Deconvolution

- convolution / filtering
- multiplication
- separable functions
- rotations
- circular symmetry
- deconvolution
Quiz 2

Tuesday, November 13, from 3pm to 5pm in **room 34-501**.

No lecture on November 13.

The exam is closed book. No electronic devices. You may use two 8.5x11” sheets of notes (front and back).

Coverage: lectures, labs, recitations, and homeworks up to and including November 12.

Practice quiz is posted.
Assume that $F[k]$ is the product of $F_a[k]$ times $F_b[k]$. Find an expression for $f[n]$ in terms of $f_a[n]$ and $f_b[n]$. 

\[
 f[n] = \sum_{k=0}^{N-1} F[k] e^{j \frac{2\pi k}{N} n} \\
= \sum_{k=0}^{N-1} F_a[k] F_b[k] e^{j \frac{2\pi k}{N} n} \\
= \sum_{k=0}^{N-1} F_a[k] \left( \frac{1}{N} \sum_{m=0}^{N-1} f_b[m] e^{-j \frac{2\pi k}{N} m} \right) e^{j \frac{2\pi k}{N} n} \\
= \frac{1}{N} \sum_{m=0}^{N-1} f_b[m] \sum_{k=0}^{N-1} F_a[k] e^{j \frac{2\pi k}{N} (n-m)} \tag{\text{would be } f_a[n-m] \text{ if } 0 \leq n-m < N} \\
= \frac{1}{N} \sum_{m=0}^{N-1} f_b[m] f_a[(n-m) \mod N] \equiv \frac{1}{N} \left( f_a \ast f_b \right)[n]
\]
Superposition View of Conventional Convolution

\[(fa \ast fb)[n] = \sum_{m=0}^{N-1} fb[m] fa[n-m]\]

Superposition View of Circular Convolution

\[
(f_a \star f_b)[n] = \sum_{m=0}^{N-1} f_b[m] f_a[(n - m) \mod N]
\]

\[
= f_b[0] f_a[n \mod N] + f_b[1] f_a[(n-1) \mod N] + f_b[2] f_a[(n-2) \mod N]
\]
The parts of the conventional convolution that would fall outside the DFT window “alias” to points inside the DFT window.

\[
(f_a * f_b)
\]

\[
(f_a \circledast f_b)
\]
2D Convolution: Conventional versus Circular

The same sort of differences result for convolution of images.

Conventional convolution: direct form (space domain) or DTFT

Circular convolution: DFT
The output of conventional convolution can be bigger than the input, while that of circular convolution aliases to the same size as the input.
Zero Padding

Differences between conventional and circular convolution can be reduced by padding the signal (1D or 2D) with a neutral border.

Here “neutral” is black.

In more natural images, a gray shade may introduce fewer artifacts. Could automatically choose an appropriate color by extending edge conditions.
Check Yourself

Not all convolutions blur.

\[
\begin{bmatrix}
-1 & 0 & 1 \\
-2 & 0 & 2 \\
-1 & 0 & 1 \\
\end{bmatrix}
\]

Which image below was generated as shown above?

1. A
2. B
3. C
4. none
Not all convolutions blur.

\[
\begin{bmatrix}
  -1 & 0 & 1 \\
  -2 & 0 & 2 \\
  -1 & 0 & 1 \\
\end{bmatrix}
\]

Which image below was generated as shown above? 2. B

Separable Functions

Separable functions of two (Cartesian) dimensions can be written as the product of two functions of one dimension.

\[ f(x, y) = f_x(x)f_y(y) \]

The Fourier transform of a separable function is separable.

\[
F(\omega_x, \omega_y) = \int \int f(x, y)e^{-j\omega_xx}e^{-j\omega_yy} \, dx \, dy \\
= \int \int f_x(x)f_y(y)e^{-j\omega_xx}e^{-j\omega_yy} \, dx \, dy \\
= \int \left( \int f_x(x)e^{-j\omega_xx} \, dx \right)f_y(y)e^{-j\omega_yy} \, dy \\
= \int \left( F_x(\omega_x) \right)f_y(y)e^{-j\omega_yy} \, dy \\
= F_x(\omega_x) \int f_y(y)e^{-j\omega_yy} \, dy \\
= F_x(\omega_x)F_y(\omega_y)
\]

\[ f_x(x)f_y(y) \overset{\text{CTFT}}{\leftrightarrow} F_x(\omega_x)F_y(\omega_y) \]
The Fourier transform of a separable function is separable.

