2-D periodic

Alternate definitions:

- $\frac{1}{MN}$ ***in inverse definition instead, or*** $\frac{1}{\sqrt{MN}}$ ***in forward and inverse definitions ("unitary")***
- ***doesn't matter as long as consistent***

2-D DISCRETE FOURIER TRANSFORM

RELATION OF THE DFT TO THE CFT

- One view of the DFT is as an *approximation* to the CFT
- " recipe" to convert CFT to DFT:

1. sample f(x,y)

$$f(x,y) \cdot \frac{1}{XY} comb(x/X, y/Y)$$

2. truncate to MX x NY

$$f(x,y) \cdot \frac{1}{XY} comb(x/X, y/Y) \cdot rect(x/MX, y/NY)$$

3. make periodic

$$\left[f(x,y) \cdot \frac{1}{XY} comb(x/X, y/Y) \cdot rect(x/MX, y/NY) \right] ** \frac{1}{MX \cdot NY} comb(x/MX, y/NY) = f_p(m,n)$$

, i.e. the periodic extension of a 2-D array f(m,n) with sample intervals X=Y=1

2-D DISCRETE FOURIER TRANSFORM

4. take CFT

replicate (aliasing occurs here) ↓ — smooth (leakage occurs here) ↓ — sample ↓

$$[F(u,v) ** comb(uX, vY) ** MX \cdot NY \operatorname{sinc}(uMX, vNY)] \cdot comb(uMX, vNY)$$
$$= F_p(k, l)$$

, i.e. the periodic extension of a 2-D array F(k,l) with sample intervals 1/X=1/Y=1

- The arrays f and F are both discrete and periodic in space and spatial frequency, respectively

2-D DISCRETE FOURIER TRANSFORM

CALCULATION OF DFT

- Both arrays $f(m,n)$ and $F(k,l)$ are

  periodic (period = $M \times N$) and

  sampled ($X \times Y$ in space, $1/MX \times 1/NY$ in frequency)

- In the CFT, if one function has compact support (i.e. it is space- or frequency-limited), the other must have ∞ support

- Therefore, aliasing will occur with the DFT, either in space or frequency. If we want the DFT to closely approximate the CFT, aliasing must be minimized in both domains

- The Fast Fourier Transform (FFT) is an efficient algorithm to calculate the DFT that takes advantage of the periodicities in the complex exponential

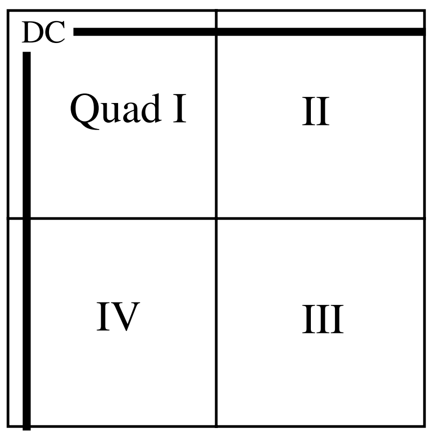
  Can use 1-D FFT for 2-D DFT (later)

2-D DISCRETE FOURIER TRANSFORM

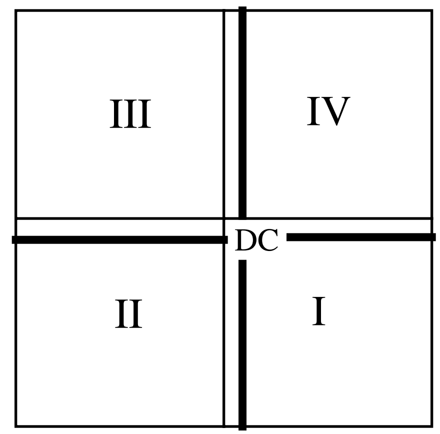
ARRAY COORDINATES

- The DC term (u=v=0) is at (0,0) in the raw output of the DFT (e.g. the Matlab function "fft2")

raw output of DFT



reordered output of DFT



- Reordering puts the spectrum into a "physical" order (the same as seen in optical Fourier transforms) (e.g. the Matlab function "fftshift")
- N and M are commonly powers of 2 for the FFT. Therefore, the DC term is at (M/2,N/2) in the reordered format for (0,0) indexing and at (M/2+1,N/2+1) for (1,1) indexing

