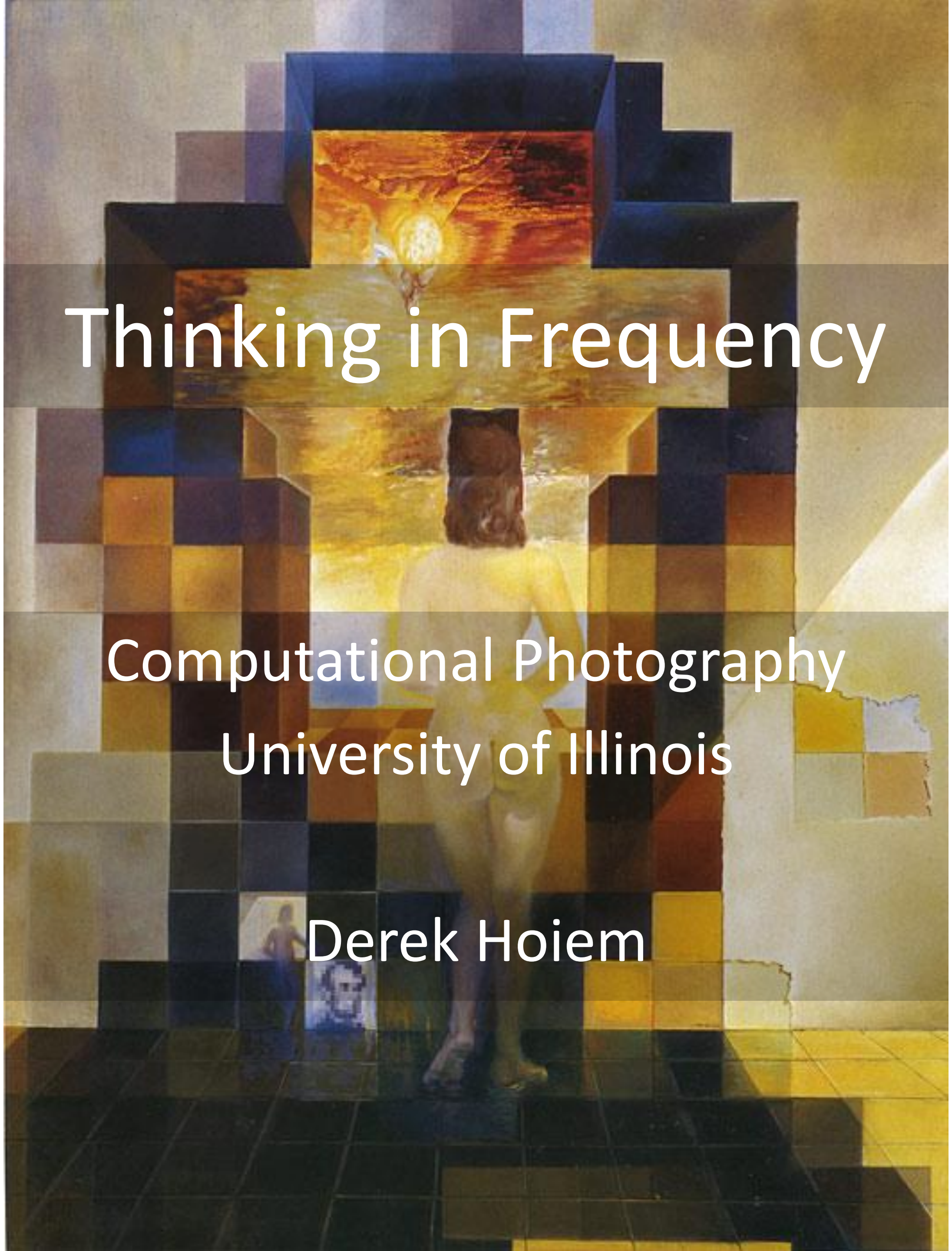
09/03/13



Thinking in Frequency

Computational Photography
University of Illinois

Derek Hoiem

Administrative

Website: http://courses.engr.illinois.edu/cs498dh3/

Doodle for tutorial: http://doodle.com/f7bien4sadwfvz64

Doodle for office hours: http://doodle.com/e9hzmy65k2icdi64

Bulletin board: https://piazza.com/class/hkvgrp10k92250

Project 1: due in ~2 weeks (see project page)

Talk to me after class about any questions or concerns

About me

Raised in “upstate” NY

Rochester
Buffalo
Niagara Falls
Syracuse
Utica
Oswego
Watertown
Ogdensburg
Massena
Plattsburgh
Glens Falls
Saratoga Springs
Schenectady
Troy
Albany
Kingston
Poughkeepsie
Newburgh
Middletown
Yonkers
New Rochelle
New York
Ithaca
Elmira
Binghamton
Jamestown
New York

About me

1998-2002
Undergrad at SUNY Buffalo
B.S., EE and CSE

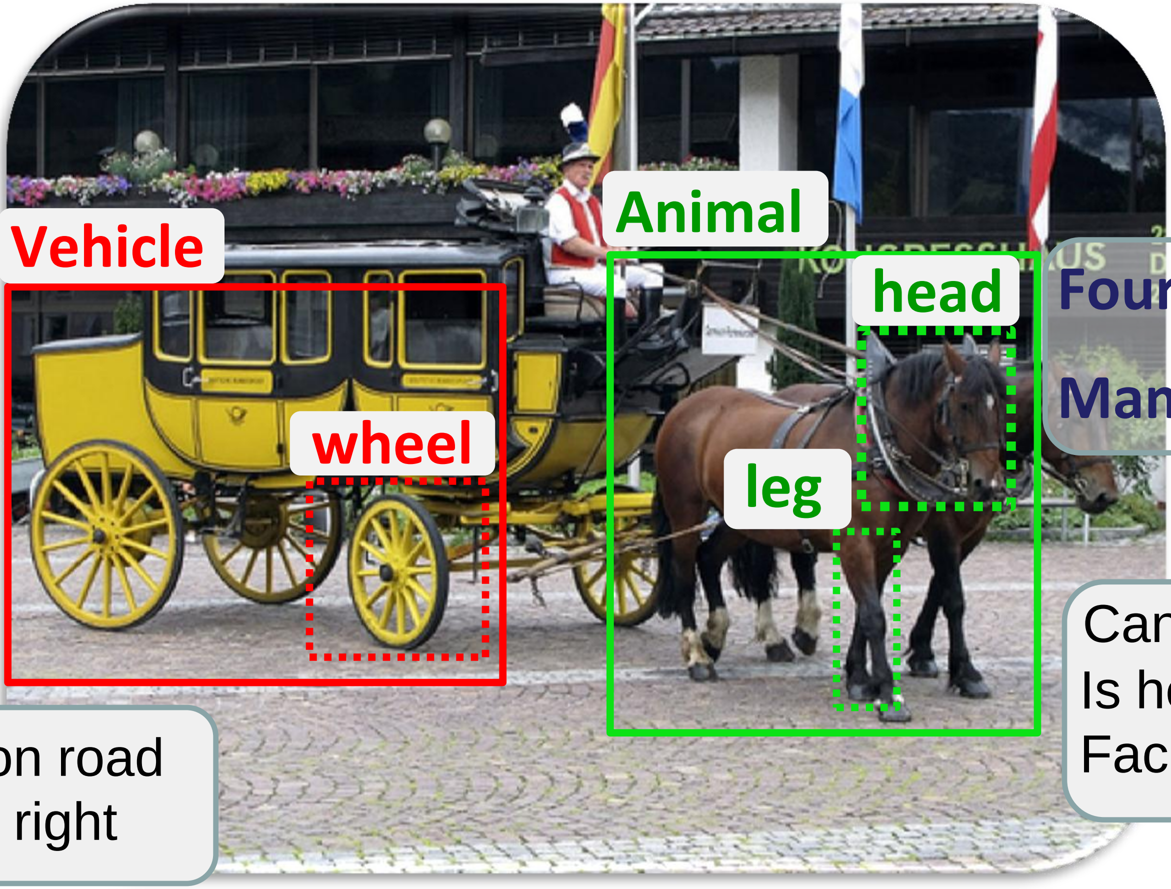
2002-2007
Grad at Carnegie Mellon
Ph.D. in Robotics

2007-2008
Postdoc at Beckman Institute

2009-
Assistant Prof in CS at UIUC

My research

My Research



Four-legged

Mammal

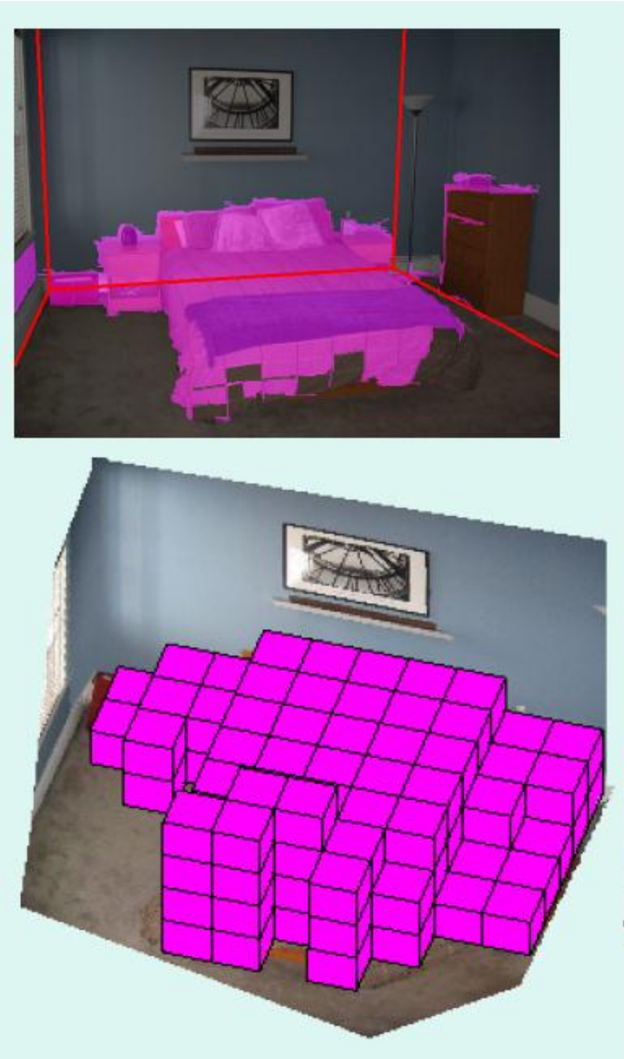
Can run, jump
Is herbivorous
Facing right

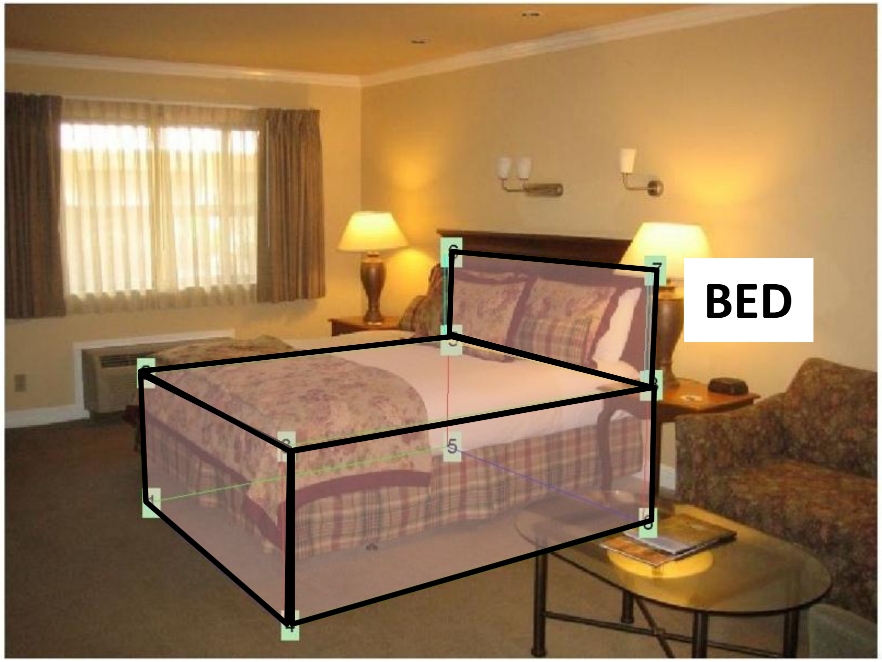
Move on road
Facing right

Farhadi et al. 2010

My Research

Recovering 3D layout and context

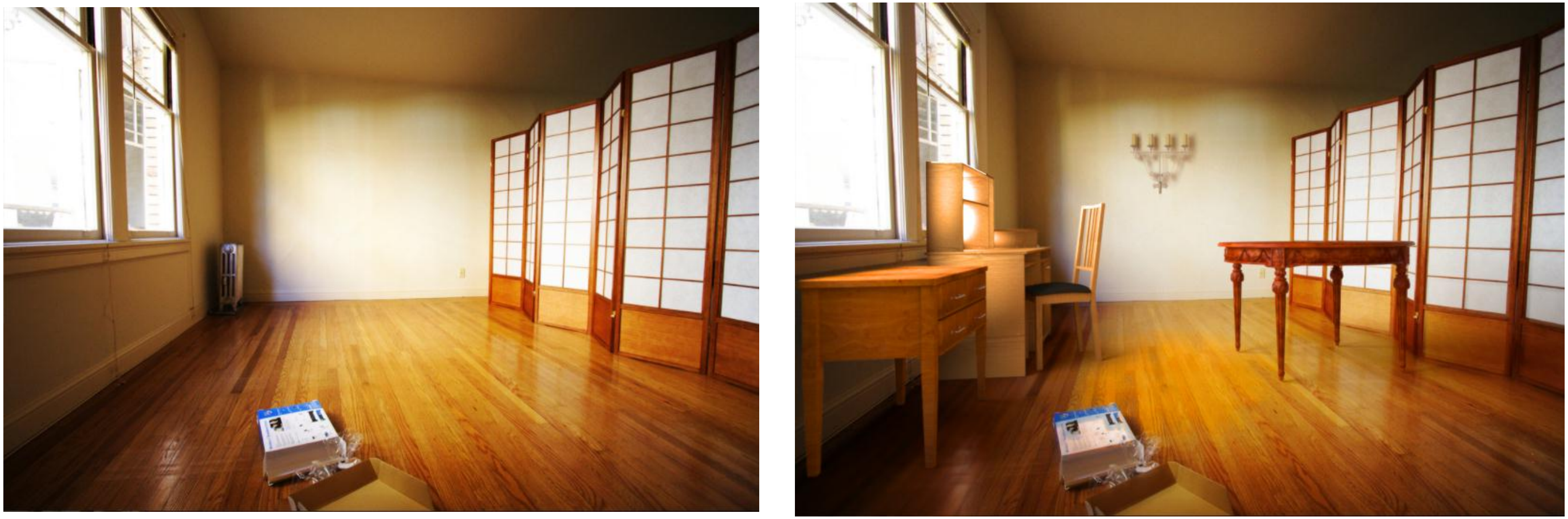




Hedau et al. 2009, 2010

My Research

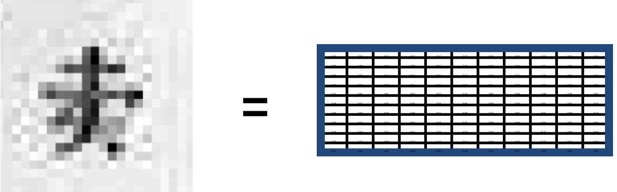
Editing images as if they were 3D scenes



Karsch et al. 2011

Last class

- Image is a matrix of numbers



- Linear filtering is a dot product at each position
  - Can smooth, sharpen, translate (among many other uses)

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- Be aware of details for filter size, extrapolation, cropping
  - Filter size should be large enough so that values at edges of filter are near 0
  - Careful to distinguish between bandwidth/sigma of Gaussian (how broad the function is) with the filter size (where you cut it off)

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:

```
gradx(y,x) = im(y,x+1)-im(y,x) for each x, y
```

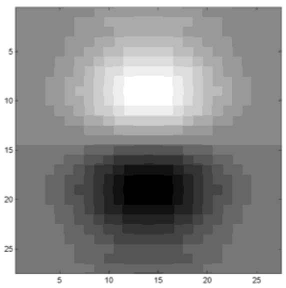
Review: questions

3. Fill in the blanks:

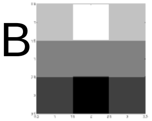
Filtering Operator

```
a) _ = D * B
b) A = _ * _
c) F = D * _
d) _ = D * D
```

