Thinking in Frequency

Computer Vision

University of Illinois

Derek Hoiem

Dali: "Gala Contemplating the Mediterranean Sea" (1976)

Review: image filtering

$$g[\cdot,\cdot] \; \frac{1}{9} \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

$$f[.,.] \qquad\qquad h[.,.]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$$h[m,n] = \sum_{k,l} g[k,l] \, f[m+k,n+l]$$

Review: image filtering

$$g[\cdot,\cdot]\ \frac{1}{9}$$

1	1	1
1	1	1
1	1	1

$$f[.,.]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$$h[.,.]$$

	0	10							

$$h[m,n] = \sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Review: image filtering

$$g[\cdot,\cdot] \; \frac{1}{9}$$

1	1	1
1	1	1
1	1	1

$$f[.,.]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$$h[.,.]$$

		0	10	20					

$$h[m,n] = \sum_{k,l} g[k,l] \, f[m+k,n+l]$$

Review: image filtering

$$g[\cdot\,,\cdot\,]\ \frac{1}{9}$$

| 1	1	1
1	1	1
1	1	1

$$f[.,.]$$

$$h[.,.]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

		0	10	20	30				

$$h[m,n] = \sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Review: image filtering

$$g[\cdot,\cdot]\ \frac{1}{9}$$

1	1	1
1	1	1
1	1	1

$$f[\cdot,\cdot]$$

| 0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$$h[\cdot,\cdot]$$

	0	10	20	30	30				

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k, n+l]$$

Review: image filtering

$$g[\cdot,\cdot]\ \frac{1}{9}$$

1	1	1
1	1	1
1	1	1

$$f[.,.]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$$h[.,.]$$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

$$h[m,n] = \sum_{k,l} g[k,l]\, f[m+k,n+l]$$

Review: image filtering in spatial domain
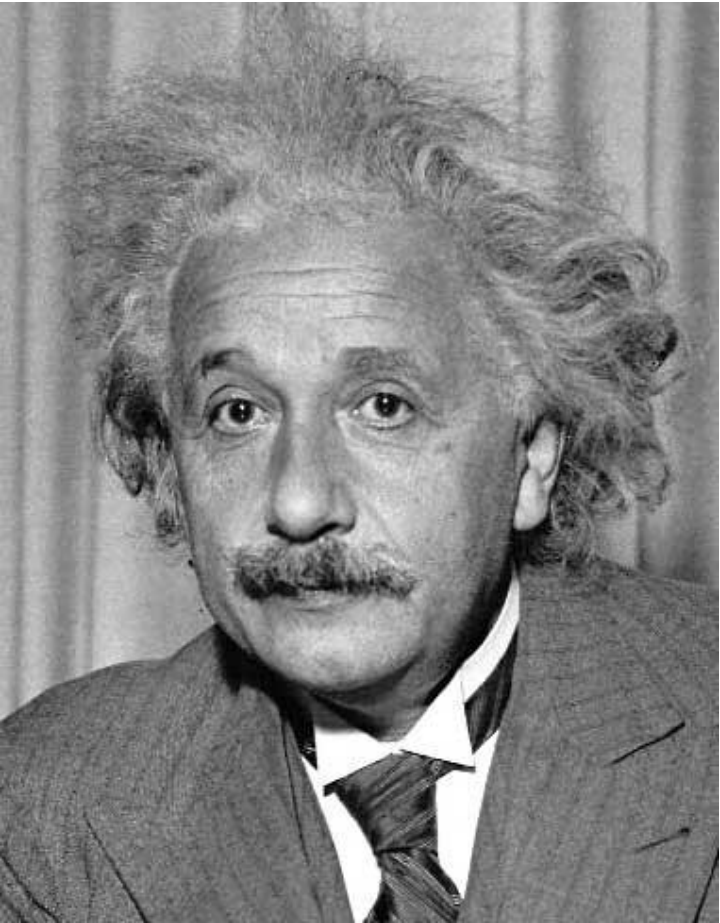
1	0	-1
2	0	-2
1	0	-1

Sobel
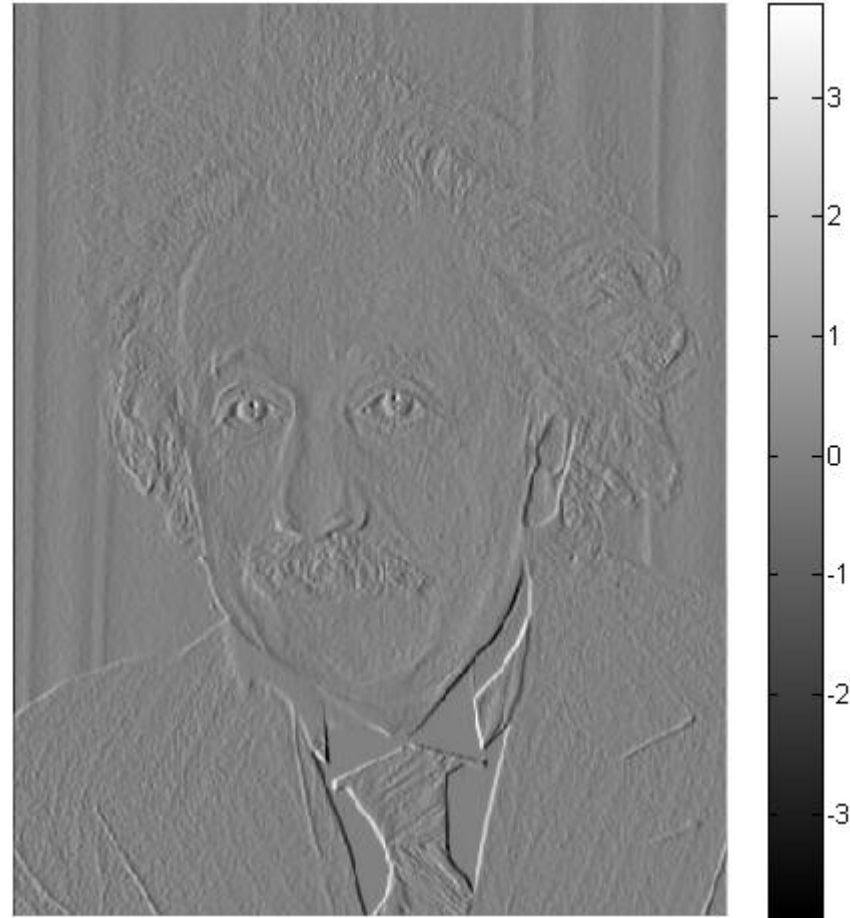
Today's Class

- Fourier transform and frequency domain
  - Frequency view of filtering
  - Sampling

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

Gaussian
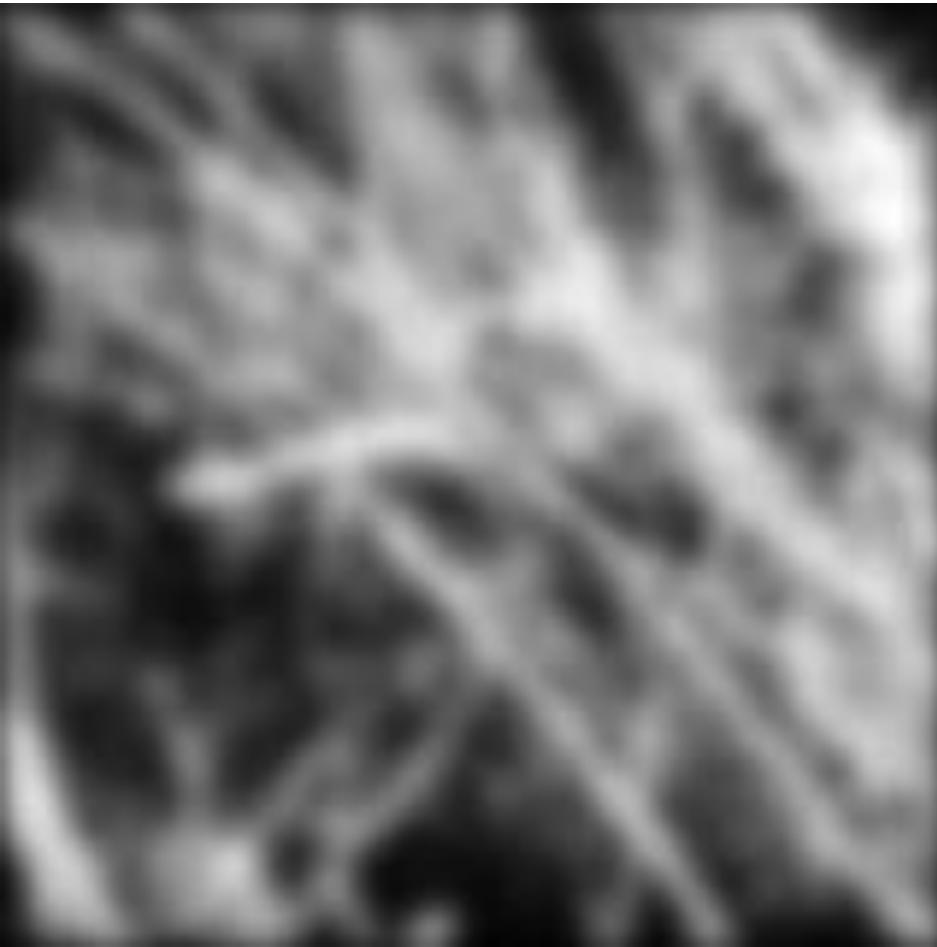
Box filter
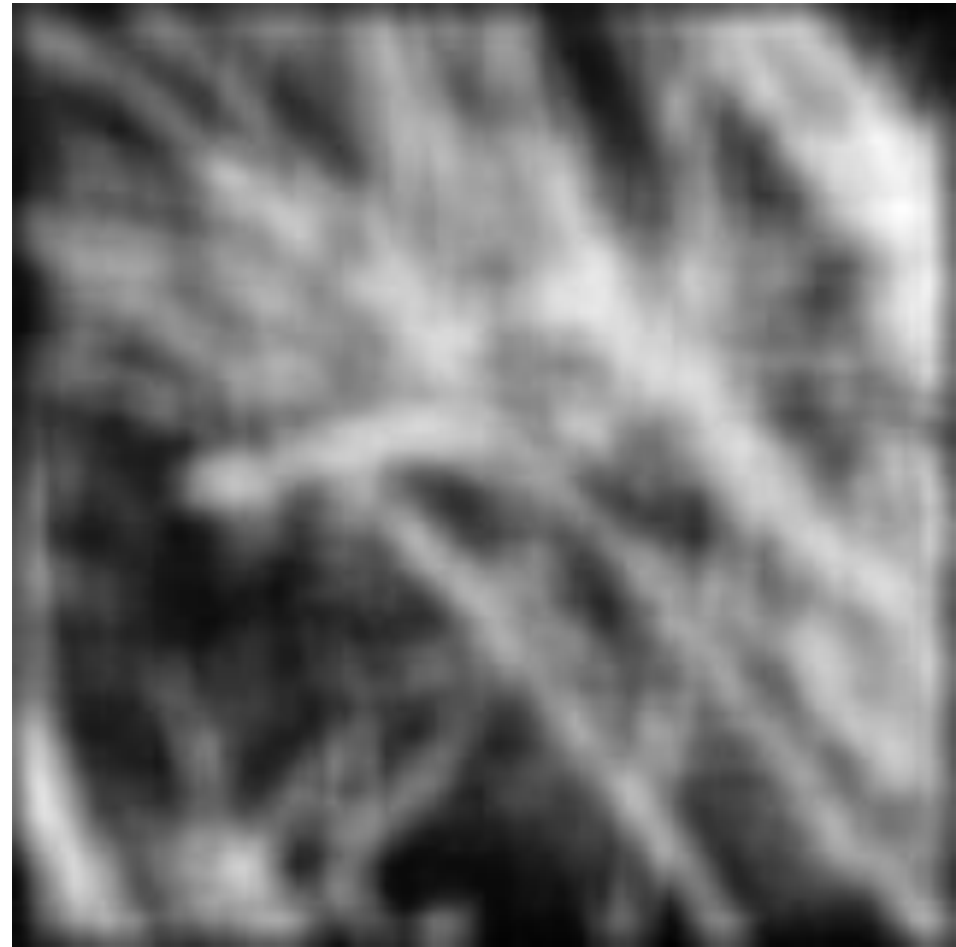
Hybrid Images
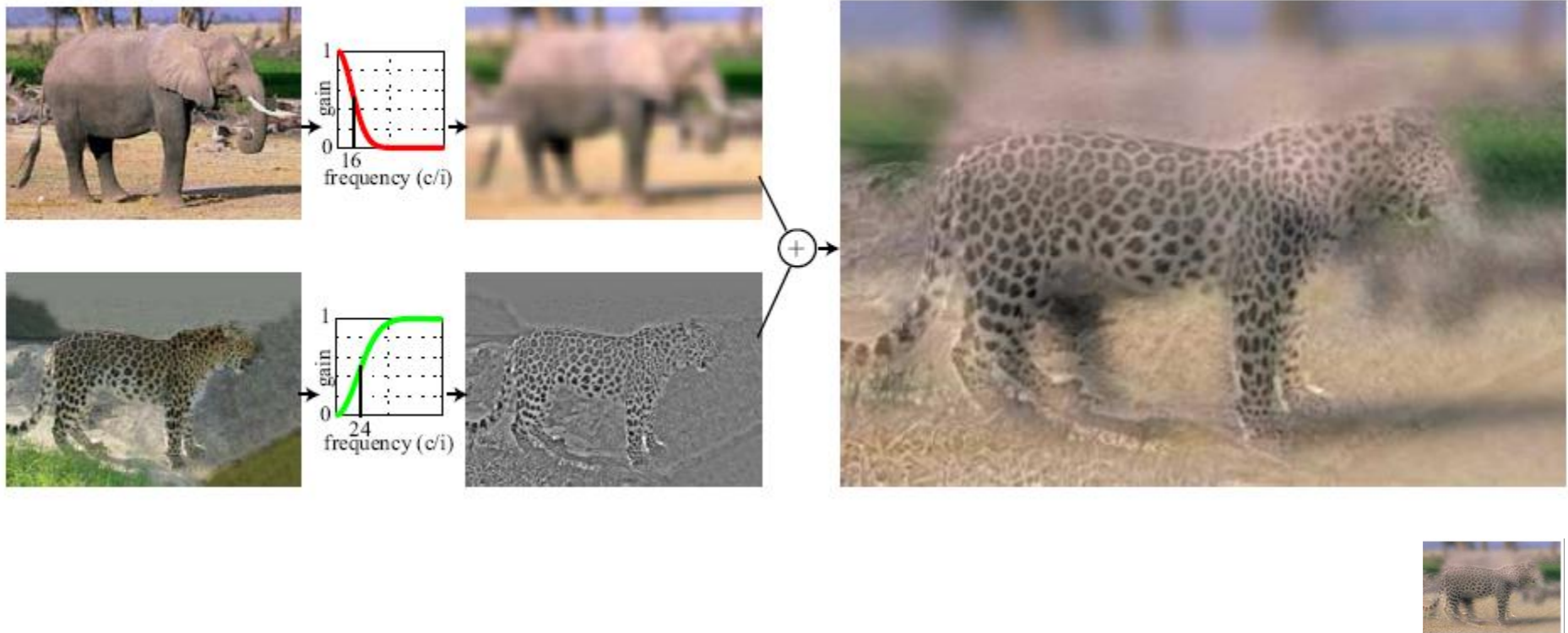
- A. Oliva, A. Torralba, P.G. Schyns, "Hybrid Images," SIGGRAPH 2006

