Midterm is on Thursday!

Project presentations are May 17th, 22nd and 24th

Next week there is a strike on campus. Class is therefore cancelled on Tuesday.

Please work on your presentations instead!
From results on practice midterm: Half of the class needs another review!
Spatial frequency

(a) High-frequency square wave
(b) Low-frequency square wave
(c) High-contrast sinusoidal spatial grid
(d) Low-contrast sinusoidal spatial grid
1D Frequency Spectrum

Graphs of $\sin(0.5x)$, $\sin(x)$, and $\sin(2x)$ over the interval $[0, 15]$.
Frequency spectrum
2D Fourier Transform
Power Spectrum

- 2D Fourier transform of the *Dunes* photograph
Sampling

- Both in space and time
- Select acquisition speed
- Select sensor size / Optical magnification
- Aliasing
Nyquist sampling
= twice the frequency
Aliasing
Aliasing

• (Image downsized around four times)
Histogram shows distribution of gray-levels in an image.
Let's plot the histogram of this synthetic image.

Histogram Processing:
- Gray-level distribution can be judged by measuring a histogram.
- For a B-bit image, initialize 2^B counters with 0.
- Loop over all pixels (x,y).
- When encountering gray level \( f(x,y) = i \), increment the counter number \( i \).
- With proper normalization, the histogram can be interpreted as an estimate of the probability density function (pdf) of the underlying random variable (the gray-level).
- You can also use fewer, larger bins to trade off amplitude.
Histogram Processing:

- Distribution of gray levels can be judged by measuring a histogram.

For B-bit image:

- Initialize 2^B counters with 0.
- Loop over all pixels x, y.
- When encountering gray level \( f(x, y) = i \), increment the counter number i.

With proper normalization, the histogram can be interpreted as an estimate of the probability density function (pdf) of the underlying random variable (the graylevel).

You can also use fewer, larger bins to trade off amplitude.
Grayscale Histogram

Example:
Color Histogram
Image enhancement

- In the pixel domain, for example:

- Make image “better” for a specific application – The idea of “better” is somewhat subjective

- We distinguish two domains:
  - Spatial or Pixel domain:
  - Frequency Domain:

- For this section: Pixel Domain
  - Operations on single pixel at a time
  - Operations on groups of pixels (neighborhoods)
Convolution

- Let $f$ be the image and $g$ be the kernel. The output of convolving $f$ with $g$ is denoted $f * g$.

$$(f * g)[m,n] = \sum_{k,l} f[m-k, n-l] g[k, l]$$

- Convention: kernel is “flipped”
- MATLAB: conv2 vs. filter2 (also imfilter)
Convolution properties

- **Linearity**: \( \text{filter}(f_1 + f_2) = \text{filter}(f_1) + \text{filter}(f_2) \)

- **Shift invariance**: same behavior regardless of pixel location: \( \text{filter}(\text{shift}(f)) = \text{shift}(\text{filter}(f)) \)

- Theoretical result: any linear shift-invariant operator can be represented as a convolution
Overview of Spatial Filtering

- Local linear operations on an image

Input: \( f(x,y) \), Output: \( g(x,y) \)

\[
g(x, y) = w_1 f(x-1, y-1) + w_2 f(x-1, y) + \cdots + w_8 f(x+1, y-1) + w_9 f(x+1, y+1)
\]
What to do at the edges of the image

- An important point: **Edge Effects**
  - To compute all pixel values in the output image, we need to fill in a “border”

Mask dimension = $2M+1$

Border dimension = $M$
Image Enhancement: Spatial Filtering Operation

• An important point: **Edge Effects** *(Ex.: 5x5 Mask)*
  
  – How to fill in a “border”
  
  • Zeros (Ringing)
  • Replication (Better)
  • Reflection (“Best”)

\[
\begin{array}{cccc}
  d & c & a & b \\
b & a & a & b \\
b & a & a & b \\
d & c & c & d \\
\end{array}
\]

• **Procedure:**
  
  – Replicate row-wise
  – Replicate column-wise
  – Apply filtering
  – Remove borders

Source: S. Marschner
MATLAB methods:

- clip filter (black): `imfilter(f, g, 0)`
- wrap around: `imfilter(f, g, 'circular')`
- copy edge: `imfilter(f, g, 'replicate')`
- reflect across edge: `imfilter(f, g, 'symmetric')`
What happens:

- Filter kernel operations on matrix (image)
- “Drag and Stamp” operation
- Using MATLAB: filter2(A,B)
  - use: ‘same’, ‘valid’ and ‘full’ parameters (ex: filter2(A,B,’full’))
Filtering example

- MATLAB code how to apply filter kernel:
Gaussian blur filtering

- Filter with Gaussian kernel
- Filtering twice with a certain Gaussian is the same as filtering with this Gaussian convolved with itself -> MORE BLUR
Gaussian Kernel

\[ G_{\sigma} = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \]

- Constant factor at front makes volume sum to 1 (can be ignored, as we should re-normalize weights to sum to 1 in any case)
Choosing kernel width

- Gaussian filters have infinite support, but discrete filters use finite kernels.
Example code MATLAB

ImageO = imread('MushroomDownload.jpg');
Image = ImageO(:,:,1);

H = fspecial('motion',20,45);
MotionBlur = imfilter(Image,H,'replicate');
subplot(2,1,2), imagesc(MotionBlur),axis equal, axis off, colormap gray
subplot(2,1,1), imagesc(Image),axis equal, axis off, colormap gray
Blurring can be used for denoising

**Figure 3.37** (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a $3 \times 3$ averaging mask. (c) Noise reduction with a $3 \times 3$ median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

- (Example from course textbook Gonzales & Wood)
Try this out on noisy data with median, gaussian etc...
Note: it’s better to vectorize your code in MATLAB

You can read up on this at: http://www.mathworks.com/help/matlab/matlab_prog/vectorization.html
Fourier Transform

- Examples of spatial filtering in frequency space
Some Fourier Transform Pairs

<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>$a\text{sinc}(a\omega)$</td>
</tr>
<tr>
<td>tent</td>
<td>$\text{tent}(x/a)$</td>
<td>$a\text{sinc}^2(a\omega)$</td>
</tr>
<tr>
<td>Gaussian</td>
<td>$G(x; \sigma)$</td>
<td>$\frac{\sqrt{2\pi}}{\sigma} G(\omega; \sigma^{-1})$</td>
</tr>
<tr>
<td>Laplacian of Gaussian</td>
<td>$(\frac{1}{\sigma^2} - \frac{1}{\sigma^2})G(x; \sigma)$</td>
<td>$-\frac{\sqrt{2\pi}}{\sigma} \omega^2 G(\omega; \sigma^{-1})$</td>
</tr>
<tr>
<td>Gabor</td>
<td>$\cos(\omega_0 x) G(x; \sigma)$</td>
<td>$\frac{\sqrt{2\pi}}{\sigma} G(\omega \pm \omega_0; \sigma^{-1})$</td>
</tr>
<tr>
<td>unsharp mask</td>
<td>$(1 + \gamma)\delta(x) - \gamma G(x; \sigma)$</td>
<td>$\frac{(1 + \gamma) - \sqrt{2\pi}}{\sigma} G(\omega; \sigma^{-1})$</td>
</tr>
<tr>
<td>windowed sinc</td>
<td>$\text{re}(x/(aW)/\sin(x/a))$</td>
<td>(see Figure 3.29)</td>
</tr>
</tbody>
</table>

# Fourier Transform Pairs

<table>
<thead>
<tr>
<th>Function, $f(t)$</th>
<th>Fourier Transform, $F(\omega)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)e^{j\omega t} , d\omega$</td>
<td>$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} , dt$</td>
</tr>
<tr>
<td>$f(t-t_0)$</td>
<td>$F(\omega)e^{-j\omega t_0}$</td>
</tr>
<tr>
<td>$f(t)e^{j\omega_0 t}$</td>
<td>$F(\omega - \omega_0)$</td>
</tr>
<tr>
<td>$f(\alpha t)$</td>
<td>$\frac{1}{</td>
</tr>
<tr>
<td>$F(t)$</td>
<td>$2\pi f(-\omega)$</td>
</tr>
<tr>
<td>$\frac{d^n f(t)}{dt^n}$</td>
<td>$(j\omega)^n F(\omega)$</td>
</tr>
<tr>
<td>$(-j\omega)^n f(t)$</td>
<td>$\frac{d^n F(\omega)}{d\omega^n}$</td>
</tr>
<tr>
<td>$\int_{-\infty}^{t} f(\tau)d\tau$</td>
<td>$F(\omega) + \pi F(0) \delta(\omega)$</td>
</tr>
<tr>
<td>$\delta(t)$</td>
<td>1</td>
</tr>
<tr>
<td>$e^{j\omega_0 t}$</td>
<td>$2\pi \delta(\omega - \omega_0)$</td>
</tr>
<tr>
<td>$\text{sgn}(t)$</td>
<td>$\frac{2}{j\omega}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function, $f(t)$</th>
<th>Fourier Transform, $F(\omega)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{1}{\pi t}$</td>
<td>$\text{sgn}(\omega)$</td>
</tr>
<tr>
<td>$u(t)$</td>
<td>$\pi \delta(\omega) + \frac{1}{j\omega}$</td>
</tr>
<tr>
<td>$\sum_{n=-\infty}^{\infty} F_n e^{j\omega_0 t}$</td>
<td>$2\pi \sum_{n=-\infty}^{\infty} F_n \delta(\omega - n\omega_0)$</td>
</tr>
<tr>
<td>$\text{rect}\left(\frac{t}{\tau}\right)$</td>
<td>$\frac{\tau}{2} \text{Sa}\left(\frac{\omega\tau}{2}\right)$</td>
</tr>
<tr>
<td>$B \frac{\text{Sa}\left(\frac{Bt}{2}\right)}{2}$</td>
<td>$\text{rect}\left(\frac{\omega}{B}\right)$</td>
</tr>
<tr>
<td>$\text{tri}(t)$</td>
<td>$\text{Sa}^2\left(\frac{\omega}{2}\right)$</td>
</tr>
<tr>
<td>$A \cos\left(\frac{\pi t}{2\tau}\right) \text{rect}\left(\frac{t}{2\tau}\right)$</td>
<td>$\frac{A\pi}{\tau} \cos(\omega t)\left(\frac{\pi}{2\tau}\right)^2 - \omega^2$</td>
</tr>
<tr>
<td>$\cos(\omega_0 t)$</td>
<td>$\pi \left[ \delta(\omega - \omega_0) + \delta(\omega + \omega_0) \right]$</td>
</tr>
<tr>
<td>$\sin(\omega_0 t)$</td>
<td>$\frac{\pi}{j} \left[ \delta(\omega - \omega_0) - \delta(\omega + \omega_0) \right]$</td>
</tr>
<tr>
<td>$u(t) \cos(\omega_0 t)$</td>
<td>$\frac{\pi}{2} \left[ \delta(\omega - \omega_0) + \delta(\omega + \omega_0) \right] + \frac{j\omega}{\omega_0^2 - \omega^2}$</td>
</tr>
<tr>
<td>$u(t) \sin(\omega_0 t)$</td>
<td>$\frac{\pi}{2j} \left[ \delta(\omega - \omega_0) - \delta(\omega + \omega_0) \right] + \frac{\omega^2}{\omega_0^2 - \omega^2}$</td>
</tr>
<tr>
<td>$u(t)e^{-\alpha t} \cos(\omega_0 t)$</td>
<td>$\frac{(\alpha + j\omega)}{\omega_0^2 + (\alpha + j\omega)^2}$</td>
</tr>
</tbody>
</table>
Lets do an experiment with spatial frequency imaging

• Take an image of the line pairs with your cell phone

• If you don't have one, come up here and take them with a camera

• Download to your computer and Fourier transform the data. Can you recognize the spatial frequencies from the image?