2-D DISCRETE FOURIER TRANSFORM

SAMPLE INTERVALS

- Constraints

  product of physical sample intervals in x and u, y and v: $XU = 1/M$, $YV = 1/N$

  sampling (replication) frequency in u and v: $u_s = 1/X$, v: $v_s = 1/Y$

  folding frequency in u and v: $u_f = 1/2X$, $v_f = 1/2Y$

- For images, a convenient, normalized set of units is

  $X = Y = 1$ pixel

- Therefore,

  $U = 1/M$ cycles/pixel, $u_s = 1$ cycle/pixel, $u_f = 1/2$ cycle/pixel

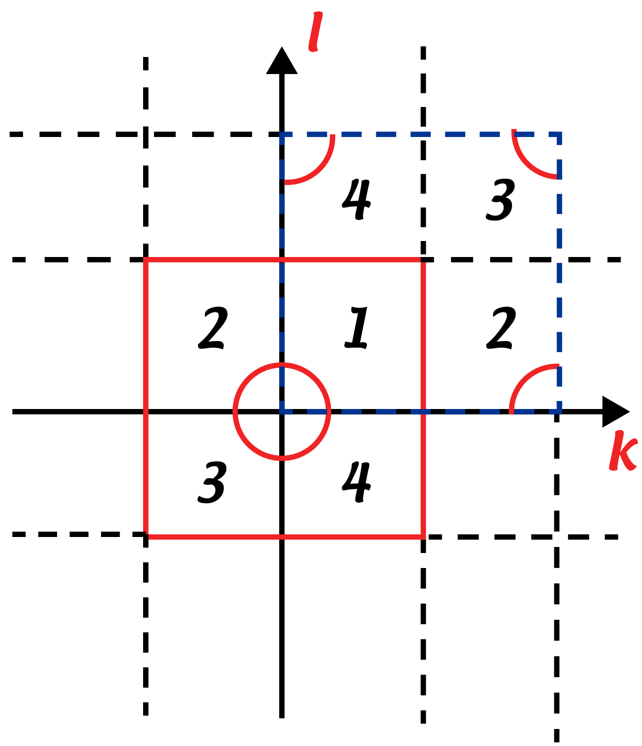
  $V = 1/N$ cycles/pixel, $v_s = 1$ cycle/pixel, $v_f = 1/2$ cycle/pixel

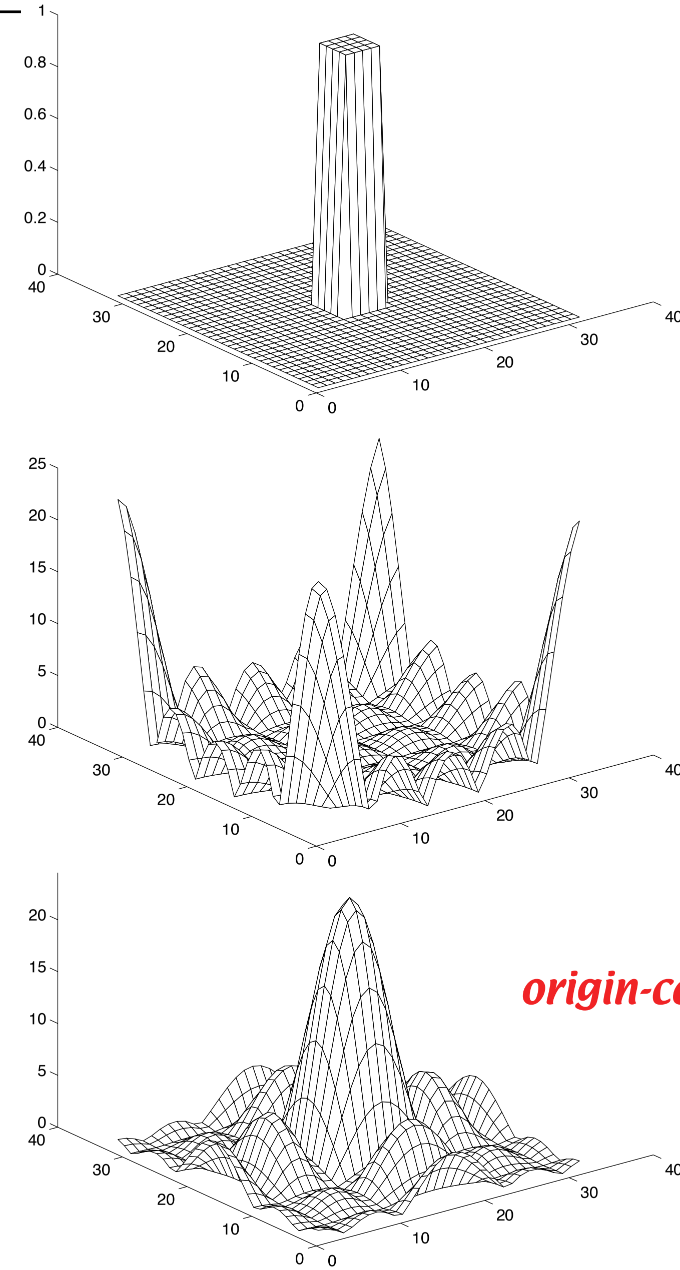
- Note, in reodered DFT format, u_f and v_f are along the first row and columns of the array

