Frequencies
Outline

• Logistics (HW due Sat)

• Frequency analysis
Logistics

• http://www.ics.uci.edu/~dramanan/teaching/cs216_spring15/project.html
Motivation

Representing a signal as a linear combination of impulse functions

\[ \text{signal} = \ldots + \text{impulse} + \text{impulse} + \text{impulse} + \ldots \]

..allows us to characterize linear-shift invariant (LSI) systems by their impulse response

What about other basis functions?

\[ \text{signal} = .1 \ast \text{function} + .3 \ast \text{function} + \ldots \]
1. Efficient representation

Real-world signals tend to be smooth and slow-changing, meaning we only need to encode low-frequency sinusoids (basis of jpeg compression)

2. Sinusoids are eigenfunctions of LSI (LTI) systems

What’s the eigenvalue?
Eigenfunctions of LTI systems

To compute output for a N-length input, multiply vector of N basis coefficients and vector of N system eigenvalues.

What’s complexity of this operation?

O(N) if we ignore conversion into/from sinusoidal basis
O(N^2) with it (though we can get to O(NlogN) with tricks)
Complex exponentials

\[ e^{j\theta} = \cos \theta + j \sin \theta \]

Sometimes “i” instead of “j”

\[ e^{j\theta} = \overline{(\cos \theta + j \sin \theta)} \]
\[ = \cos \theta - j \sin \theta \]

Complex conjugate
Discrete Fourier Transform (DFT)

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

N-long sequence \( \rightarrow \) N-long sequence

How much is the conversion?
Discrete Fourier Transform (DFT)

\[
F[k = 0] = \frac{1}{N} \sum_{u=0}^{N} f[u]
\]

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

\ldots

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

\ldots

\[
F[k = N] = ?
\]
Linear transformation
Complex matrix

\[
F[m,n] = f[k,l] \cdot e^{-\pi i km/M} + \ldots
\]

\[
f[u] \xrightarrow{\text{f}[k]} F[k]
\]

N-long sequence \quad N-long sequence

How do we go the other direction?
Self-inverting transforms

\[ F = Uf \]
\[ (U^*)U = I \]
\[ (U^*)F = f \]
Inverse DFT

\[ f[u] = \sum_{k=0}^{N-1} F[k] e^{j2\pi ku/N} \quad \text{(IDFT)} \]

\[ F[k] = \frac{1}{N} \sum_{u=0}^{N-1} f[u] e^{-j2\pi ku/N} \quad \text{(DFT)} \]

\[ F[k] \rightarrow f[u] \]

N-long sequence \quad N-long sequence

Sometimes the 1/N is moved or factored (really need \sqrt{N} to make transform self-inverting)
Frequency Spectra
Properties of DFT

- Periodic: $F[k+N] = F[k]$. (we only need to encode $N$ values; often use $k = [-N/2…N/2]$)

- Symmetric for real-valued $f[u]$: $F[k] = F[-k]^*$ (we only really need $N/2$ values)

- Eigenfunction property: the DFT of the convolution of two functions is the product of their DFTs (convolution in time domain is multiplication in frequency domain)

- DFT of a Gaussian with variance $\sigma$ is a another Gaussian with variance $1/\sigma$
Magnitude and phase

\[ F[k] = \frac{1}{N} \sum_{u=0}^{N-1} f[u] e^{-j2\pi ku} \]

How do we interpret the real and imaginary parts of \( F = a + jb \)?

\[ a \cos\left(\frac{-2\pi ku}{N}\right) + jb \sin\left(\frac{-2\pi ku}{N}\right) \]
Magnitude and phase

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

How do we interpret the real and imaginary parts of $F = a + j b$?

$$r e^{\phi} e^{-\frac{2\pi k u}{N}} = r e^{-\frac{2\pi k u}{N}} + \phi$$

$$r \cos\left( -\frac{2\pi k u}{N} + \phi \right) + r \sin\left( -\frac{2\pi k u}{N} + \phi \right)$$
Proof of convolution theory