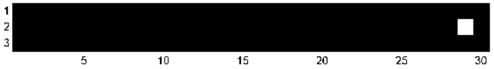
A



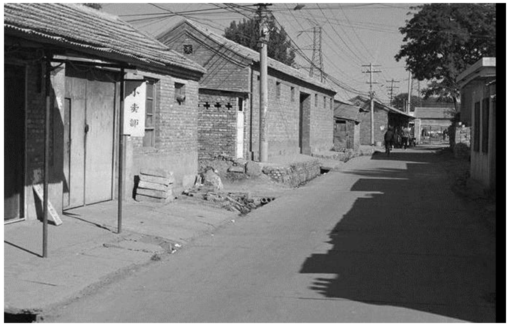
B



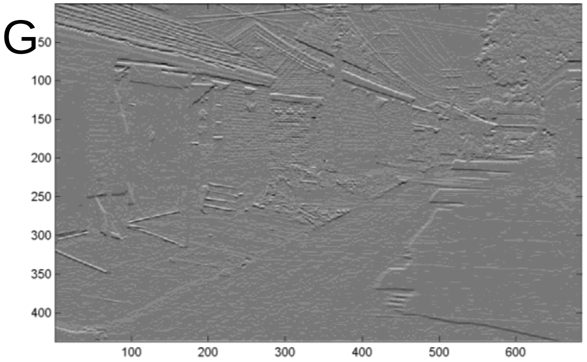
E



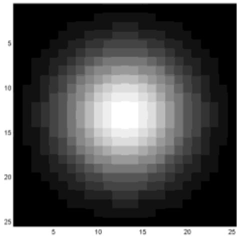
F



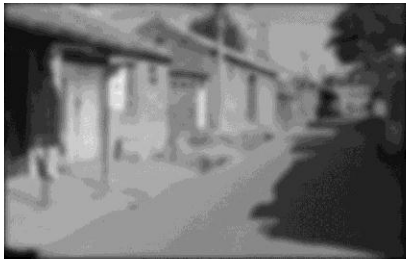
G



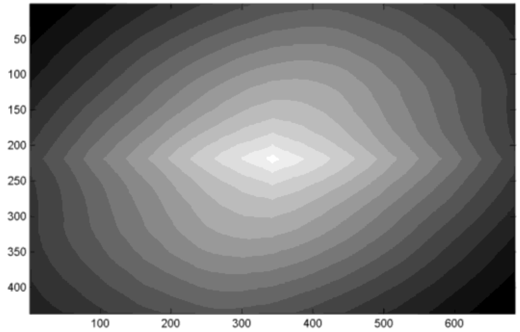
C



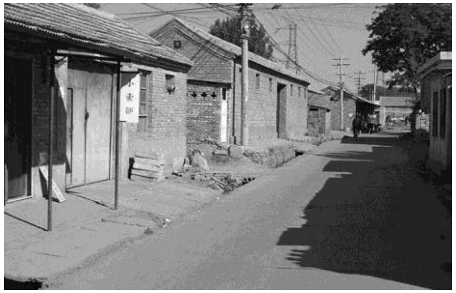
H



I



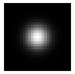
D

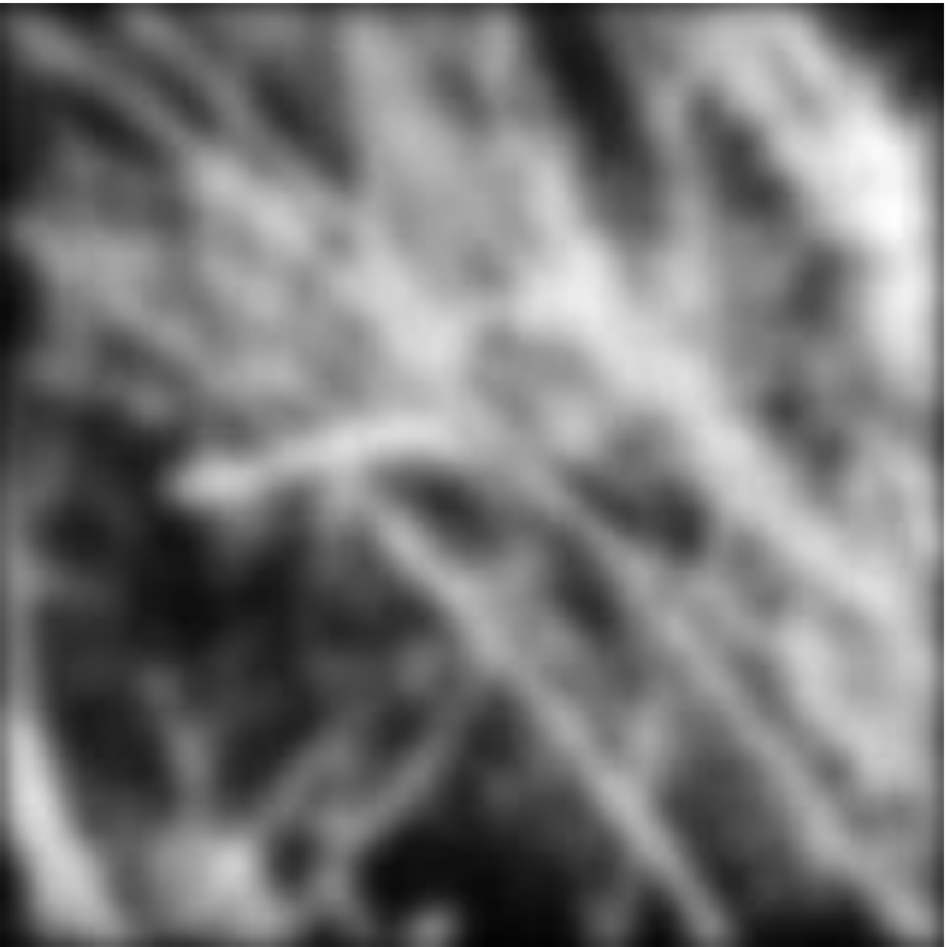


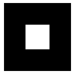
Today’s class

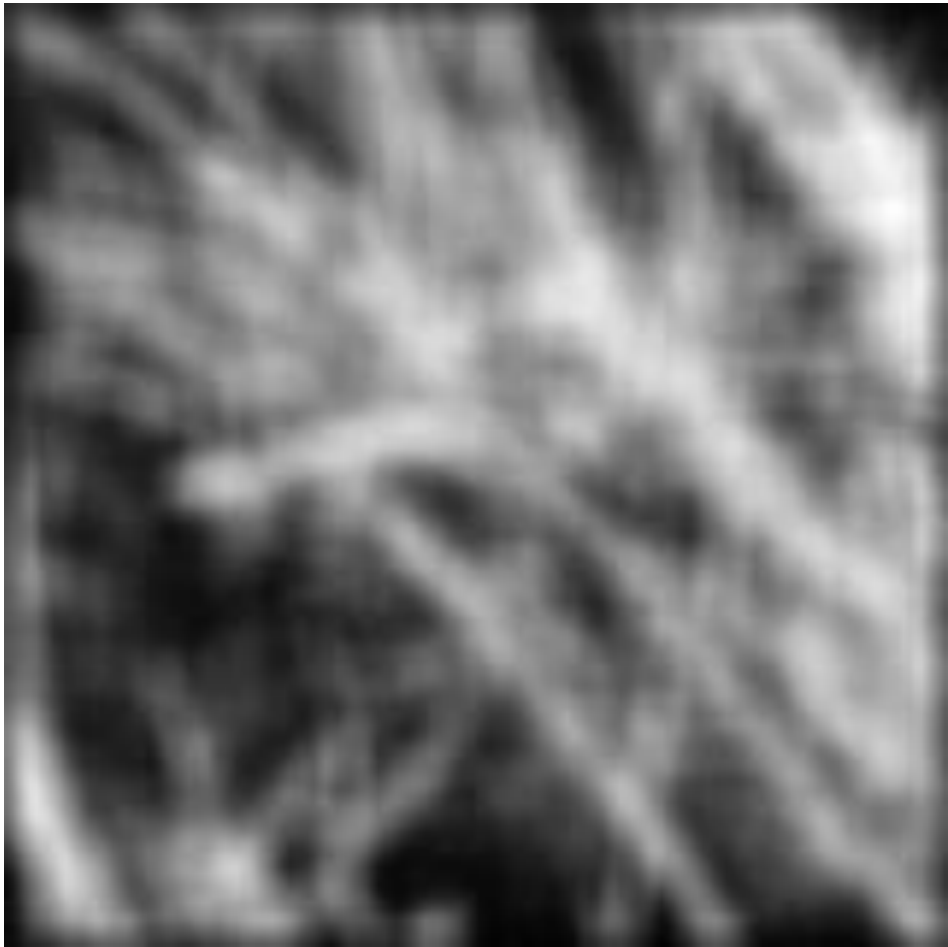
- Fourier transform and frequency domain
  - Frequency view of filtering
  - Another look at hybrid images
  - Sampling

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

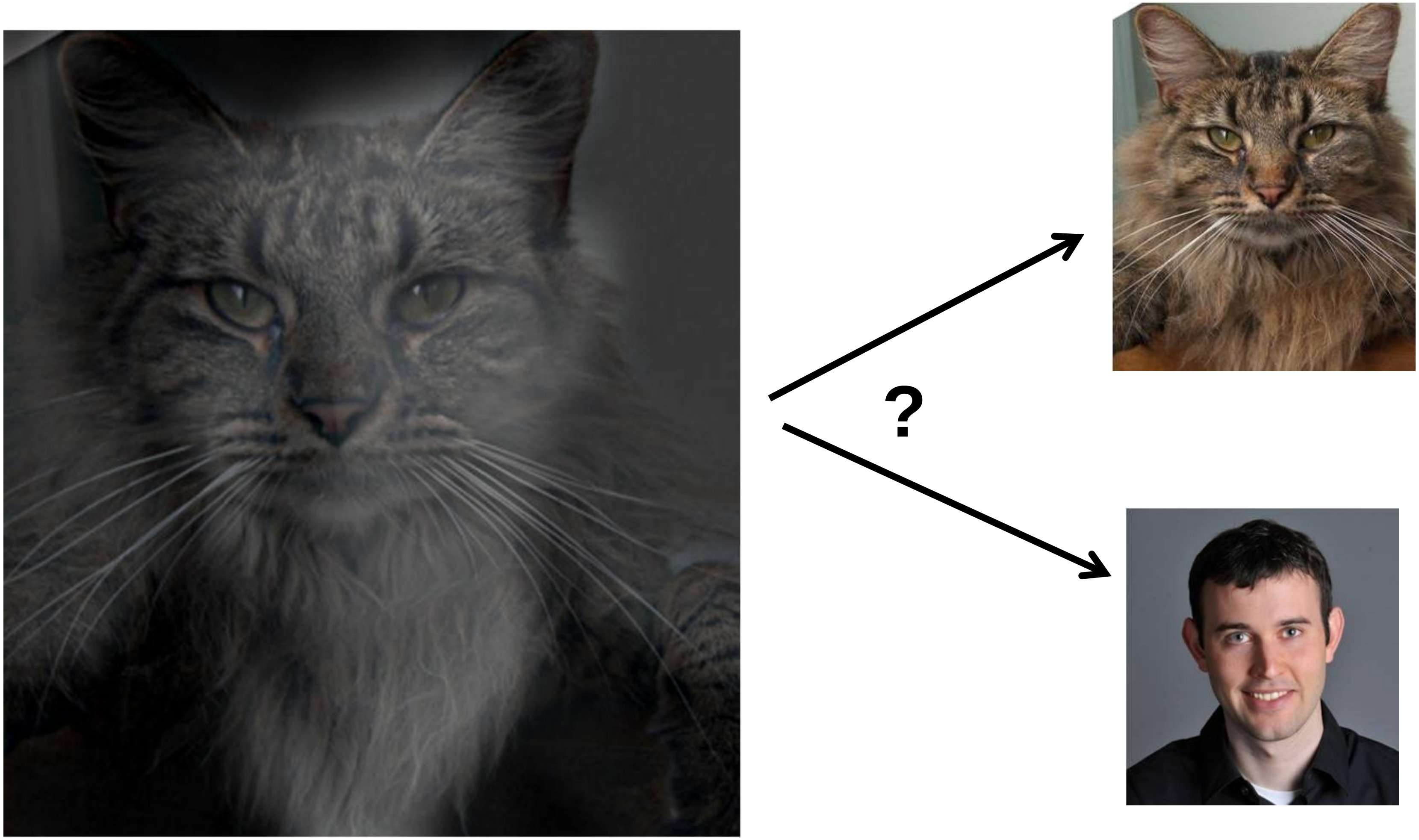
Gaussian 



Box filter 



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?

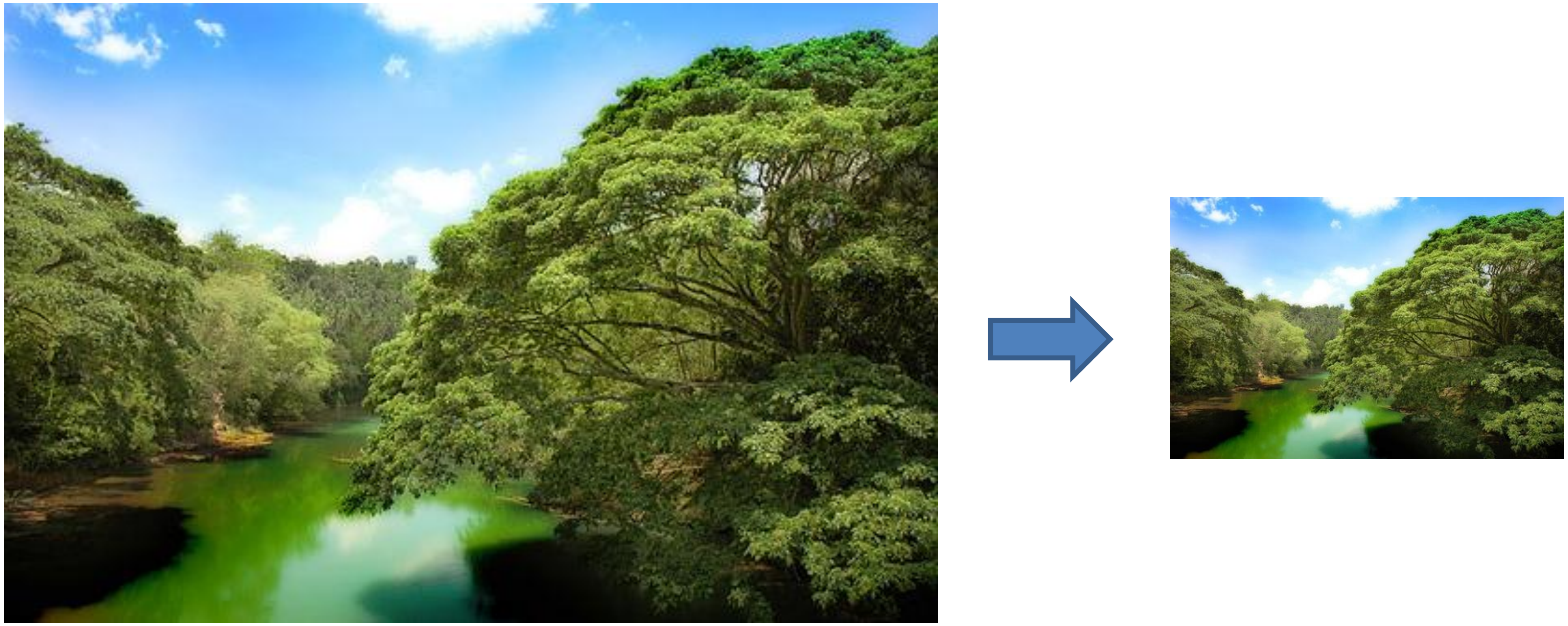


Image: http://www.flickr.com/photos/igorms/136916757/

Thinking in terms of frequency

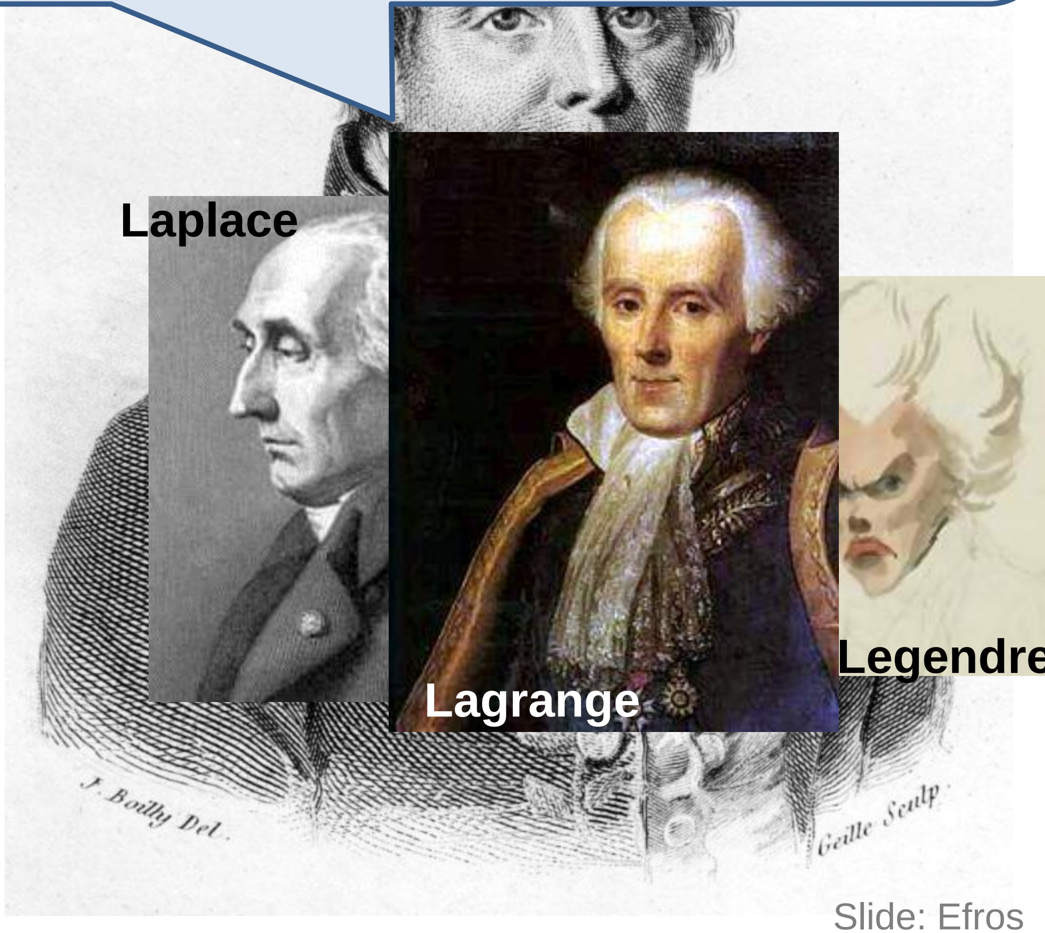
Jean Baptiste Joseph Fourier (1768-1830)

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



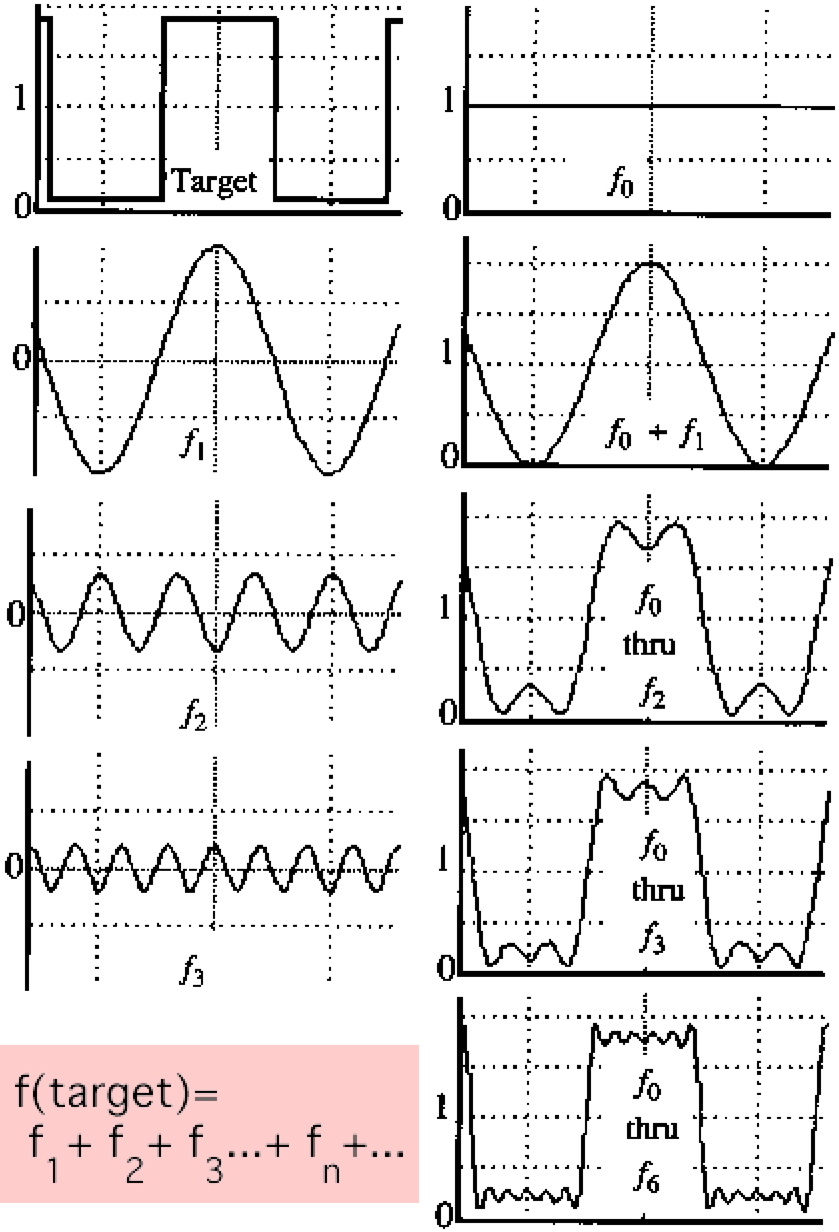
Slide: Efros

A sum of sines

Our building block:

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

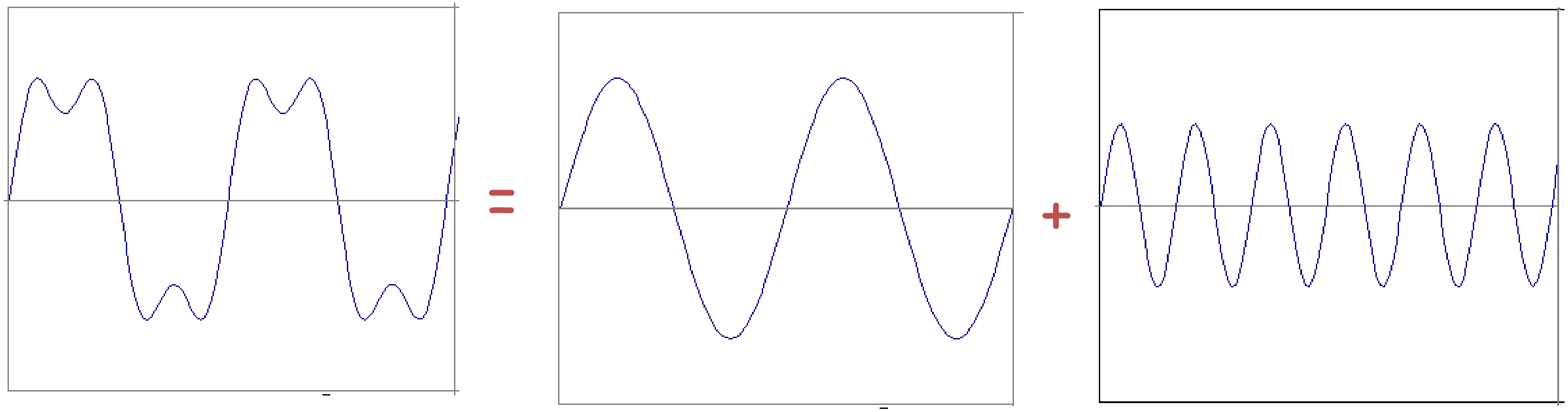
Add enough of them to get any signal *f(x)* you want!