2-D DISCRETE FOURIER TRANSFORM

Reordering the 2-D DFT

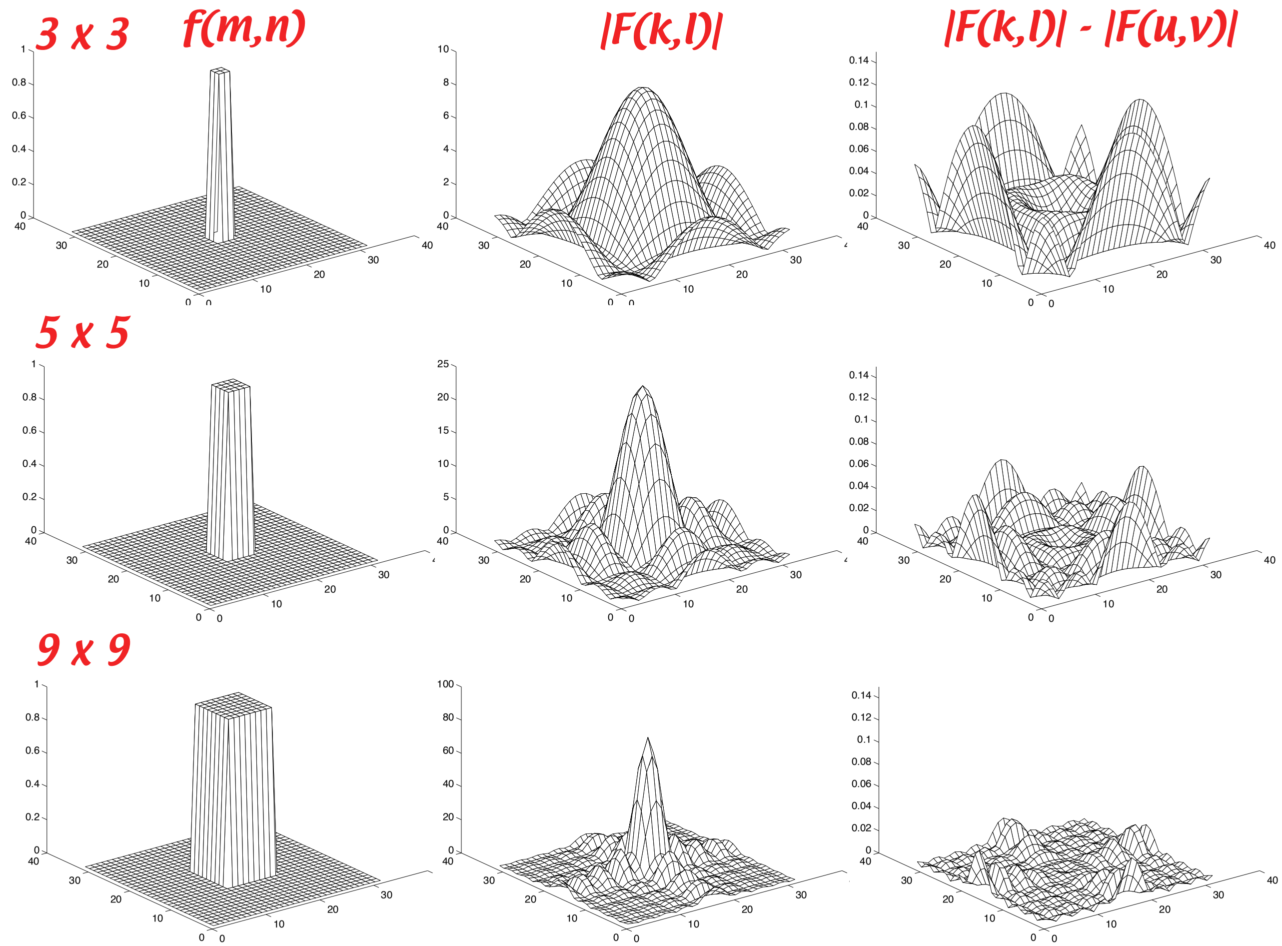
- "origin-centered" display





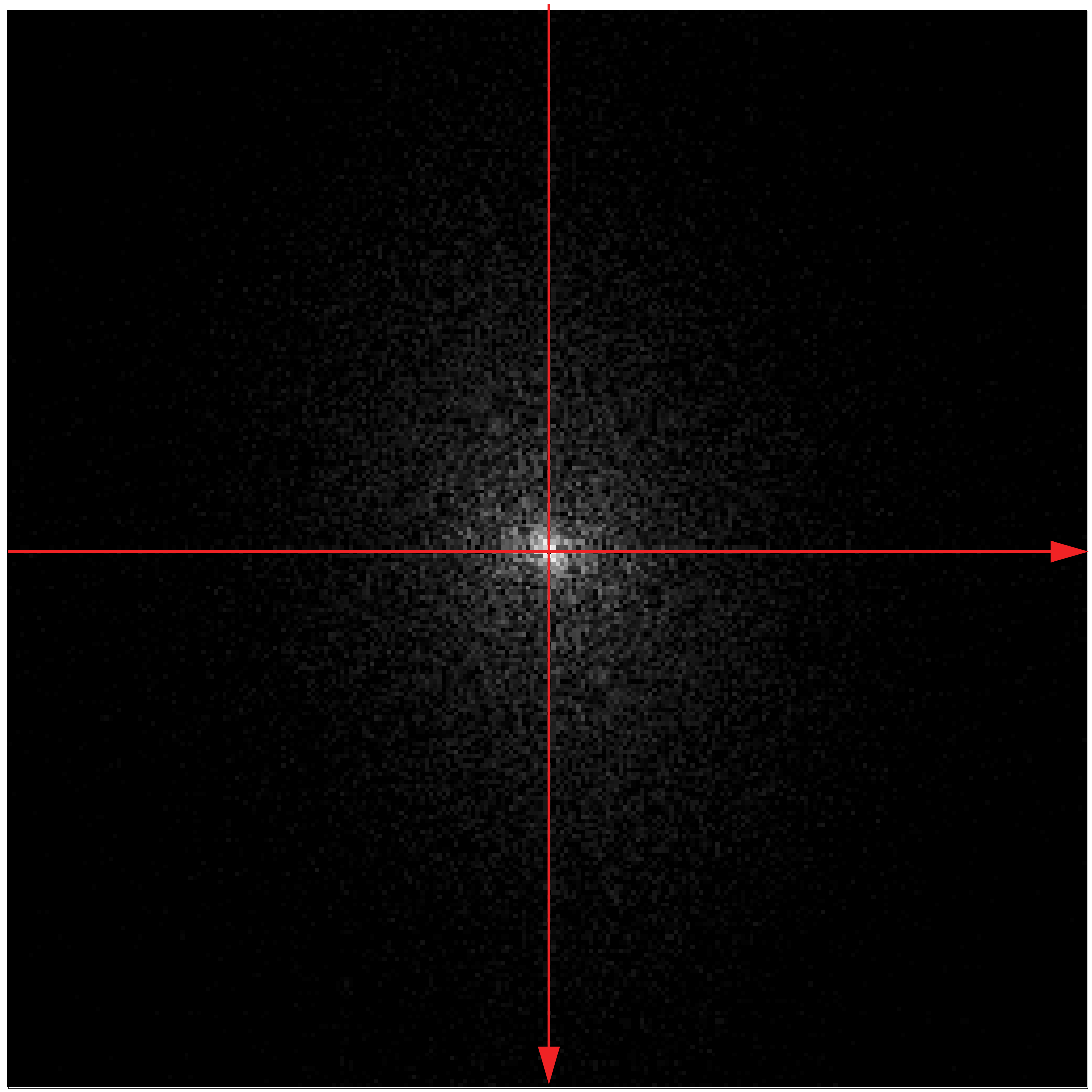
Aliasing in the frequency domain

