6.3000: Signal Processing

2D Fourier Transforms 1

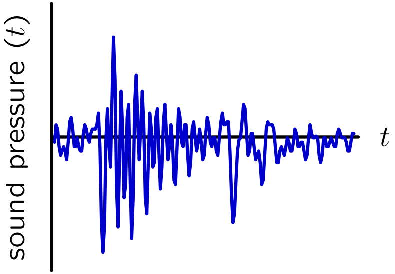
- Introduction to 2D Signal Processing
- 2D Fourier Representations

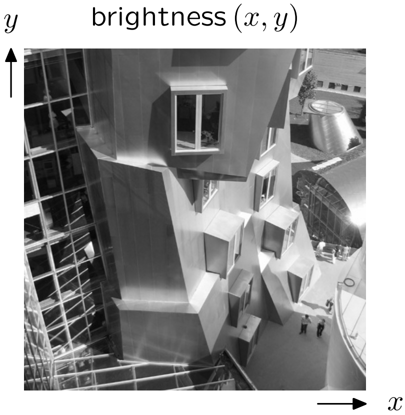
April 22, 2025

Signals

Signals are functions that are used to convey information.

– may have 1 or 2 or 3 or even more **independent variables**





A 1D signal has a one-dimensional domain.

We have usually thought of the domain as time t or discrete time n.

A 2D signal has a two-dimensional domain.

We will usually think of the domain as x and y or n_x and n_y.

Fourier Representations

From "Continuous Time" to "Continuous Space."

One dimensional CTFT:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)\, e^{-j\omega t}\, dt$$

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)\, e^{j\omega t}\, d\omega$$

Two dimensional CTFT:

$$F(\omega_x, \omega_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)\, e^{-j(\omega_x x + \omega_y y)}\, dx\, dy$$

$$f(x, y) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(\omega_x, \omega_y)\, e^{j(\omega_x x + \omega_y y)}\, d\omega_x\, d\omega_y$$

integrals $\rightarrow$ double integrals; sum of x and y exponents in kernel function.

Fourier Representations

From "Discrete Time" to "Discrete Space."

One dimensional DTFT:

$$F(\Omega) = \sum_{n=-\infty}^{\infty} f[n]\, e^{-j\Omega n}$$

$$f[n] = \frac{1}{2\pi} \int_{2\pi} F(\Omega)\, e^{j\Omega n} d\Omega$$

Two dimensional DTFT:

$$F(\Omega_x, \Omega_y) = \sum_{n_x=-\infty}^{\infty} \sum_{n_y=-\infty}^{\infty} f[n_x, n_y]\, e^{-j(\Omega_x n_x + \Omega_y n_y)}$$

$$f[n_x, n_y] = \frac{1}{4\pi^2} \int_{2\pi} \int_{2\pi} F(\Omega_x, \Omega_y)\, e^{j(\Omega_x n_x + \Omega_y n_y)} d\Omega_x d\Omega_y$$

double integrals; double sums; sum of x and y exponents in kernel function.

Fourier Representations

From 1D DFT to 2D DFT.

One dimensional DFT:

$$F[k] = \frac{1}{N} \sum_{n=0}^{N-1} f[n] \, e^{-j \frac{2\pi k}{N} n}$$

$$f[n] = \sum_{k=0}^{N-1} F[k] \, e^{j \frac{2\pi k}{N} n}$$

Two dimensional DFT:

$$F[k_x, k_y] = \frac{1}{N_x N_y} \sum_{n_x=0}^{N_x-1} \sum_{n_y=0}^{N_y-1} f[n_x, n_y] \, e^{-j\left(\frac{2\pi k_x}{N_x} n_x + \frac{2\pi k_y}{N_y} n_y\right)}$$

$$f[n_x, n_y] = \sum_{k_x=0}^{N_x-1} \sum_{k_y=0}^{N_y-1} F[k_x, k_y] \, e^{j\left(\frac{2\pi k_x}{N_x} n_x + \frac{2\pi k_y}{N_y} n_y\right)}$$

double sums; sum of x and y exponents in kernel function.

Importance of Orthogonality

Fourier series represent periodic signals as weighted sum of **basis functions**.

$$f[n] = \sum_{k=0}^{N-1} F[k] e^{j\frac{2\pi}{N}kn}$$

We "sifted" out the l^{th} component by multiplying both sides by $e^{-j\frac{2\pi}{N}ln}$ and summing over a period.

$$\sum_{n=0}^{N-1} f[n] e^{-j\frac{2\pi}{N}ln} = \sum_{n=0}^{N-1}\sum_{k=0}^{N-1} F[k] e^{j\frac{2\pi}{N}kn} e^{-j\frac{2\pi}{N}ln} = \sum_{k=0}^{N-1} F[k] \sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}(k-l)n}$$

$$= \sum_{k=0}^{N-1} F[k] N\delta\Big[(k-l) \bmod N\Big] = NF[l]$$

This sifting provided an explicit "analysis" formula for the coefficients:

$$F[k] = \frac{1}{N}\sum_{n=0}^{N-1} f[n] e^{-j\frac{2\pi}{N}kn}$$

Orthogonality of the basis functions is key to Fourier decomposition.

