6.003 Signal Processing

Week 10, Lecture A:
2D Signal Processing (I): 2D Fourier Representation
Signals: Functions Used to Convey Information

• Signals may have 1 or 2 or 3 or even more independent variables.

A 1D signal has a one-dimensional domain.
We usually think of it as time t or discrete time n.

A 2D signal has a two-dimensional domain.
We usually think of the domains as x and y or \( n_x \) and \( n_y \) (or r and c).
Signals from physical systems are often of continuous domain:
• continuous time – measured in seconds, etc
• continuous spatial coordinates – measured in meters, cm, etc

Computations usually manipulate functions of discrete domain:
• discrete time – measured in samples
• discrete spatial coordinates – measured in pixels
Sampling

Continuous “time” (CT) versus discrete “time” (DT)

\[ x(t) \]

\[ x[n] = x(n\Delta T) \]

\[ \Delta T \text{ (seconds / sample)} = \text{sampling interval} \]
\[ f_s \text{ (samples / second)} = \text{sampling rate} \]

Important for computational manipulation of physical data.
- digital representations of audio signals (as in MP3)
- digital representations of images (as in JPEG)
From Time to Space

So far, our signals have been a function of time: f(t), f[n]

Now, start to consider functions of space: f(x, y), f[r, c]

Our goal is still the same:
• Extract meaningful information from a signal,
• Manipulate information in a signal.
We still resort to Fourier representations for these purposes.

Turn now to development of “frequency domain” representations in 2D.
Fourier Representations
From “Continuous Time” to “Continuous Space.”

1D Continuous-Time Fourier Transform

\[
F(\omega) = \int_{-\infty}^{\infty} f(t) \cdot e^{-j\omega t} \, dt \quad \text{Analysis equation}
\]

\[
f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) \cdot e^{j\omega t} \, d\omega \quad \text{Synthesis equation}
\]

Two dimensional CTFT:

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

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

\(x\) and \(y\) are continuous spatial variables (units: cm, m, etc.)

\(\omega_x\) and \(\omega_y\) are spatial frequencies (units: radians / length)

- integrals \(\rightarrow\) double integrals;
- sum of \(x\) and \(y\) exponents in kernal function.
Fourier Representations

From “Discrete Time” to “Discrete Space.”

1D Discrete-Time Fourier Transform

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

Analysis equation

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

Synthesis equation

Two dimensional DTFT:

\[
F(\Omega_r, \Omega_c) = \sum_{r=-\infty}^{\infty} \sum_{c=-\infty}^{\infty} f[r, c] \cdot e^{-j(\Omega_r r + \Omega_c c)}
\]

\(r\) and \(c\) are discrete spatial variables (units: pixels)

\[
f[r, c] = \frac{1}{4\pi^2} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} F(\Omega_r, \Omega_c) \cdot e^{j(\Omega_r r + \Omega_c c)} \, d\Omega_r \, d\Omega_c
\]

\(\Omega_r\) and \(\Omega_c\) are spatial frequencies (units: radians / pixel)

• sum → double sums; integral → double integrals;
• sum of \(r\) and \(c\) exponents in kernal function.
Fourier Representations

1D DFT to 2D DFT

1D Discrete Fourier Transform

\[ F[k] = \frac{1}{N} \sum_{n=0}^{N-1} f[n] \cdot e^{-j \frac{2\pi k}{N} n} \]  
\text{Analysis equation}

\[ f[n] = \sum_{k=0}^{N-1} F[k] \cdot e^{j \frac{2\pi k}{N} n} \]  
\text{Synthesis equation}

Two dimensional DFT:

\[ F[k_r,k_c] = \frac{1}{RC} \sum_{r=0}^{R-1} \sum_{c=0}^{C-1} f[r,c] \cdot e^{-j\left(\frac{2\pi k_r}{R} r + \frac{2\pi k_c}{C} c\right)} \]  
\(r\) and \(c\) are discrete spatial variables (units: pixels)

\[ f[r,c] = \sum_{r=0}^{R-1} \sum_{c=0}^{C-1} F[k_r,k_c] \cdot e^{-j\left(\frac{2\pi k_r}{R} r + \frac{2\pi k_c}{C} c\right)} \]  
\(k_r\) and \(k_c\) are integers representing frequencies
Orthogonality

DFT basis functions are orthogonal to each other in 1D and 2D.