\[ f_x(x)f_y(y) \overset{\text{CTFT}}{\iff} F_x(\omega_x)F_y(\omega_y) \]

But shouldn’t multiplication in time \( \rightarrow \) convolution in frequency?

\[ f_x(x)f_y(y) \overset{\text{CTFT}}{\iff} F_x(\omega_x) \ast F_y(\omega_y) \]
Separable Functions

The Fourier transform of a separable function is separable.

\[ f_x(x)f_y(y) \overset{\text{CTFT}}{\Leftrightarrow} F_x(\omega_x)F_y(\omega_y) \]

But shouldn’t multiplication in time \( \rightarrow \) convolution in frequency?

\[ f_x(x)f_y(y) \overset{\text{CTFT}}{\Leftrightarrow} F_x(\omega_x) * F_y(\omega_y) \]

We should convolve a 2D function with a 2D function to get a 2D function. If \( f_x(x) \) represents a 2D function, then that function is constant in the \( y \) direction. Then

\[ f_x(x) \overset{\text{2D-FT}}{\Leftrightarrow} F_x(\omega_x)\delta(\omega_y) \]

Similarly

\[ f_y(y) \overset{\text{2D-FT}}{\Leftrightarrow} F_y(\omega_y)\delta(\omega_x) \]

Then

\[ F(\omega_x, \omega_y) = \left( F_x(\omega_x)\delta(\omega_y) \right) * \left( F_x(\omega_y)\delta(\omega_x) \right) \]
Separable Functions

The convolution filter in the previous example was separable.

Let

\[
\begin{align*}
    f_x[n_x] & \quad f_y[n_y] \\
    n_x & \quad n_y
\end{align*}
\]

\[
\begin{align*}
    f_x[n_x]f_y[n_y] \quad & \quad 2D-FT \quad F_x(\Omega_x)F_y(\Omega_y) \\
    F_x(\Omega_x) = j2 \sin \Omega_x \\
    F_y(\Omega_y) = 2 + 2 \cos \Omega_y \\
    F(\Omega_x, \Omega_y) = j2 \sin \Omega_x(2 + 2 \cos \Omega_y)
\end{align*}
\]
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

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

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 \(f_2(r, \theta) = f_r(r, \theta - \psi)\),

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

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

Example: Find the DFT of a cosine wave.

\[ f[n_x, n_y] \]

Magnitude

Angle
Rotation of Images

Example: Find the DFT of a cosine wave.

\[ f[n_x, n_y] \]

Magnitude

Angle

DFT(rows)

\[ n_x \quad \text{DFT(rows)} \quad n_y \]

\[ n_x \quad k_x \]

\[ n_y \quad k_x \]
Rotation of Images

Example: Find the DFT of a cosine wave.
Rotation of Images

Example: Find the DFT of a cosine wave.
### Rotation of Images

Example: Find the DFT of a cosine wave.

\[ f[n_x, n_y] \]

Magnitude

Angle

DFT(rows)

\[ k_x \]

\[ n_x \]

\[ n_y \]
Rotation of Images

Example: Find the DFT of a cosine wave.
Rotation of Images

Example: Find the DFT of a cosine wave.

\[
\begin{align*}
ny & \quad f[n_x, n_y] \\
\text{Magnitude} & \quad n_y \quad \text{DFT(rows)} \\
\text{Angle} & \quad n_y \\
n_x & \quad k_x \\
n_x & \quad k_x
\end{align*}
\]
Rotation of Images

Example: Find the DFT of a cosine wave.

\( f[n_x, n_y] \)

\( n_y \)  \( n_x \)

Magnitude

\( n_y \)  \( k_x \)

DFT(rows)

\( n_y \)  \( n_x \)

Angle
Rotation of Images

Example: Find the DFT of a cosine wave.
Rotation of Images

Example: Find the DFT of a cosine wave.