Orthogonality

The form of the 2D Fourier kernel preserves orthogonality.

1D DFT basis functions: $\phi_k[n] = e^{j\frac{2\pi}{N}kn}$

"Inner product" of 1D basis functions:

$$\sum_n \phi_k^*[n]\phi_l[n] = \sum_{n=0}^{N-1} e^{-j\frac{2\pi}{N}kn} e^{j\frac{2\pi}{N}ln} = \sum_{n=0}^{N-1} e^{-j\frac{2\pi}{N}(k-l)n} = N\delta\Big[(k-l) \bmod N\Big]$$

2D DFT basis functions: $\phi_{k_x,k_y}[n_x,n_y] = e^{j\frac{2\pi}{N_x}k_x n_x} e^{j\frac{2\pi}{N_y}k_y n_y}$

"Inner product" of 2D basis functions:

$$\sum_{n_x,n_y} \phi_{k_x,k_y}^*[n_x,n_y]\phi_{l_x,l_y}[n_x,n_y] = \sum_{n_x,n_y} e^{-j\left(\frac{2\pi}{N_x}k_x n_x + \frac{2\pi}{N_y}k_y n_y\right)} e^{j\left(\frac{2\pi}{N_x}l_x n_x + \frac{2\pi}{N_y}l_y n_y\right)}$$

$$= \left(\sum_{n_x} e^{-j\frac{2\pi}{N_x}(k_x-l_x)n_x}\right)\left(\sum_{n_y} e^{-j\frac{2\pi}{N_y}(k_y-l_y)n_y}\right)$$

$$= N_x N_y \delta\Big[(k_x-l_x) \bmod N_x\Big]\delta\Big[(k_y-l_y) \bmod N_y\Big]$$

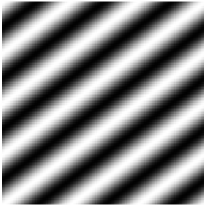
Check Yourself

The 2D Fourier basis functions have the following form.

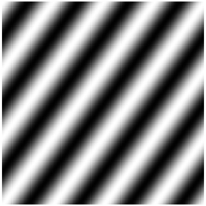
$$\phi_{k_x,k_y}[n_x,n_y] = e^{j\left(\frac{2\pi k_x}{N_x}n_x+\frac{2\pi k_y}{N_y}n_y\right)}$$

Which (if any) of the following images show the real part of one of the basis functions $\phi_{k_x,k_y}[n_x,n_y]$?

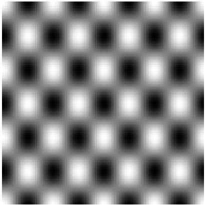
A



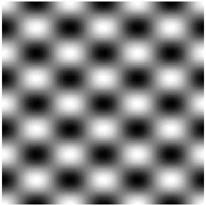
B



C



D



What values of k_x and k_y correspond to basis function?

2D Discrete Fourier Transform

Finding a 2D DFT.

Example: Find the DFT of a **2D unit sample**.

$$f_0[n_x, n_y] = \delta[n_x]\delta[n_y] = \begin{cases} 1 & n_x = 0 \text{ and } n_y = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned} F_0[k_x, k_y] &= \frac{1}{N_x N_y} \sum_{n_x=0}^{N_x-1} \sum_{n_y=0}^{N_y-1} \delta[n_x]\delta[n_y]\, e^{-j\left(\frac{2\pi k_x}{N_x} n_x + \frac{2\pi k_y}{N_y} n_y\right)} \\ &= \frac{1}{N_x N_y} \sum_{n_x=0} \sum_{n_y=0} e^{-j\left(\frac{2\pi k_x}{N_x} 0 + \frac{2\pi k_y}{N_y} 0\right)} \\ &= \frac{1}{N_x N_y} \end{aligned}$$

$$\delta[n_x]\delta[n_y] \quad \overset{\text{DFT}}{\Longrightarrow} \quad \frac{1}{N_x N_y}$$

This is a perfectly fine way to compute a Fourier Transform.
But there are other methods that provide additional insights.