Sketch; show that complex exponentials are eigenvalues of LSI systems

\[ g[u] = \sum_v h[v] f[u - v], \quad f[u] = e^{j\omega u} \]

\[ = e^{j\omega u} \sum_v h[v] e^{-j\omega v} \]

\[ = e^{j\omega u} H(w) \quad \text{where} \quad H(w) = \sum_v h[v] e^{-j\omega v} \]
Extensions to 2D

\[ F[k, l] = \frac{1}{N^2} \sum_{u,v=0}^{N} f[u, v] e^{-2j\pi \left( \frac{ku+lv}{N} \right)} \]

\[
\begin{bmatrix} k \\ l \end{bmatrix}^T \begin{bmatrix} u \\ v \end{bmatrix} = ||a|| ||b|| \cos \theta
\]

magnitude of \([k, l]\) gives the frequency
To get some sense of what basis elements look like, we plot a basis element --- or rather, its real part --- as a function of $x,y$ for some fixed $u,v$. We get a function that is constant when $(ux+vy)$ is constant. The magnitude of the vector $(u, v)$ gives a frequency, and its direction gives an orientation. The function is a sinusoid with this frequency along the direction, and constant perpendicular to the direction.

Here $u$ and $v$ are larger than in the previous slide.
To get some sense of what basis elements look like, we plot a basis element --- or rather, its real part --- as a function of $x,y$ for some fixed $u,v$. We get a function that is constant when $(ux+vy)$ is constant. The magnitude of the vector $(u,v)$ gives a frequency, and its direction gives an orientation. The function is a sinusoid with this frequency along the direction, and constant perpendicular to the direction.

Here $u$ and $v$ are larger than in the previous slide.
Extension to 2D
Extension to 2D

in Matlab, check out: imagesc(log(abs(fftshift(fft2(im)))));
Fourier analysis in images

Intensity Image

Fourier Image

http://sharp.bu.edu/~slehar/fourier/fourier.html#filtering
Signals can be composed

http://sharp.bu.edu/~slehar/fourier/fourier.html#filtering
More: http://www.cs.unm.edu/~brayer/vision/fourier.html
Let’s build up a signal by randomly adding in Fourier transform coefficients.
Now, an analogous sequence of images, but selecting Fourier components in descending order of magnitude.
<table>
<thead>
<tr>
<th>Name</th>
<th>Signal</th>
<th>Transform</th>
</tr>
</thead>
<tbody>
<tr>
<td>impulse</td>
<td>( \delta(x) )</td>
<td>( 1 )</td>
</tr>
<tr>
<td>shifted impulse</td>
<td>( \delta(x-u) )</td>
<td>( e^{-j\omega u} )</td>
</tr>
<tr>
<td>box filter</td>
<td>( \text{box}(x/a) )</td>
<td>( \text{asinc}(a\omega) )</td>
</tr>
<tr>
<td>tent</td>
<td>( \text{tent}(x/a) )</td>
<td>( \text{asinc}^2(a\omega) )</td>
</tr>
<tr>
<td>Gaussian</td>
<td>( G(x;\sigma) )</td>
<td>( \sqrt{2\pi} \frac{G(\omega;\sigma^{-1})}{\sigma} )</td>
</tr>
<tr>
<td>Laplacian of Gaussian</td>
<td>( \frac{x^2}{\sigma^4} - \frac{1}{\sigma^2} G(x;\sigma) )</td>
<td>( -\sqrt{\frac{2\pi}{\sigma}} \omega^2 G(\omega;\sigma^{-1}) )</td>
</tr>
<tr>
<td>Gabor</td>
<td>( \cos(\omega_0 x) G(x;\sigma) )</td>
<td>( \sqrt{\frac{2\pi}{\sigma}} G(\omega \pm \omega_0;\sigma^{-1}) )</td>
</tr>
<tr>
<td>unsharp mask</td>
<td>( (1+\gamma) \delta(x) - \gamma G(x;\sigma) )</td>
<td>( (1+\gamma) - \sqrt{\frac{2\pi}{\sigma}} G(\omega;\sigma^{-1}) )</td>
</tr>
<tr>
<td>windowed sinc</td>
<td>( r \cos(x/(aW)) ) ( \frac{\sin(x/a)}{x/a} )</td>
<td>(see Figure 3.29)</td>
</tr>
</tbody>
</table>