1D DFT basis functions: \( \phi_k[n] = e^{-j \frac{2\pi k}{N} n} \)

"Inner product" of 1D basis functions: See slide #9 of Lec 3B.

\[
\sum_{n=0}^{N-1} \phi^*_k[n] \phi_l[n] = \sum_{n=0}^{N-1} e^{j \frac{2\pi k}{N} n} \cdot e^{-j \frac{2\pi l}{N} n} = \sum_{n=0}^{N-1} e^{j \frac{2\pi (k-l)}{N} n} = \begin{cases} 
N & \text{if } k = l \\
0 & \text{otherwise}
\end{cases} 
(0 \leq k, l < N)
\]

2D DFT basis functions: \( \phi_{k_r,k_c}[r, c] = e^{-j \frac{2\pi k_r}{R} r} e^{-j \frac{2\pi k_c}{C} c} \)

"Inner product" of 2D basis functions:

\[
\sum_{r=0}^{R-1} \sum_{c=0}^{C-1} \phi^*_{k_r,k_c}[r, c] \phi_{l_r,l_c}[r, c] = \sum_{r=0}^{R-1} \sum_{c=0}^{C-1} \left( e^{j \frac{2\pi k_r}{R} r} \cdot e^{j \frac{2\pi k_c}{C} c} \right) \cdot \left( e^{-j \frac{2\pi l_r}{R} r} \cdot e^{-j \frac{2\pi l_c}{C} c} \right)
\]

\[
= \sum_{r=0}^{R-1} e^{j \frac{2\pi (k_r-l_r)}{R} r} \sum_{c=0}^{C-1} e^{j \frac{2\pi (k_c-l_c)}{C} c} = \begin{cases} 
RC & \text{if } k_r = l_r \text{ and } k_c = l_c \\
0 & \text{otherwise}
\end{cases} 
(0 \leq k_r, l_r < R, 0 \leq k_c, l_c < C)
\]
Python Representation of 2D Images

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** as in lists: e.g., a[-1] = a[len(a)-1]
- 2D indices address row then column:

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

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 be mindful of this difference.
Python Representation of 2D Images

We provide methods in the `lib6003.image` module for saving/loading PNG images:

- `png_read(filename)` loads an image from a file into a NumPy array
- `png_write(array, filename)` saves image data to a file
- `show_image(array)` displays an image

Arrays are indexed in row, column order, and the values are brightness, usually in the range [0, 1]

Example: Make a white square on a black background.

```python
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')
```

We also provide functions for computing 2D DFTs in the `lib6003.fft` module: `fft2` and `ifft2`

In the frequency domain, index 0, 0 into a Numpy array corresponds to $k_r = 0, k_c = 0$ (the DC component).
Check yourself!

The 2D DFT basis functions have the form

$$\phi_{k_r,k_c}[r,c] = e^{-j \frac{2\pi k_r}{R} r} e^{-j \frac{2\pi k_c}{C} c}$$

Which (if any) of the following images show the real part of one of the basis functions $\phi_{k_r,k_c}[r,c]$?

(0,0) is at top left corner, black correspond to lowest value, white correspond to highest value.

What values of $k_r$ and $k_c$ correspond to each basis function?
Check yourself!

The 2D DFT basis functions have the form:

\[ \phi_{k_r,k_c}[r,c] = e^{-j\frac{2\pi k_r}{R} r} e^{-j\frac{2\pi k_c}{C} c} \]

\[ = \cos \left( \frac{2\pi k_r}{R} r + \frac{2\pi k_c}{C} c \right) - j\sin \left( \frac{2\pi k_r}{R} r + \frac{2\pi k_c}{C} c \right) \]

If \( \frac{2\pi k_r}{R} r + \frac{2\pi k_c}{C} c \) is constant, the real and imaginary parts will be constant.

Example: Let \( k_r = 3 \) and \( k_c = -4 \) when \( R = C = 128 \).

Then the exponent is \( \frac{2\pi 3}{128} r - \frac{2\pi 4}{128} c \). This exponent is zero if \( 3r = 4c \).

If the exponent is zero, then cosine is at its peak value of 1.