2D Discrete Fourier Transform

Alternatively, implement a 2D DFT as a sequence of 1D DFTs.

$$F[k_x,k_y] = \frac{1}{N_xN_y}\sum_{n_y=0}^{N_y-1}\sum_{n_x=0}^{N_x-1} f[n_x,n_y]\, e^{-j\left(\frac{2\pi k_x}{N_x}n_x+\frac{2\pi k_y}{N_y}n_y\right)}$$

$$= \underbrace{\frac{1}{N_y}\sum_{n_y=0}^{N_y-1}\underbrace{\left(\frac{1}{N_x}\sum_{n_x=0}^{N_x-1} f[n_x,n_y]\, e^{-j\frac{2\pi k_x}{N_x}n_x}\right)}_{\text{first take DFTs of rows}} e^{-j\frac{2\pi k_y}{N_y}n_y}}_{\text{then take DFTs of resulting columns}}$$

Start with a 2D function of space $f[n_x, n_y)$.

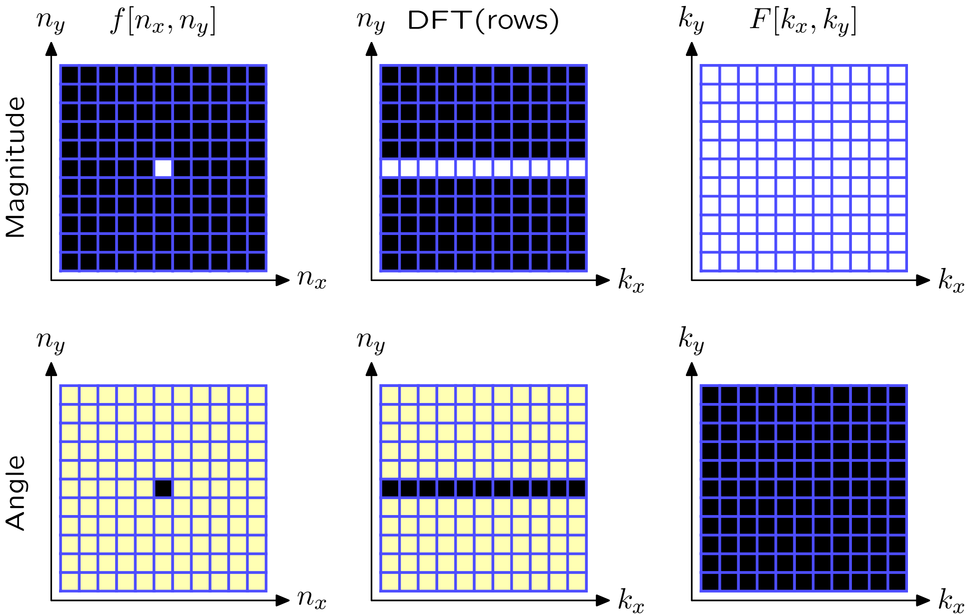
- Replace each row by the DFT of that row.
- Replace each column by the DFT of that column.

The result is $F[k_x, k_y]$, the 2D DFT of $f[n_x, n_y]$.

Could just as well start with columns and then do rows.

2D Discrete Fourier Transform

Example: Find the DFT of a 2D unit sample.



2D Discrete Fourier Transform

Example: Find the DFT of a constant.