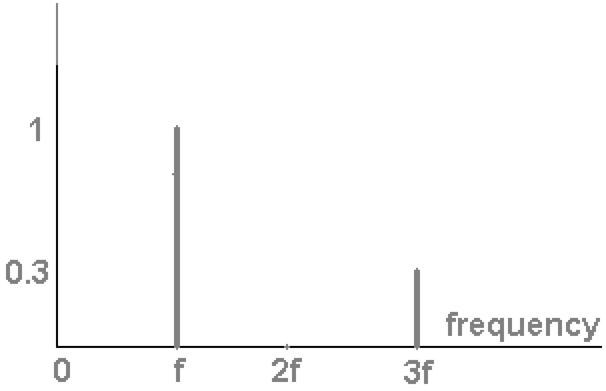


f(target)=
$f_1 + f_2 + f_3 ... + f_n + ...$

Frequency Spectra

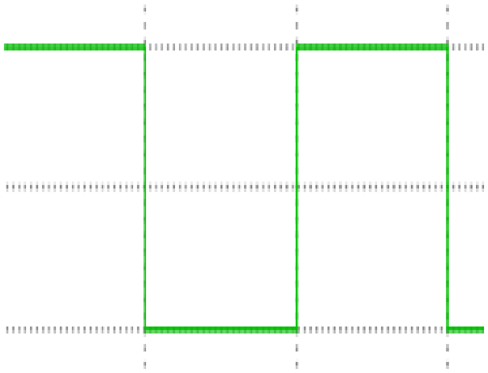
- example : $g(t) = \sin(2\pi f\ t) + (1/3)\sin(2\pi(3f)\ t)$



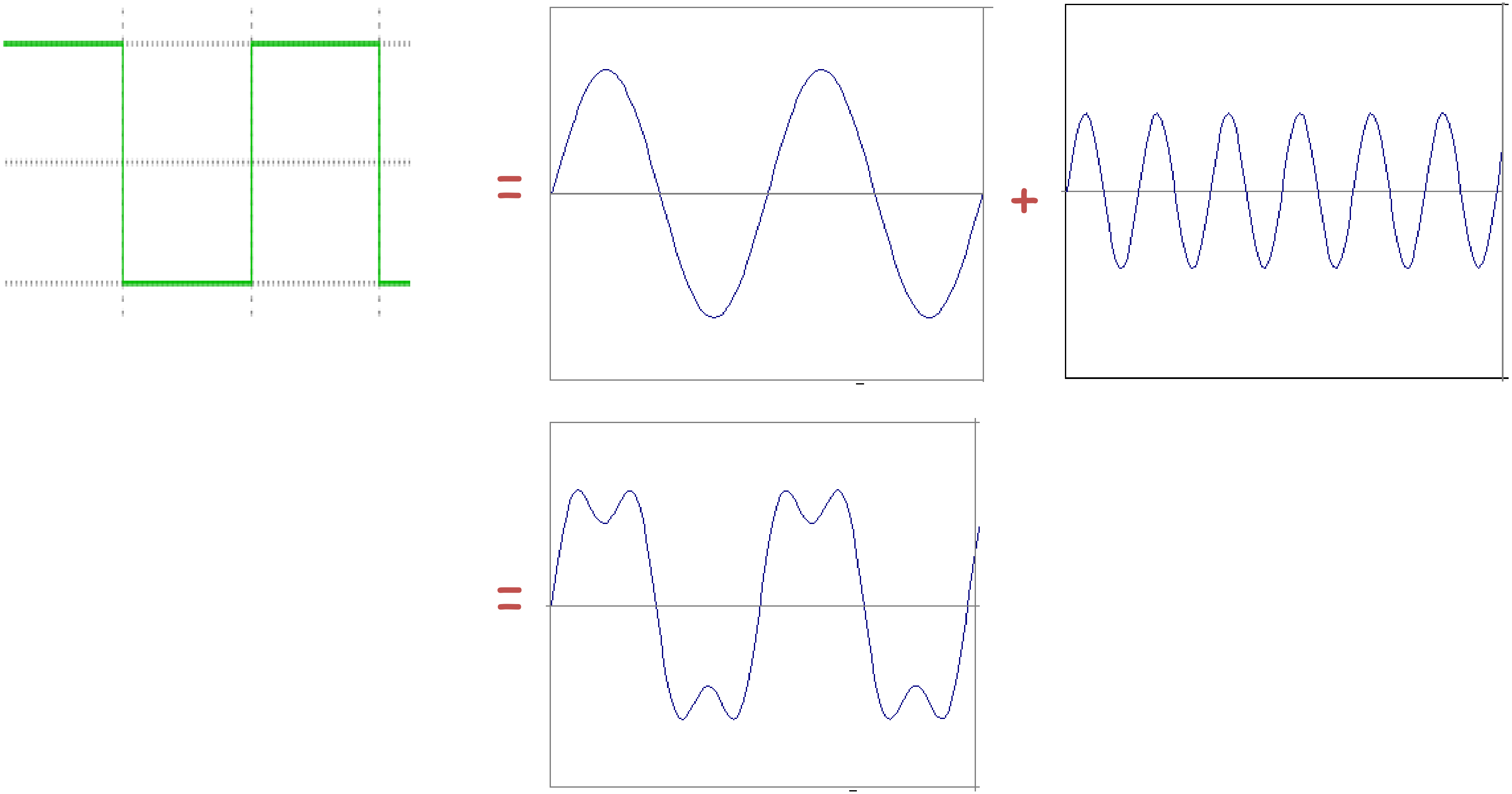


Slides: Efros

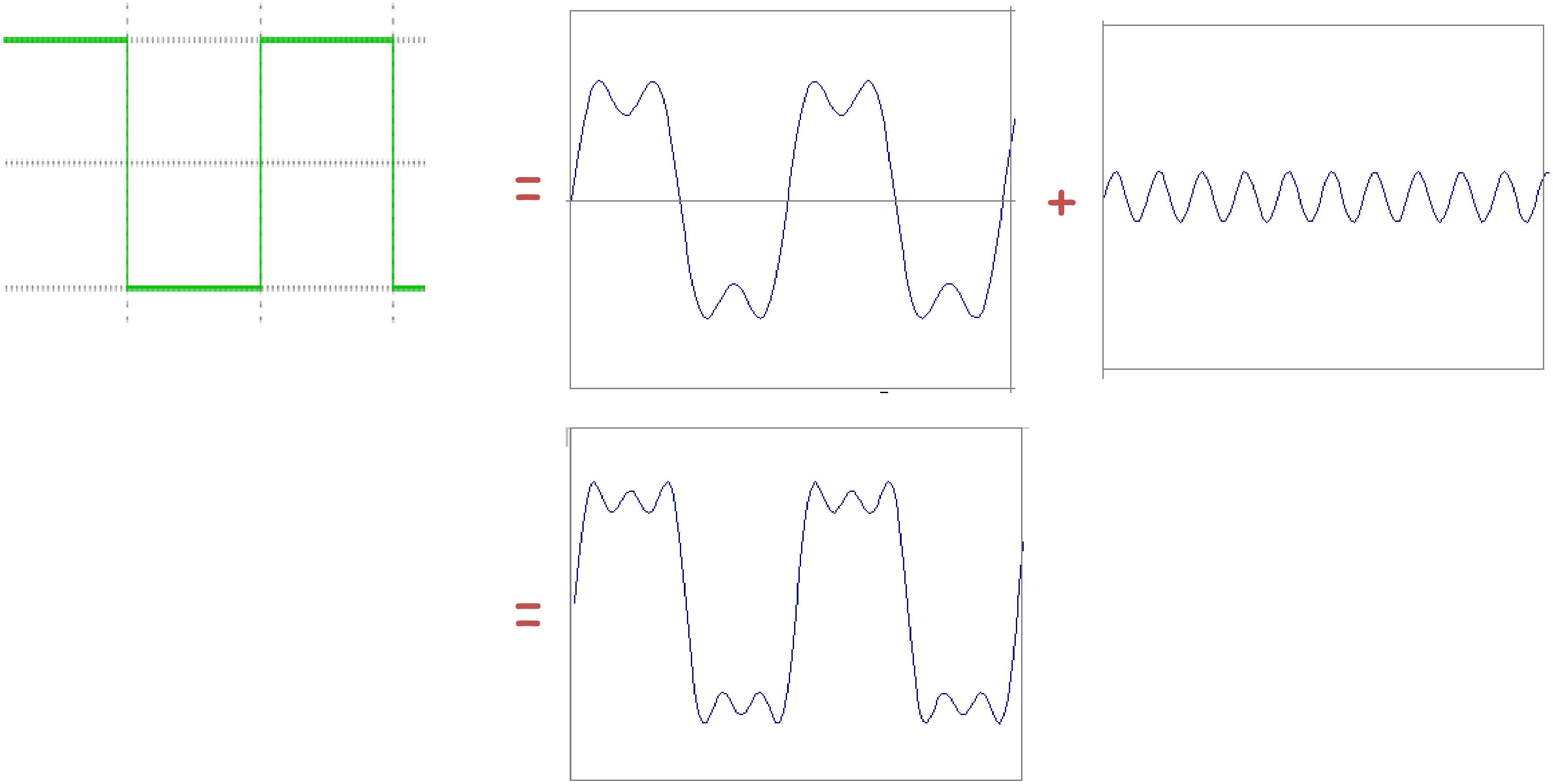
Frequency Spectra



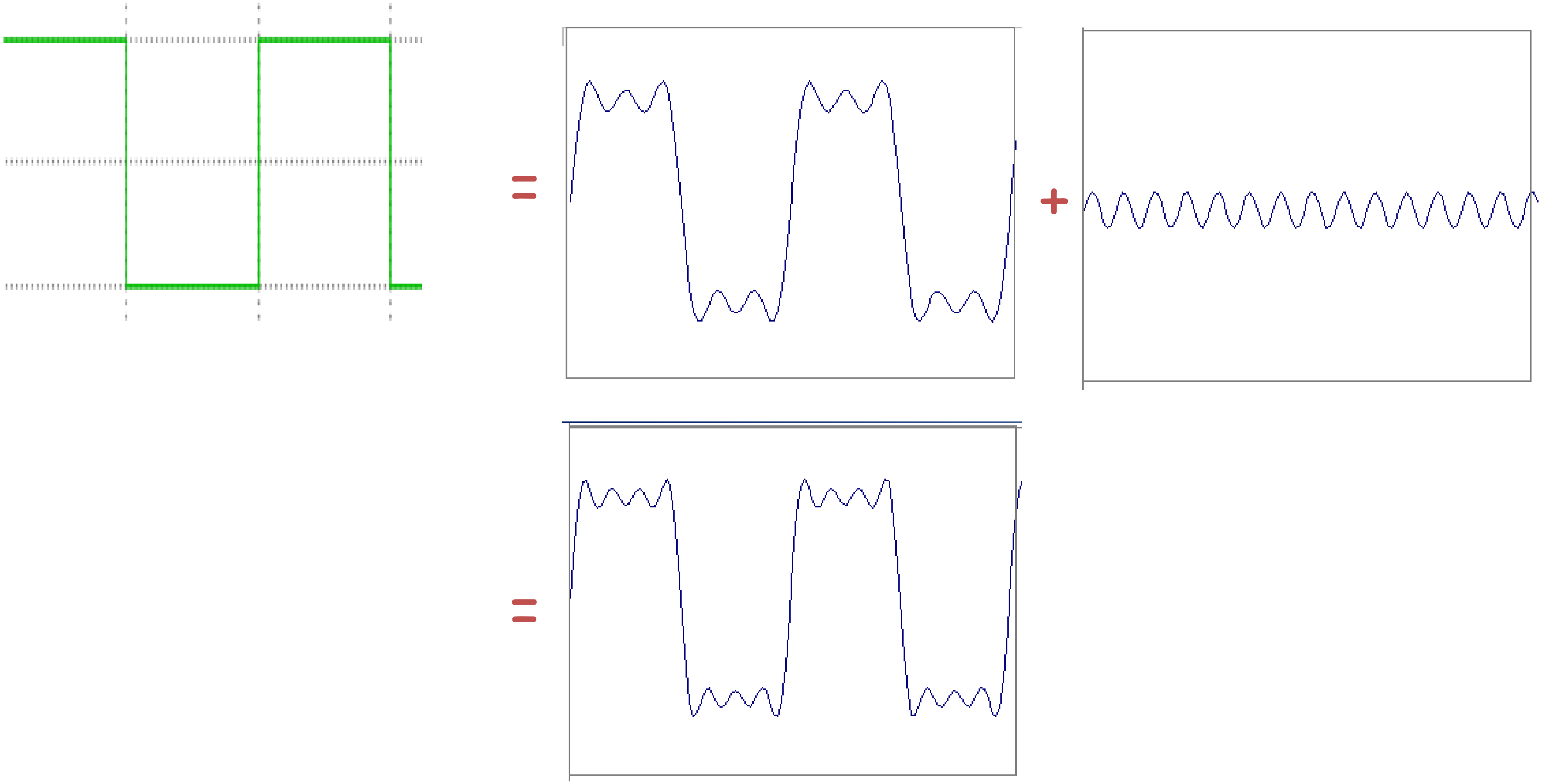
Frequency Spectra



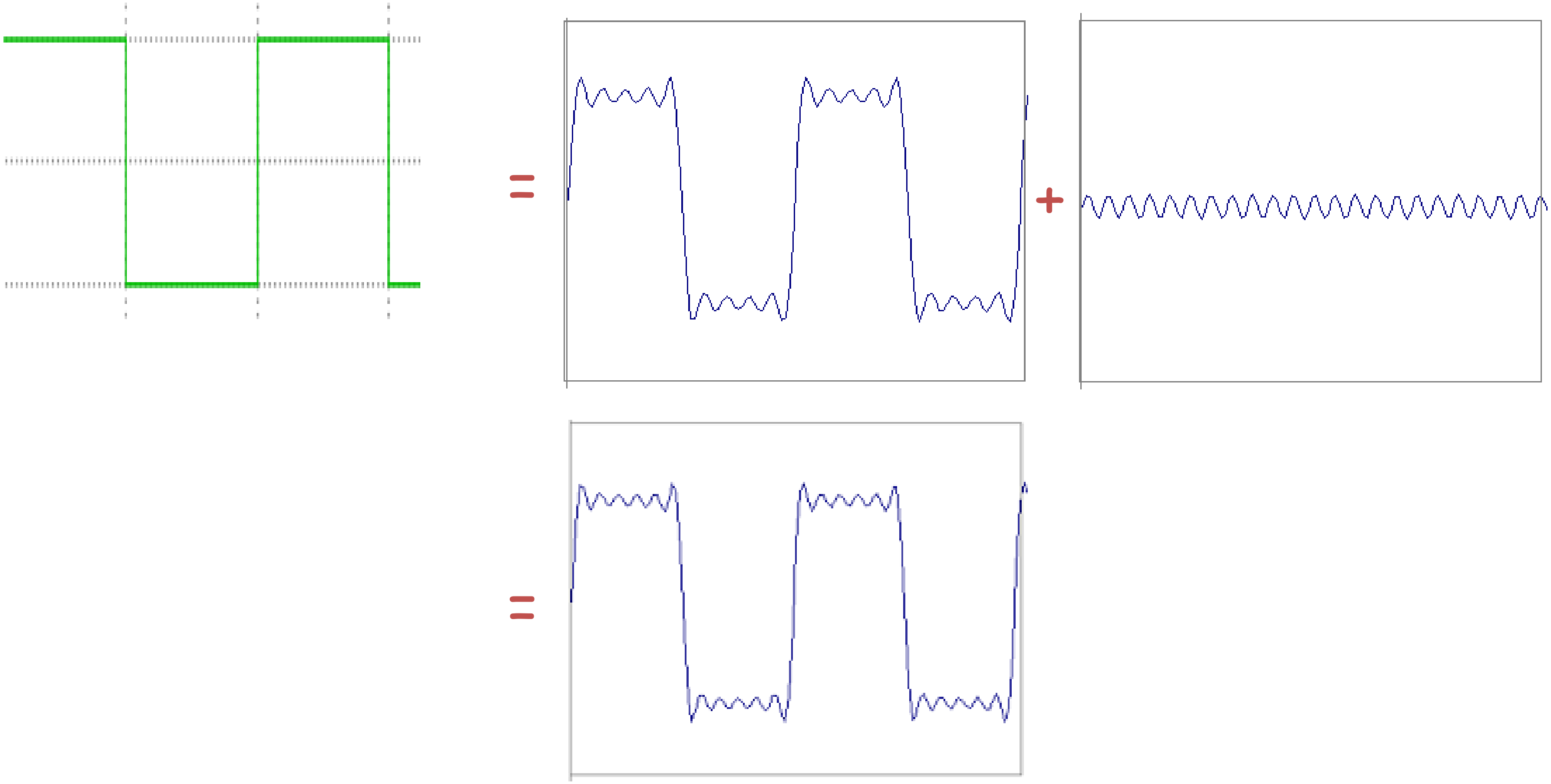
Frequency Spectra



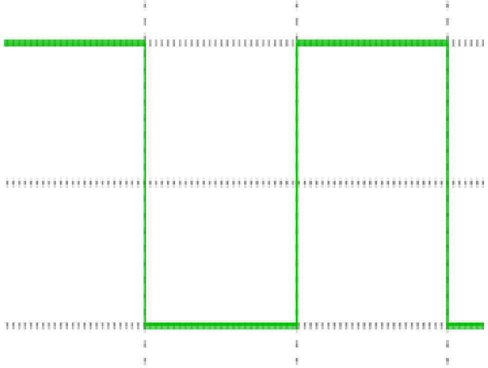
Frequency Spectra



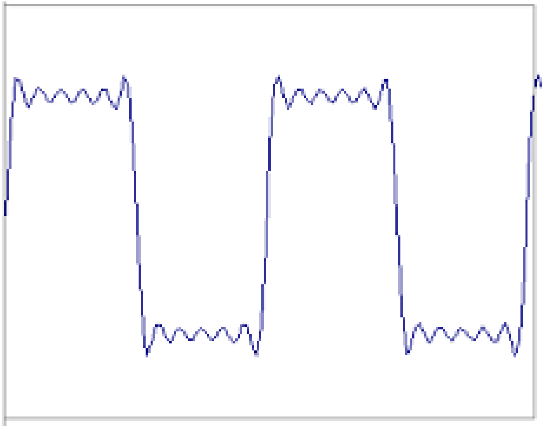
Frequency Spectra



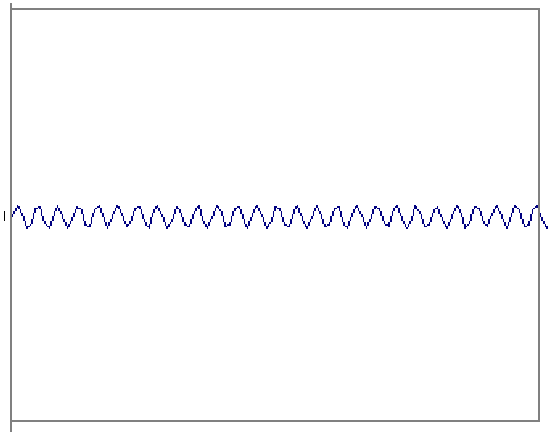
Frequency Spectra



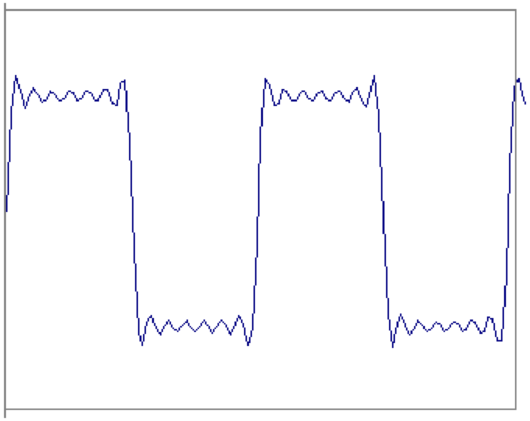
=



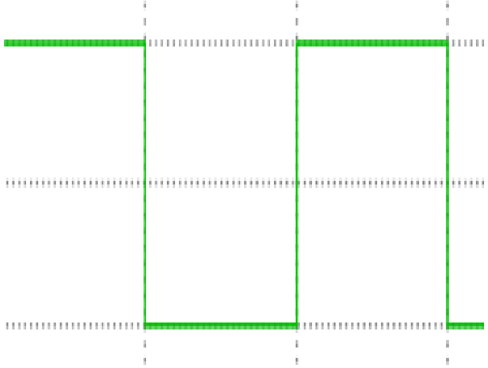
+



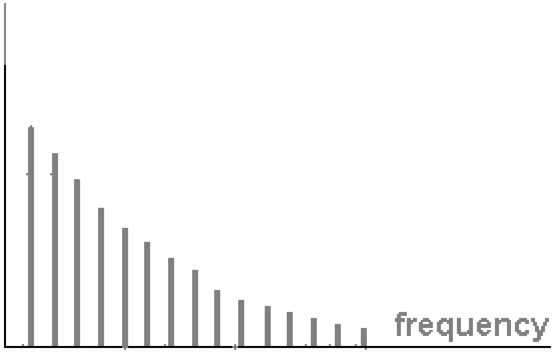
=



Frequency Spectra

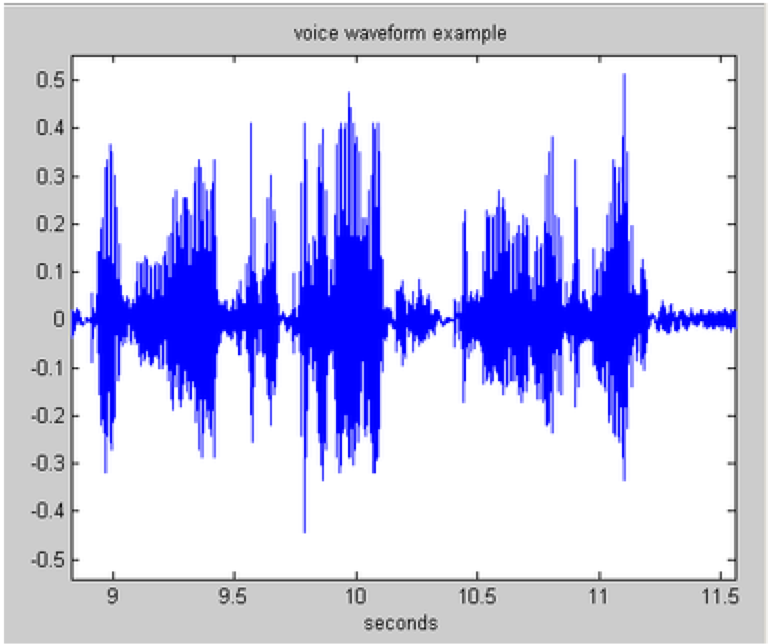


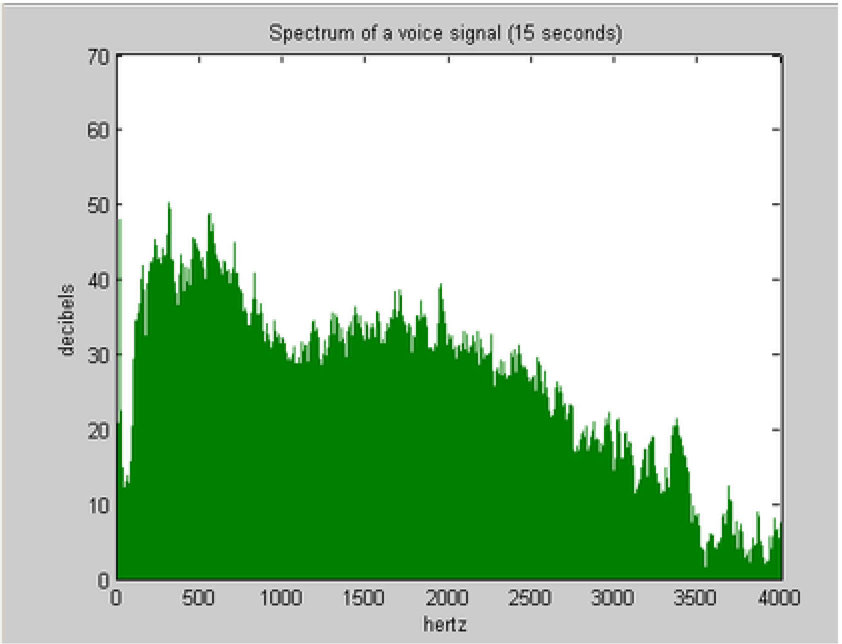
$$= \boxed{A\sum_{k=1}^{\infty}\frac{1}{k}\sin(2\pi kt)}$$



Example: Music

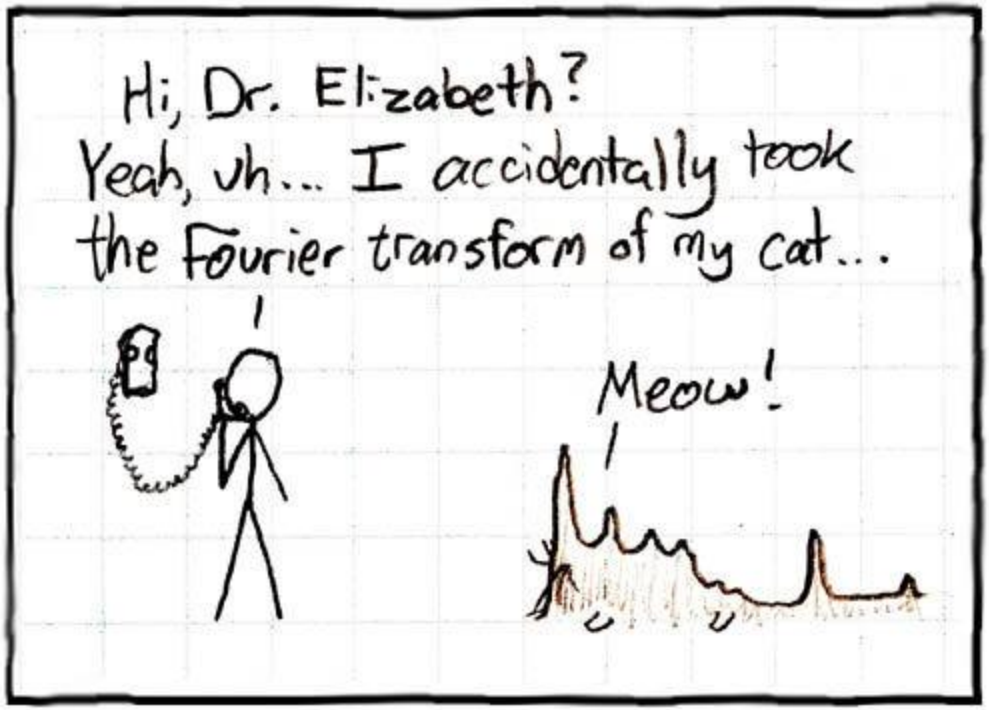
- We think of music in terms of frequencies at different magnitudes





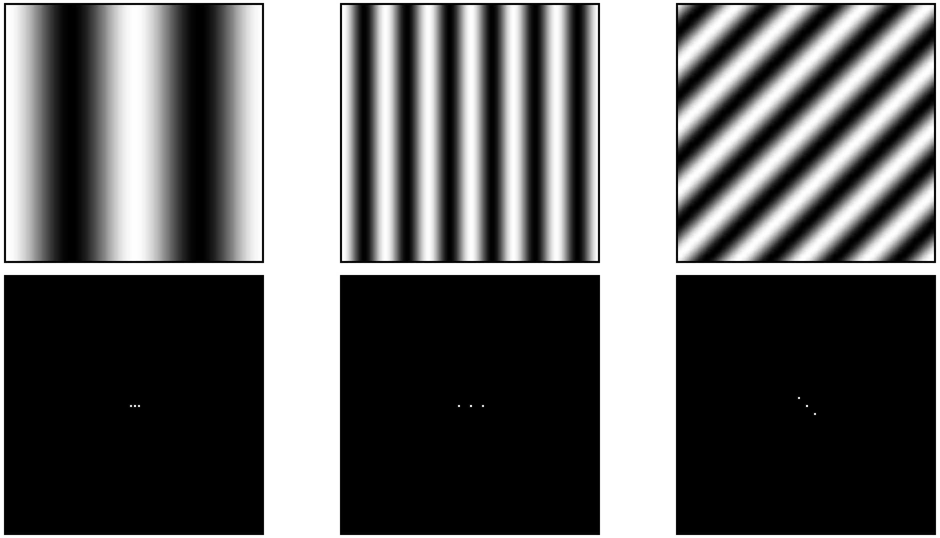
Other signals

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


Hi, Dr. Elizabeth?
Yeah, uh... I accidentally took the Fourier transform of my cat...
Meow!

xkcd.com

Fourier analysis in images

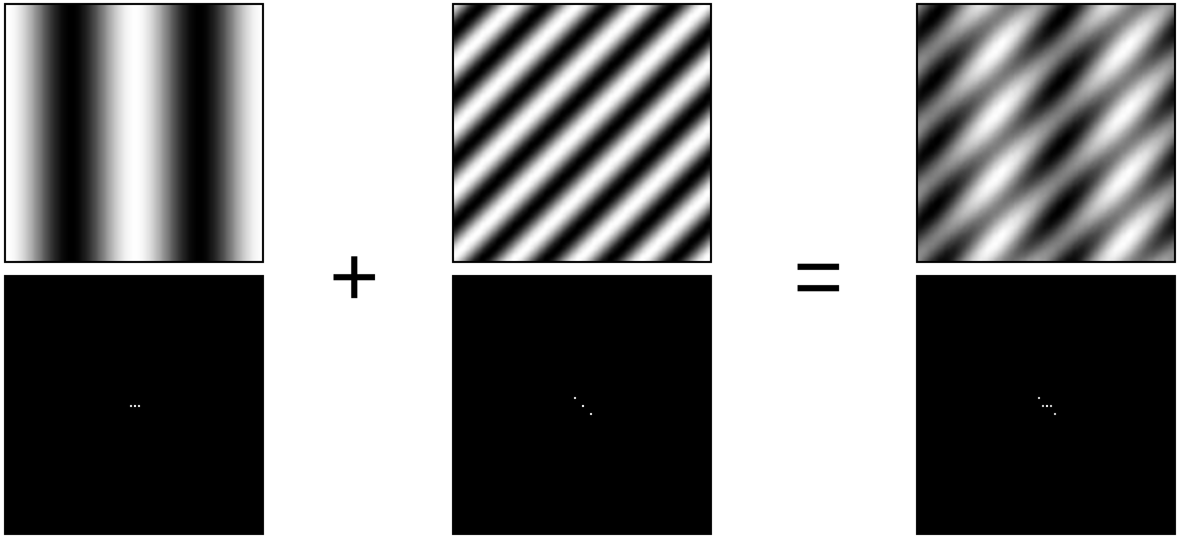


Intensity Image

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

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

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

Computing the Fourier Transform

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

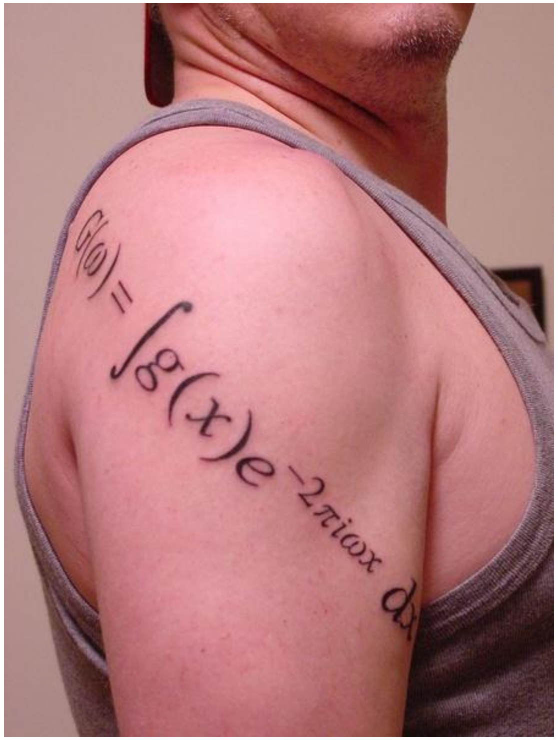
Continuous

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

Discrete

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

k=-N/2..N/2



(options for if you can't remember this)

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

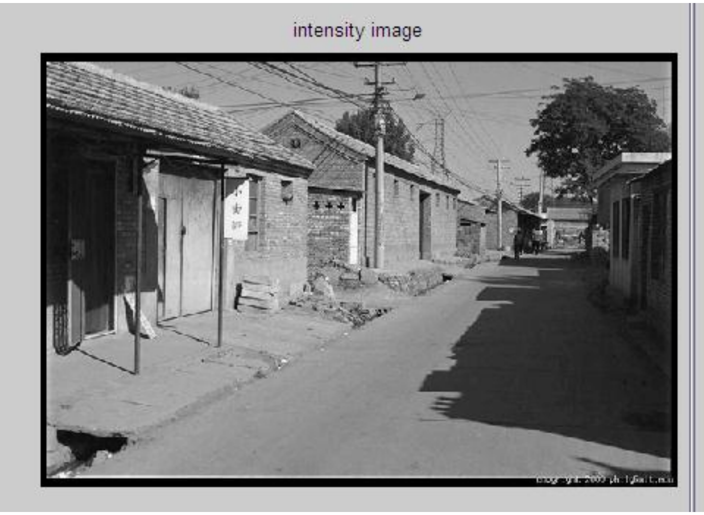
- Fourier transform of a real signal is symmetric about the origin

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

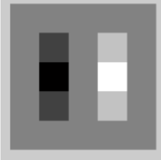
See Szeliski Book (3.4)

Filtering in spatial domain

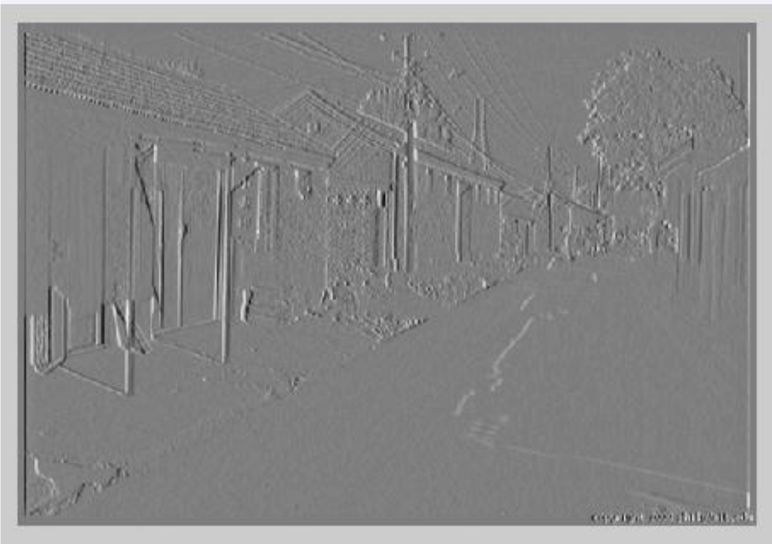
1	0	-1
2	0	-2
1	0	-1



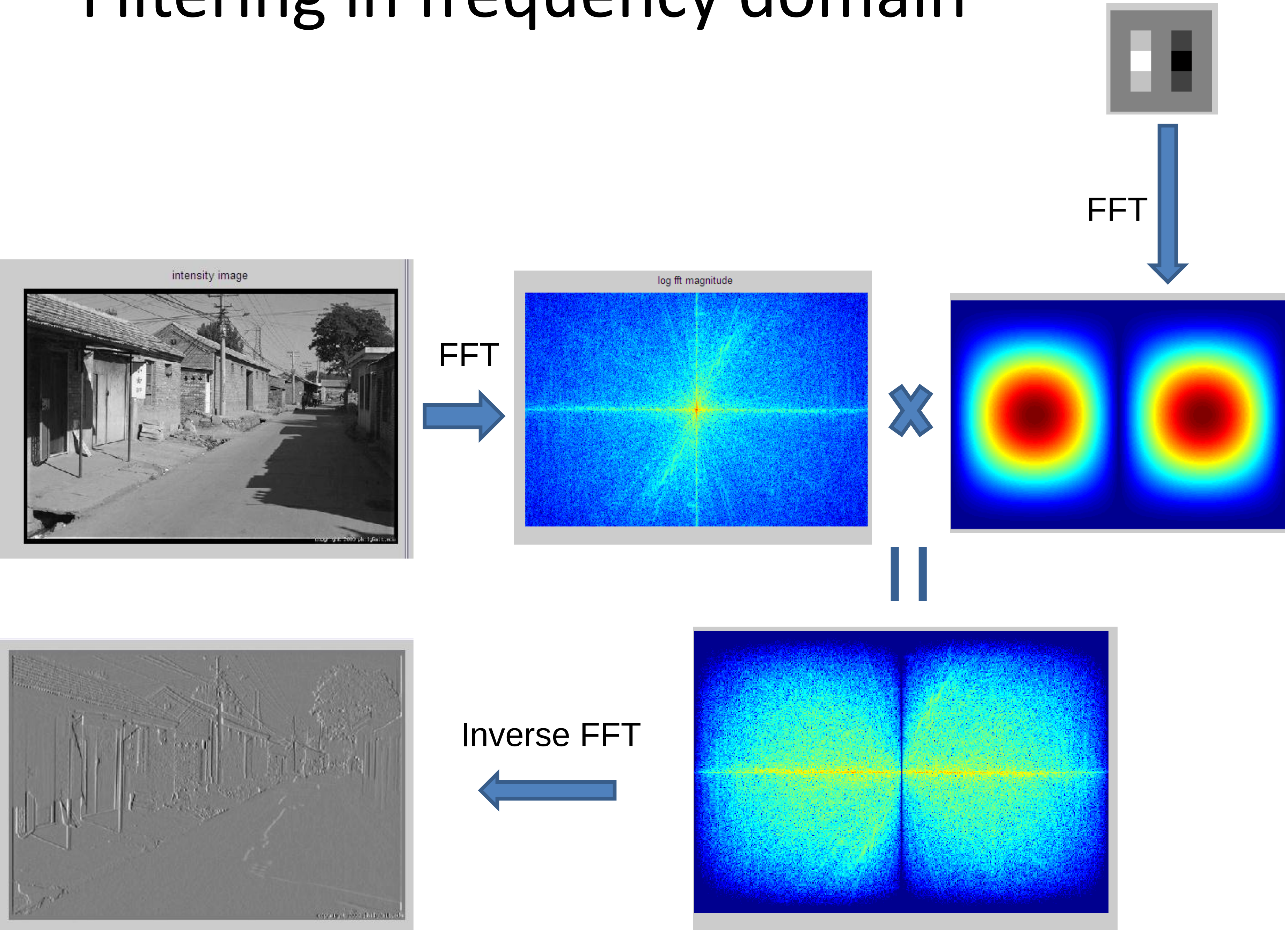
*



=



Filtering in frequency domain



FFT

FFT

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 = 30; % 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
```

Introduce yourselves

Questions

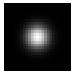
Which has more information, the phase or the magnitude?

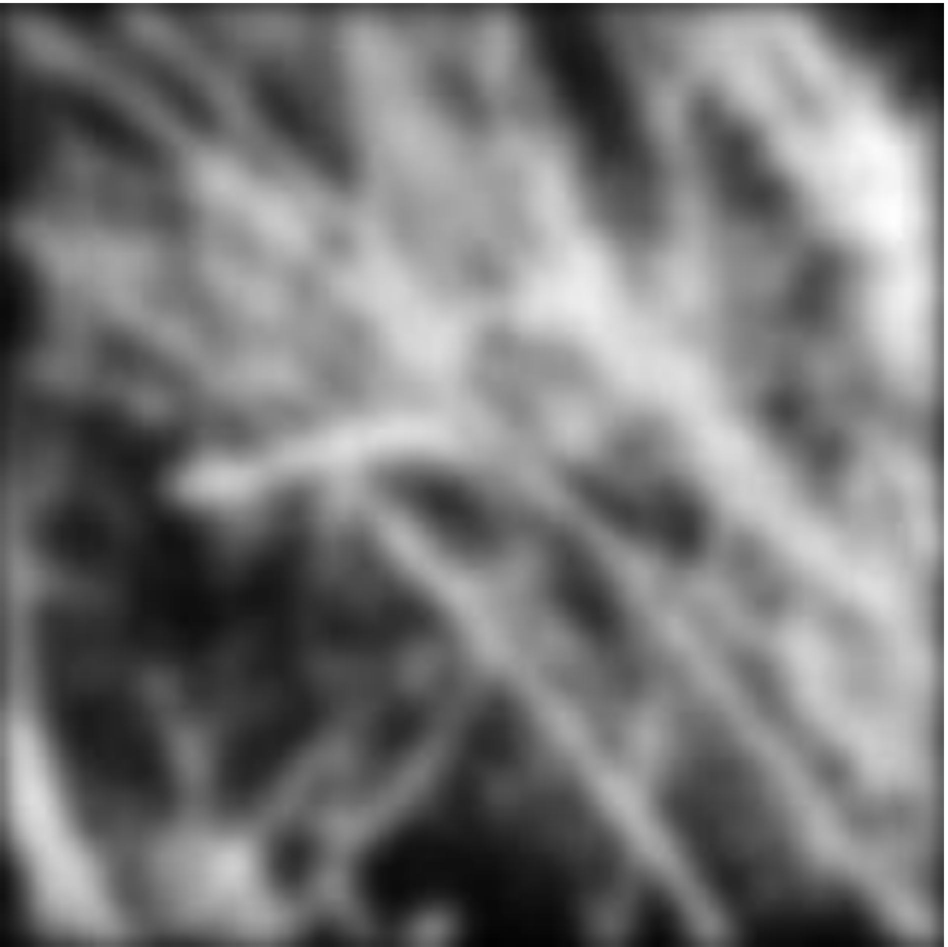
What happens if you take the phase from one image and combine it with the magnitude from another image?

Filtering

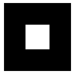
Why does the Gaussian give a nice smooth image, but the square filter give edgy artifacts?

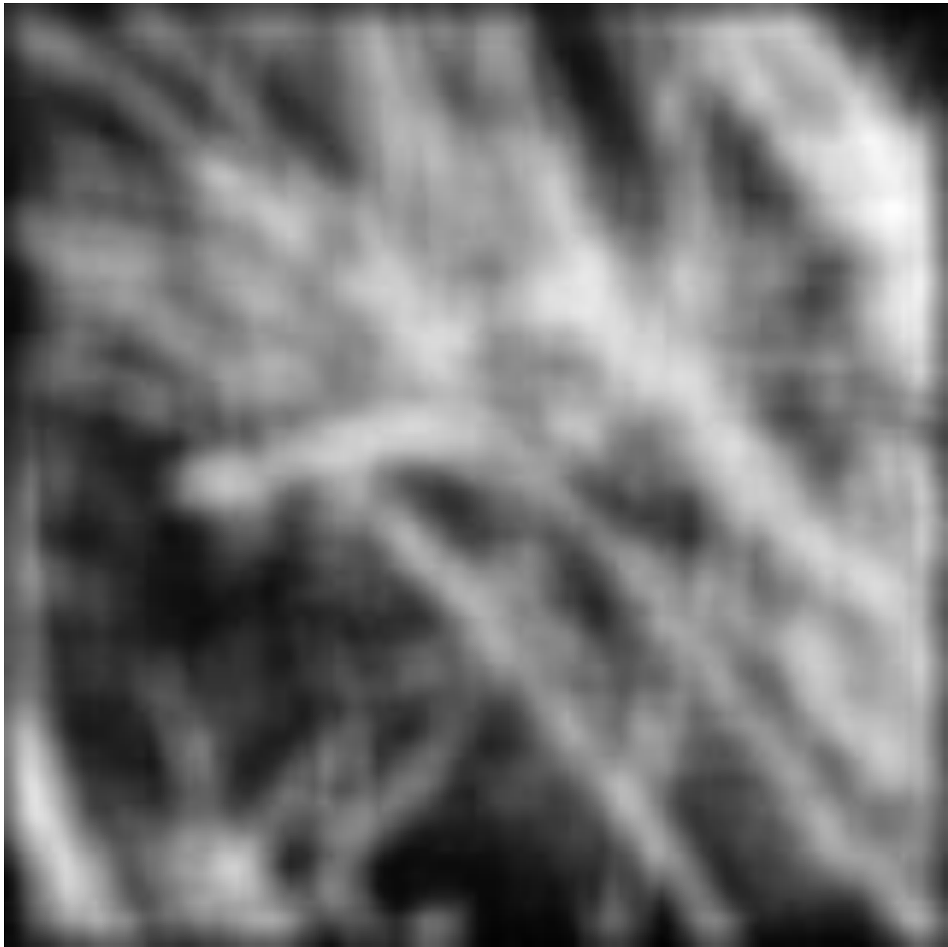
Gaussian



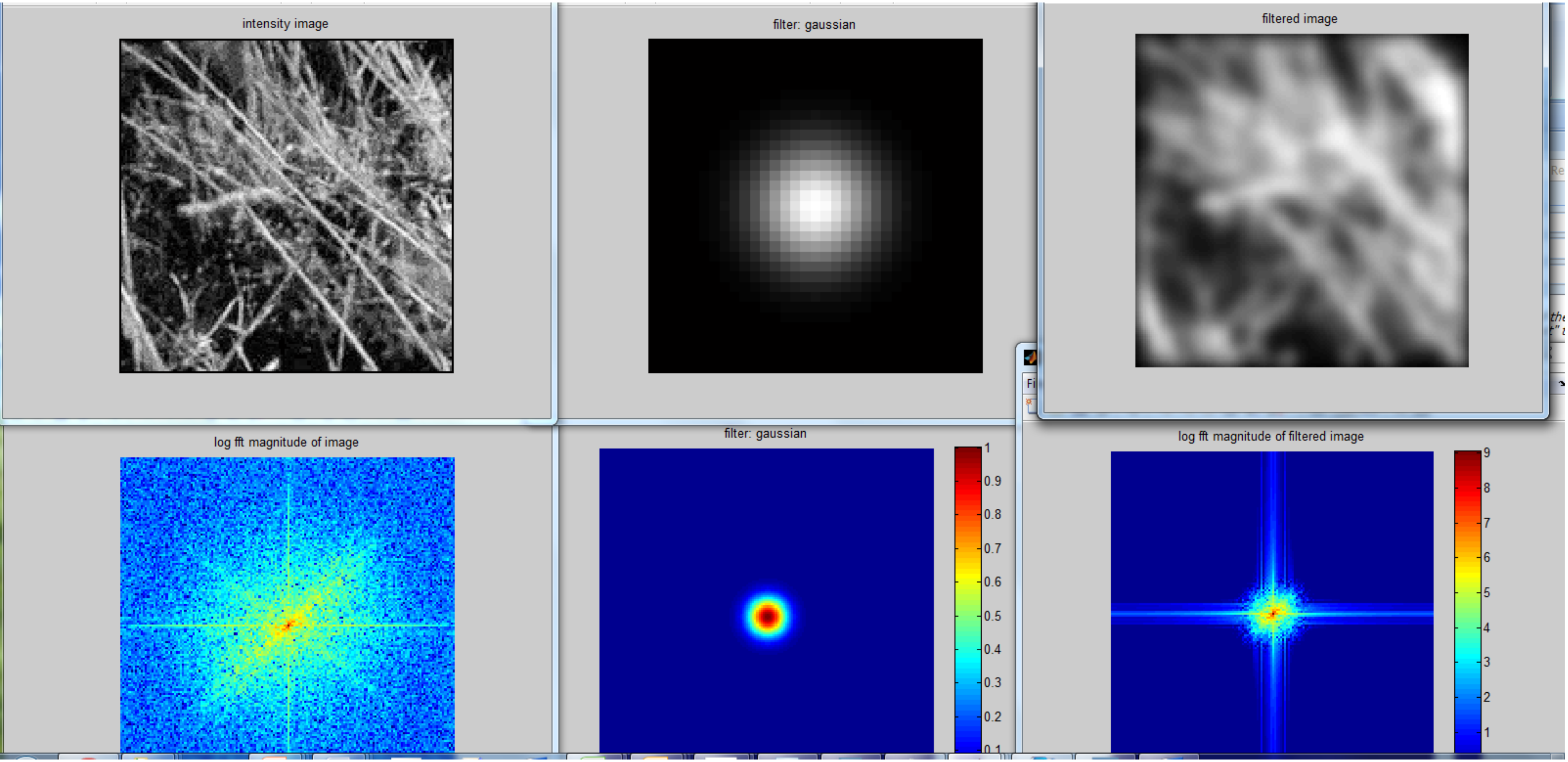


Box filter

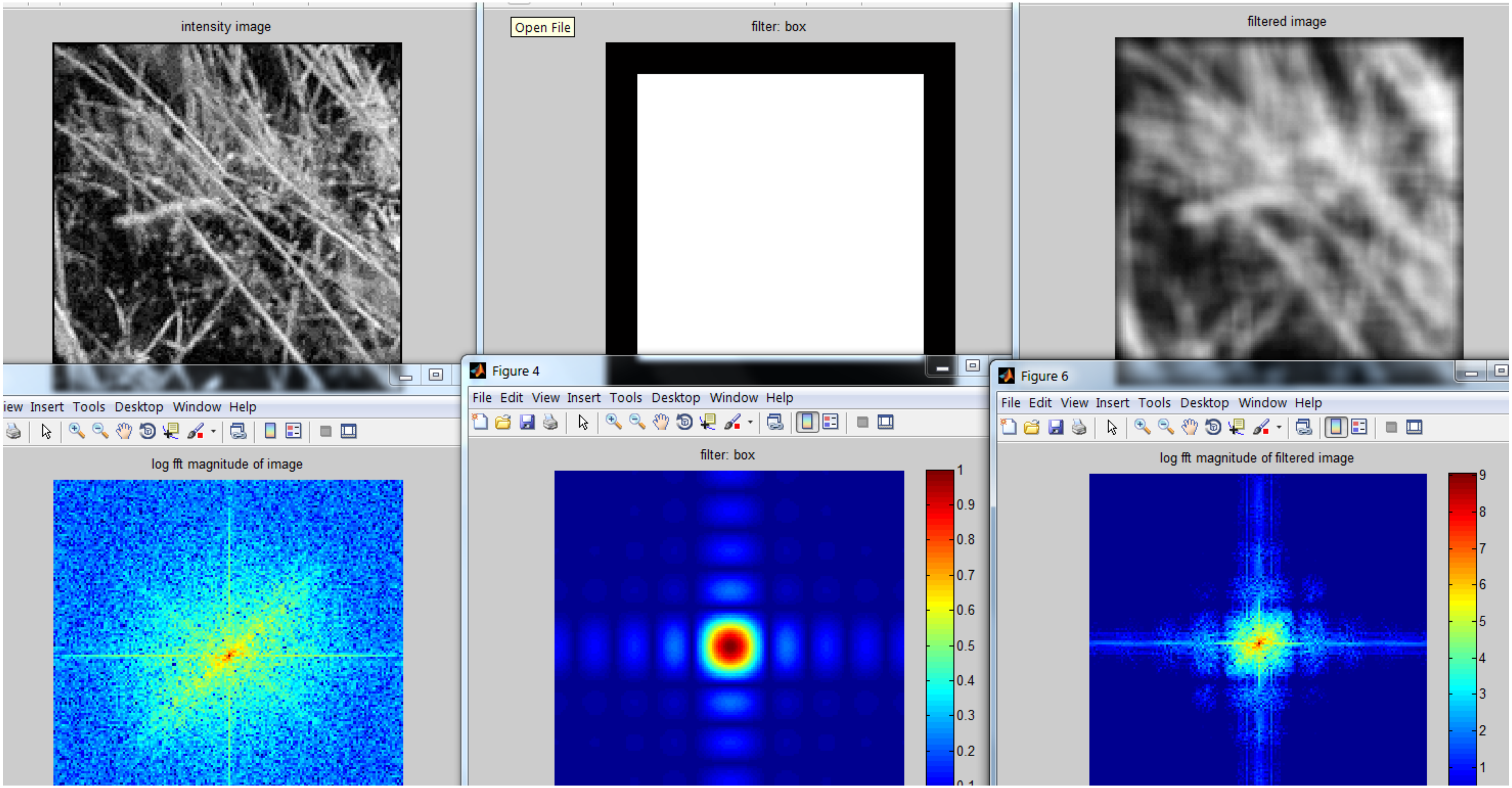




Gaussian


intensity image
filter: gaussian
filtered image
log fft magnitude of image
filter: gaussian
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
log fft magnitude of filtered image
9
8
7
6
5
4
3
2
1

Box Filter


intensity image
Open File
filter: box
filtered image
Figure 4
Figure 6
iew Insert Tools Desktop Window Help
File Edit View Insert Tools Desktop Window Help
File Edit View Insert Tools Desktop Window Help
log fft magnitude of image
filter: box
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
log fft magnitude of filtered image
9
8
7
6
5
4
3
2
1

Sampling

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

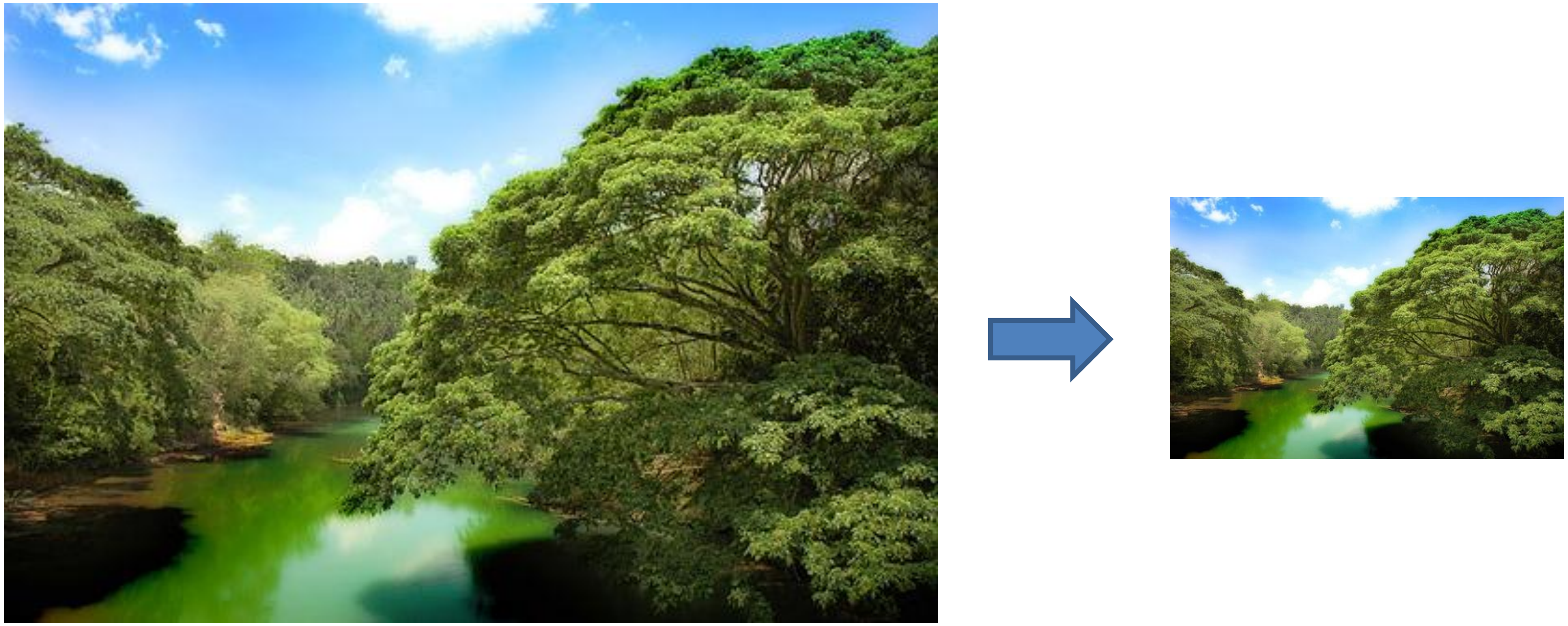
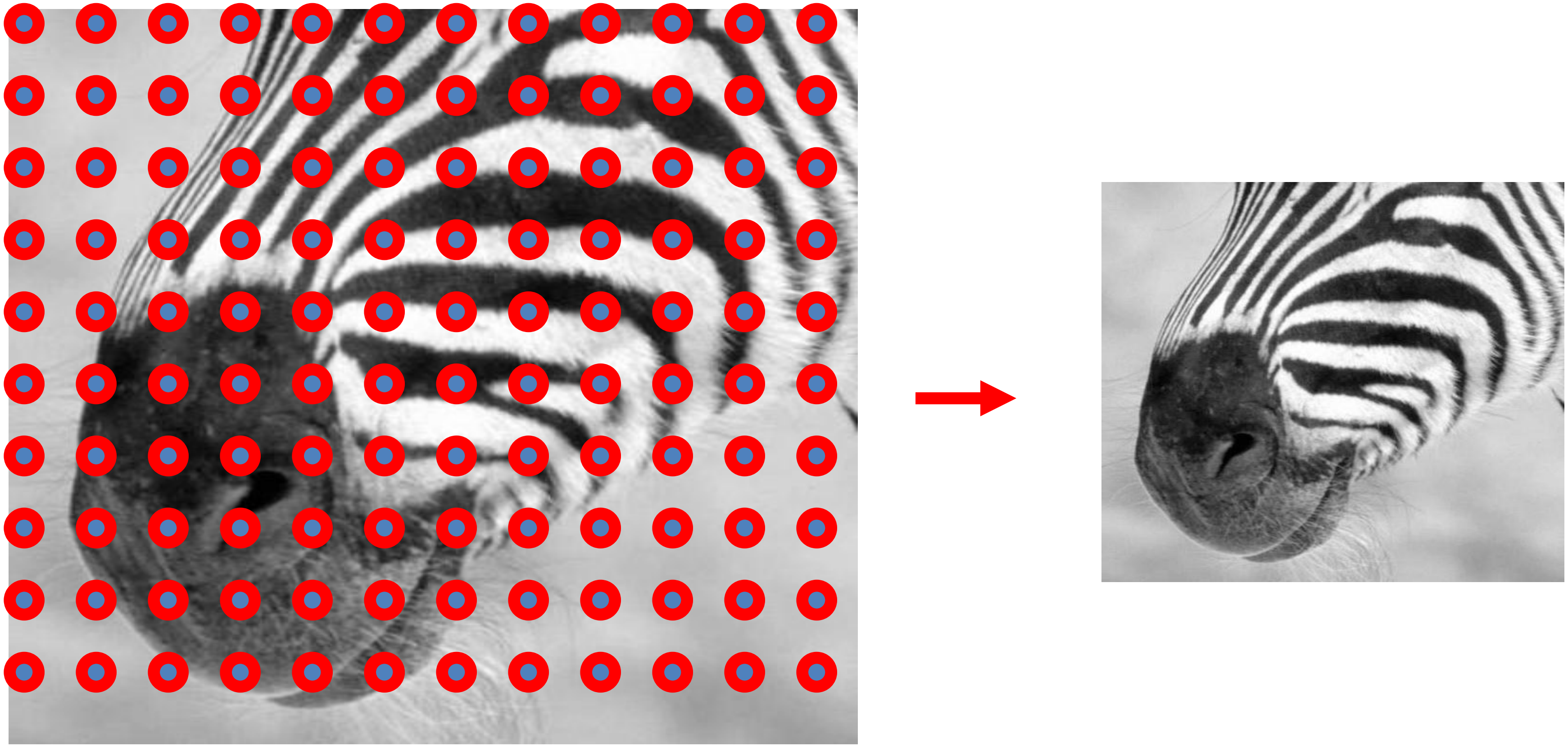


Image: http://www.flickr.com/photos/igorms/136916757/

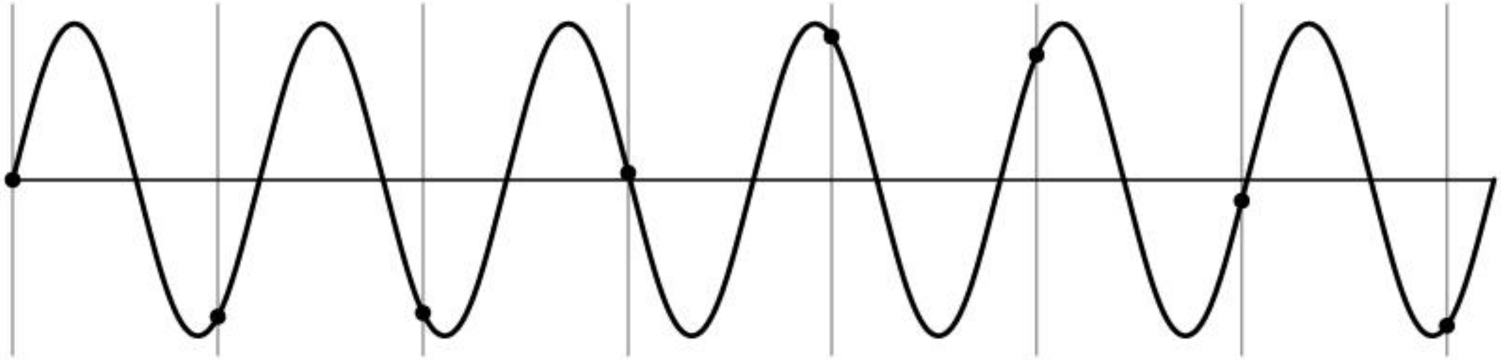
Subsampling by a factor of 2



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

Aliasing problem

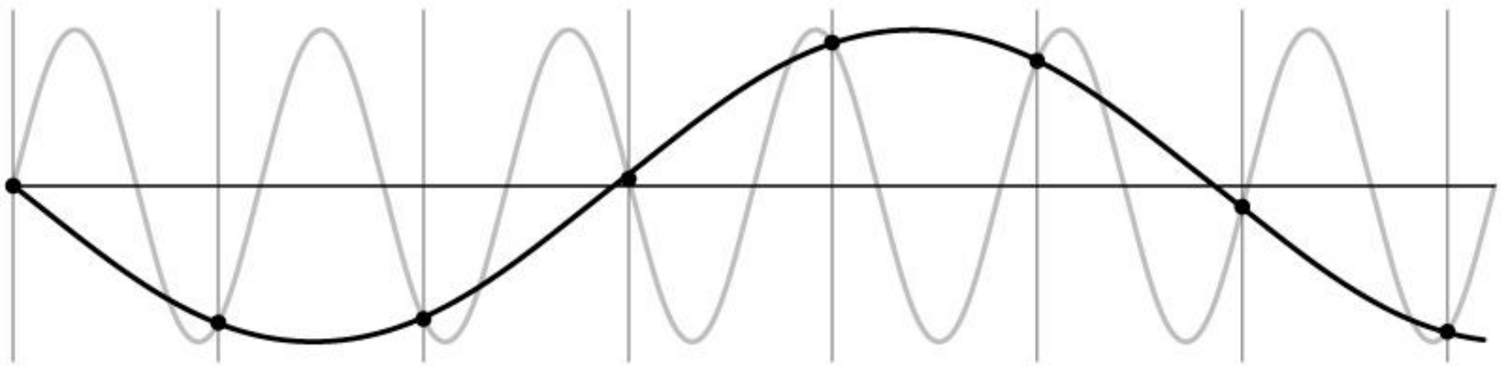
- 1D example (sinewave):



Source: S. Marschner

Aliasing problem

- 1D example (sinewave):



Source: S. Marschner

Aliasing problem

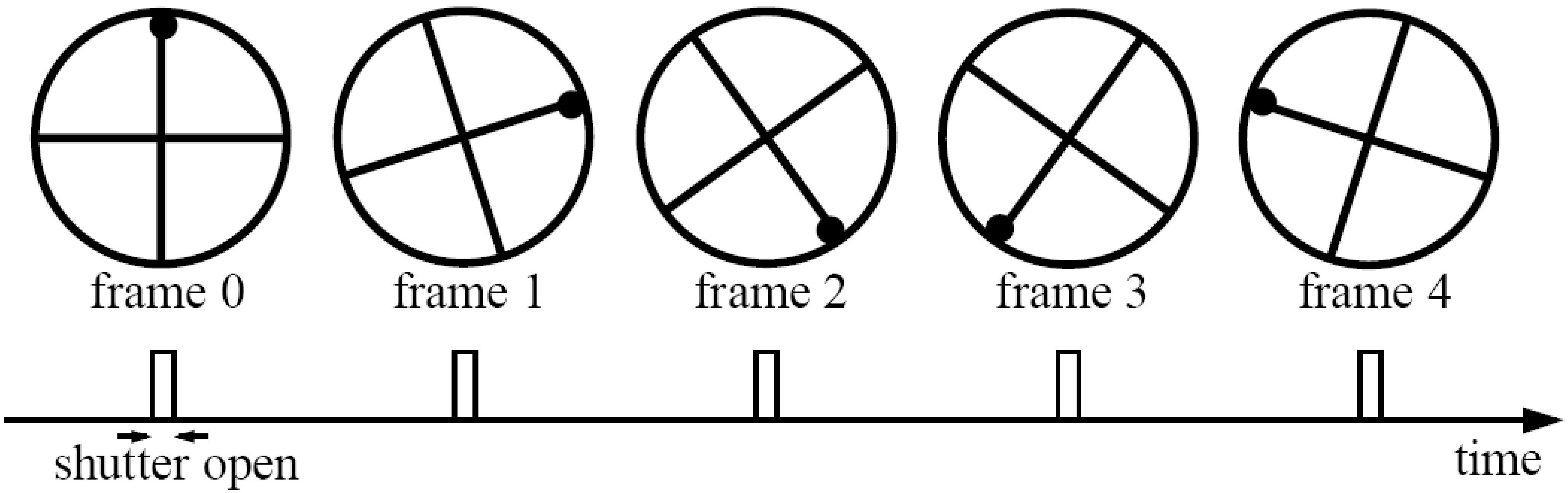
- Sub-sampling may be dangerous….
- Characteristic errors may appear:
  - “Wagon wheels rolling the wrong way in movies”
  - “Checkerboards disintegrate in ray tracing”
  - “Striped shirts look funny on color television”

Source: D. Forsyth

Aliasing in video

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

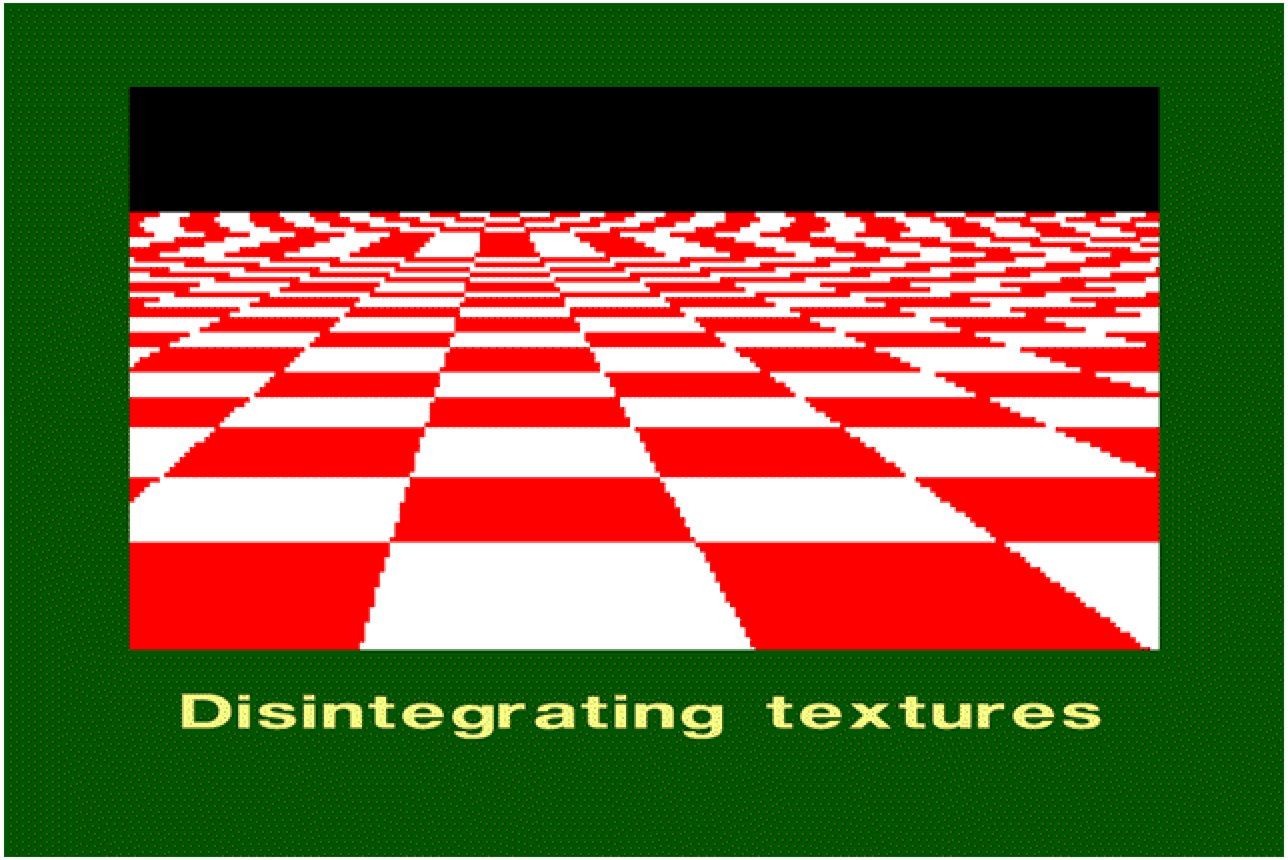
If camera shutter is only open for a fraction of a frame time (frame time = 1/30 sec. for video, 1/24 sec. for film):



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

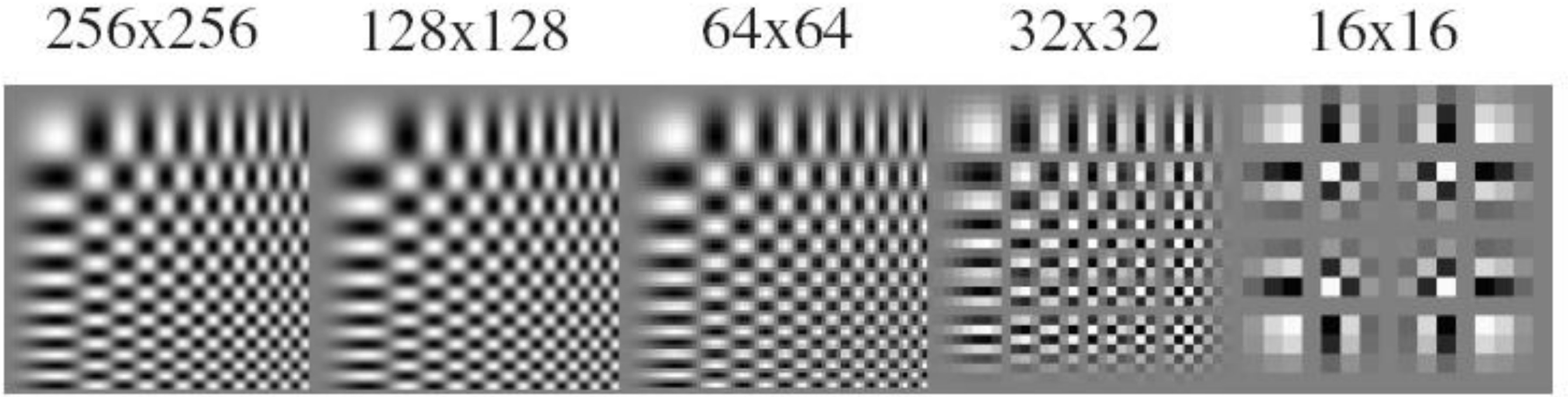
Slide by Steve Seitz

Aliasing in graphics



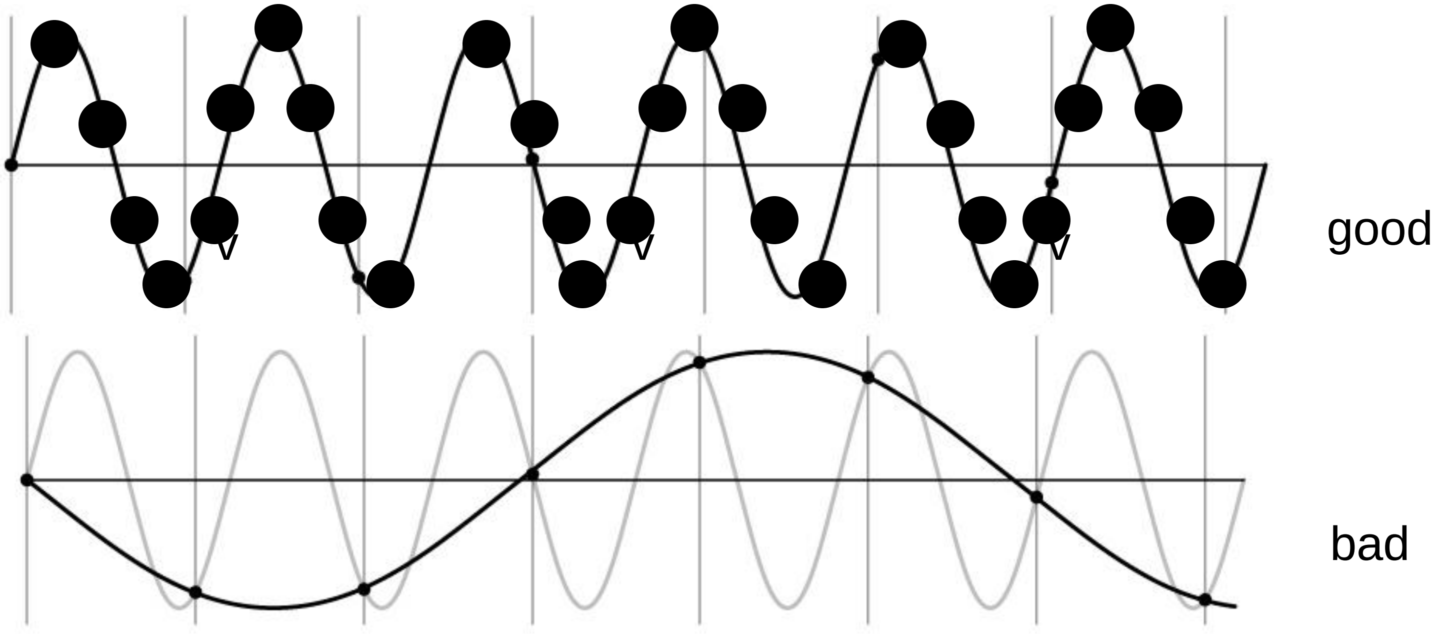
Source: A. Efros

Sampling and aliasing



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



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

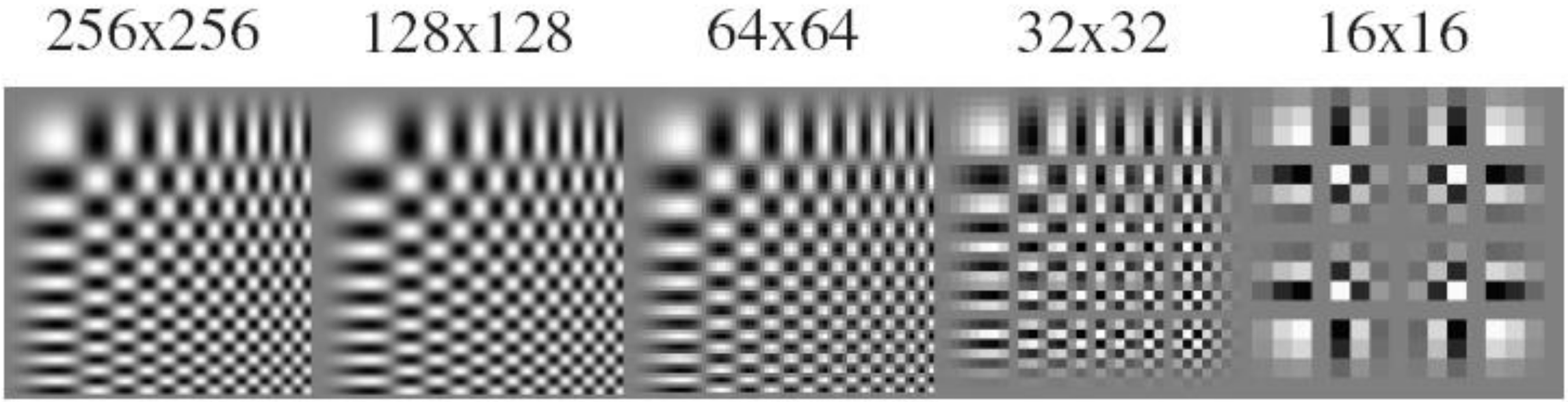
1. Start with image(h, w)
2. Apply low-pass filter

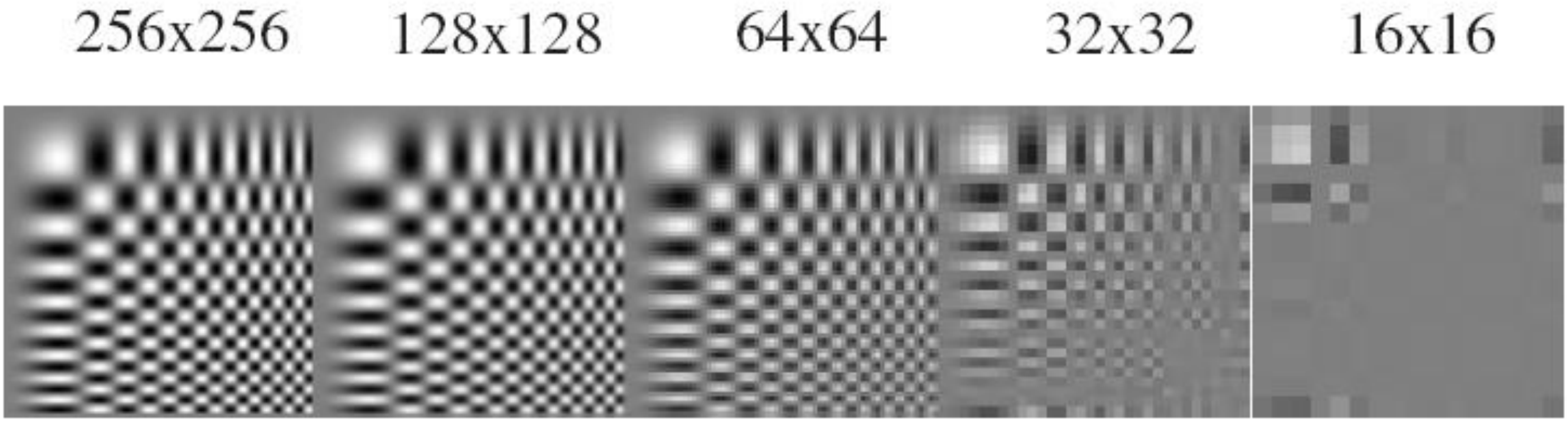
   ```
   im_blur = imfilter(image, fspecial(‘gaussian’, 7, 1))
   ```

3. Sample every other pixel

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

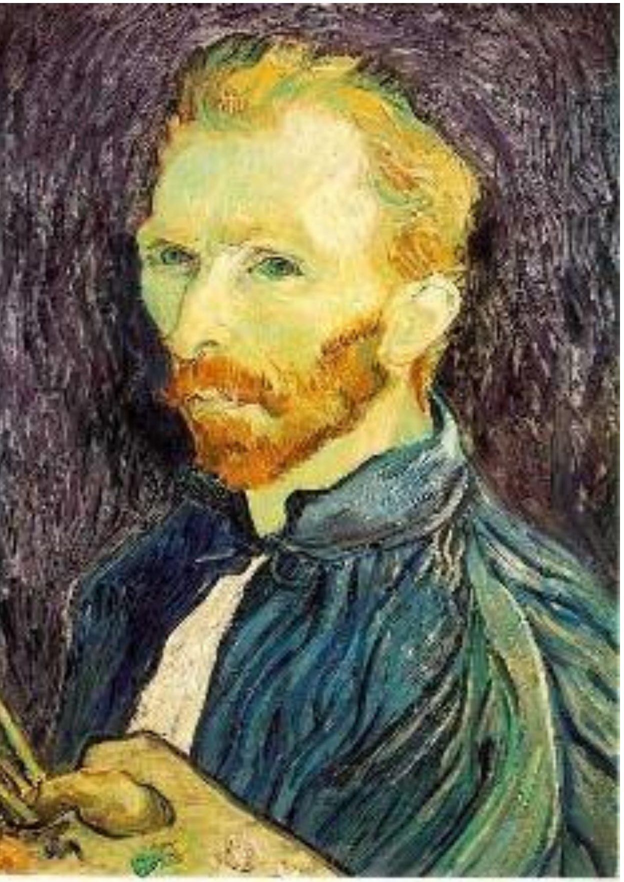
Anti-aliasing



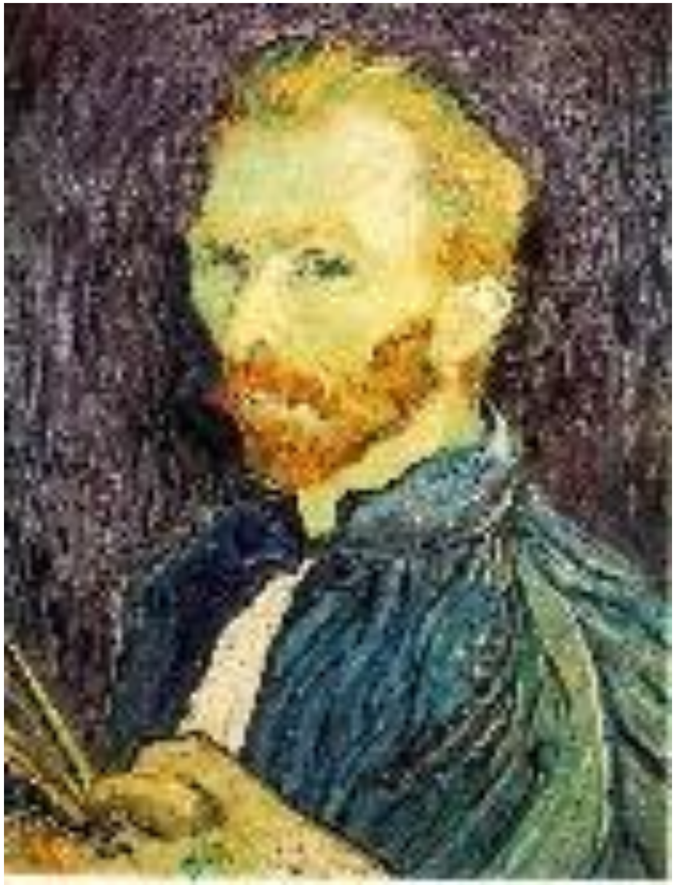


Forsyth and Ponce 2002

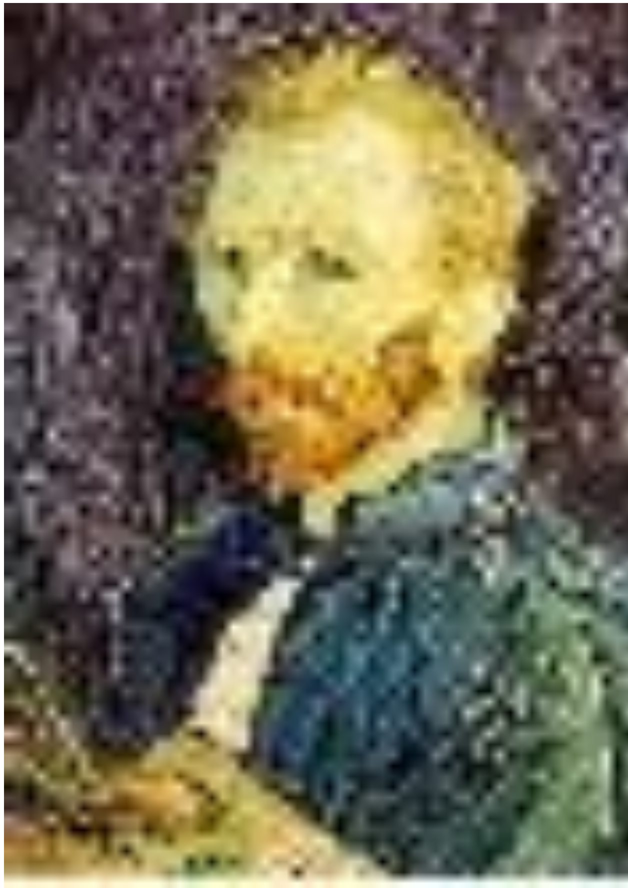
Subsampling without pre-filtering



1/2



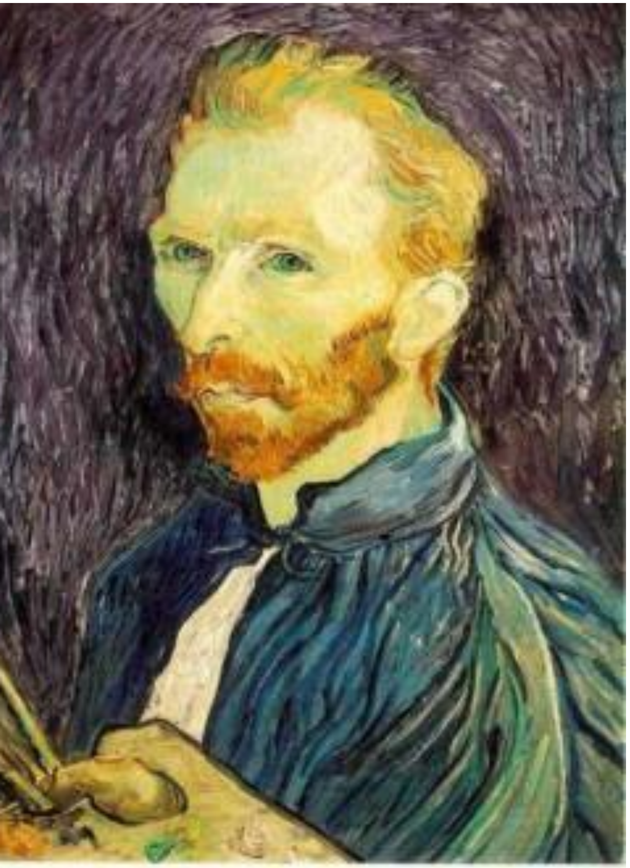
1/4 (2x zoom)



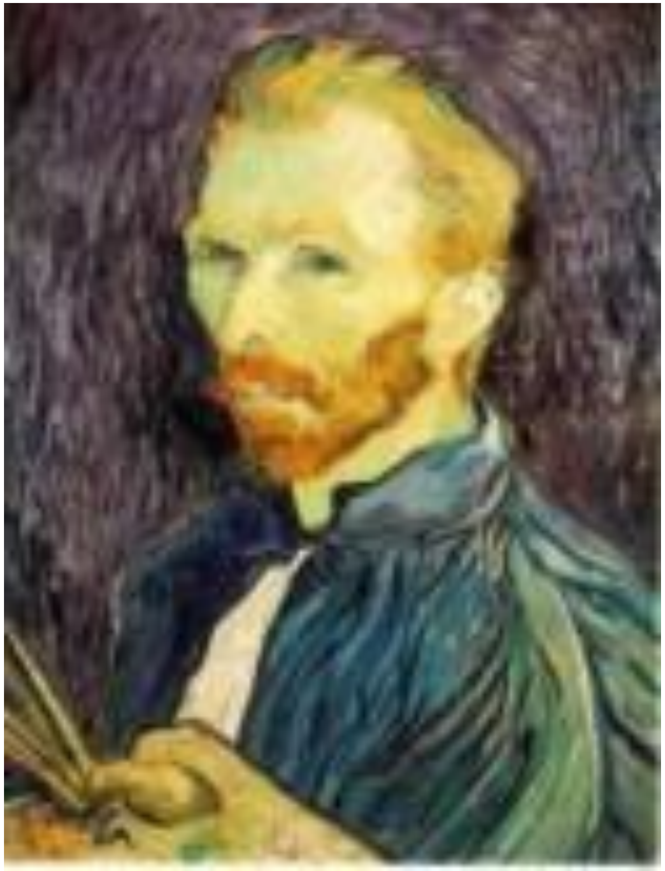
1/8 (4x zoom)

Slide by Steve Seitz

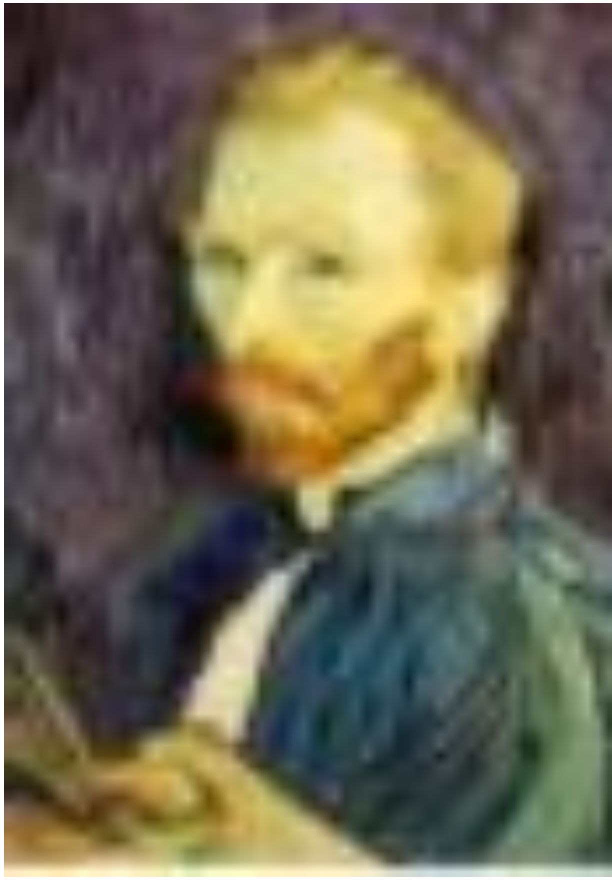
Subsampling with Gaussian pre-filtering



Gaussian 1/2



G 1/4



G 1/8

Slide by Steve Seitz

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

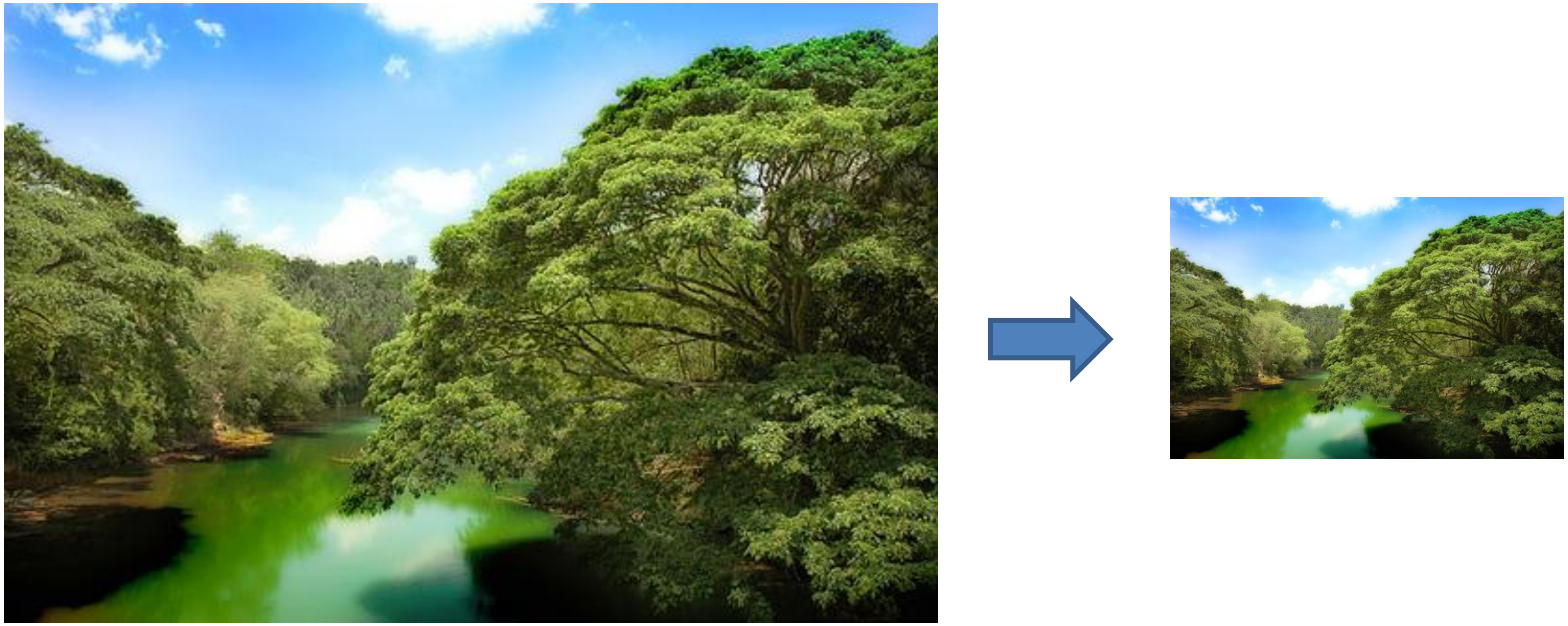
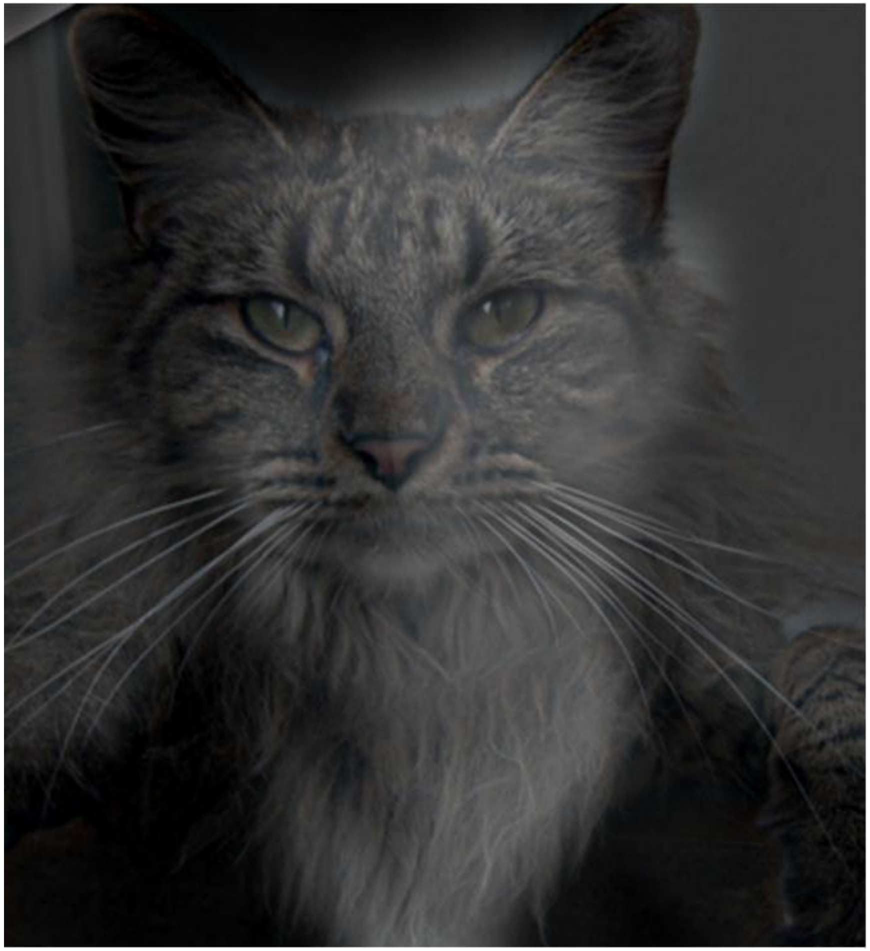


Image: http://www.flickr.com/photos/igorms/136916757/

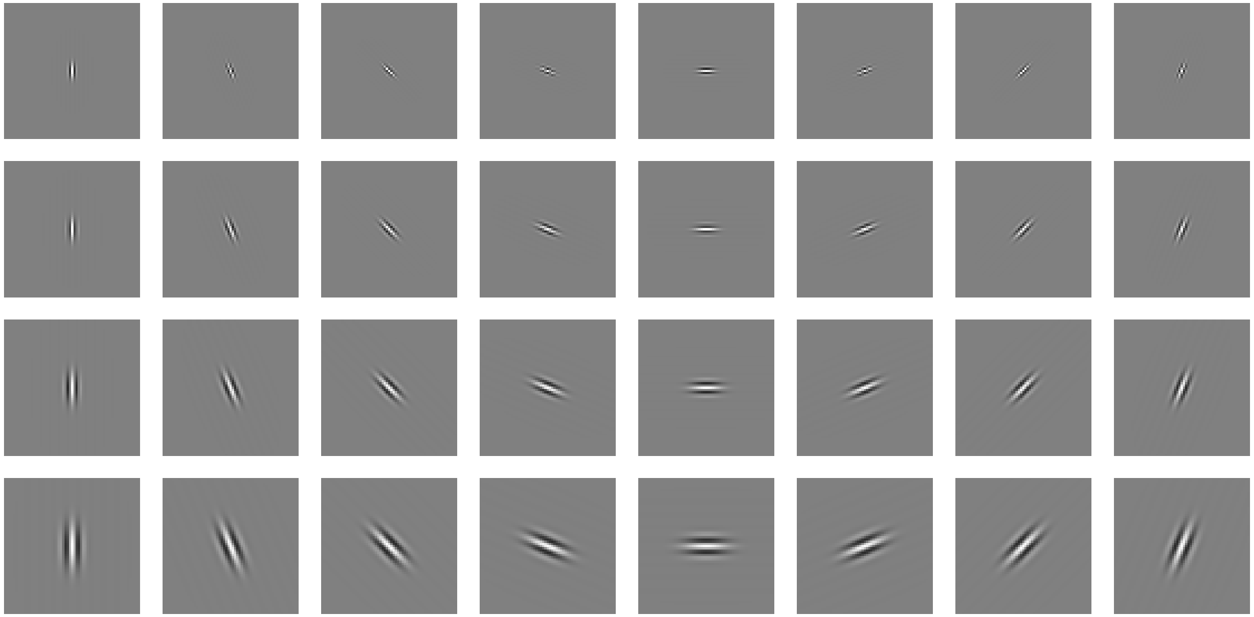
Why do we get different, distance-dependent interpretations of hybrid images?



?

Clues from Human Perception

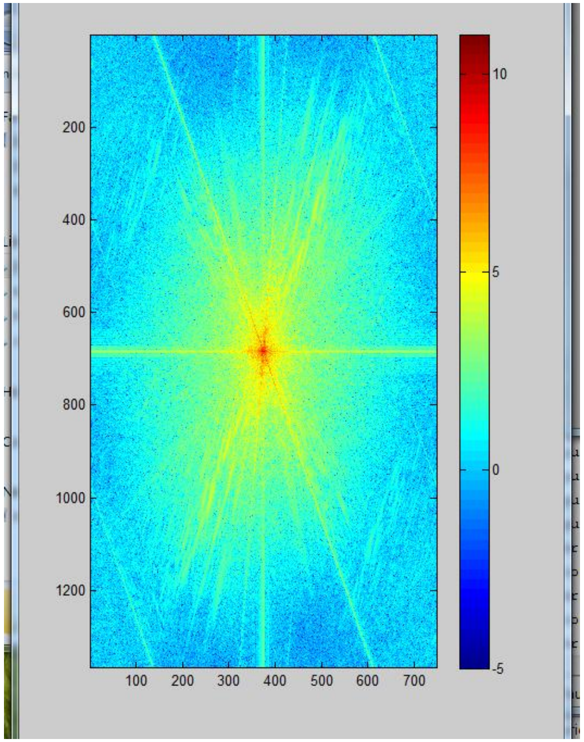
- Early processing in humans filters for various orientations and scales of frequency
- Perceptual cues in the mid frequencies dominate perception
- When we see an image from far away, we are effectively subsampling it