- ***DFT of discrete approximation to a rect(x/W,y/W) function***

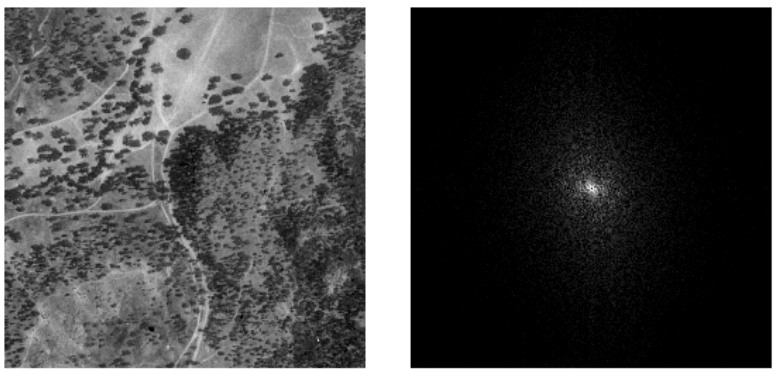


2-D DISCRETE FOURIER TRANSFORM

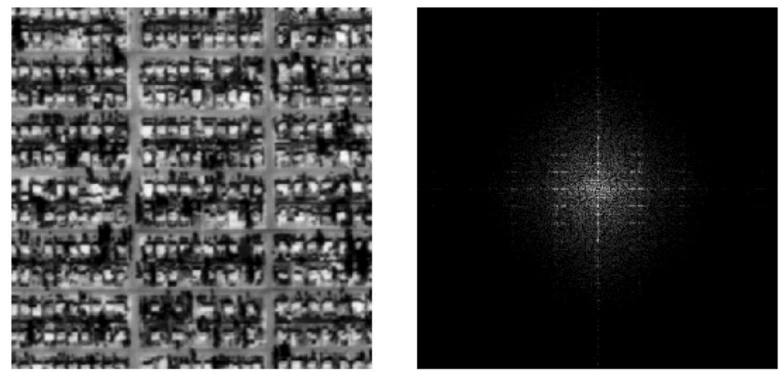
Digital image power spectrum (squared amplitude of F) coordinates