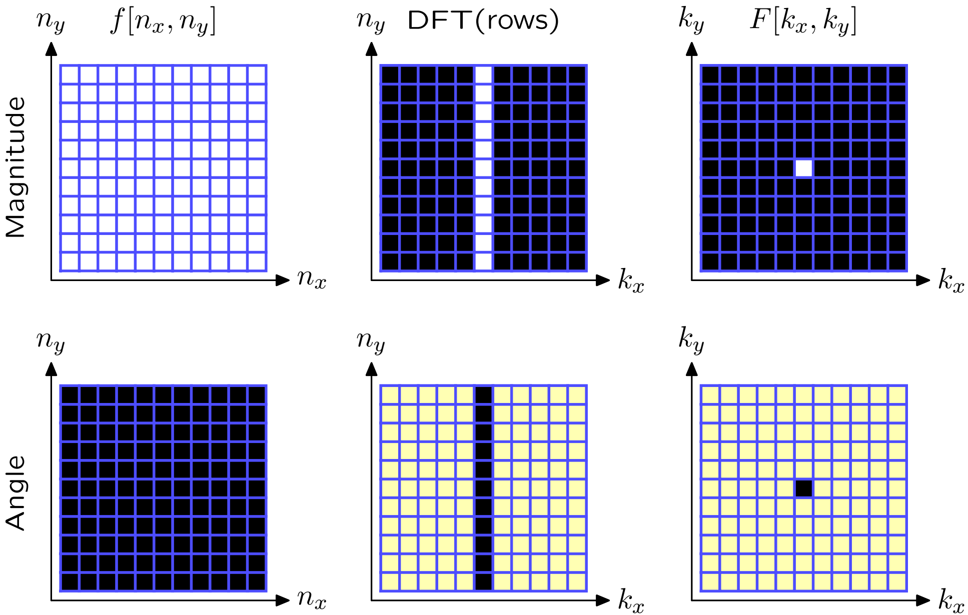
$$f_1[n_x, n_y] = 1$$

$$\begin{aligned} F_1[k_x, k_y] &= \frac{1}{N_x N_y} \sum_{n_x=0}^{N_x-1} \sum_{n_y=0}^{N_y-1} e^{-j\left(\frac{2\pi k_x}{N_x} n_x + \frac{2\pi k_y}{N_y} n_y\right)} \\ &= \left(\frac{1}{N_x} \sum_{n_x=0}^{N_x-1} e^{-j\frac{2\pi k_x}{N_x} n_x} \right) \left(\frac{1}{N_y} \sum_{n_y=0}^{N_y-1} e^{-j\frac{2\pi k_y}{N_y} n_y} \right) \\ &= \delta[k_x]\delta[k_y] \end{aligned}$$

$$1 \quad \stackrel{\text{DFT}}{\Longrightarrow} \quad \delta[k_x]\delta[k_y]$$

2D Discrete Fourier Transform

Example: Find the DFT of a constant.



2D Discrete Fourier Transform

Example: Find the DFT of a vertical line.

$$f_v[n_x, n_y] = \delta[n_x] = \begin{cases} 1 & n_x = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$F_v[k_x, k_y] = \frac{1}{N_x N_y} \sum_{n_x=0}^{N_x-1} \sum_{n_y=0}^{N_y-1} \delta[n_x]\, e^{-j\left(\frac{2\pi k_x}{N_x} n_x + \frac{2\pi k_y}{N_y} n_y\right)}$$

$$= \frac{1}{N_x N_y} \sum_{n_x=0}^{0} \sum_{n_y=0}^{N_y-1} e^{-j\left(\frac{2\pi k_x}{N_x} 0 + \frac{2\pi k_y}{N_y} n_y\right)} = \frac{1}{N_x N_y} \sum_{n_y=0}^{N_y-1} e^{-j\frac{2\pi k_y}{N_y} n_y}$$

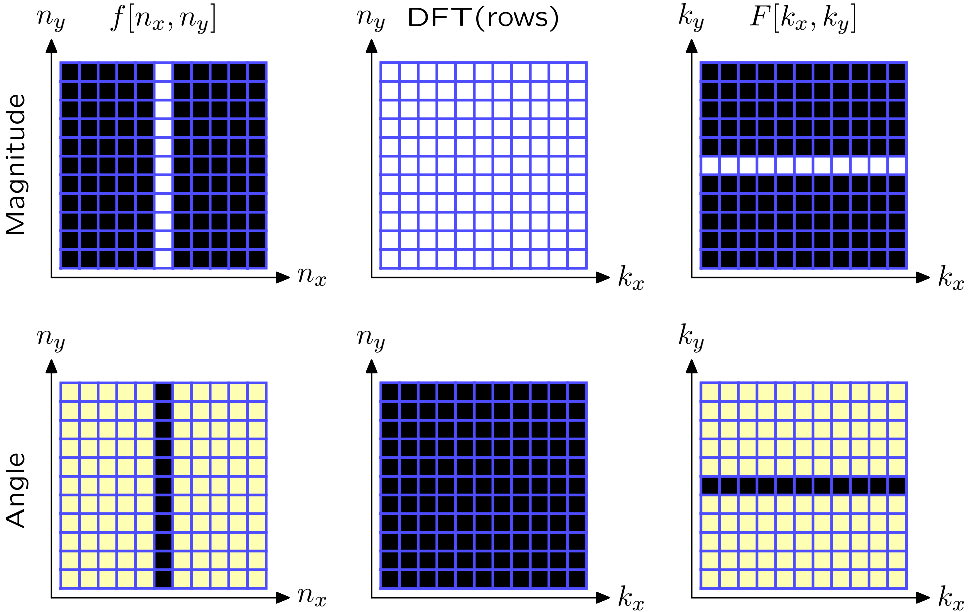
But $\displaystyle\sum_{n_y=0}^{N_y-1} e^{-j\frac{2\pi k_y}{N_y} n_y} = \begin{cases} N_y & k_y = 0 \\ 0 & \text{otherwise} \end{cases}$

$$F_v[k_x, k_y] = \frac{1}{N_x N_y} N_y \delta[k_y] = \frac{1}{N_x} \delta[k_y]$$

$$\delta[n_x] \quad \stackrel{\text{DFT}}{\Longrightarrow} \quad \frac{1}{N_x} \delta[k_y]$$

2D Discrete Fourier Transform

Example: Find the DFT of a vertical line.