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/
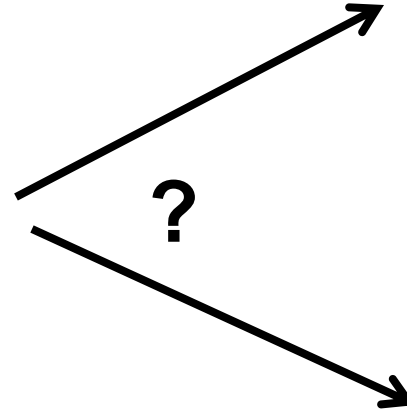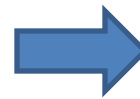
Thinking in terms of frequency

Jean Baptiste Joseph Fourier (1768-1830)

had crazy idea (1807):

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

...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.

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

- But it's (mostly) true!
  - called Fourier Series
  - there are some subtle restrictions

Laplace

Lagrange

Legendre
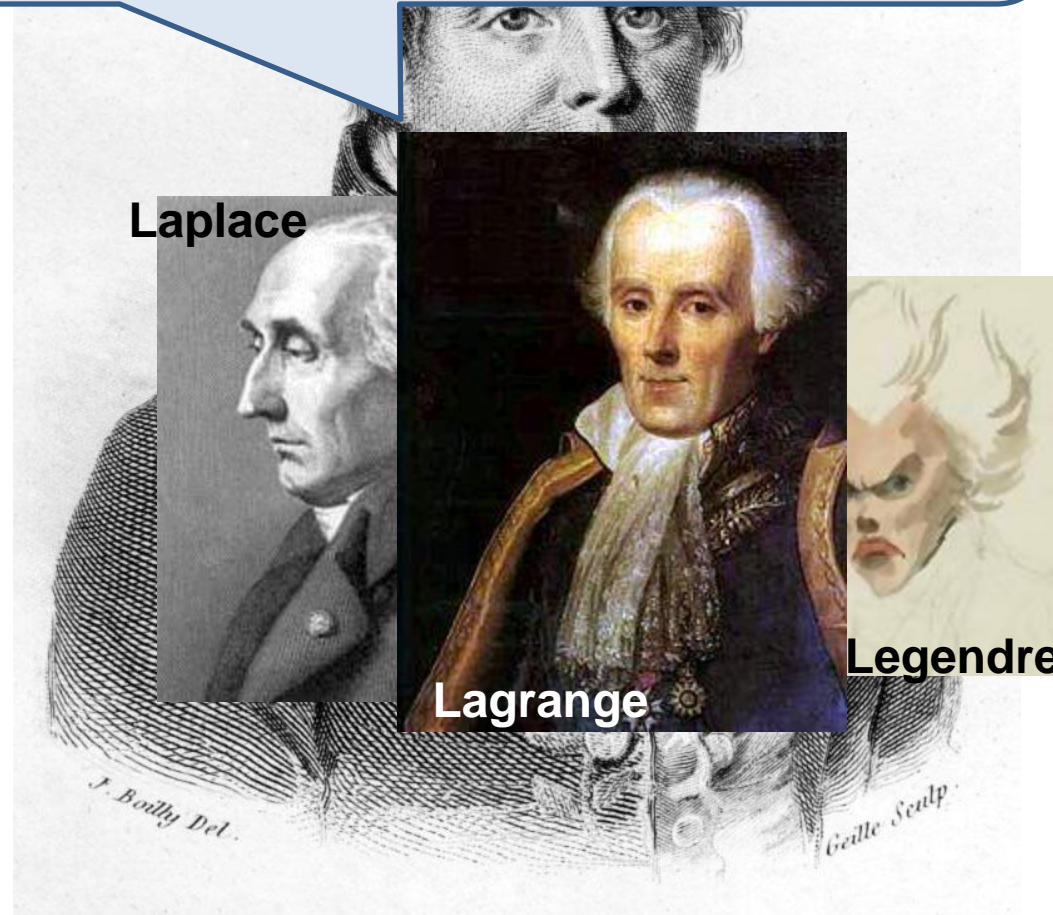
A sum of sines

Our building block:

$$A \sin(\omega x + \phi)$$

Add enough of them to get any signal *f(x)* you want!
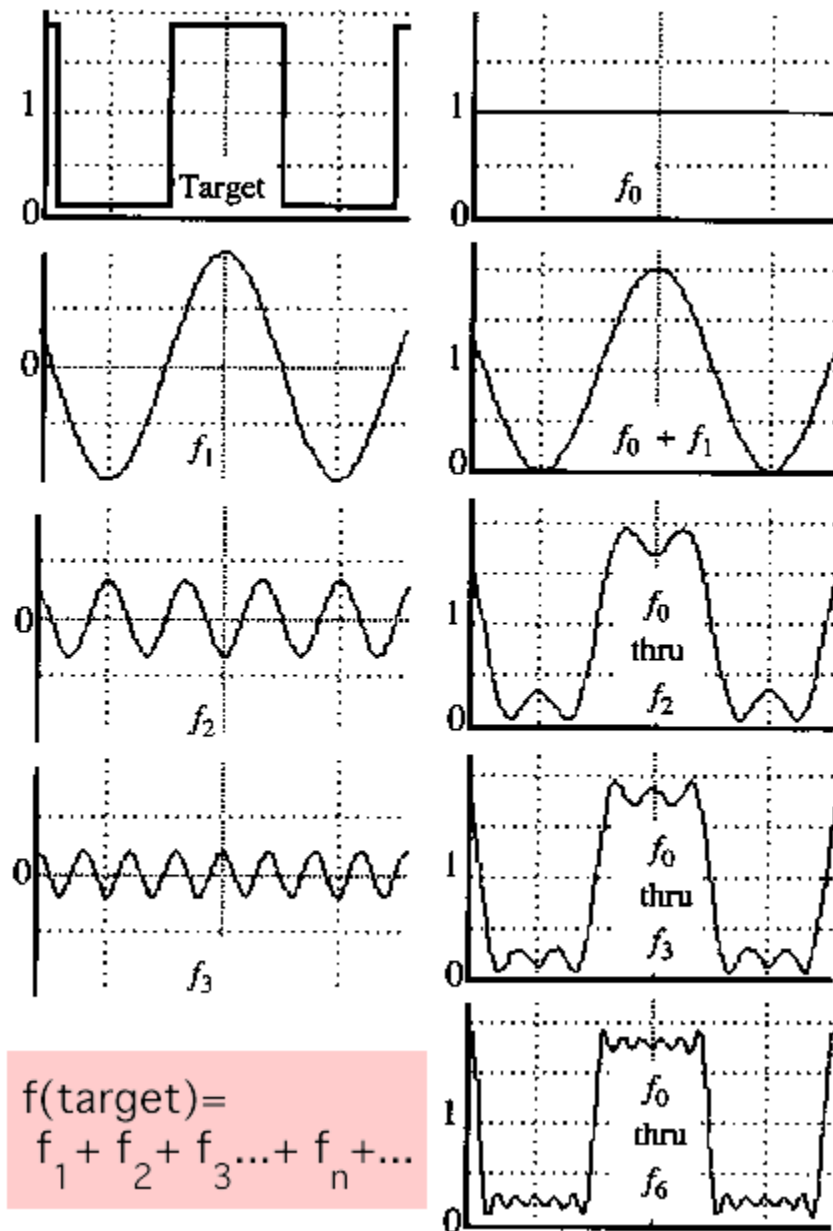
f(target)=
$f_1 + f_2 + f_3 \ldots + f_n + \ldots$

Frequency Spectra

- example : $g(t) = \sin(2\pi f\, t) + (1/3)\sin(2\pi(3f)\, t)$
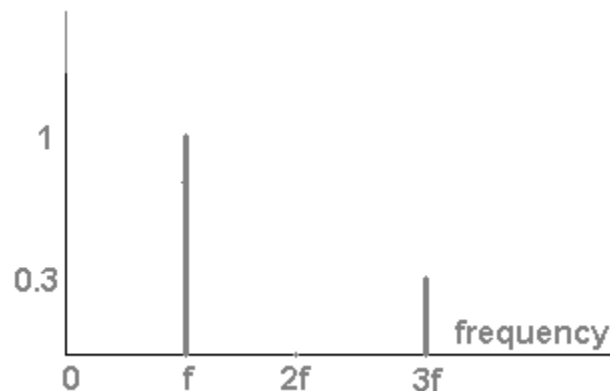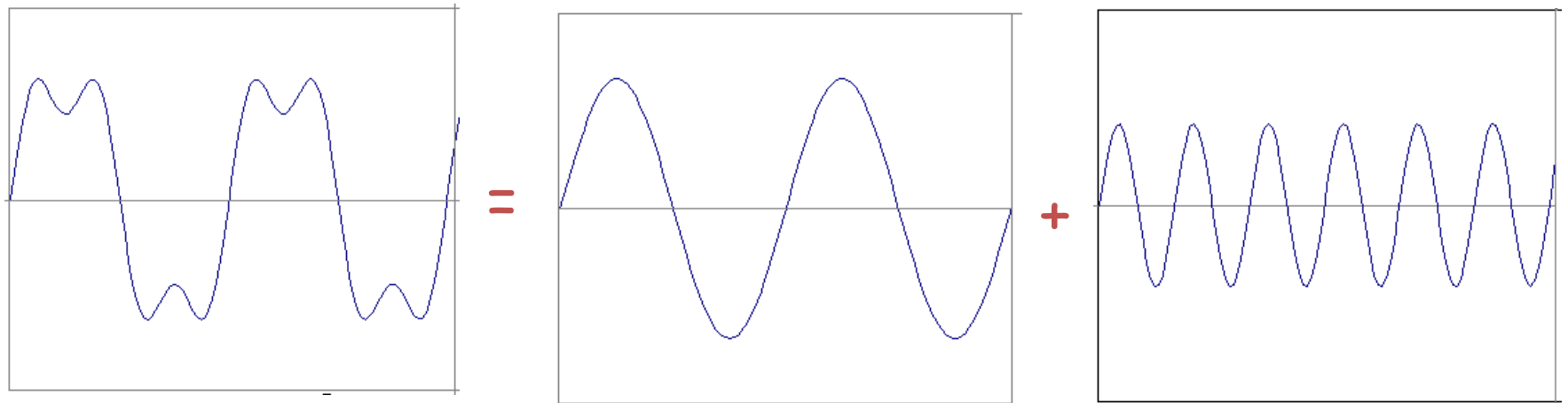
Frequency Spectra
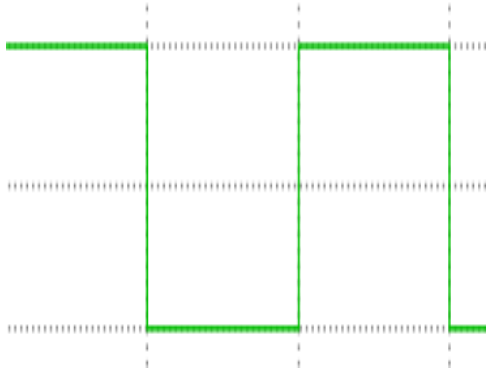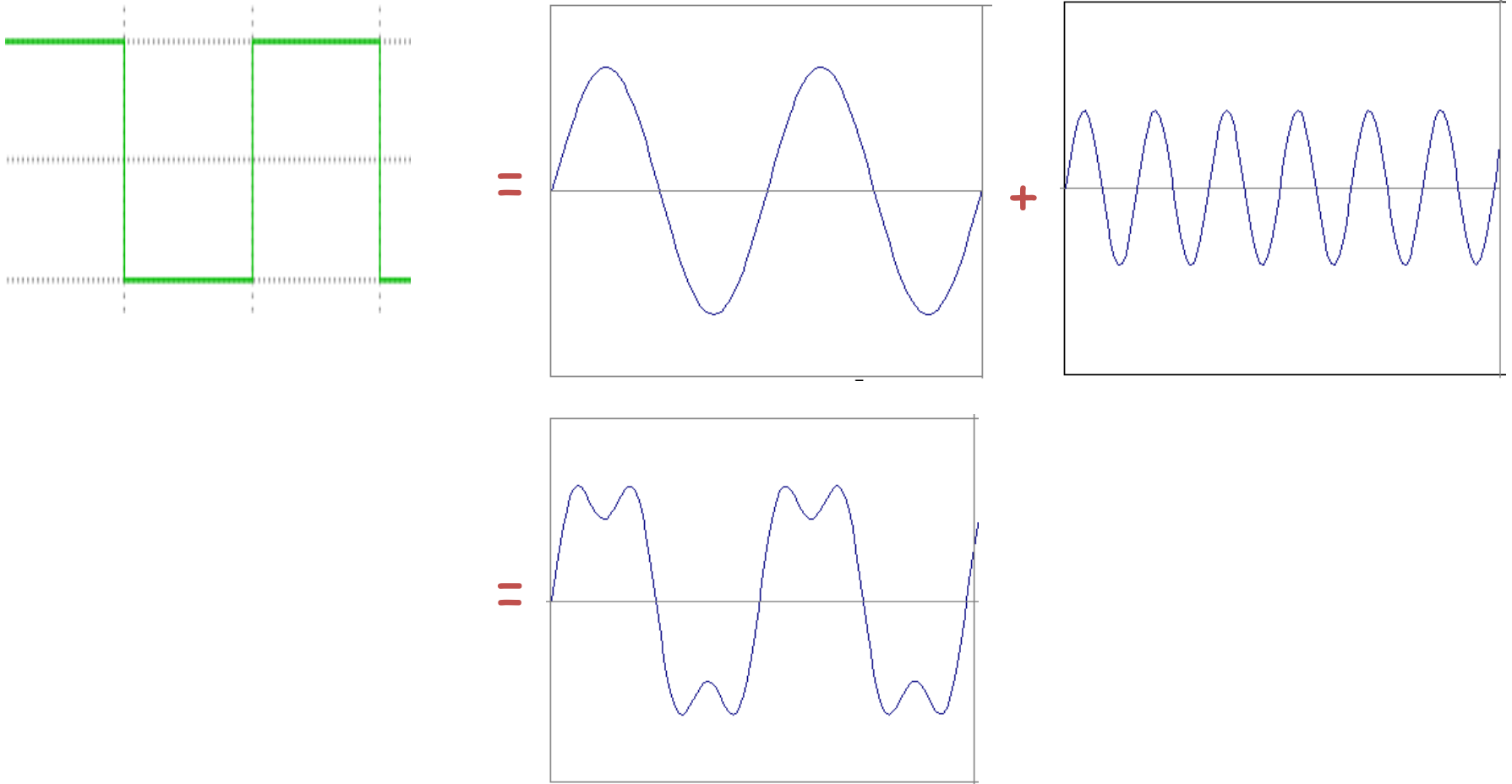
Frequency Spectra
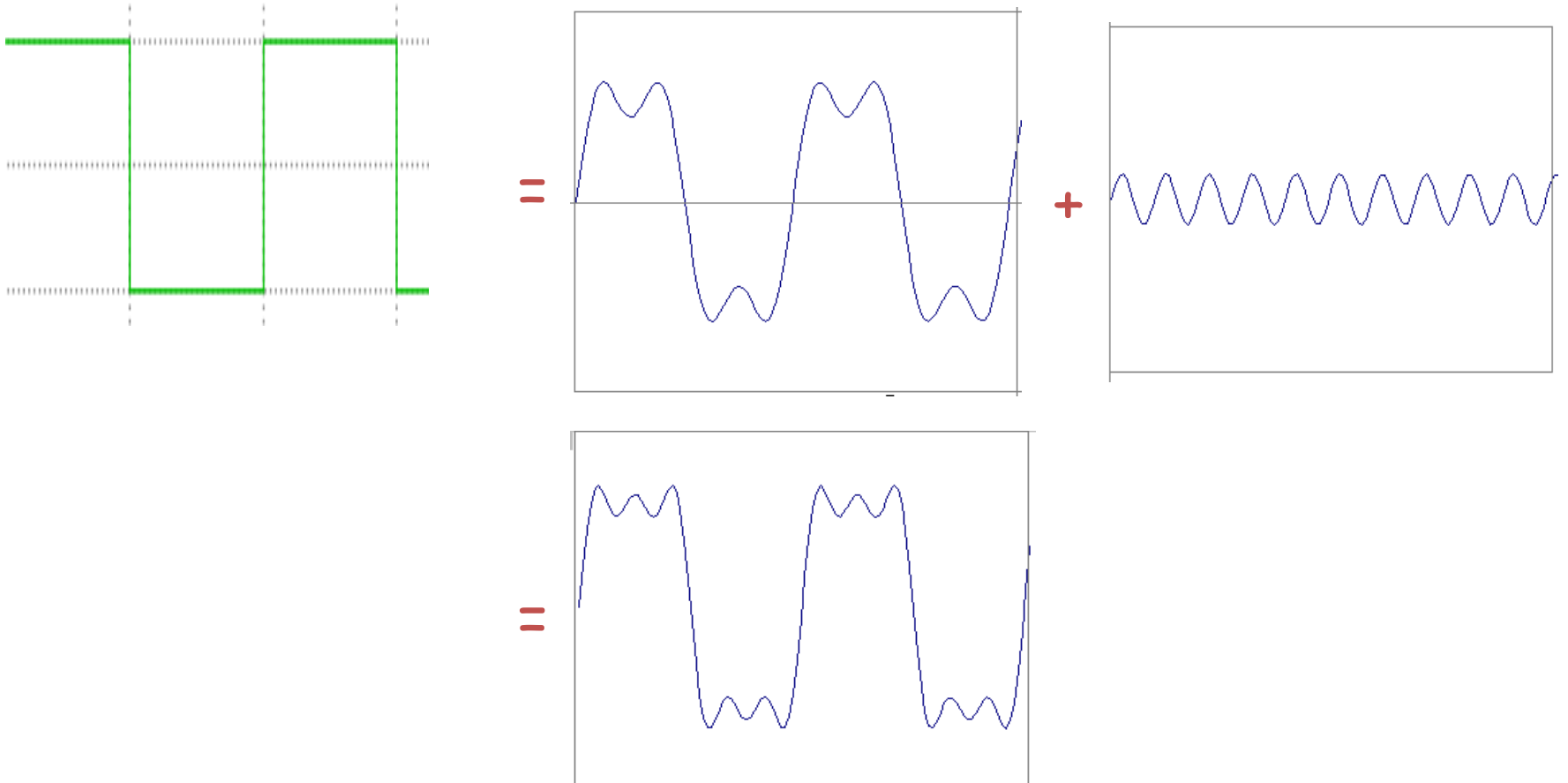
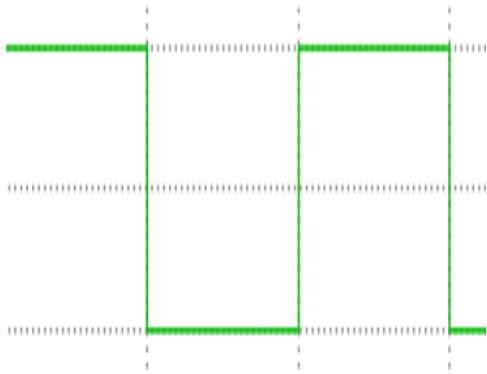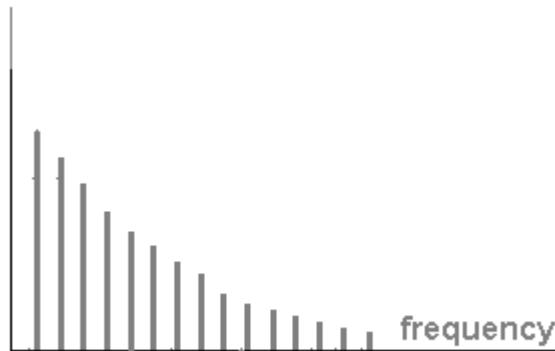# Frequency Spectra
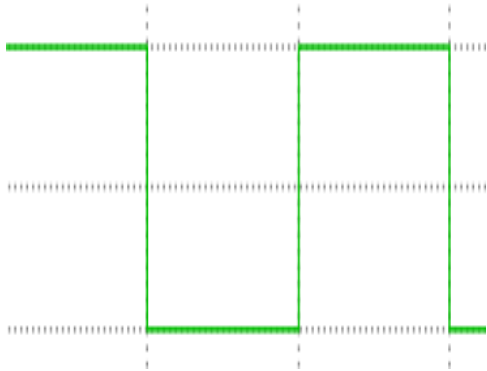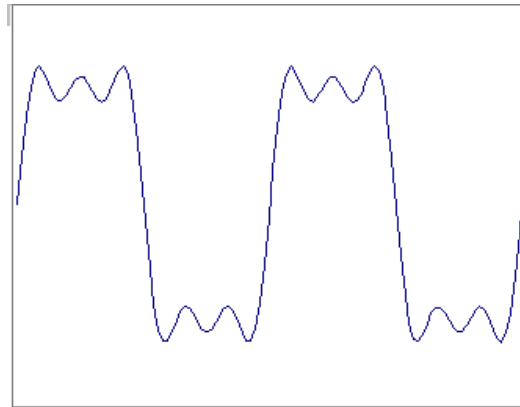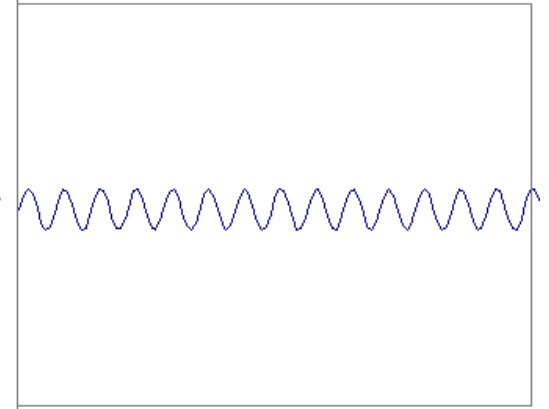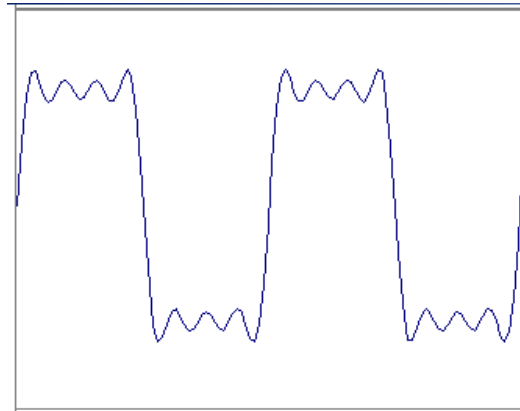
Frequency Spectra
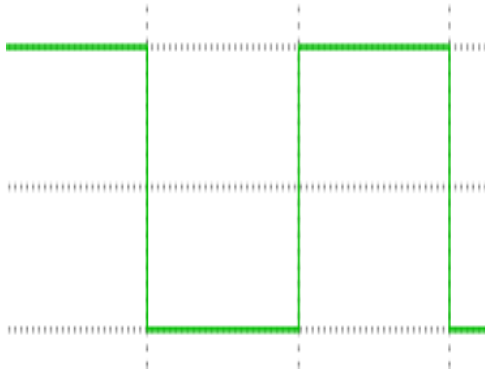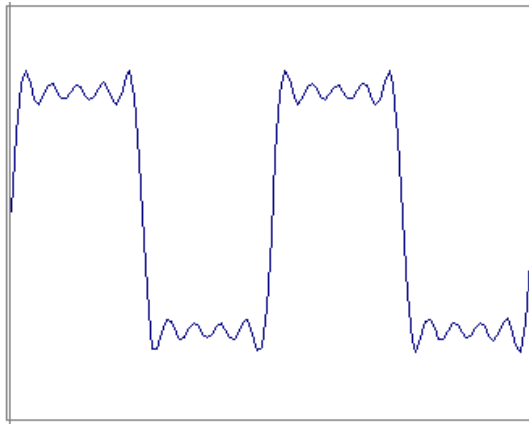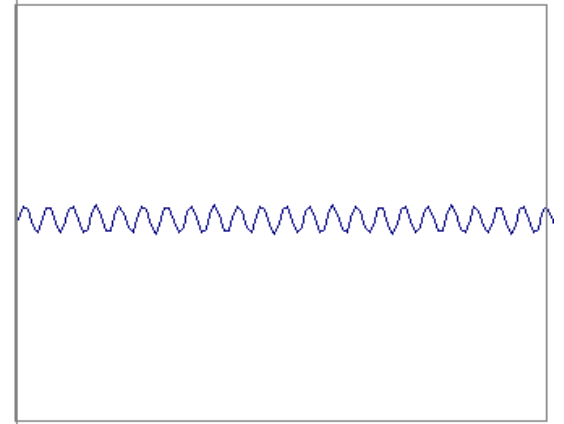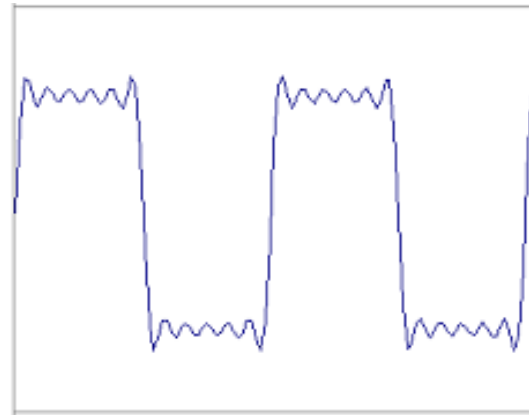
Frequency Spectra
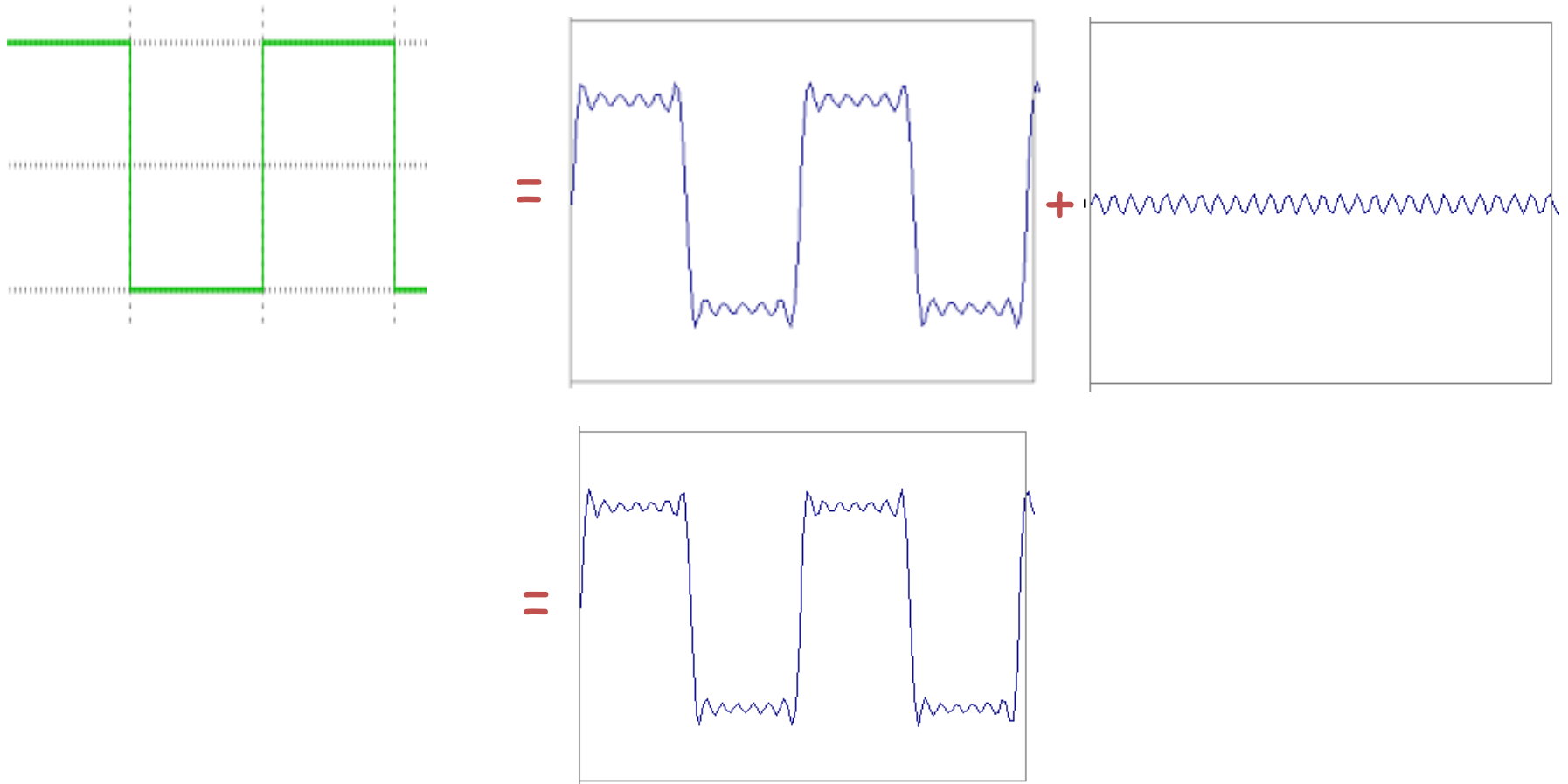
Frequency Spectra

Frequency Spectra
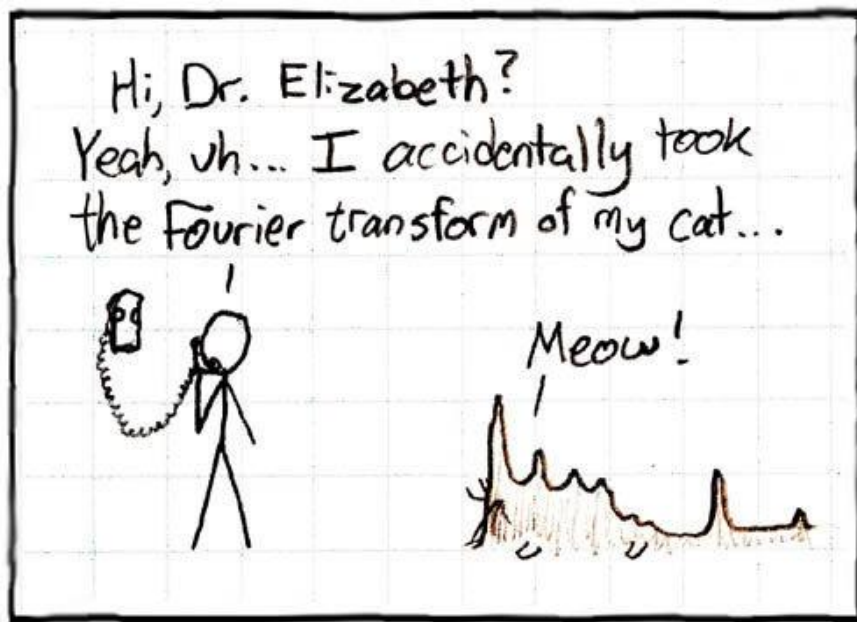
$$= A\sum_{k=1}^{\infty}\frac{1}{k}\sin(2\pi kt)$$

Example: Music

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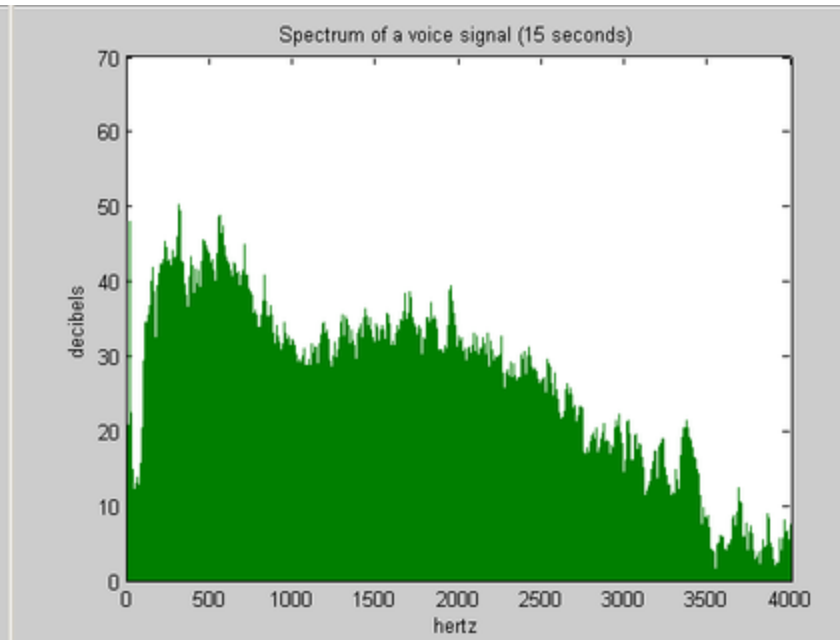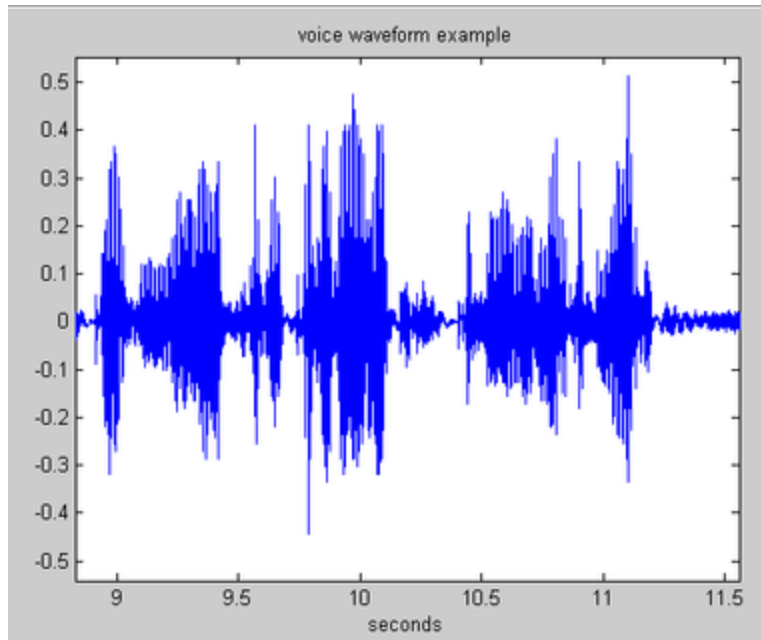
Other signals

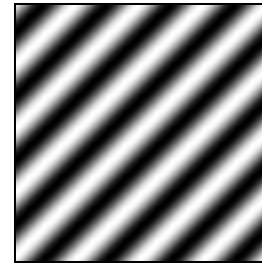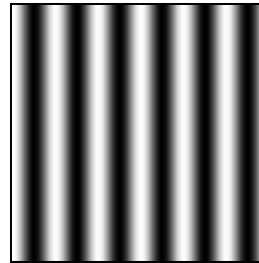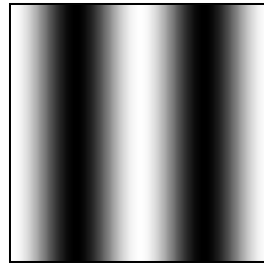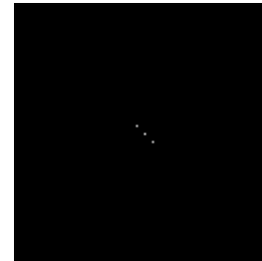- We can also think of all kinds of other signals the same way

Cats(?)