Early Visual Processing: Multi-scale edge and blob filters

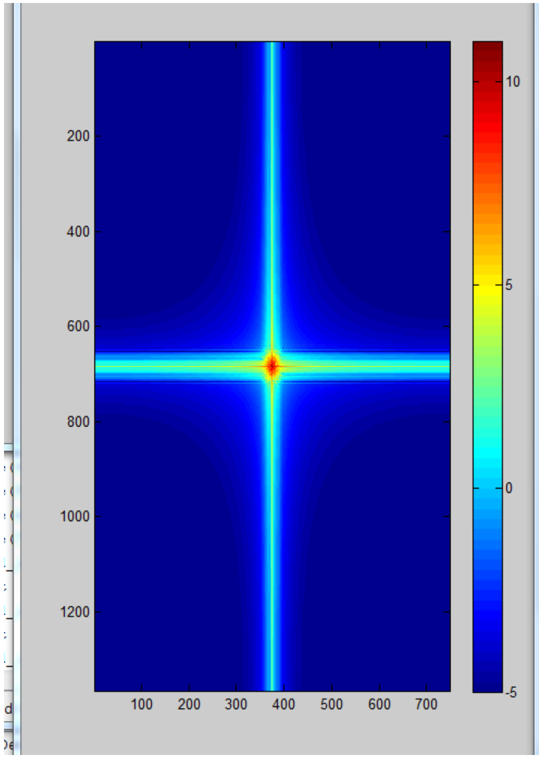
Hybrid Image in FFT

Hybrid Image



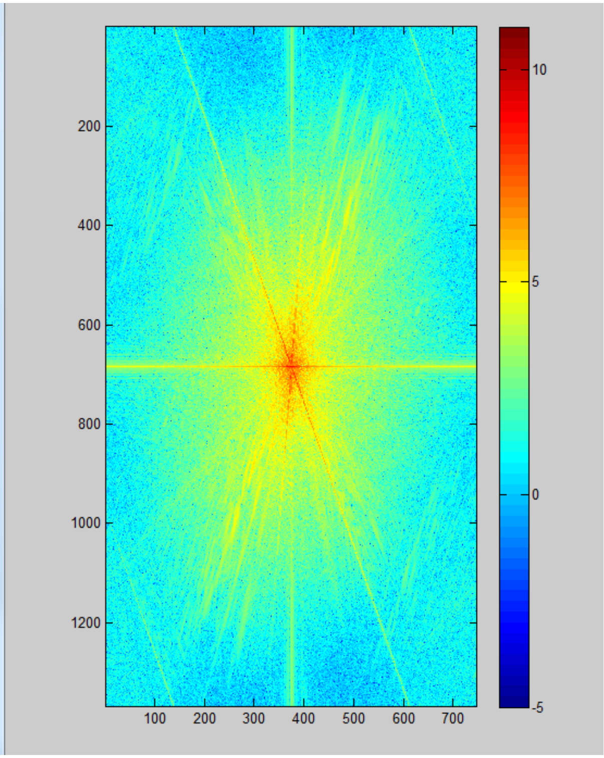
=

Low-passed Image



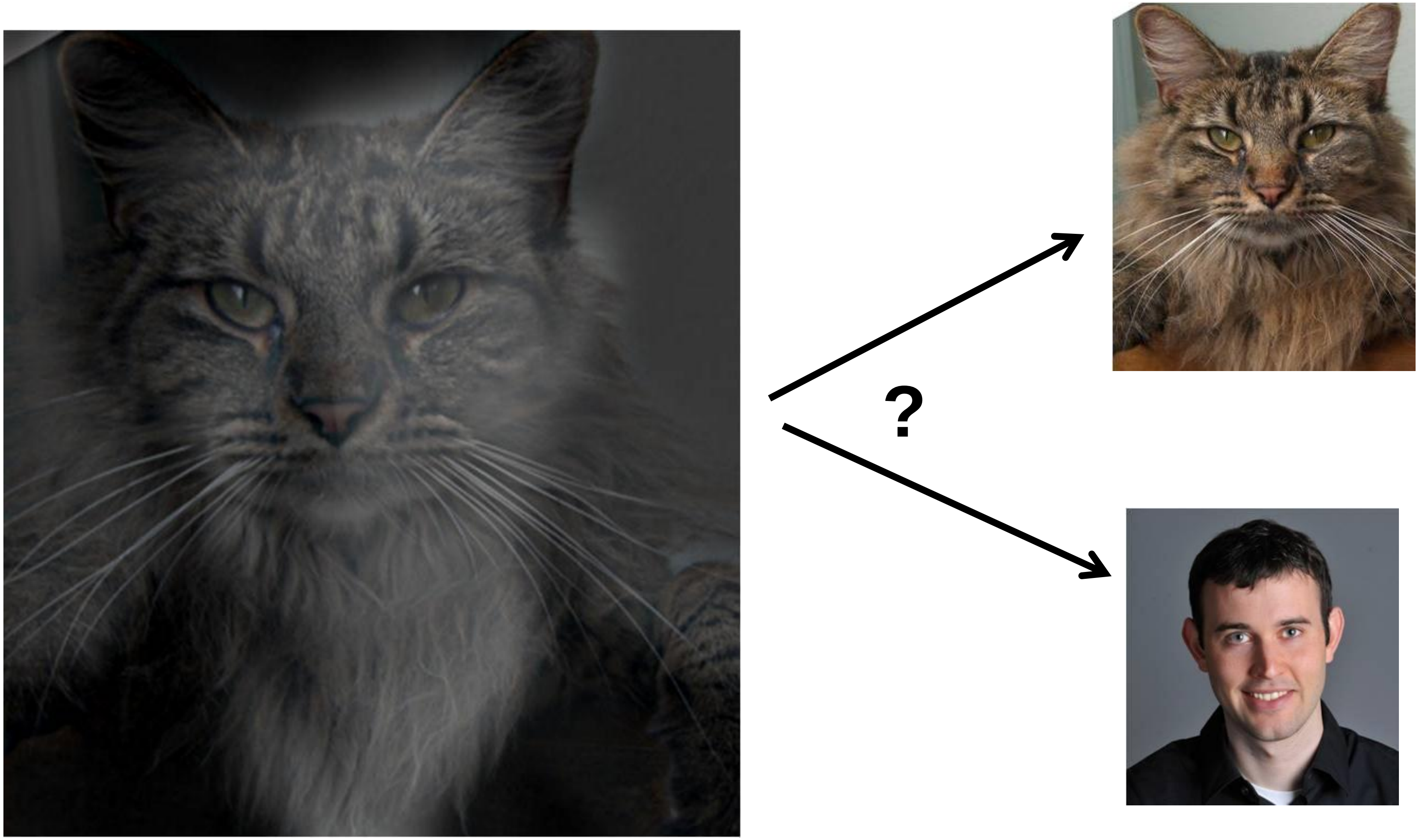
+

High-passed Image



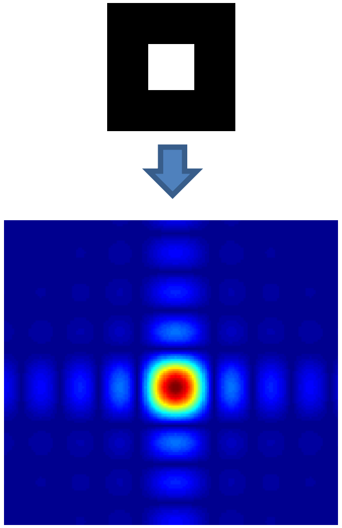
Perception

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



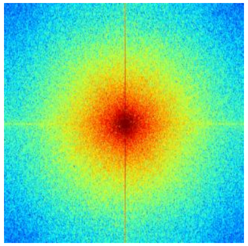
Things to Remember

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

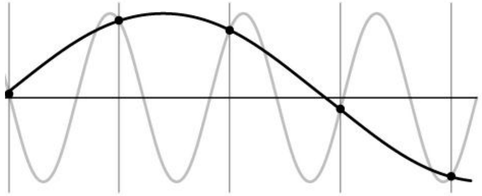


- Can be faster to filter using FFT for large images (N logN vs. N^2 for auto-correlation)

- Images are mostly smooth
  - Basis for compression



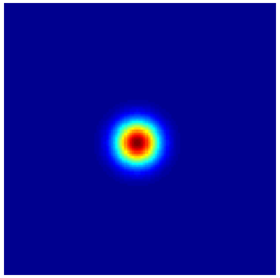
- Remember to low-pass before sampling



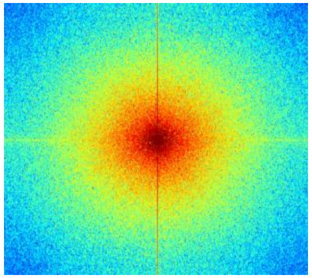
Take-home question

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

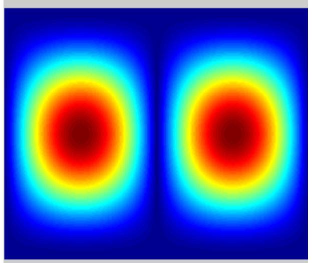
1



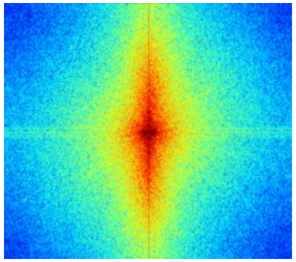
2



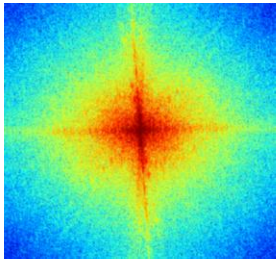
3



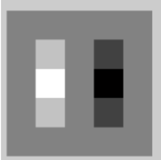
4



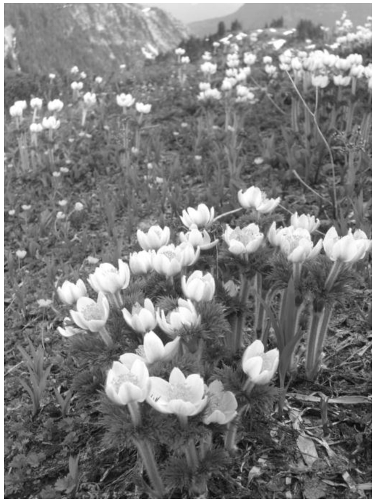
5



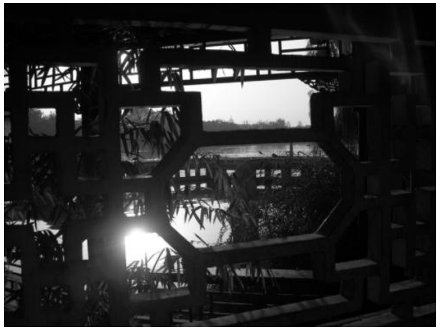
A



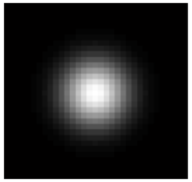
B



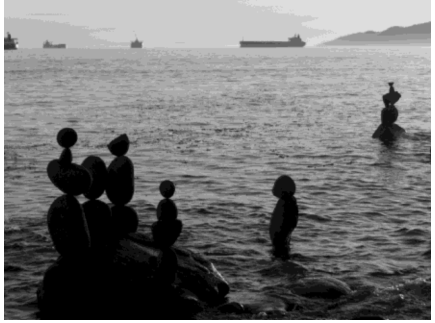
C



D



E



Next class: applications of filtering

- Denoising
- Template matching
- Image pyramids
- Compression

Questions