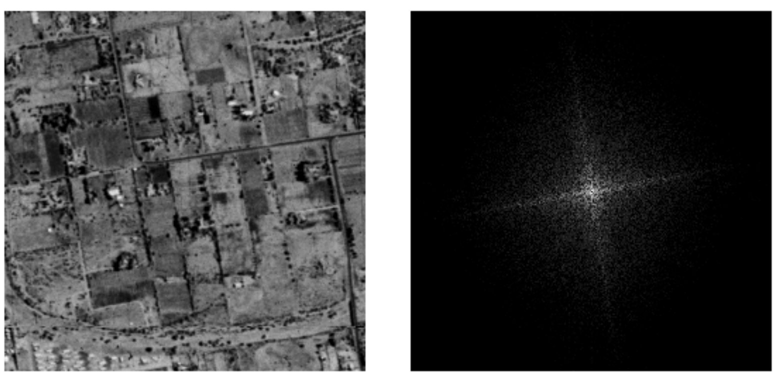
EXAMPLES OF IMAGE POWER SPECTRA



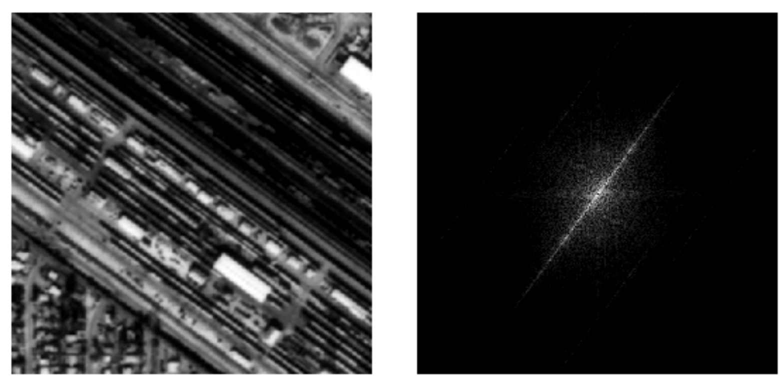
desert



streets



fields



railroad

2-D DISCRETE FOURIER TRANSFORM

DISPLAY OF POWER SPECTRA

- Large dynamic range

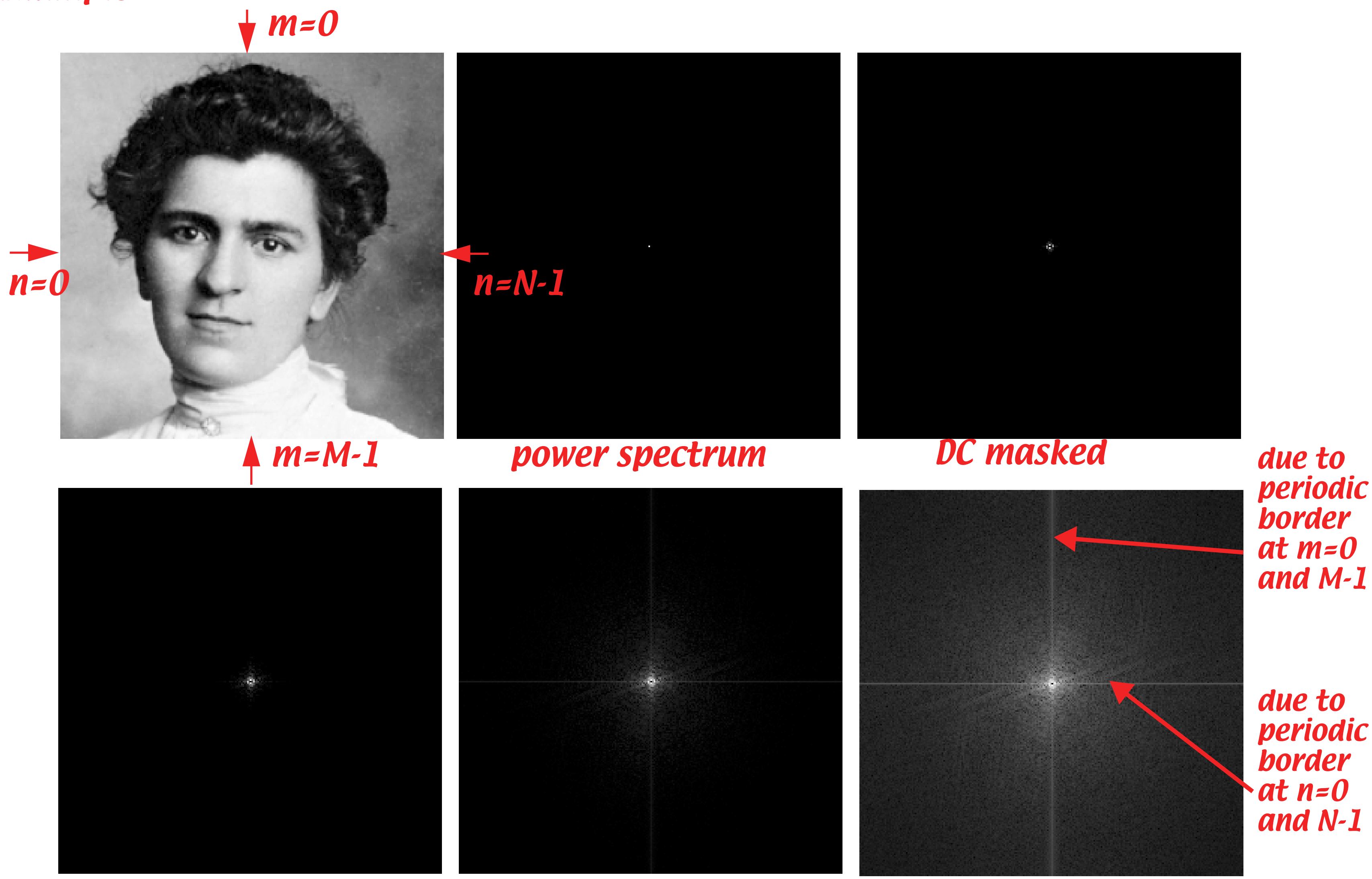
  amplitude at zero-frequency dominates

Power Spectra Display

- Mask zero-frequency term to zero
- Contrast stretch with square-root transform
- Repeat constrast stretch as needed

2-D DISCRETE FOURIER TRANSFORM

Example



$\sqrt[2]{2}$ $\sqrt[4]{2}$ $\sqrt[8]{2}$

2-D DISCRETE FOURIER TRANSFORM

MATRIX REPRESENTATION

This section is from lecture notes by my late friend and colleague, Professor Steve Park, of the College of William and Mary, Virginia

- **Compact notation**
- **Generalizable to other transforms**
- **DFT definition** $F(k,l) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)\cdot e^{-j2\pi\left(\frac{mk}{M}+\frac{nl}{N}\right)}$

let

$$W_M(m,k) = e^{-j2\pi\left(\frac{mk}{M}\right)}$$

$$W_N(n,l) = e^{-j2\pi\left(\frac{nl}{N}\right)}$$

, where W_M is M x M, W_N is N x N

then $F(k,l) = \frac{1}{MN}\sum_{m=0}^{M-1} W_M(m,k)\sum_{n=0}^{N-1} f(m,n)W_N(n,l) = \frac{1}{MN}W_M f W_N$,

which is the forward transform

2-D DISCRETE FOURIER TRANSFORM

- ***Note that***

$$W_M^* W_M = W_M W_M^* = M I_M \text{ (M x M identity matrix)}$$

$$W_N^* W_N = W_N W_N^* = N I_N \text{ (N x N identity matrix)}$$

then,

$$W_M^* F W_N^* = \frac{1}{MN}(W_M^* W_M) f (W_N W_N^*)$$

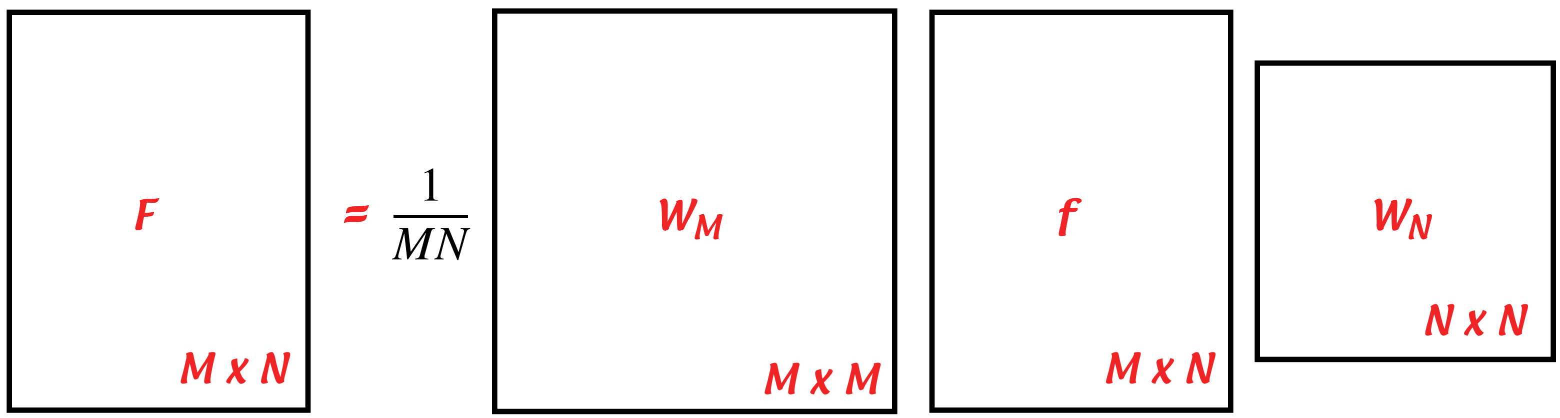
$$= \frac{1}{MN}(M I_M) f (N I_N)$$

, which is the inverse transform

$$= f$$

2-D DISCRETE FOURIER TRANSFORM

Matrix Dimensionality Diagram (M > N)