Table 3.2 Some useful (continuous) Fourier transform pairs: The dashed line in the Fourier transform of the shifted impulse indicates its (linear) phase. All other transforms have zero phase (they are real-valued). Note that the figures are not necessarily drawn to scale but are drawn to illustrate the general shape and characteristics of the filter or its response. In particular, the Laplacian of Gaussian is drawn inverted because it resembles more a “Mexican hat”, as it is sometimes called.
Man-made Scene
Can change spectrum, then reconstruct

Local change in one domain, courses global change in the other
Low and High Pass filtering
Eigenfunctions

- Sinusoidal functions (or rather, complex exponentials) are eigenfunctions of linear shift invariant operators.

\[ \text{Linear-Shift-Invariant Operator} \rightarrow *A \]
Now consider doing this for all possible frequencies. What do we call the set of \{a, b, \ldots\} and \{A, B, \ldots\}?
Linear-Shift-Invariant Operator
2D convolution theorem example

\[ f(x, y) \ast h(x, y) = g(x, y) \]

\[ |F(s_x, s_y)| \times |H(s_x, s_y)| = |G(s_x, s_y)| \]
Fast Fourier transform (FFT)

“The most influential numerical algorithms of our lifetime”

Allows us to compute (I)FTs in $N\log N$ instead of $N^2$ (standard matrix multiplication)
Filtering

Why does the Gaussian give a nice smooth image, but the square filter give edgy artifacts?

Gaussian

Box filter
Recall...

Spatial domain

\[ f(x) \]

\[ \text{box}(x) \]

Frequency domain

\[ F(s) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi sx} dx \]

\[ \text{sinc}(s) \]

\[ \text{gauss}(s; 1/\sigma) \]

\[ \text{sinc}(s) \]

\[ \text{box}(x) \]
Gaussian
Box Filter
Low-pass, Band-pass, High-pass filters

low-pass:

High-pass / band-pass:
Edges in images
What does blurring take away?
What does blurring take away?

smoothed (5x5 Gaussian)
High-Pass filter

smoothed – original
Image “Sharpening”

What does blurring take away?

Let’s add it back:

original

smoothed (5x5)

detail

original

detail

sharpened
Unsharp mask filter

\[ f + \alpha(f - f \ast g) = (1+\alpha)f - \alpha f \ast g = f \ast ((1+\alpha)e - g) \]

- **Unit impulse**: Image
- **Blurred image**: Gaussian filtered image
- **Unit impulse (identity)**: Laplacian of Gaussian
Hybrid Images


Gaussian Filter!

Laplacian Filter!

Gaussian

unit impulse

Laplacian of Gaussian
Showing, at full resolution, the information captured at each level of a Gaussian (top) and Laplacian (bottom) pyramid.

Phase and Magnitude

• Curious fact
  – all natural images have about the same magnitude transform
  – hence, phase seems to matter, but magnitude largely doesn’t

• Demonstration
  – Take two pictures, swap the phase transforms, compute the inverse - what does the result look like?
This is the magnitude transform of the cheetah pic.
This is the phase transform of the cheetah pic!
This is the magnitude transform of the zebra pic
his is the phase transform of the zebra pic
Reconstruction with zebra phase, cheetah magnitude
Reconstruction with cheetah phase, zebra magnitude
Phase and Magnitude

Image with cheetah phase (and zebra magnitude)

Image with zebra phase (and cheetah magnitude)