xkcd.com
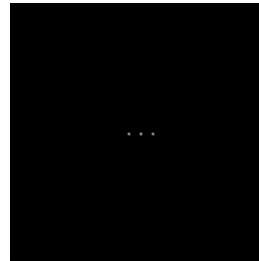
FFT of
my cat
Tesla
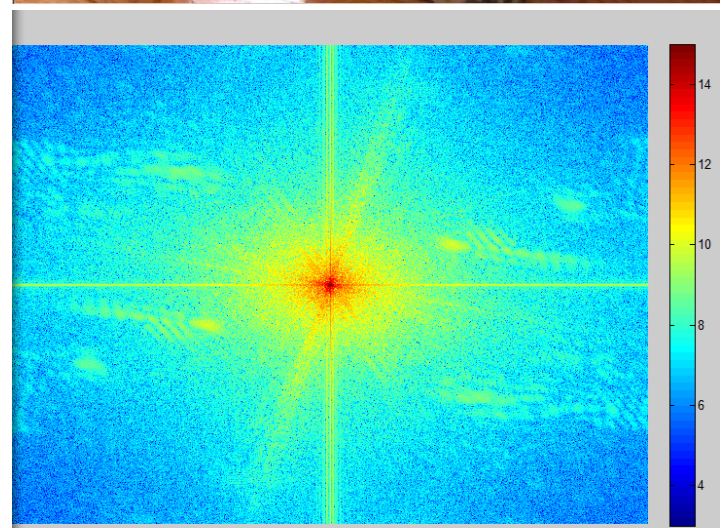
Fourier analysis in images

Intensity Image

Fourier Image

Signals can be composed

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

Strong Vertical Frequency
(Sharp Horizontal Edge)
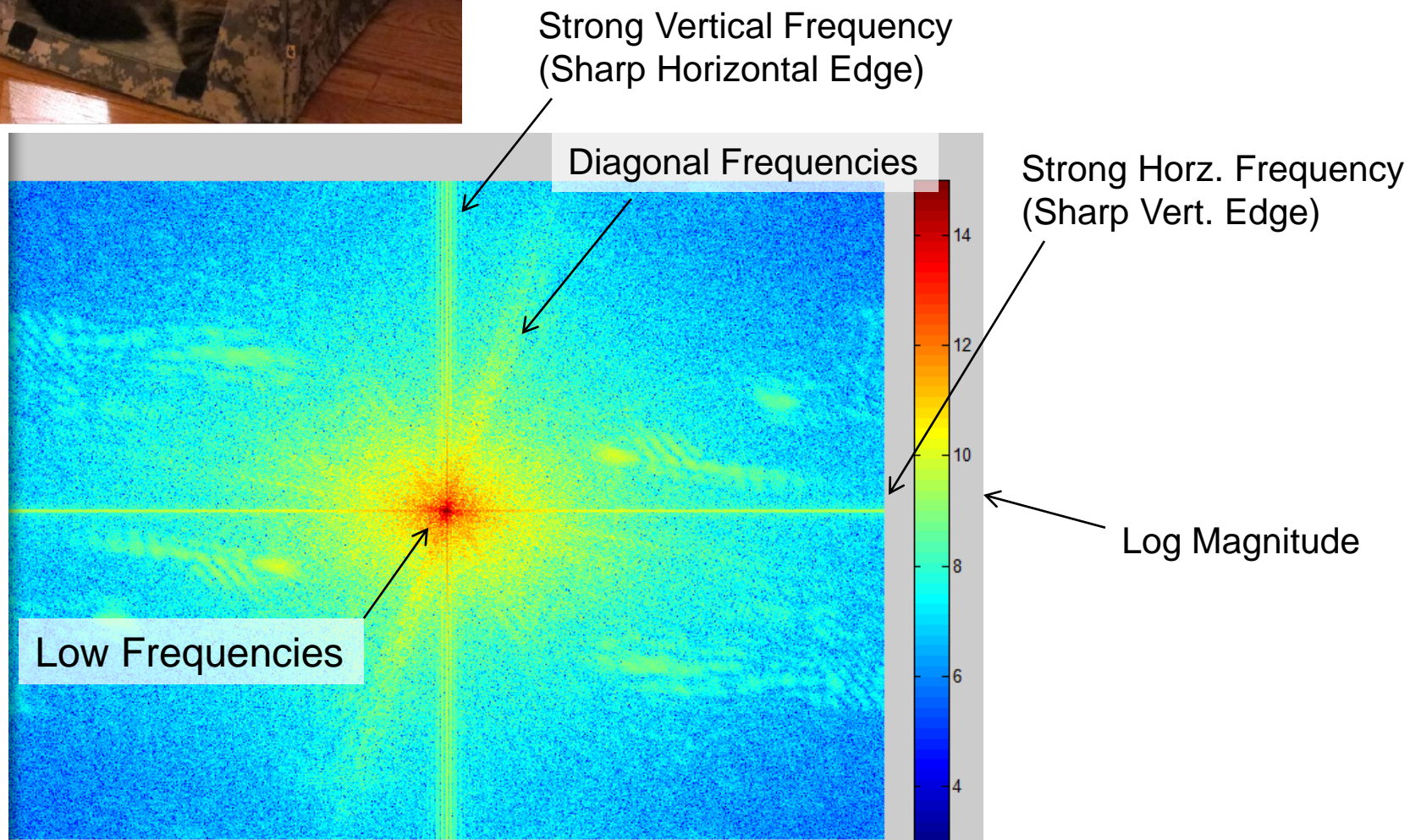
Diagonal Frequencies

Strong Horz. Frequency
(Sharp Vert. Edge)

Log Magnitude

Low Frequencies
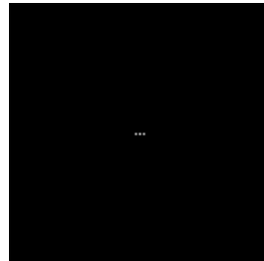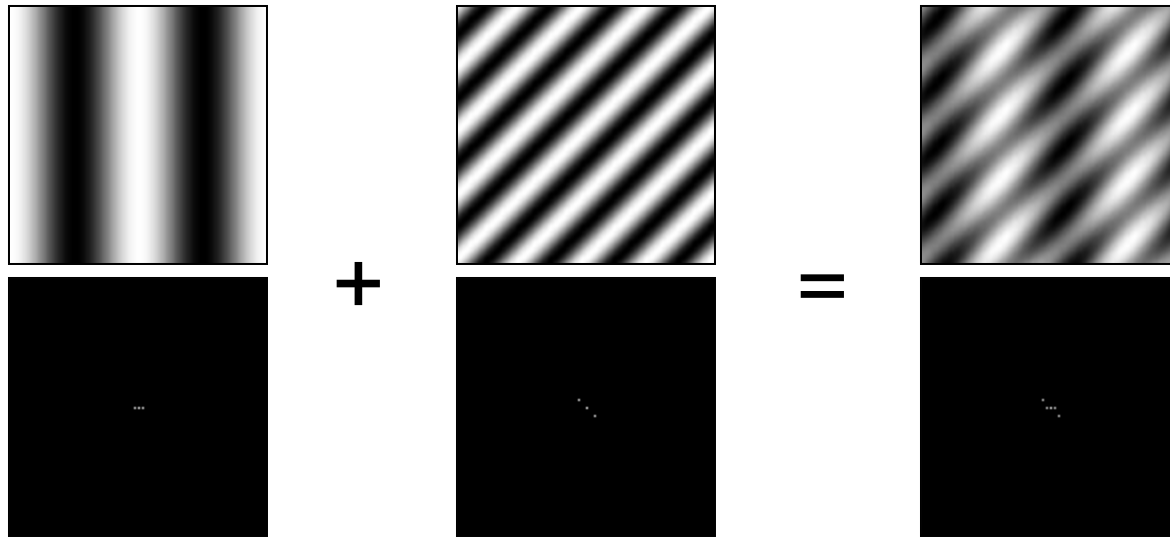
Fourier Transform

- 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

Amplitude: $A = \pm\sqrt{R(\omega)^2 + I(\omega)^2}$

Phase: $\phi = \tan^{-1}\dfrac{I(\omega)}{R(\omega)}$

Euler's formula: $e^{inx} = \cos(nx) + i\sin(nx)$

Computing the Fourier Transform

$$H(\omega) = \mathcal{F}\{h(x)\} = Ae^{j\phi}$$

Continuous

$$H(\omega) = \int_{-\infty}^{\infty} h(x)e^{-j\omega x}\,dx$$

Discrete

$$H(k) = \frac{1}{N}\sum_{x=0}^{N-1} h(x)e^{-j\frac{2\pi k x}{N}} \qquad k = \text{-}N/2..N/2$$

Fast Fourier Transform (FFT): NlogN

The Convolution Theorem

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

$$\mathrm{F}[g * h] = \mathrm{F}[g]\,\mathrm{F}[h]$$

- The inverse Fourier transform of the product of two Fourier transforms is the convolution of the two inverse Fourier transforms

$$\mathrm{F}^{-1}[gh] = \mathrm{F}^{-1}[g] * \mathrm{F}^{-1}[h]$$

- **Convolution** in spatial domain is equivalent to **multiplication** in frequency domain!

Properties of Fourier Transforms

- Linearity $\mathcal{F}[ax(t) + by(t)] = a\mathcal{F}[x(t)] + b\mathcal{F}[y(t)]$

- Fourier transform of a real signal is symmetric about the origin

- The energy of the signal is the same as the energy of its Fourier transform

See Szeliski Book (3.4)