$$F = \frac{1}{MN} W_M f W_N$$

- Diagram for inverse transform is similar, except no 1/MN factor
- Note, this representation is possible because the 2-D DFT kernel is **separable**, i.e. $e^{-j2\pi\left(\frac{mk}{M} + \frac{nl}{N}\right)} = e^{-j2\pi\left(\frac{mk}{M}\right)} e^{-j2\pi\left(\frac{nl}{N}\right)}$

CALCULATING THE 2-D DFT

$$F = \frac{1}{MN} W_M f W_N$$

- ***Step 1***

write image as $f = [f_1|f_2| \ldots |f_N]$ ***where*** $f_1, f_2, \ldots f_N$ ***are the image columns of length M***

then,

$$F = \frac{1}{N}\left[\frac{1}{M} W_M f_1 | \frac{1}{M} W_M f_2 | \ldots | \frac{1}{M} W_M f_N\right] W_N$$

$$= \frac{1}{N}[F_1|F_2| \ldots |F_N] W_N$$

where each column is a 1-D DFT of length M of the image columns

2-D DISCRETE FOURIER TRANSFORM

- **Step 2**

form matrix transpose $F^t = \frac{1}{N} W_N^t \begin{bmatrix} F_1^t \\ \hline F_2^t \\ \hline \cdots \\ \hline F_N^t \end{bmatrix}$ ***note, W is symmetric*** $W_N^t = W_N$

- **Step 3**

partition image matrix by columns

$$\begin{bmatrix} F_1^t \\ \hline F_2^t \\ \hline \cdots \\ \hline F_N^t \end{bmatrix} = [g_1 | g_2 | \ldots | g_M]$$

, where each column is an array of length N

2-D DISCRETE FOURIER TRANSFORM

then $F^t = \left[\frac{1}{N}W_N g_1 \middle| \frac{1}{N}W_N g_2 \middle| \; \ldots \; \middle| \frac{1}{N}W_N g_N\right]$

where each column is a 1-D DFT of length N

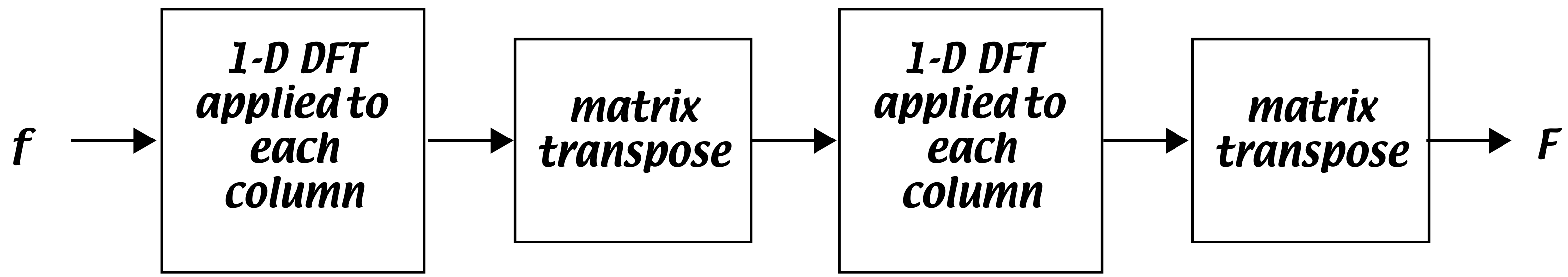
therefore $F^t = [G_1 | G_2 | \; \ldots \; | G_M]$

- ***Step 4***

transpose F^t to get F

2-D DISCRETE FOURIER TRANSFORM

Calculating the 2-D DFT - Summary



N 1-D DFTs of length M
$N(M\log_2 M)$ operations

M 1-D DFTs of length N
$M(N\log_2 N)$ operations

- $N(M\log_2 M) + M(N\log_2 N) = MN\log_2(MN)$ total operations

  assumes 1-D FFT is used and M,N are powers of 2

- Compares to M^2N^2 total operations for "brute force" 2-D DFT