Thus the real part of the 2D basis function is 1 along the line \( r = \frac{4}{3} c \).

Therefore the real part of the 2D basis function will be 1 along the lines \( r = -\frac{k_c}{k_r} c \).
The 2D DFT basis functions have the form

\[ \phi_{k_r,k_c}[r,c] = e^{-j \frac{2\pi}{R} kr} e^{-j \frac{2\pi}{C} kc} \]

\[ \text{Re}(\phi_{k_r,k_c}[r,c]) = \cos \left( \frac{2\pi}{R} kr + \frac{2\pi}{C} kc \right) \]

Which (if any) of the following images show the real part of one of the basis functions \( \phi_{k_r,k_c}[r,c] \)?  
A and B

(0,0) is at top left corner, black correspond to lowest value, white correspond to highest value.

What values of \( k_r \) and \( k_c \) correspond to each basis function?

A: (4,3) or (-4,-3); B: (3,4) or (-3,-4); C: none; D: none
Fourier Transform Pairs

In 1D, we found that it was useful to know how the transforms of simple shapes looked (for example delta $\rightarrow$ constant), in part because it was often possible to use that understanding to simplify thinking about bigger problems.

The same will be true in 2D!

The rest of today: 2D Fourier analysis of simple shapes.
2D Discrete Fourier Transform

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

\[ f[r, c] = \delta[r]\delta[c] = \begin{cases} 1, & r = 0 \text{ and } c = 0 \\ 0, & \text{otherwise} \end{cases} \]

\[ F[k_r, k_c] = \frac{1}{RC} \sum_{r=0}^{R-1} \sum_{c=0}^{C-1} \delta[r]\delta[c] \cdot e^{-j\left(\frac{2\pi k_r}{R} r + \frac{2\pi k_c}{C} c\right)} \]

\[ = \frac{1}{RC} e^{-j\left(\frac{2\pi k_r}{R} 0 + \frac{2\pi k_c}{C} 0\right)} \]

\[ = \frac{1}{RC} \]

\[ \delta[r]\delta[c] \xrightarrow{\text{DFT}} \frac{1}{RC} \]

1D unit sample

\[ f[n] = \delta[n] = \begin{cases} 1, & n = 0 \\ 0, & \text{otherwise} \end{cases} \]

\[ F[k] = \frac{1}{N} \sum_{n=0}^{N-1} \delta[n] \cdot e^{-j\frac{2\pi k n}{N}} = \frac{1}{N} \]
2D Discrete Fourier Transform

Generally, implement a 2D DFT as a sequence of 1D DFTs:

$$F[k_r, k_c] = \frac{1}{RC} \sum_{r=0}^{R-1} \sum_{c=0}^{C-1} f[r, c] \cdot e^{-j\left(\frac{2\pi k_r r}{R} + \frac{2\pi k_c c}{C}\right)}$$

$$= \frac{1}{R} \sum_{r=0}^{R-1} \left( \frac{1}{C} \sum_{c=0}^{C-1} f[r, c] \cdot e^{-j\frac{2\pi k_c c}{C}} \right) \cdot e^{-j\frac{2\pi k_r r}{R}}$$

- First, obtain the DFT for each row
- Then, take the DFT of each resulting columns

Alternatively, we can start with columns and then do rows just as well.
2D Discrete Fourier Transform

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

\[ f[r, c] \]

Magnitude

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

\[ f[r, c] \]
2D Discrete Fourier Transform

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

\[ f[r, c] \quad \xrightarrow{\text{DFT (rows)}} \quad k_c \]

Magnitude

\[ r \quad \xrightarrow{c} \quad k_c \]

Angle

\[ r \quad \xrightarrow{c} \quad k_c \]
2D Discrete Fourier Transform

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

\[ f[r, c] \]
Example: Find the DFT of a 2D unit sample:

- **Magnitude**
  - **$f[r, c]$**
  - **DFT (rows)**

- **Angle**
  - **$f[r, c]$**
  - **DFT (rows)**
2D Discrete Fourier Transform

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

\[ f[r, c] \rightarrow \text{DFT(rows)} \rightarrow k_c \]

Magnitude

\( r \rightarrow \) \( k_c \)

Angle

\( r \rightarrow \) \( k_c \)
2D Discrete Fourier Transform

Example: Find the DFT of a 2D unit sample:
Example: Find the DFT of a 2D unit sample:
2D Discrete Fourier Transform

Example: Find the DFT of a 2D unit sample:
2D Discrete Fourier Transform

Example: Find the DFT of a 2D constant.

\[ f[r, c] = 1 \]

\[
F[k_r, k_c] = \frac{1}{RC} \sum_{r=0}^{R-1} \sum_{c=0}^{C-1} 1 \cdot e^{-j \left( \frac{2\pi k_r r}{R} + \frac{2\pi k_c c}{C} \right)} = \frac{1}{RC} \sum_{r=0}^{R-1} e^{-j \frac{2\pi k_r r}{R}} \sum_{c=0}^{C-1} e^{-j \frac{2\pi k_c c}{C}}
\]

And we know:

\[
\sum_{c=0}^{C-1} e^{-j \frac{2\pi k_c c}{C}} = \begin{cases} C, & k_c = 0 \\ 0, & \text{otherwise} \end{cases} \quad \sum_{r=0}^{R-1} e^{-j \frac{2\pi k_r r}{R}} = \begin{cases} R, & k_r = 0 \\ 0, & \text{otherwise} \end{cases}
\]

\[
F[k_r, k_c] = \frac{1}{RC} \cdot R \cdot \delta[k_r] \cdot C \cdot \delta[k_c] = \delta[k_c] \delta[k_r]
\]

1D constant

\[ f[n] = 1 \]

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

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

\[
= \begin{cases} 1, & k = 0 \\ 0, & \text{otherwise} \end{cases}
\]

\[ DFT \quad 1 \rightarrow \delta[k_r] \delta[k_c] \]
2D Discrete Fourier Transform

Example: Find the DFT of a constant.

\[ f[r, c] \]
2D Discrete Fourier Transform

Example: Find the DFT of a constant.

$f[r, c]$
2D Discrete Fourier Transform

Example: Find the DFT of a constant.

\[ f[r, c] \]

\[ \text{DFT (rows)} \]

Magnitude

\[ r \rightarrow c \rightarrow k_c \]

\[ r \rightarrow r \]

Angle

\[ r \rightarrow c \rightarrow k_c \]

\[ r \rightarrow r \]
Example: Find the DFT of a constant.
2D Discrete Fourier Transform

Example: Find the DFT of a constant.

$$f[r, c]$$

DFT (rows)

$$F[k_r, k_c]$$
2D Discrete Fourier Transform

Example: Find the DFT of a constant.

\[ f[r, c] \]

DFT(rows)

\[ F[k_r, k_c] \]
2D Discrete Fourier Transform

Example: Find the DFT of a constant.
2D Discrete Fourier Transform

Example: Find the DFT of a constant.
2D Discrete Fourier Transform

Example: Find the DFT of a vertical line.

\[ f[r, c] = \delta[c] = \begin{cases} 1, & c = 0 \\ 0, & \text{otherwise} \end{cases} \]

\[ F[k_r, k_c] = \frac{1}{RC} \sum_{r=0}^{R-1} \sum_{c=0}^{C-1} \delta[c] \cdot e^{-j\left(\frac{2\pi k_r r}{R} + \frac{2\pi k_c c}{C}\right)} = \frac{1}{RC} \sum_{r=0}^{R-1} e^{-j\left(\frac{2\pi k_r r}{R} + \frac{2\pi k_c 0}{C}\right)} = \frac{1}{RC} \sum_{r=0}^{R-1} e^{-j\left(\frac{2\pi k_r r}{R}\right)} \]

And:

\[ \sum_{r=0}^{R-1} e^{-j\frac{2\pi k_r}{R} r} = \begin{cases} R, & k_r = 0 \\ 0, & \text{otherwise} \end{cases} \]

\[ F[k_r, k_c] = \frac{1}{RC} \cdot R \cdot \delta[k_r] = \frac{1}{C} \delta[k_r] \]

\[ \delta[c] \xrightarrow{\text{DFT}} \frac{1}{C} \delta[k_r] \]
2D Discrete Fourier Transform

Example: Find the DFT of a vertical line:

$$f[r, c]$$
Example: Find the DFT of a vertical line:

\( f[r, c] \)
Example: Find the DFT of a vertical line:

$$f[r, c]$$

DFT (rows)

Magnitude

Angle
Example: Find the DFT of a vertical line:
2D Discrete Fourier Transform

Example: Find the DFT of a vertical line:

\[ f[r, c] \rightarrow \text{DFT (rows)} \rightarrow F[k_r, k_c] \]

**Magnitude**

- Input: \( f[r, c] \)
- DFT (rows): \( k_c \)
- Output: \( F[k_r, k_c] \)

**Angle**

- Input: \( f[r, c] \)
- DFT (rows): \( k_c \)
- Output: \( F[k_r, k_c] \)
2D Discrete Fourier Transform

Example: Find the DFT of a vertical line:

\[ f[r, c] \xrightarrow{\text{DFT}} F[k_r, k_c] \]
2D Discrete Fourier Transform

Example: Find the DFT of a horizontal line:

\[ f[r, c] = \delta[r] = \begin{cases} 1, & r = 0 \\ 0, & \text{otherwise} \end{cases} \]

\[ F[k_r, k_c] = \frac{1}{RC} \sum_{r=0}^{R-1} \sum_{c=0}^{C-1} \delta[r] \cdot e^{-j\left(\frac{2\pi k_r r}{R} + \frac{2\pi k_c c}{C}\right)} = \frac{1}{RC} \sum_{c=0}^{C-1} e^{-j\left(\frac{2\pi k_r r}{R} + \frac{2\pi k_c c}{C}\right)} = \frac{1}{RC} \sum_{c=0}^{C-1} e^{-j\left(\frac{2\pi k_c c}{C}\right)} \]

Since:

\[ \sum_{c=0}^{C-1} e^{-j\frac{2\pi k_c c}{C}} = \begin{cases} C, & k_c = 0 \\ 0, & \text{otherwise} \end{cases} \]

\[ F[k_r, k_c] = \frac{1}{RC} \cdot C \cdot \delta[k_c] = \frac{1}{R} \delta[k_c] \]

\[ \delta[r] \xrightarrow{DFT} \frac{1}{R} \delta[k_c] \]
Example: Find the DFT of a horizontal line:

\[ f[r, c] \]
Example: Find the DFT of a horizontal line:
Example: Find the DFT of a horizontal line:

\[ f[r, c] \]  

Magnitude

\[ r \]  

Angle

\[ r \]  

DFT (rows)

\[ k_c \]
2D Discrete Fourier Transform

Example: Find the DFT of a horizontal line:

$f[r, c] \quad DFT(\text{rows})$

Magnitude

Angle
Example: Find the DFT of a horizontal line:

\[ f[r, c] \quad \text{DFT (rows)} \]

Magnitude

\[ r \quad c \quad k_c \]

Angle

\[ r \quad c \quad k_c \]
2D Discrete Fourier Transform

Example: Find the DFT of a horizontal line:

\[ f[r, c] \quad \text{DFT (rows)} \quad F[k_r, k_c] \]
2D Discrete Fourier Transform

Example: Find the DFT of a horizontal line:

\[ f[r, c] \]

\[ \text{DFT(rows)} \]

\[ F[k_r, k_c] \]
2D Discrete Fourier Transform

Example: Find the DFT of a horizontal line:

\[ f[r, c] \]

\[ \text{DFT (rows)} \]

\[ F[k_r, k_c] \]
2D Discrete Fourier Transform

Example: Find the DFT of a horizontal line:

\[ f[r, c] \rightarrow \text{DFT (rows)} \rightarrow F[k_r, k_c] \]
2D Discrete Fourier Transform

Example: Find the DFT of a horizontal line:

\[ f[r, c] \]

\[ \text{DFT (rows)} \]

\[ F[k_r, k_c] \]
2D Discrete Fourier Transform

Example: Find the DFT of a horizontal line:

\[ f[r, c] \quad \xrightarrow{\text{DFT}} \quad F[k_r, k_c] \]
Circularly Translating/Shifting an Image

Effect of image translation/shifting on its Fourier transform.