Magnitude

Angle
Rotation of Images

Example: Find the DFT of a cosine wave.

\[ f[n_x, n_y] \]
Rotation of Images

Example: Find the DFT of a cosine wave.
Rotation of Images

Example: Find the DFT of a cosine wave.
Rotation of Images

Example: Find the DFT of a cosine wave.

\[ f[n_x, n_y] \]

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

\[ F[k_x, k_y] \]
Rotation of Images

Example: Find the DFT of a cosine wave.

\[ f[n_x, n_y] \]

\[ n_y \quad \text{DFT(rows)} \quad k_y \quad F[k_x, k_y] \]

\[ n_x \quad n_y \quad k_x \quad k_y \]

Magnitude

Angle

\[ f[n_x, n_y] \]

\[ n_y \quad \text{DFT(rows)} \quad k_y \quad F[k_x, k_y] \]

\[ n_x \quad n_y \quad k_x \quad k_y \]
Rotation of Images

Example: Find the DFT of a cosine wave.

\[ f[n_x, n_y] \]

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

\[ k_y \quad F[k_x, k_y] \]

Magnitude

Angle
Rotation of Images

Example: Find the DFT of a cosine wave.
Rotation of Images

Example: Find the DFT of a cosine wave.
Example: Find the DFT of a cosine wave.
Rotation of Images

Example: Find the DFT of a cosine wave.
Rotation of Images

Example: Find the DFT of a cosine wave.

\[ f[n_x, n_y] \]

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

\[ F[k_x, k_y] \]
Rotation of Images

Example: Find the DFT of a cosine wave.
Rotation of Images

Example: Find the DFT of a cosine wave.
Rotation of Images

Example: Find the DFT of a cosine wave.

\( f[n_x, n_y] \)

Magnitude

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

\( F[k_x, k_y] \)

Angle
Rotation of Images

Example: Find the DFT of a cosine wave.
Rotation of Images

Example: Find the DFT of a rotated cosine wave.

\[ f[n_x, n_y] \]

Magnitude

Angle
Rotation of Images

Example: Find the DFT of a rotated cosine wave.
Example: Find the DFT of a rotated cosine wave.
Rotation of Images

Example: Find the DFT of a rotated cosine wave.
Rotation of Images

Example: Find the DFT of a rotated cosine wave.
Example: Find the DFT of a rotated cosine wave.
Rotation of Images

Example: Find the DFT of a rotated cosine wave.
Rotation of Images

Example: Find the DFT of a rotated cosine wave.

$$f[n_x, n_y]$$

Magnitude

$$\text{DFT(rows)}$$

Angle
Rotation of Images

Example: Find the DFT of a rotated cosine wave.

\[ f[n_x, n_y] \]

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

\[ n_x \quad k_x \]

\[ n_x \quad n_y \quad n_x \quad n_y \]

Magnitude

Angle
Rotation of Images

Example: Find the DFT of a rotated cosine wave.
Rotation of Images

Example: Find the DFT of a rotated cosine wave.

\[ n_y \quad f[n_x, n_y] \]

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

Magnitude

\[ n_x \quad k_x \]

Angle

\[ n_x \quad k_x \]
Rotation of Images

Example: Find the DFT of a rotated cosine wave.
Rotation of Images

Example: Find the DFT of a rotated cosine wave.

\[ f[n_x, n_y] \]

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

\[ F[k_x, k_y] \]
Rotation of Images

Example: Find the DFT of a rotated cosine wave.
Rotation of Images

Example: Find the DFT of a rotated cosine wave.

\[ f[n_x, n_y] \] \hspace{1cm} DFT(\text{rows}) \hspace{1cm} F[k_x, k_y]
Rotation of Images

Example: Find the DFT of a rotated cosine wave.

\[ f[n_x,n_y] \rightarrow n_y \text{ DFT (rows)} \rightarrow k_y \]

\[ F[k_x,k_y] \]

Magnitude

Angle
Rotation of Images

Example: Find the DFT of a rotated cosine wave.
Rotation of Images

Example: Find the DFT of a rotated cosine wave.

Magnitude

$ny \quad f[nx, ny]$  $ny \quad \text{DFT(rows)}$  $ky \quad F[kx, ky]$

Angle

$ny \quad \text{DFT(rows)}$  $ny \quad F[kx, ky]$
Rotation of Images

Example: Find the DFT of a rotated cosine wave.
Rotation of Images

Example: Find the DFT of a rotated cosine wave.
Rotation of Images

Example: Find the DFT of a rotated cosine wave.
Rotation of Images

Example: Find the DFT of a rotated cosine wave.
Rotation of Images

Example: Find the DFT of a rotated cosine wave.
Rotation of Images

Example: Find the DFT of a rotated cosine wave.
Rotation of Images

Example: Find the DFT of a rotated cosine wave.
Rectangular Versus Circular Symmetry

A 2D lowpass filter that passes frequencies in the range $-\omega_c < \omega_x < \omega_c$ and $-\omega_c < \omega_y < \omega_c$ is separable and mathematically convenient.