2D Discrete Fourier Transform

Example: Find the DFT of a horizontal line.

$$f_h[n_x, n_y] = \delta[n_y] = \begin{cases} 1 & n_y = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$F_h[k_x, k_y] = \frac{1}{N_x N_y} \sum_{n_x=0}^{N_x-1} \sum_{n_y=0}^{N_y-1} \delta[n_y]\, e^{-j\left(\frac{2\pi k_x}{N_x} n_x + \frac{2\pi k_y}{N_y} n_y\right)}$$

$$= \frac{1}{N_x N_y} \sum_{n_x=0}^{N_x-1} \sum_{n_y=0}^{0} e^{-j\left(\frac{2\pi k_x}{N_x} n_x + \frac{2\pi k_y}{N_y} 0\right)} = \frac{1}{N_x N_y} \sum_{n_x=0}^{N_x-1} e^{-j\frac{2\pi k_x}{N_x} n_x}$$

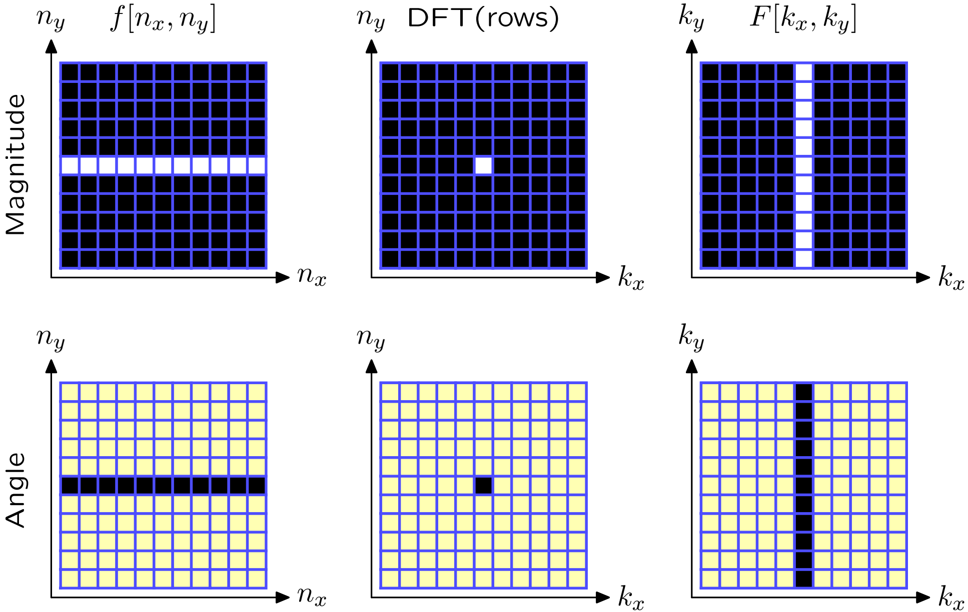
But $\displaystyle\sum_{n_x=0}^{N_x-1} e^{-j\frac{2\pi k_x}{N_x} n_x} = \begin{cases} N_x & k_x = 0 \\ 0 & \text{otherwise} \end{cases}$

$$F_h[k_x, k_y] = \frac{1}{N_x N_y} N_x \delta[k_x] = \frac{1}{N_y} \delta[k_x]$$

$$\delta[n_y] \quad \stackrel{\text{DFT}}{\Longrightarrow} \quad \frac{1}{N_y} \delta[k_x]$$

2D Discrete Fourier Transform

Example: Find the DFT of a horizontal line.



Translating (Shifting) an Image

Effect of image translation (shifting) on its Fourier transform.

Assume that $f_0[n_x, n_y] \overset{\text{DFT}}{\Longrightarrow} F_0[k_x, k_y]$.

Find the 2D DFT of $f_1[n_x, n_y] = f_0[n_x - n_{x0}, n_y - n_{y0}]$

$$F_1[k_x, k_y] = \sum_{kx} \sum_{ky} f_1[n_x, n_y] e^{-j\frac{2\pi kx}{Nx} nx} e^{-j\frac{2\pi ky}{Ny} ny}$$

$$= \sum_{kx} \sum_{ky} f_0[n_x - n_{x0}, n_y - n_{y0}] e^{-j\frac{2\pi kx}{Nx} nx} e^{-j\frac{2\pi ky}{Ny} ny}$$

Let $l_x = n_x - n_{x0}$ and $l_y = n_y - n_{y0}$. Then

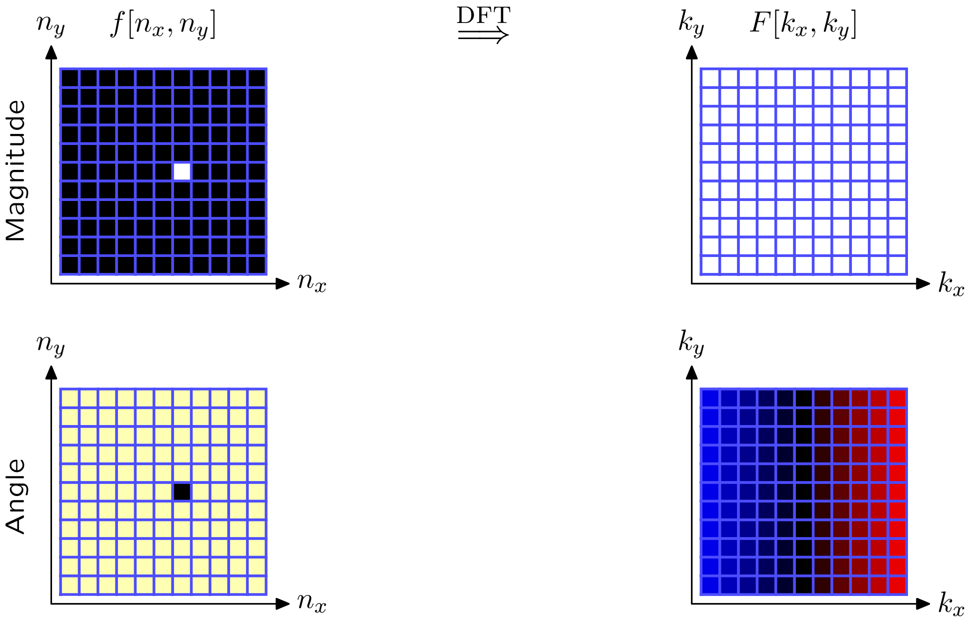
$$F_1[k_x, k_y] = \sum_{lx} \sum_{ly} f_0[l_x, l_y] e^{-j\frac{2\pi kx}{Nx} (lx + n_{x0})} e^{-j\frac{2\pi ky}{Ny} (ly + n_{y0})}$$

$$= e^{-j\frac{2\pi kx}{Nx} n_{x0}} e^{-j\frac{2\pi ky}{Ny} n_{y0}} F_0[k_x, k_y]$$

Translating an image adds linear (in k_x, k_y) **phase** to its transform.

2D Discrete Fourier Transform

Example: Find the DFT of a shifted 2D unit sample.



where blue represents positive phase and red represents negative phase

Using Python

Calculating DFTs is most efficient in NumPy (Numerical Python).

- NumPy arrays are **homogeneous**: their elements are of the same type
- Numpy operators (`+`, `-`, `abs`, `.real`, `.imag`) combine **elements** to create new arrays. e.g., `(f+g)[n]` is `f[n]+g[n]`.
- 2D Numpy arrays can be **indexed by tuples**: e.g., `f[r,c] = f[r][c]`.
- 2D Numpy arrays support **negative indices**: e.g., `f[-1] = f[len(f)-1]`
- 2D indices address **row then column**.

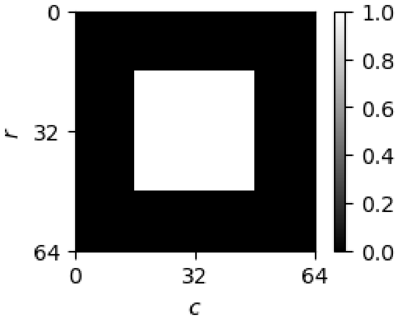
$$\begin{array}{ccccc} f[0,0] & f[0,1] & f[0,2] & f[0,3] & \cdots \\ f[1,0] & f[1,1] & f[1,2] & f[1,3] & \cdots \\ f[2,0] & f[2,1] & f[2,2] & f[2,3] & \cdots \\ f[3,0] & f[3,1] & f[3,2] & f[3,3] & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{array}$$

NumPy indexing is consistent with **linear algebra** (row first then column with rows increasing downward and columns increasing to the right). But it differs from **physical mathematics** (x then y with x increasing to the right and y increasing upward). You may do calculations either way, but row,column is often less confusing.