Assume that \[ f_0[r, c] \xrightarrow{DFT} F_0[k_r, k_c] \]

Find the 2D DFT of \( f_1[r, c] = f_0[(r - r_0) \ mod \ R, (c - c_0) \ mod \ C] \)

\[
F_1[k_r, k_c] = \frac{1}{RC} \sum_{r=0}^{R-1} \sum_{c=0}^{C-1} f_1[r, c] \cdot e^{-j\left(\frac{2\pi k_r r}{R} + \frac{2\pi k_c c}{C}\right)}
\]

Let: \( l_r = (r - r_0) \ mod \ R, l_c = (c - c_0) \ mod \ C. \) Then:

\[
F_1[k_r, k_c] = \frac{1}{RC} \sum_{l_r=0}^{R-1} \sum_{l_c=0}^{C-1} f_0[l_r, l_c] \cdot e^{-j\left(\frac{2\pi k_r (l_r + r_0) \ mod \ R)}{R} + \frac{2\pi k_c (l_c + c_0) \ mod \ C}{C}\right)}
\]

\[= e^{-j\frac{2\pi k_r}{R} r_0} \cdot e^{-j\frac{2\pi k_c}{C} c_0} \cdot \frac{1}{RC} \sum_{l_r=0}^{R-1} \sum_{l_c=0}^{C-1} f_0[l_r, l_c] \cdot e^{-j\left(\frac{2\pi k_r l_r}{R} + \frac{2\pi k_c l_c}{C}\right)} = e^{-j\frac{2\pi k_r}{R} r_0} \cdot e^{-j\frac{2\pi k_c}{C} c_0} \cdot F_0[k_r, k_c] \]

Circularly translating an image adds linear phase to its transform.
2D Discrete Fourier Transform

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

\[
f[r, c] = \delta[r]\delta[c] \Rightarrow F[k_r, k_c] = \frac{1}{RC}
\]
2D Discrete Fourier Transform

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

\[ f[r, c] = \delta[r]\delta[c - 1] \Rightarrow F[k_r, k_c] = e^{-j\frac{2\pi k_c}{c} \cdot 1} \frac{1}{RC} \]
2D Discrete Fourier Transform

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

\[
f[r, c] = \delta[r] \delta[c - 2] \quad \Rightarrow \quad F[k_r, k_c] = e^{-j \frac{2\pi k_c}{c} \cdot 2} \frac{1}{RC}
\]
2D Discrete Fourier Transform

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

\[
f[r, c] = \delta[r] \delta[c - 3] \quad \rightarrow \quad F[k_r, k_c] = e^{-j\frac{2\pi k_c}{c - 3}} \frac{1}{RC}
\]
2D Discrete Fourier Transform

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

\[ f[r, c] = \delta[r] \delta[c] \rightarrow \mathcal{F}[k_r, k_c] = \frac{1}{RC} \]
2D Discrete Fourier Transform

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

\[ f[r, c] = \delta[r + 1] \delta[c] \Rightarrow F[k_r, k_c] = e^{-j \frac{2\pi k_r}{R} \left( \frac{1}{1} \right)} \frac{1}{RC} \]
Example: Find the DFT of a shifted 2D unit sample:

\[ f[r, c] = \delta[r + 2]\delta[c] \rightarrow F[k_r, k_c] = e^{-j\frac{2\pi kr}{R}(-2)} \frac{1}{RC} \]
2D Discrete Fourier Transform

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

\[
f[r, c] = \delta[r + 3] \delta[c] \quad \xrightarrow{\text{DFT}} \quad F[k_r, k_c] = e^{-j\frac{2\pi k_r}{R}(3)} \frac{1}{RC}
\]
Summary

Introduced 2D signal processing:
• Mostly simple extensions of 1D ideas
• Some small differences

Introduced 2D Fourier representations:
• Fourier kernel comprises the sum of an $x$ part and a $y$ part (or $r, c$)
• Basis functions look like sinusoids turned at angles determined by the ratio of $k_c$ to $k_r$.

Multiple 2D Fourier Transform pairs:
• 2D unit sample $\Rightarrow$ 2D constant
• 2D constant $\Rightarrow$ 2D unit sample
• vertical line $\Rightarrow$ horizontal line
• horizontal line $\Rightarrow$ vertical line

(Circularly) translating an image does not change the magnitude but adds linear phase to its transform.