Filtering in spatial domain

1	0	-1
2	0	-2
1	0	-1

intensity image

* =
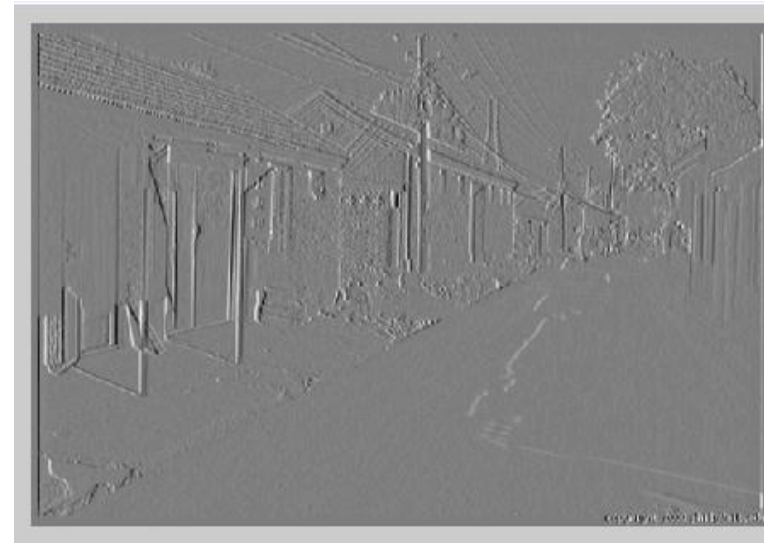
Filtering in frequency domain

FFT

intensity image

FFT

log fft magnitude

X

=

Inverse FFT
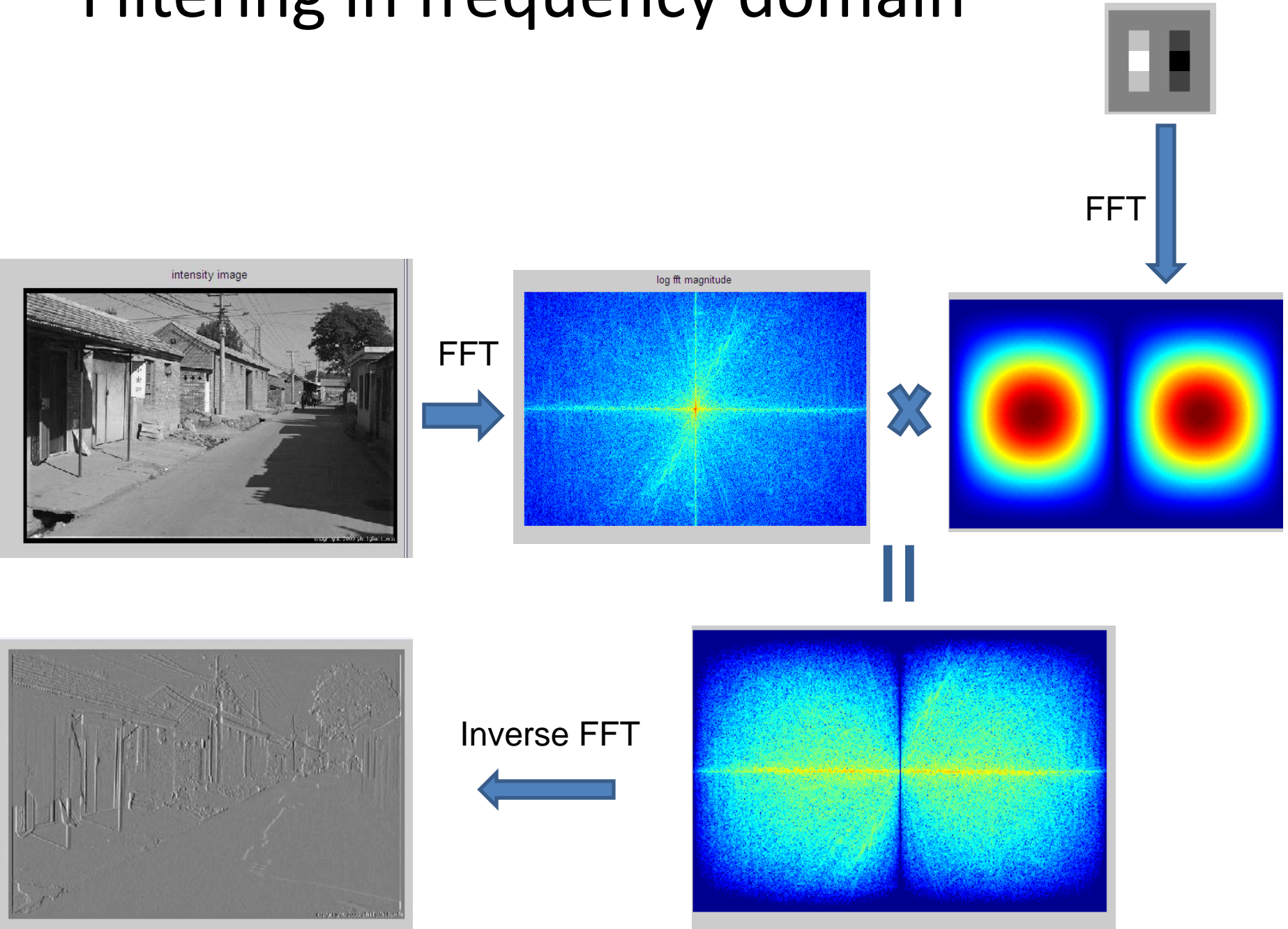
Fourier Matlab demo

FFT in Matlab

- ## Filtering with fft

```
im = ... % "im" should be a gray-scale floating point image
[imh, imw] = size(im);
fftsize = 1024; % should be order of 2 (for speed) and include padding
im_fft = fft2(im, fftsize, fftsize); % 1) fft im with padding
hs = 50; % filter half-size
fil = fspecial('gaussian', hs*2+1, 10);
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

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

Questions

Which has more information, the phase or the magnitude?

What happens if you take the phase from one image and combine it with the magnitude from another image?

Filtering

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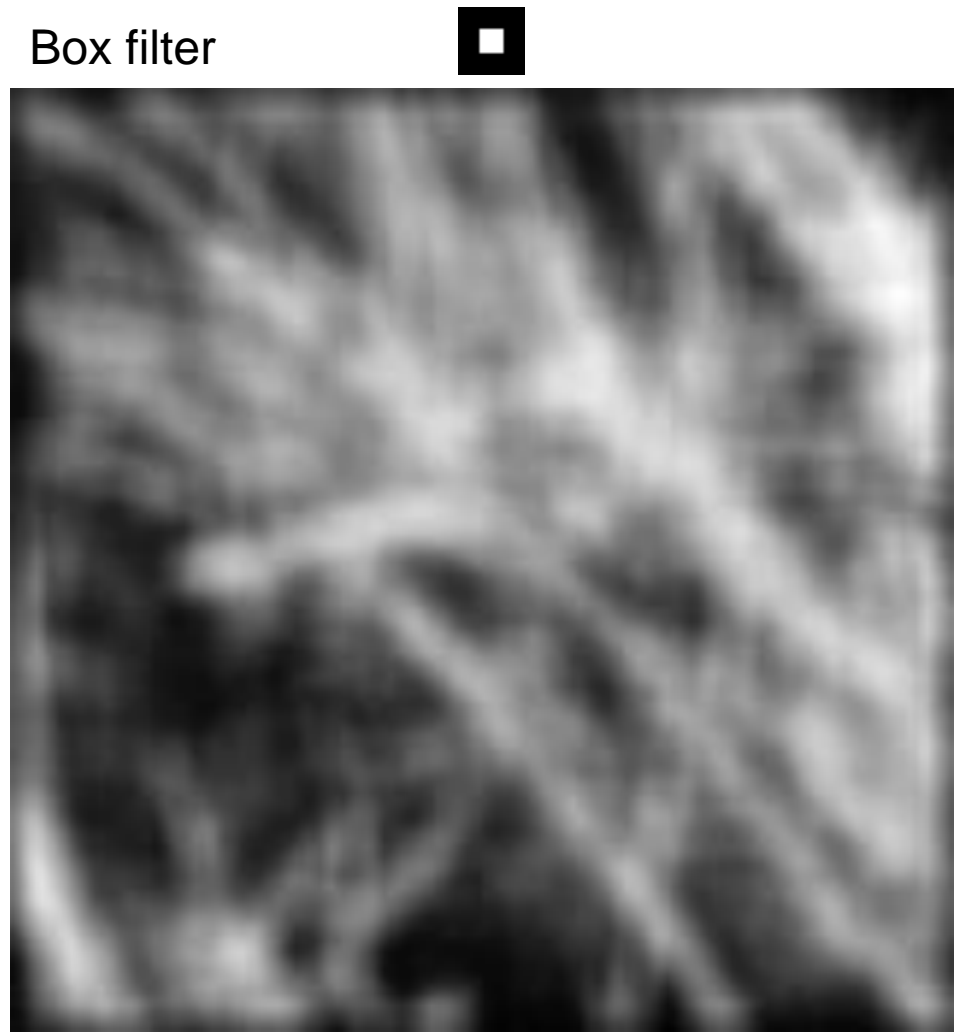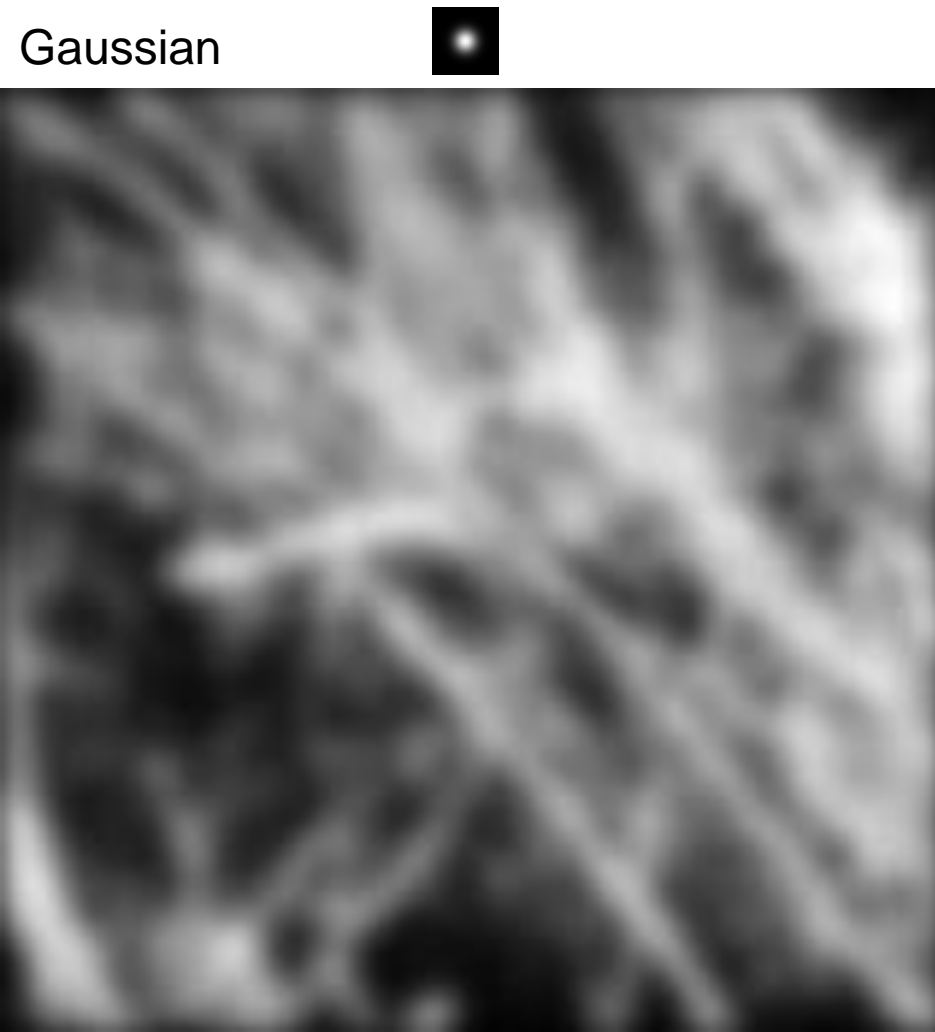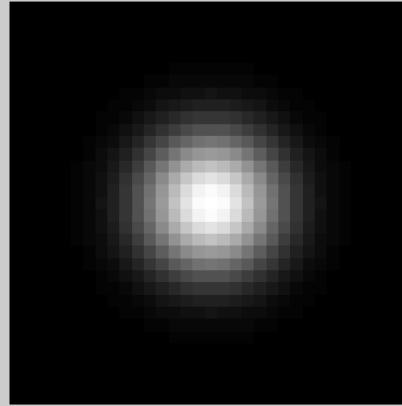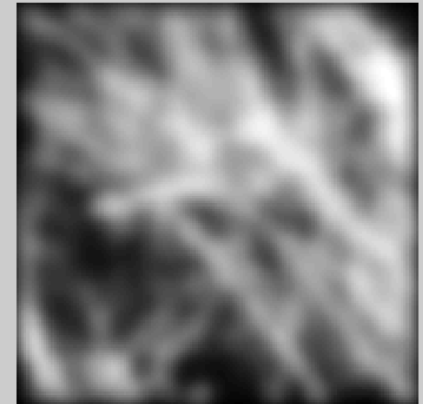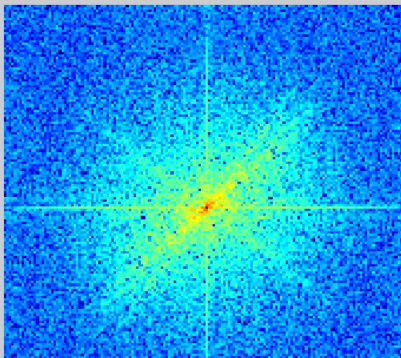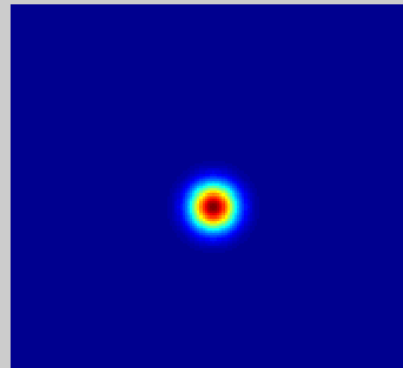
Gaussian

Box filter

Gaussian
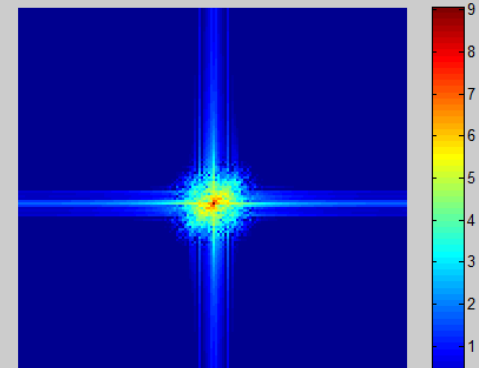
Box Filter
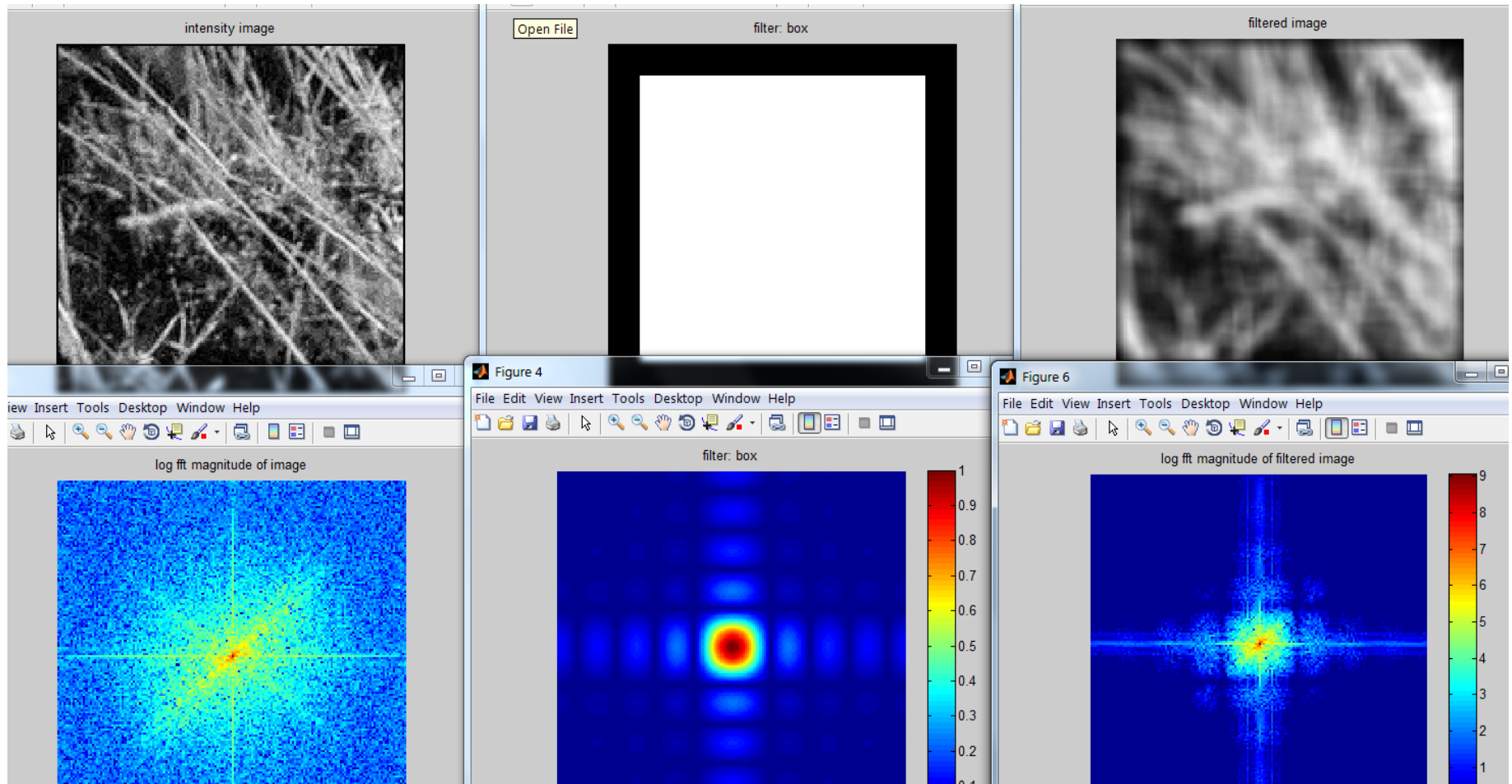
Sampling

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

Image: http://www.flickr.com/photos/igorms/136916757/
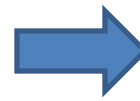
Subsampling by a factor of 2

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

Aliasing problem
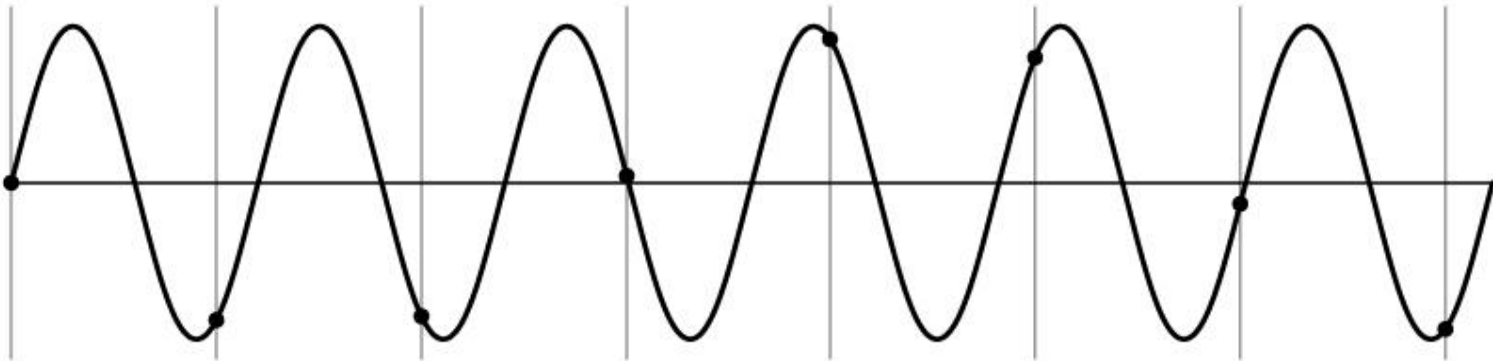
- 1D example (sinewave):

Aliasing problem

- 1D example (sinewave):

Source: S. Marschner
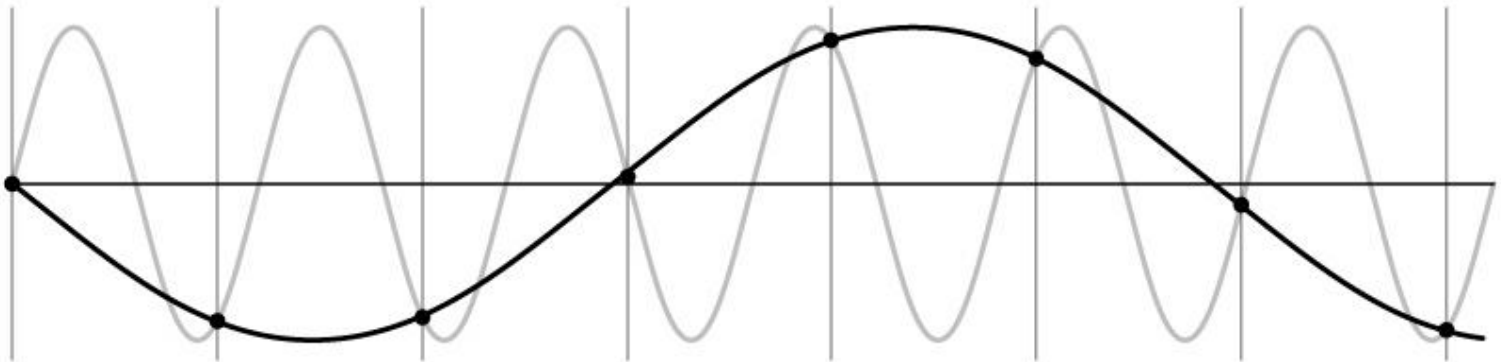
Aliasing problem

- 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):

frame 0   frame 1   frame 2   frame 3   frame 4

shutter open   time
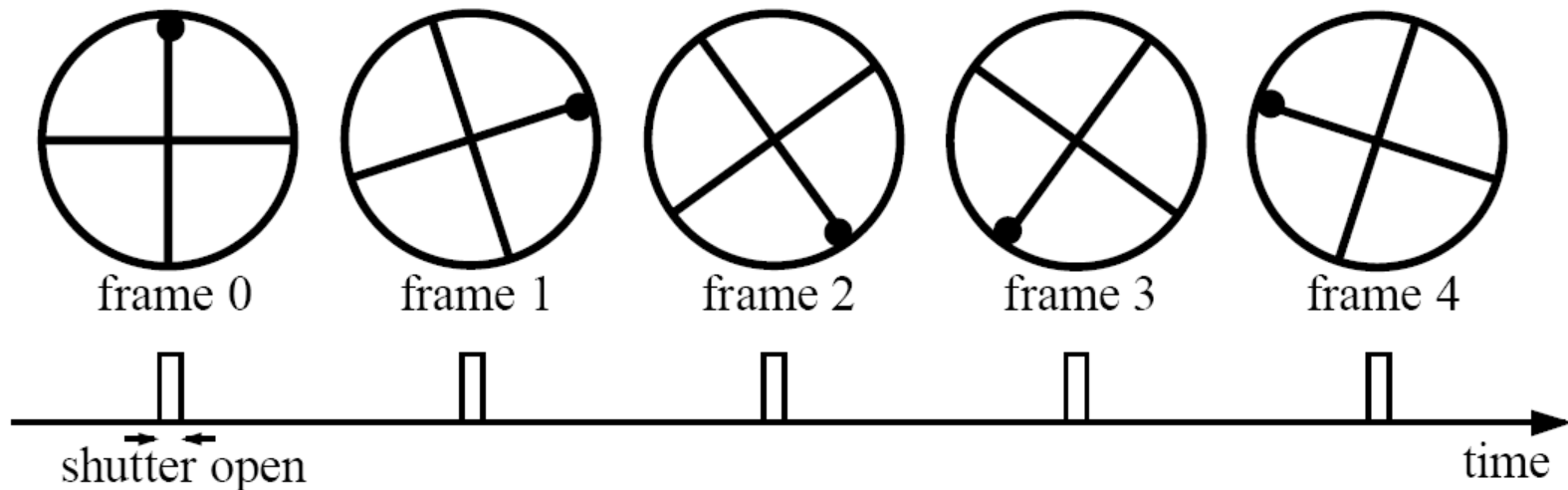
Without dot, wheel appears to be rotating slowly backwards! (counterclockwise)

Aliasing in graphics

Disintegrating textures

Source: A. Efros

Sampling and aliasing

256x256 128x128 64x64 32x32 16x16
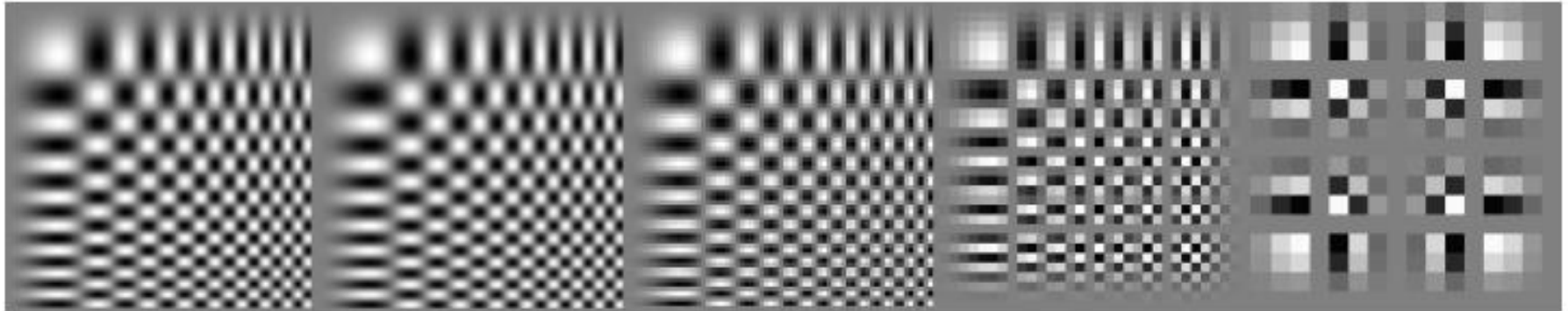
Nyquist-Shannon Sampling Theorem

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

good

bad
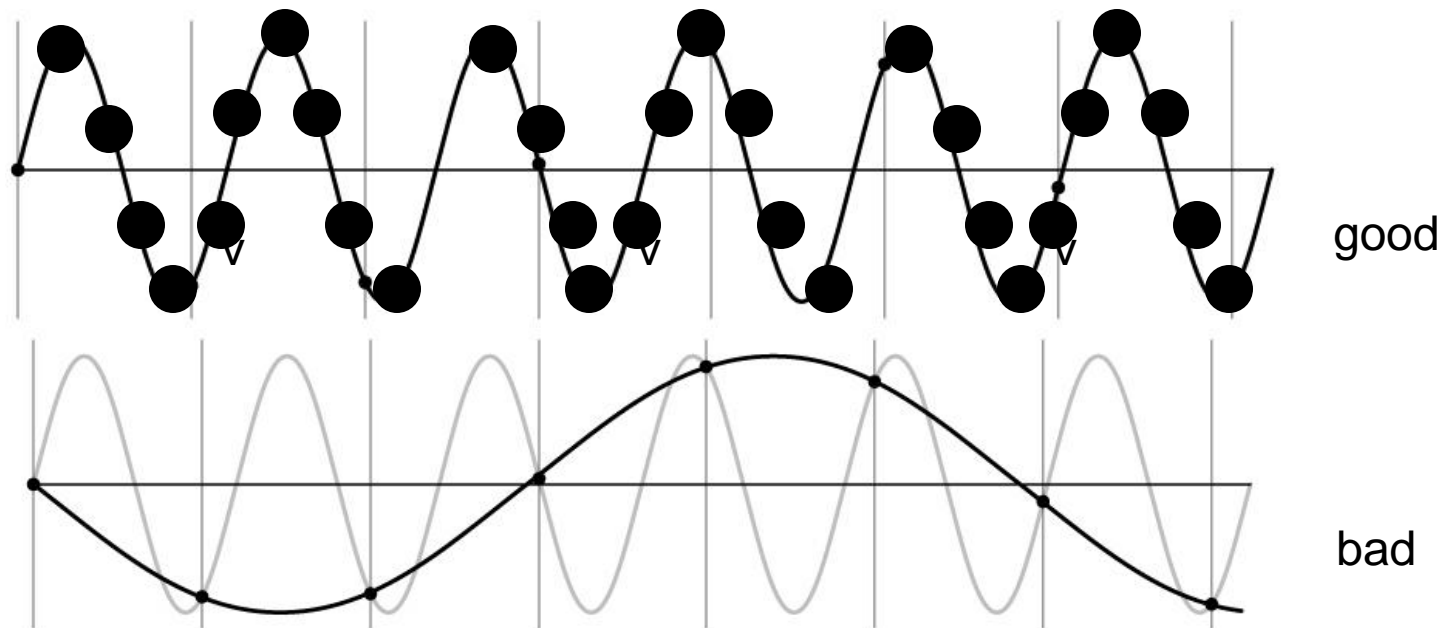
Anti-aliasing

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 factor of 2

1. Start with image(h, w)

2. Apply low-pass filter

    im_blur = imfilter(image, fspecial('gaussian', 7, 1))

3. Sample every other pixel

    im_small = im_blur(1:2:end, 1:2:end);

Anti-aliasing

256x256   128x128   64x64   32x32   16x16
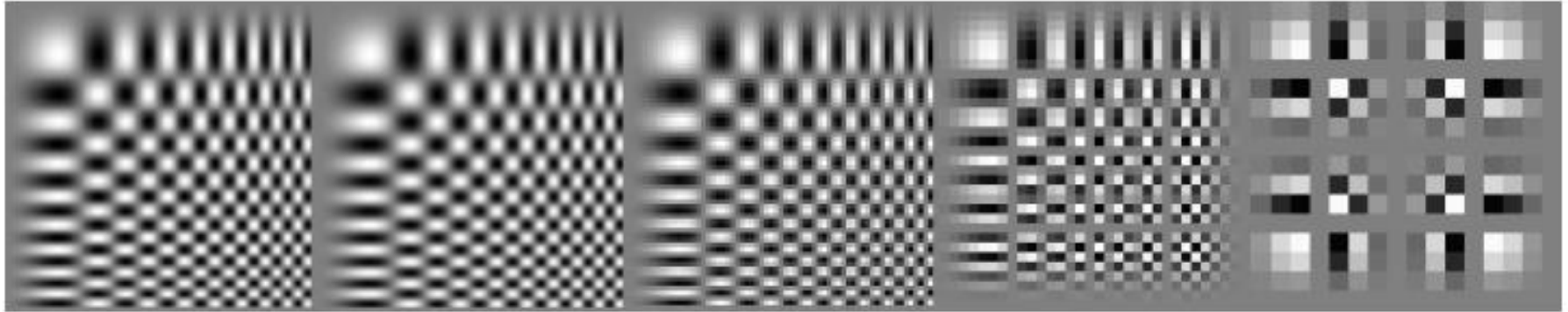
256x256   128x128   64x64   32x32   16x16
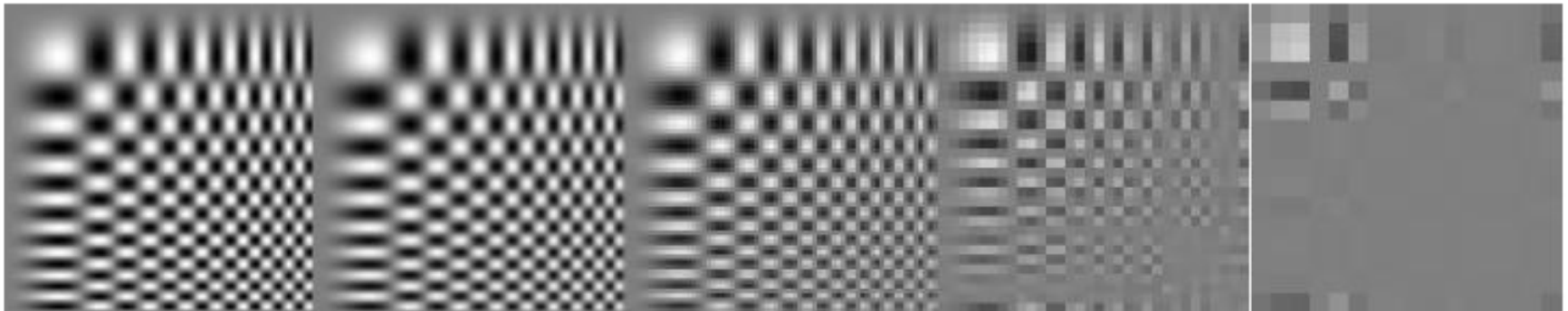
Subsampling without pre-filtering

1/2   1/4   (2x zoom)   1/8   (4x zoom)

Subsampling with Gaussian pre-filtering

Gaussian 1/2

G 1/4

G 1/8

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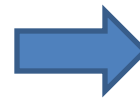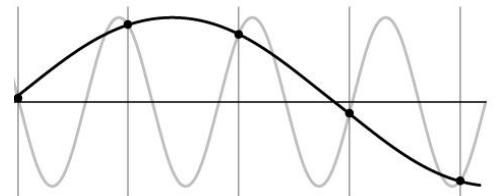
Why do we get different, distance-dependent interpretations of hybrid images?
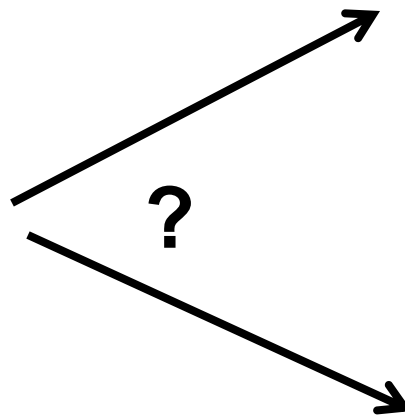
Clues from Human Perception

- 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
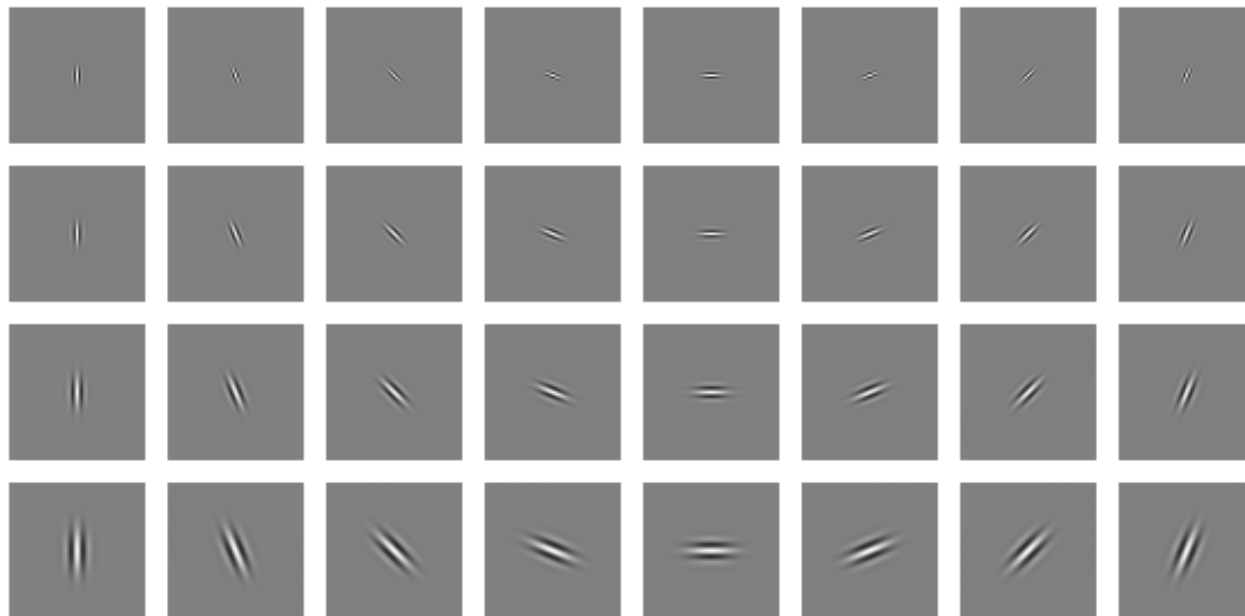
Early Visual Processing: Multi-scale edge and blob filters

Hybrid Image in FFT

Hybrid Image
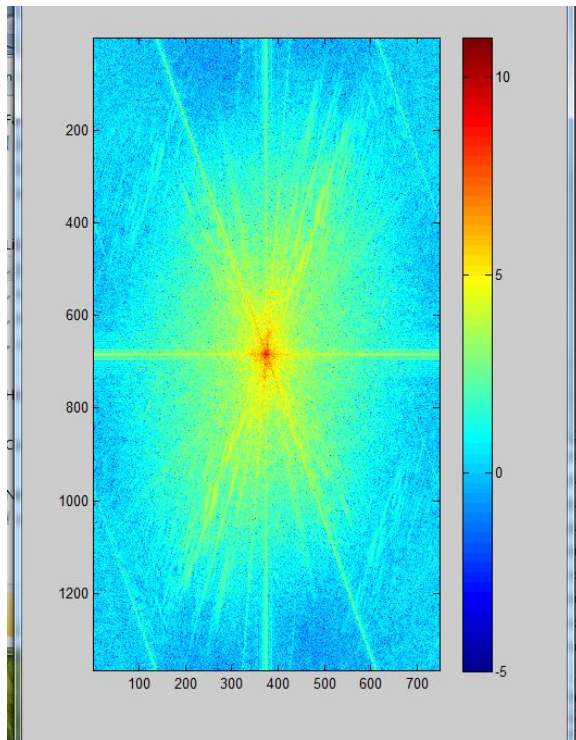
Low-passed Image

High-passed Image
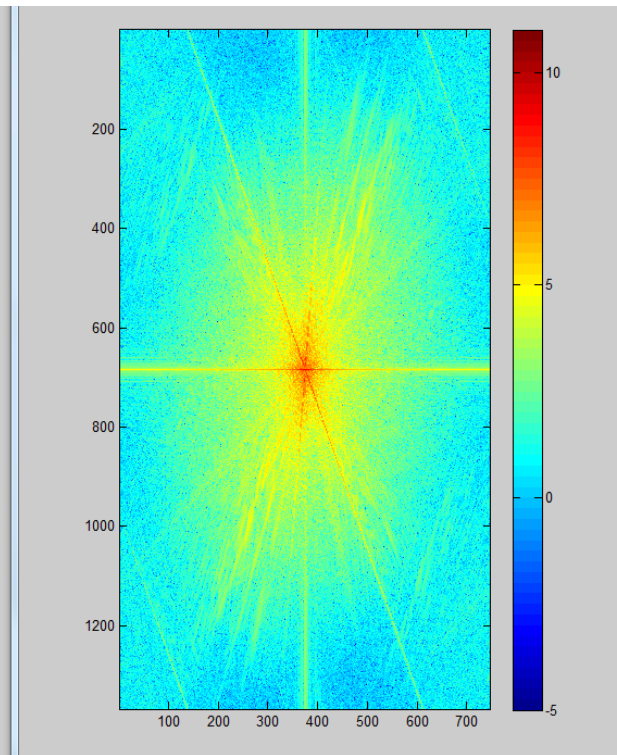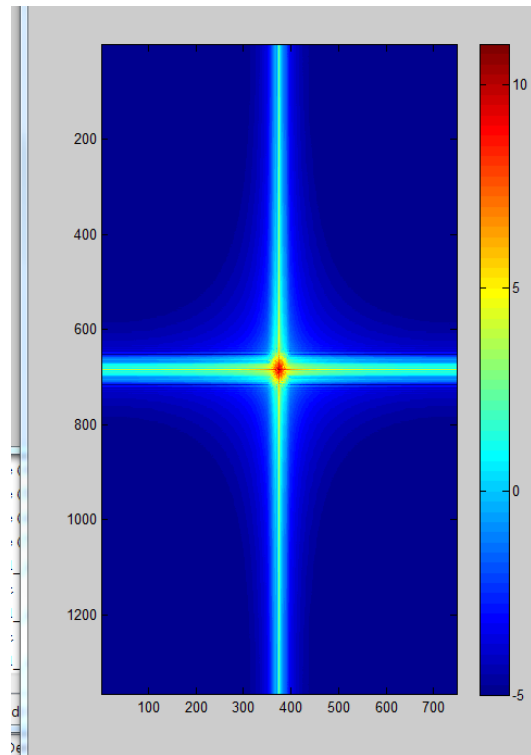
Perception

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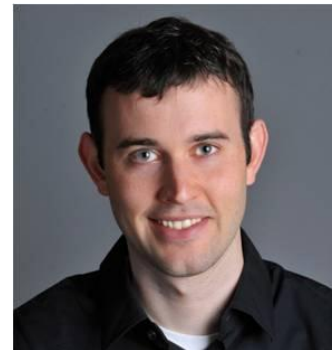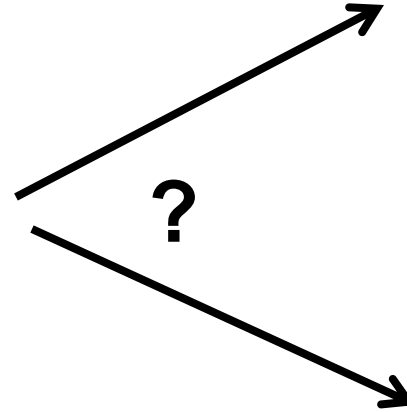
Things to Remember

- 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
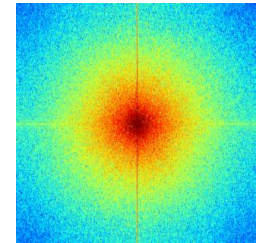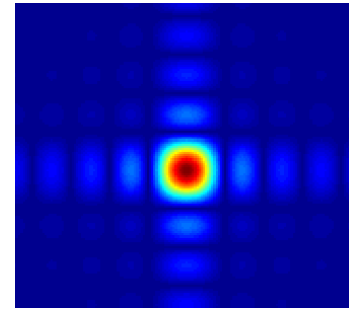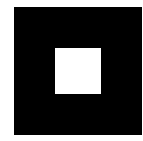
Practice question

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

Next class

- Template matching

- Image Pyramids

- Filter banks and texture

- Denoising, Compression