Thinking in Frequency

Computer Vision
Brown

James Hays
(with modifications by Ricardo Fabbri)
Recap of Wednesday

linear filtering
convolution
differential filters
filter types
boundary conditions.
Review: questions

1. Write down a 3x3 filter that returns a positive value if the average value of the 4-adjacent neighbors is less than the center and a negative value otherwise.

2. Write down a filter that will compute the gradient in the x-direction:

   \[ \text{grad}_x(y,x) = \text{im}(y,x+1) - \text{im}(y,x) \] for each \( x, y \)
Review: questions

3. Fill in the blanks:

a) \_
   = D \times B
b) A = \_ \times \_
c) F = D \times \_
d) \_ = D \times D

Filtering Operator

E
F
G
H
I

Slide: Hoiem
Today’s Class

• Fourier transform and frequency domain
  – Frequency view of filtering
  – Hybrid images
  – Sampling

• Reminder: Read your textbook
  – Today’s lecture covers material in 3.4
Why does the Gaussian give a nice smooth image, but the square filter give edgy artifacts?
Hybrid Images

Why do we get different, distance-dependent interpretations of hybrid images?
Why does a lower resolution image still make sense to us? What do we lose?

Image: http://www.flickr.com/photos/igorms/136916757/
Thinking in terms of frequency
Jean Baptiste Joseph Fourier
Jean Baptiste Joseph Fourier

had crazy idea (1807):

Any univariate function can be rewritten as a weighted sum of sines and cosines of different frequencies.

• Don’t believe it?
  – Neither did Lagrange, Laplace, Poisson and other big wigs
  – Not translated into English until 1878!

...the manner in which the author arrives at these equations is not exempt of difficulties and...his analysis to integrate them still leaves something to be desired on the score of generality and even rigour.

Laplace

Lagrange

Legendre
Jean Baptiste Joseph Fourier

had crazy idea (1807):

Any univariate function can be rewritten as a weighted sum of sines and cosines of different frequencies.

• Don’t believe it?
  – Neither did Lagrange, Laplace, Poisson and other big wigs
  – Not translated into English until 1878!
• But it’s (mostly) true!
A sum of sines

Our building block:

\[ A \sin(\omega x + \phi) \]

Add enough of them to get any signal \( g(x) \) you want!
Frequency Spectra

• example: \( g(t) = \sin(2\pi f t) + \frac{1}{3}\sin(2\pi(3f) t) \)
Frequency Spectra
Frequency Spectra
Frequency Spectra
Frequency Spectra

= 

+ 

= 

= 
Frequency Spectra

\[ \text{Square Wave} \oplus \text{Random Noise Wave} = \text{Modulated Wave} \]
Frequency Spectra
Frequency Spectra

\[ A \sum_{k=1}^{\infty} \frac{1}{k} \sin(2\pi kt) \]
Example: Music

- We think of music in terms of frequencies at different magnitudes
Other signals

• We can also think of all kinds of other signals the same way

xkcd.com
Fourier analysis in images

Intensity Image

Fourier Image

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

\[
\begin{array}{ccc}
\text{gradient} & + & \text{pattern} \\
\text{shading} & = & \text{result}
\end{array}
\]

http://sharp.bu.edu/~slehar/fourier/fourier.html#filtering
More: http://www.cs.unm.edu/~brayer/vision/fourier.html
Fourier Transform
Fourier Transform

• Fourier transform stores the magnitude and phase at each frequency
Fourier Transform

• Fourier transform stores the magnitude and phase at each frequency
  – Magnitude encodes how much signal there is at a particular frequency
Fourier Transform

• Fourier transform stores the magnitude and phase at each frequency
  – Magnitude encodes how much signal there is at a particular frequency
  – Phase encodes spatial information (indirectly)
Fourier Transform

• Fourier transform stores the magnitude and phase at each frequency
  – Magnitude encodes how much signal there is at a particular frequency
  – Phase encodes spatial information (indirectly)
  – For mathematical convenience, this is often notated in terms of real and complex numbers

\[
\text{Amplitude: } A = \pm \sqrt{R(\omega)^2 + I(\omega)^2} \quad \text{Phase: } \phi = \tan^{-1} \frac{I(\omega)}{R(\omega)}
\]
The Convolution Theorem

\[ F[g \ast h] = F[g]F[h] \]
The Convolution Theorem

• The Fourier transform of the convolution of two functions is the product of their Fourier transforms

\[ F[g \ast h] = F[g]F[h] \]
The Convolution Theorem

• The Fourier transform of the convolution of two functions is the product of their Fourier transforms

$$F[g * h] = F[g]F[h]$$
The Convolution Theorem

• The Fourier transform of the convolution of two functions is the product of their Fourier transforms

\[ F[g * h] = F[g]F[h] \]
The Convolution Theorem

• The Fourier transform of the convolution of two functions is the product of their Fourier transforms

\[ F[g \ast h] = F[g]F[h] \]
The Convolution Theorem

- The Fourier transform of the convolution of two functions is the product of their Fourier transforms
  \[ F[g * h] = F[g]F[h] \]

- Convolution in spatial domain is equivalent to multiplication in frequency domain!
  \[ g * h = F^{-1}[F[g]F[h]] \]
Filtering in spatial domain

\[
\begin{pmatrix}
1 & 0 & -1 \\
2 & 0 & -2 \\
1 & 0 & -1 \\
\end{pmatrix}
\]
Filtering in frequency domain

- FFT
- Inverse FFT
Fourier Matlab demo
FFT in Matlab

• Filtering with fft

```matlab
im = double(imread('...'))/255;
im = rgb2gray(im); % "im" should be a gray-scale floating point image
[imh, imw] = size(im);

hs = 50; % filter half-size
fil = fspecial('gaussian', hs*2+1, 10);

fftsize = 1024; % should be order of 2 (for speed) and include padding
im_fft = fft2(im, fftsize, fftsize); % 1) fft im with padding
fil_fft = fft2(fil, fftsize, fftsize); % 2) fft fil, pad to same size as image
im_fil_fft = im_fft .* fil_fft; % 3) multiply fft images
im_fil = ifft2(im_fil_fft); % 4) inverse fft2
im_fil = im_fil(1+hs:size(im,1)+hs, 1+hs:size(im, 2)+hs); % 5) remove padding
```

• Displaying with fft

```matlab
figure(1), imagesc(log(abs(fftshift(im_fft)))), axis image, colormap jet
```
Filtering

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

Why does a lower resolution image still make sense to us? What do we lose?