Numpy Example

Make a white square on a black background.

```
import numpy
from lib6003.image import show_image
f = numpy.zeros((64,64))
for r in range(16,48):
    for c in range(16,48):
        f[r,c] = 1
show_image(f,zero_loc='topleft')
```

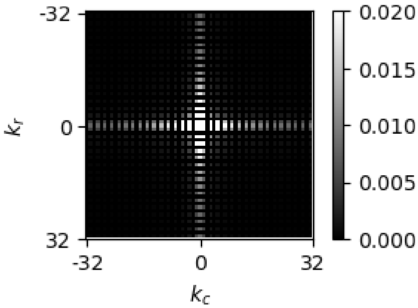


Numpy Example

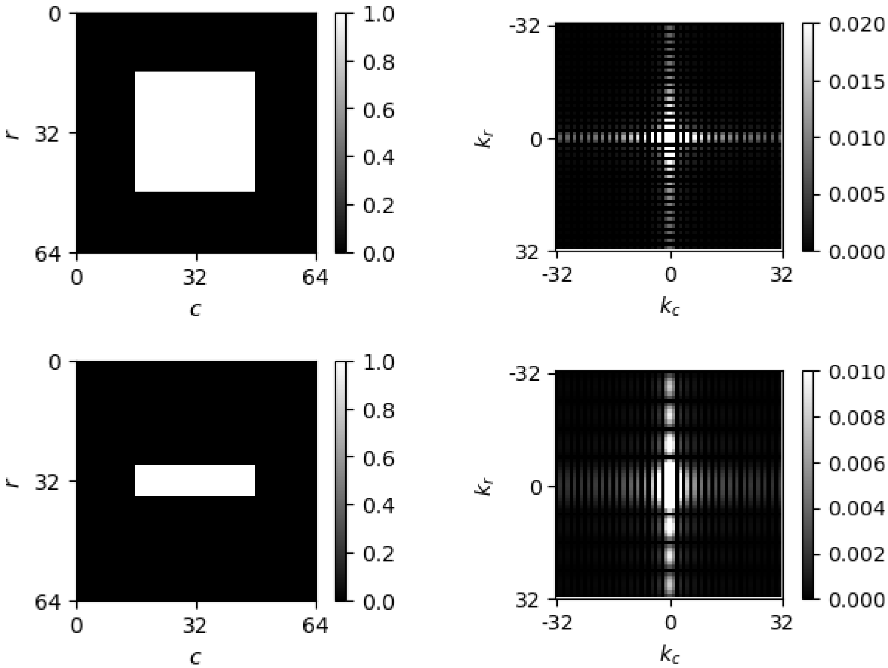
Find the 2D DFT of the square.

```
import numpy
from lib6003.image import show_image
from lib6003.fft import fft2

F = fft2(f)
show_image(numpy.abs(F),zero_loc='center',vmin=0,vmax=0.02)
```

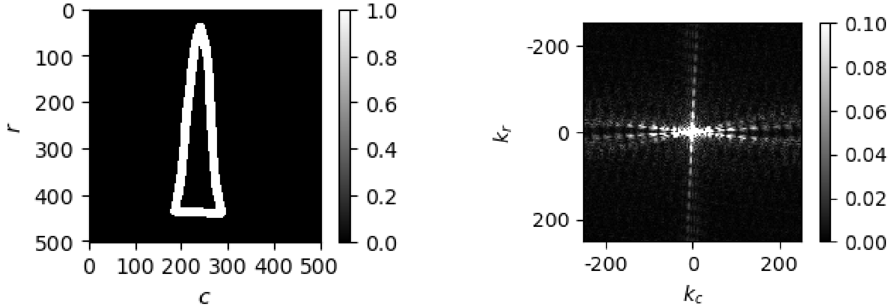


Big and Small



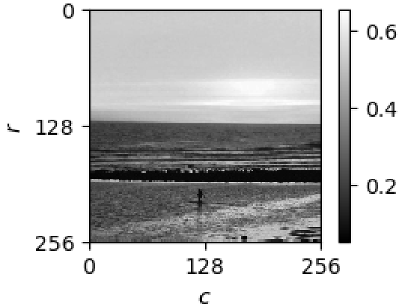
Triangle

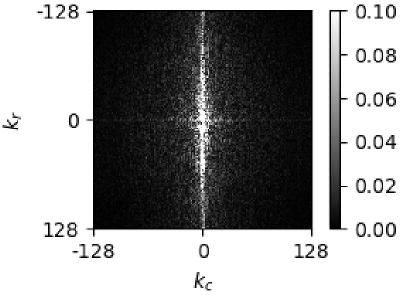
What are the dominant features of the magnitude of the DFT of a triangle?



Ocean

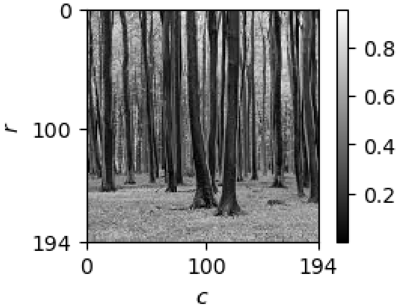
What are the dominant features of the DFT magnitude of an ocean view?

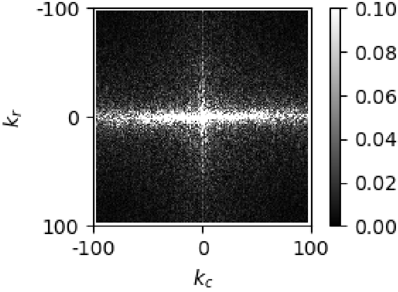




Trees

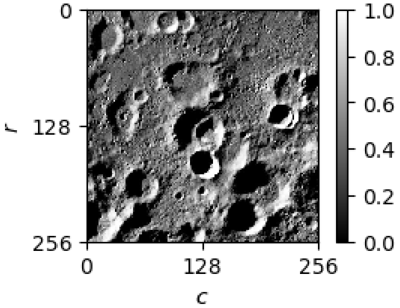
What are the dominant features of the DFT magnitude of these trees?

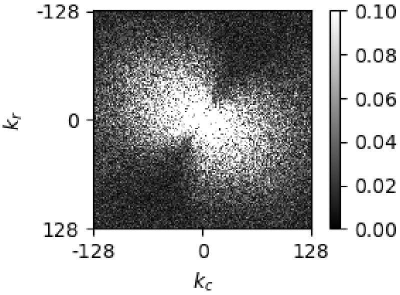




Moon

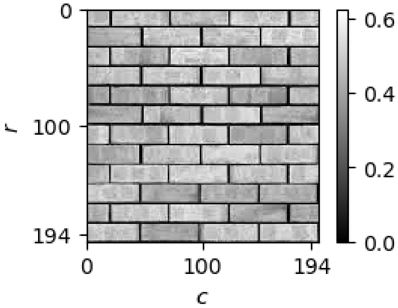
What are the dominant features of the DFT magnitude of the moon?

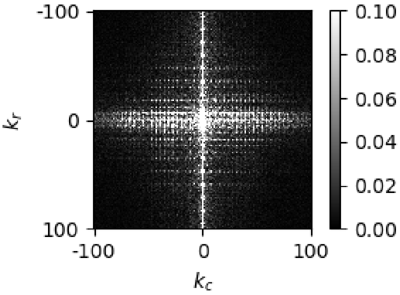




Bricks

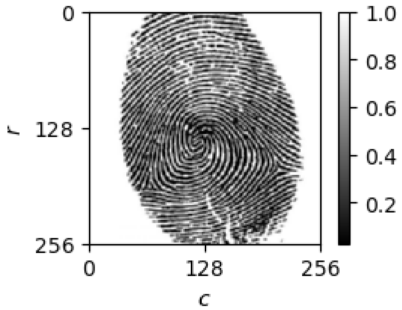
What are the dominant features of the DFT magnitude of this brick wall?

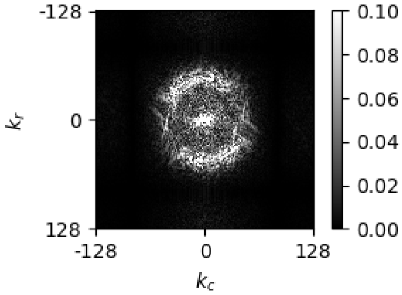




Fingerprint

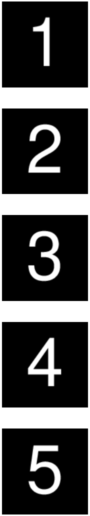
What are the dominant features of the DFT magnitude of this fingerprint?

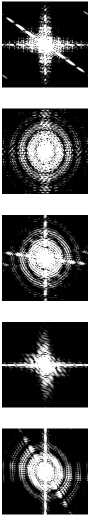




Check Yourself

Which panel on right shows the mag of the DFT of each digit on the left?





Summary

Introduced 2D signal processing.

- generally simple extensions of 1D ideas

Introduced 2D Fourier representations.

- Fourier kernel comprises the sum of an x part and a y part
- basis functions are complex exponentials

Properties of 2D DFT

- transform all of the rows then transform all of the columns
- transform all of the columns then transform all of the rows

Question of the Day

Sketch the magnitude of the 2D Fourier Transform of a checkmark.