However, the responses of such filters may be surprising.

Describe an important consequence of this square symmetry for

$$f(x, y) = \cos (0.85 \omega_c (x+y))$$

whose 2D DTFT is illustrated by the red $\times$’s above.
Deconvolving as Inverse Filtering

Since (circular) convolution by $h[n_x, n_y]$ is equivalent to filtering by $H[k_x, k_y]$, we can deconvolve by inverse filtering.

For example, if $h[n_x, n_y]$ represents blurring by convolution

$$f_o[n_x, n_y] = (f_i * h)[n_x, n_y]$$

then this blurring can be equivalently thought of as multiplication in the frequency domain:

$$F_o[k_x, k_y] = F_i[k_x, k_y]H[k_x, k_y]$$

This suggests that if we knew the blurring function, we could recover the original image from the blurred image by inverse filtering

$$F_i[k_x, k_y] = F_o[k_x, k_y]/H[k_x, k_y]$$

provided $H[k_x, k_y] \neq 0$!
Deblurring by Inverse Filtering

An original image was blurred using a gaussian kernel and then de-blurred by inverse filtering.

The inverse filtered image is nearly identical to the original.

This is not surprising. Inverse filtering is equivalent to dividing by the $H[k_n, k_y]$, which exactly compensates for the blurring that results when the image is convolved with $h[n_x, n_y]$.

In practice, we don’t generally have access to the exact blurring function. Differences can lead to errors.
Sources of Noise

Noise processes can also lead to errors.

**Shot (Poisson) noise:** The brightness of a conventional image results from the superposition of photons. Statistical variation in the number of photons gives rise to variability in image brightness. This is most important for relatively dark images (as in telescopy). The average brightness of a pixel can be equivalent to that of \( \frac{1}{10} \) of a photon. However, every instance of that pixel must contain an integer number of photons – usually 0 or 1, but never \( \frac{1}{10} \).

**Gaussian noise:** A variety of processes generate additive noise that can have a Gaussian distribution. One example is electronic noise that is associated with amplification of signals (as when the output of a photosensor is amplified prior to digitization).

**Quantization:** To reduce the number of bits used to transmit and/or store an image, the signals associated with each pixel are often quantized. For example, 8-bit representations are popular for conventional images.
Deblurring by Inverse Filtering

Inverse filtering is less effective when noise is added after blurring.

The degradation results because the "gain" of the inverse filter is largest where the signal is the smallest – which is precisely where the added noise has its largest effect.
Noise Compensation

We can mitigate noise degradation by limiting the maximum gain of the inverse filter.
Deblurring by Inverse Filtering

Here is a second example of deblurring by inverse filtering.

original  blurred  inverse filtered

As before, inverse filtering is effective because dividing by $H[k_x, k_y]$ exactly compensates for blurring by convolution by $h[n_x, n_y]$. 
Deblurring by Inverse Filtering

As before, adding noise limits the effectiveness of inverse filtering.

original

inverse filtered
Noise Compensation

Limiting the maximum gain of the inverse filter results in more effective deblurring (right panel).

Here the noise compensated inverse filter is effective in unblurring a nature images (as opposed to previous binary image of simple shapes).
Summary

2D convolution and filtering using the DFT.

**Convolution / Filtering:**
\[
\frac{1}{N_x N_y} \left( f_a \ast f_b \right) \xrightarrow{\text{DFT}} F_a \times F_b
\]

**Multiplication:**
\[
f_a \times f_b \xrightarrow{\text{DFT}} F_a \ast F_b
\]

**Separable Functions:**
\[
fx(x) fy(y) \xrightarrow{2\text{D-FT}} Fx(\omega_x) Fy(\omega_y)
\]

**Rotations:**
\[
\text{if } f(r, \theta) \xrightarrow{2\text{D-FT}} F(\omega, \phi)
\]
\[
\text{then } f(r, \theta - \psi) \xrightarrow{2\text{D-FT}} F(\omega, \phi - \psi)
\]

**Circularly Symmetric Filters**

**Deconvolution**