Review: Fourier Representations

One dimensional DTFT:
\[ F(\Omega) = \sum_{n=-\infty}^{\infty} f[n] e^{-j\Omega n} \]
\[ f[n] = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\Omega) e^{j\Omega n} d\Omega \]

Two dimensional DTFT:
\[ F(\Omega_r, \Omega_c) = \sum_{r=-\infty}^{\infty} \sum_{c=-\infty}^{\infty} f[r,c] e^{-j(\Omega_r r + \Omega_c c)} \]
\[ f[r,c] = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(\Omega_r, \Omega_c) e^{j(\Omega_r r + \Omega_c c)} d\Omega_r d\Omega_c \]

\( r \) and \( c \) are discrete spatial variables (units: pixels)
\( \Omega_r \) and \( \Omega_c \) are spatial frequencies (units: radians / pixel)

Review: Fourier Representations

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

Two dimensional DFT:
\[ F[k_r, k_c] = \frac{1}{RC} \sum_{r=0}^{R-1} \sum_{c=0}^{C-1} f[r, c] e^{-j\left(\frac{2\pi}{R} k_r r + \frac{2\pi}{C} k_c c\right)} \]
\[ f[r, c] = \sum_{k_r=0}^{R-1} \sum_{k_c=0}^{C-1} F[k_r, k_c] e^{j\left(\frac{2\pi}{R} k_r r + \frac{2\pi}{C} k_c c\right)} \]
Review: 2D DFT

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

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

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

first take DFTs of rows

then take DFTs of resulting columns

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

Simple Shapes

[Square]

[Rectangle]

[Line]

[Diagonal Line]
**Rotation of Images**

Rotating an image rotates its Fourier transform by the same angle. We can describe the CTFT relation

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

in polar coordinates by expressing points \((x, y)\) in space as \((r, \theta)\) and points \((\omega_x, \omega_y)\) in the frequency plane as \((\omega, \phi)\). Then

\[ \omega_x x + \omega_y y = \omega \cos \phi r \cos \theta + \omega \sin \phi r \sin \theta = \omega r \cos(\phi - \theta) \]

(which directly follows from the dot product relation) so that

\[ F_2(\omega, \phi) = \int \int f_r(r, \theta)e^{j\omega r \cos(\phi \theta)} r \, dr \, d\theta \]

where \(f_r(r, \theta)\) and \(F_r(\omega, \phi)\) are polar equivalents of \(f(x, y)\) and \(F(\omega_x, \omega_y)\). Rotating the image \(f_r\) by \(\psi\) results in a new image \(f_2(r, \theta) = f_r(r, \theta - \psi)\),

\[
F_2(\omega, \phi) = \int \int f_r(r, \theta - \psi)e^{-j\omega r \cos(\phi \theta)} r \, dr \, d\theta \\
= \int \int f_r(r, \lambda)e^{-j\omega r \cos(\phi \lambda - \psi)} r \, dr \, d\lambda = F_r(\omega, \phi - \psi)
\]

**Check Yourself!**

Below is a perfect isosceles triangle. What will a plot of its DFT magnitudes look like?

![Image of a perfect isosceles triangle](image)

**Check Yourself!**

Below is a picture of the ocean. What will a plot of its DFT magnitudes look like?

![Image of the ocean](image)
Magnitude and Phase

So far, we have looked at DFT coefficient magnitudes. What role do the DFT magnitudes play in perception? What role do the DFT phases play? Consider the following image:

The magnitude has some clear structure, but the phase is seemingly random. How does changing each affect the image we see in the spatial domain?

Trees image reconstructed with:

- DFT magnitudes of original, all DFT phases set to 0
- DFT phases of original, all DFT magnitudes set to 1
**Visual Perception of Phase**

Why are images so sensitive to phase?

![Waveforms](waveforms.png)

All Fourier components must have correct phase to preserve an edge. Changing the phase of just one component can have a drastic effect on an image.

Magnitude controls the amplitude of each wave, but phase has a large effect on *interference patterns* (e.g., dark regions correspond to the “troughs” of basis functions lining up).

**Science Experiment**

Consider the following 2 images:

![Images](images.png)

What will happen if we create a new image using the DFT magnitudes of the image on the left, and the DFT phases of the image on the right?

**Magnitude and Phase**

Phase plays a large role in perception of images. What about music?
After quiz 2, we’ll look closely at filtering of 2D signals. This is similar to how we thought about filtering of 1D signals:

If the system is linear and shift-invariant, with a unit-sample response \( h[r,c] \), then we can think of the output of the system in response to an arbitrary input \( f[r,c] \) as a convolution: \( (f * h)[r,c] \).

Convolution in space is equivalent to multiplication in frequency.

\[
f[r,c] = (f_a * f_b)[r,c] = \sum_{m_r=-\infty}^{\infty} \sum_{m_c=-\infty}^{\infty} f_a[m_r,m_c]f_b[r-m_r,c-m_c] e^{-j\Omega_r m_r} e^{-j\Omega_c m_c}
\]

Discrete-Time Fourier Transform (all sums are over \( -\infty, \infty \)):

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

\[
= \sum_{m_r} \sum_{m_c} \sum_{r} \sum_{c} f_a[m_r,m_c]f_b[r-m_r,c-m_c] e^{-j\Omega_r m_r} e^{-j\Omega_c m_c}
\]

\[
= \sum_{m_r} \sum_{m_c} f_a[m_r,m_c] \sum_{r} \sum_{c} f_b[r-m_r,c-m_c] e^{-j\Omega_r m_r} e^{-j\Omega_c m_c}
\]

\[
= \sum_{m_r} \sum_{m_c} f_a[m_r,m_c] e^{-j\Omega_r m_r} e^{-j\Omega_c m_c} \sum_{r} \sum_{c} f_b[r,c] e^{-j\Omega_r r} e^{-j\Omega_c c}
\]

\[
= F_a(\Omega_r,\Omega_c) F_b(\Omega_r,\Omega_c)
\]
Example: Fishing

Consider the following image:

What output is produced by the following code?

```python
kernel = numpy.zeros((61, 61))
kern[8, -16] = 1

fish = png_read('fish.png')
new = ifft2(fft2(fish) * fft2(kernel))
out = png_write(new, 'output.png')
```

Summary

Today:
- 2D frequency representations
- 2D convolution

Recitation:
- More on 2D convolution

Next week:
- Quiz week!

Then:
- Convolution, filtering, deconvolution