Throw away every other row and column to create a 1/2 size image

Subsampling by a factor of 2
Subsampling by a factor of 2

Throw away every other row and column to create a 1/2 size image
Aliasing problem

• 1D example (sinewave):
Aliasing problem

- 1D example (sinewave):
Aliasing problem

• Sub-sampling may be dangerous….

• Characteristic errors may appear:
  – “Wagon wheels rolling the wrong way in movies”
  – “Checkerboards disintegrate in ray tracing”
  – “Striped shirts look funny on color television”
Aliasing in video

Imagine a spoked wheel moving to the right (rotating clockwise). Mark wheel with dot so we can see what’s happening.

If camera shutter is only open for a fraction of a frame time (frame time = \(1/30\) sec. for video, \(1/24\) sec. for film):

Without dot, wheel appears to be rotating slowly backwards! (counterclockwise)
Aliasing in graphics

Disintegrating textures

Source: A. Efros
Sampling and aliasing
Nyquist–Shannon Sampling

- When sampling a signal at discrete intervals, the sampling frequency must be \( \geq 2 \times f_{\text{max}} \)
- \( f_{\text{max}} = \text{max frequency of the input signal} \)
- This will allow to reconstruct the original perfectly from the sampled version
Anti-aliasing

Solutions:

• Sample more often

• Get rid of all frequencies that are greater than half the new sampling frequency
  – Will lose information
  – But it’s better than aliasing
  – Apply a smoothing filter
Algorithm for downsampling by

1. Start with image\((h, w)\)
2. Apply low-pass filter
   \[
   \text{im\_blur} = \text{imfilter}(\text{image}, \text{fspecial}('\text{gaussian}', 7, 1))
   \]
3. Sample every other pixel
   \[
   \text{im\_small} = \text{im\_blur}(1:2:end, 1:2:end);
   \]
Anti-aliasing

Forsyth and Ponce 2002
Subsampling without pre-filtering

1/2  1/4  (2x zoom)  1/8  (4x zoom)
Subsampling with Gaussian pre–
Why do we get different, distance-dependent interpretations of hybrid images?
Salvador Dali invented Hybrid Images?
Salvador Dali invented Hybrid Images?

Salvador Dali
“Gala Contemplating the Mediterranean Sea, which at 30 meters becomes the portrait of Abraham Lincoln”, 1976
Clues from Human Perception

Early Visual Processing: Multi-scale edge and blob filters
Clues from Human Perception

- Early processing in humans filters for various orientations and scales of frequency
Clues from Human Perception

- Early processing in humans filters for various orientations and scales of frequency
- Perceptual cues in the mid–high frequencies dominate perception
Clues from Human Perception

• Early processing in humans filters for various orientations and scales of frequency
• Perceptual cues in the mid–high frequencies dominate perception
• When we see an image from far away, we are effectively subsampling it

Early Visual Processing: Multi-scale edge and blob filters
Campbell-Robson contrast sensitivity curve
Campbell-Robson contrast sensitivity curve
Hybrid Image in FFT

Hybrid Image

Low-passed Image + High-passed Image
Perception

Why do we get different, distance-dependent interpretations of hybrid images?
Things to Remember
Things to Remember

• Sometimes it makes sense to think of images and filtering in the frequency domain
  – Fourier analysis
Things to Remember

• Sometimes it makes sense to think of images and filtering in the frequency domain
  – Fourier analysis

• Can be faster to filter using FFT for large images (N logN vs. N² for auto-correlation)
Things to Remember

• Sometimes it makes sense to think of images and filtering in the frequency domain
  – Fourier analysis

• Can be faster to filter using FFT for large images (N logN vs. N^2 for auto-correlation)
Things to Remember

- Sometimes it makes sense to think of images and filtering in the frequency domain
  - Fourier analysis

- Can be faster to filter using FFT for large images (\(N \log N\) vs. \(N^2\) for auto-correlation)

- Images are mostly smooth
  - Basis for compression
Things to Remember

- Sometimes it makes sense to think of images and filtering in the frequency domain
  - Fourier analysis

- Can be faster to filter using FFT for large images (N logN vs. $N^2$ for auto-correlation)

- Images are mostly smooth
  - Basis for compression
Things to Remember

• Sometimes it makes sense to think of images and filtering in the frequency domain
  – Fourier analysis

• Can be faster to filter using FFT for large images (N logN vs. \( N^2 \) for auto-correlation)

• Images are mostly smooth
  – Basis for compression

• Remember to low-pass before sampling
Practice question

1. Match the spatial domain image to the Fourier magnitude image
Next class

• Template matching

• Image Pyramids

• Filter banks and texture
Sharpening revisited

• What does blurring take away?

original

smoothed (5x5)
Sharpening revisited

• What does blurring take away?

original \[\text{smoothed (5x5)}\] \[=\] detail
Sharpening revisited

• What does blurring take away?

Let’s add it back:
Sharpening revisited

• What does blurring take away?

Let's add it back: