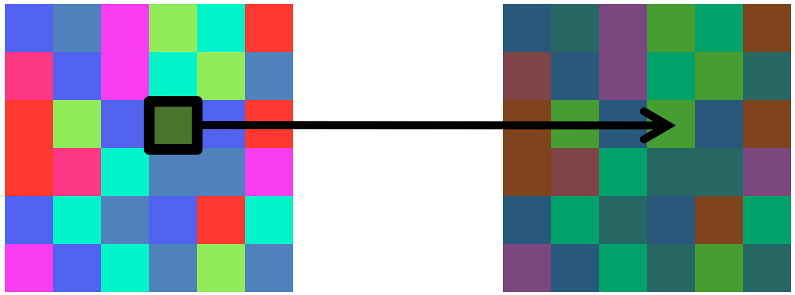
南京大學
NANJING UNIVERSITY

计算机视觉表征与识别
Chapter 3: Frequency Domain and Sampling

王利民
媒体计算课题组
http://mcg.nju.edu.cn/

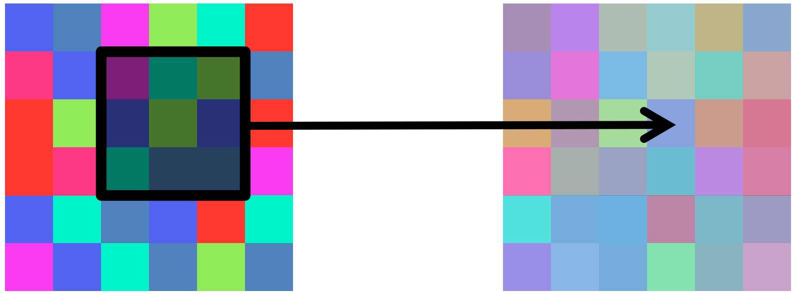
Image filtering

Point Operation



point processing

Neighborhood Operation



“filtering”

Image filtering

- Compute a function of the local neighborhood at each pixel in the image
  - Function specified by a “filter” or mask saying how to combine values from neighbors.

- Uses of filtering:
  - Enhance an image (denoise, resize, etc)
  - Extract information (texture, edges, etc)
  - Detect patterns (template matching)

Adapted from Derek Hoiem

Three views of filtering

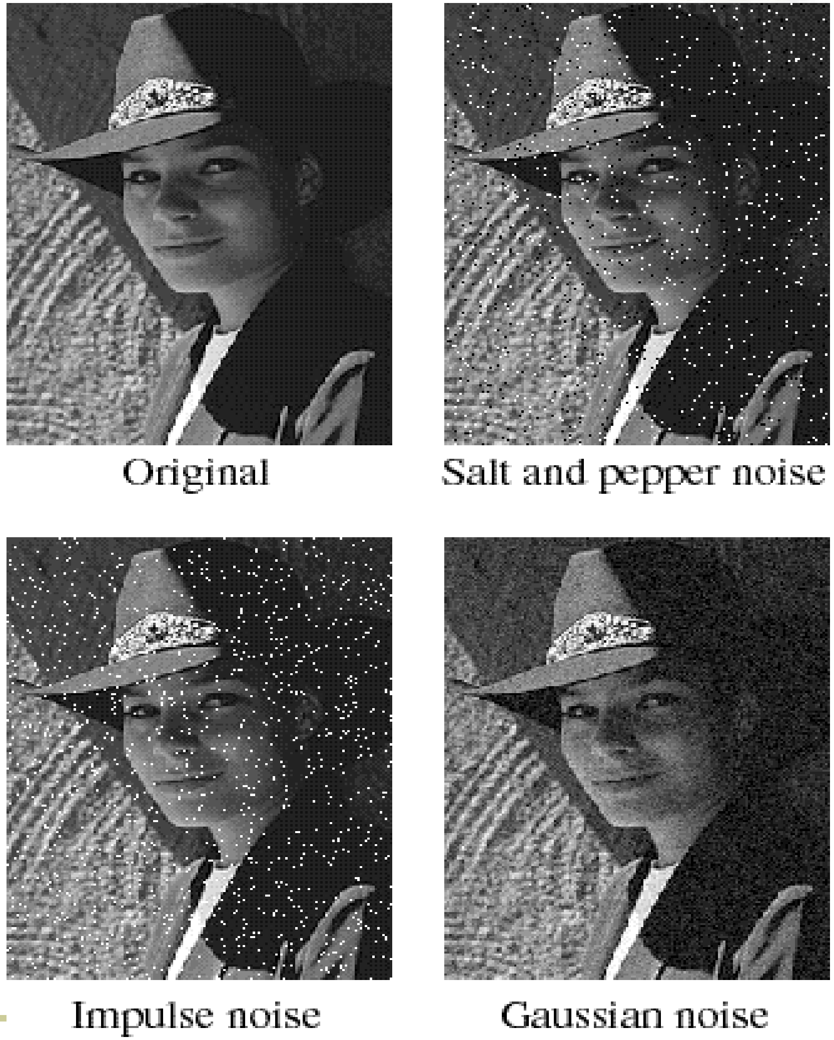
- Image filters in spatial domain
  - Filter is a mathematical operation on values of each patch
  - Smoothing, sharpening, measuring texture

- Image filters in the frequency domain
  - Filtering is a way to modify the frequencies of images
  - Denoising, sampling, image compression

- Templates and Image Pyramids
  - Filtering is a way to match a template to the image
  - Detection, coarse-to-fine registration

Common types of noise

- **Salt and pepper noise**: random occurrences of black and white pixels
- **Impulse noise:** random occurrences of white pixels
- **Gaussian noise**: variations in intensity drawn from a Gaussian normal distribution



Correlation filtering

Say the averaging window size is 2k+1 x 2k+1:

$$G[i,j] = \underbrace{\frac{1}{(2k+1)^2}}_{\text{Attribute uniform weight to each pixel}} \underbrace{\sum_{u=-k}^{k}\sum_{v=-k}^{k} F[i+u, j+v]}_{\text{Loop over all pixels in neighborhood around image pixel } F[i,j]}$$

Now generalize to allow different weights depending on neighboring pixel's relative position:

$$G[i,j] = \sum_{u=-k}^{k}\sum_{v=-k}^{k} \underbrace{H[u,v]}_{\text{Non-uniform weights}} F[i+u, j+v]$$

Averaging filter

- What values belong in the kernel *H* for the moving average example?

$F[x,y]$ $\otimes$ $H[u,v]$ → $G[x,y]$

$F[x,y]$:

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

$H[u,v]$:

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

?

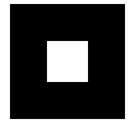
"box filter"

$G[x,y]$:

	0	20	40	60	60				

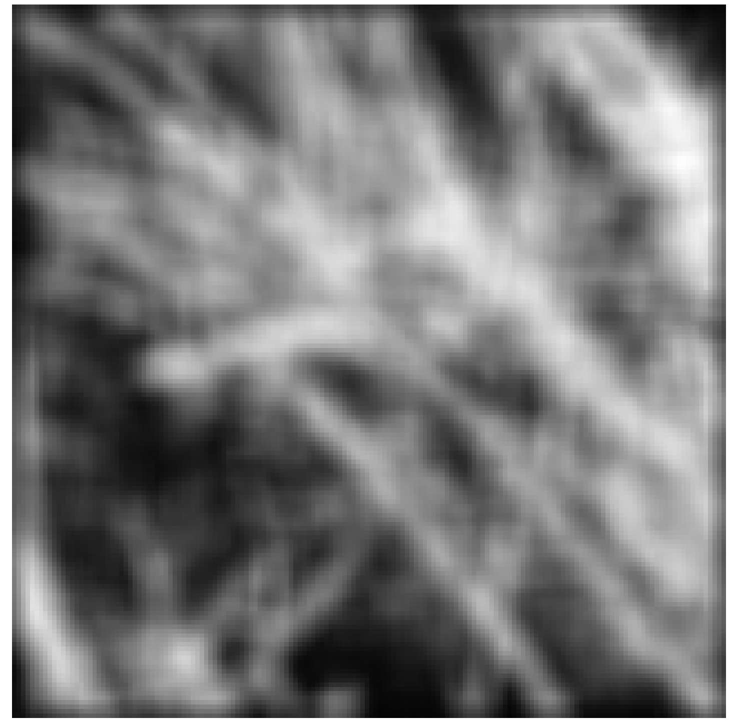
$$G = H \otimes F$$

Smoothing by averaging



depicts box filter:
white = high value, black = low value

original



filtered

What if the filter size was 5 x 5 instead of 3 x 3?

Gaussian filter

- What if we want nearest neighboring pixels to have the most influence on the output?

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

$F[x, y]$

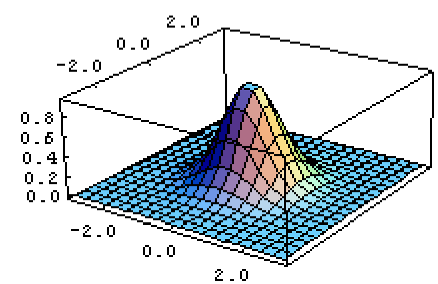
$\frac{1}{16}$

1	2	1
2	4	2
1	2	1

$H[u, v]$

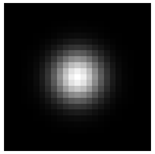
This kernel is an approximation of a 2d Gaussian function:

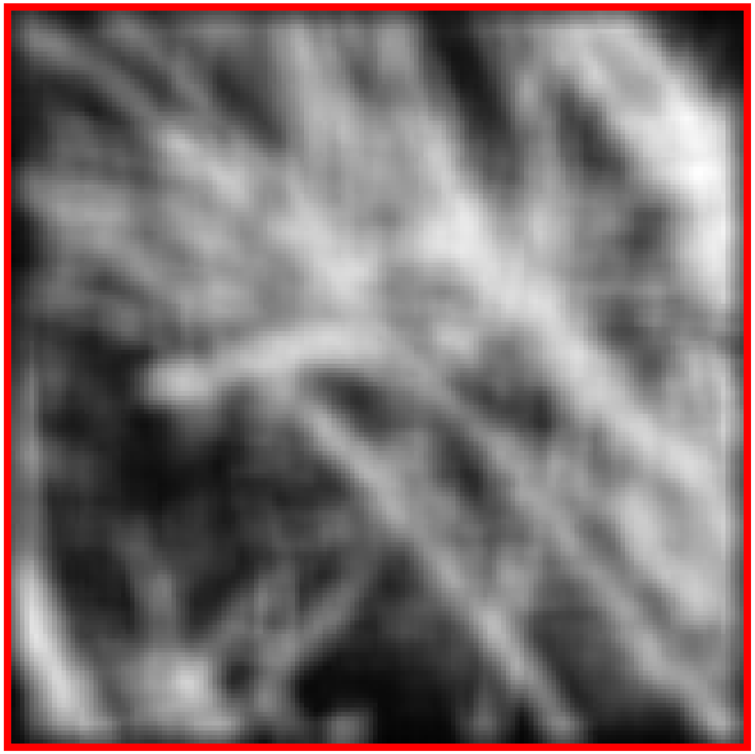
$$h(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{\sigma^2}}$$



- Removes high-frequency components from the image ("low-pass filter").

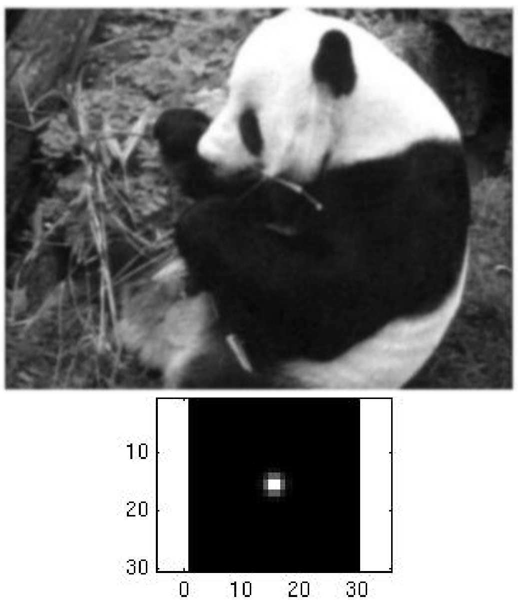
Smoothing with a Gaussian

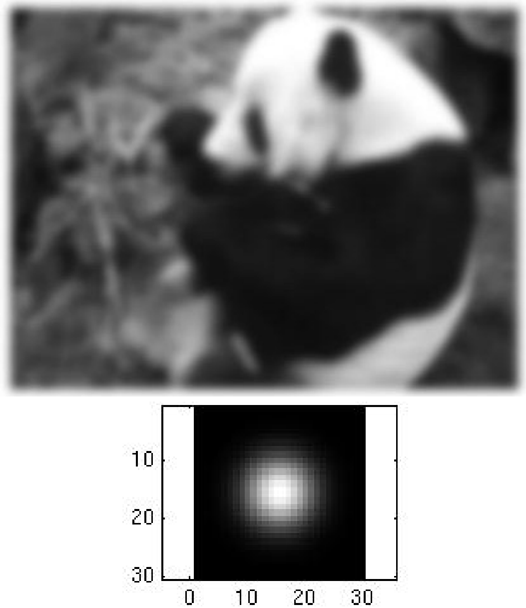




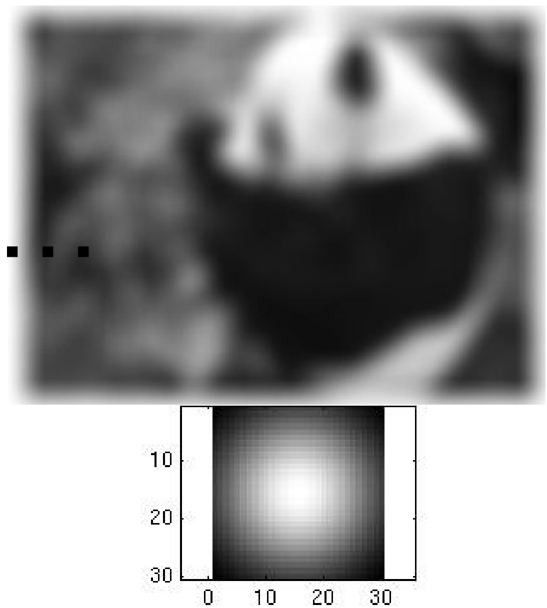
Smoothing with a Gaussian

Parameter σ is the “scale” / “width” / “spread” of the Gaussian kernel, and controls the amount of smoothing.





…




```
for sigma=1:3:10
    h = fspecial('gaussian‘, fsize,
    sigma);
    out = imfilter(im, h);
    imshow(out);
    pause;
end
```

Convolution vs correlation

Definition of discrete 2D convolution:

notice the flip

$$(f * g)(x, y) = \sum_{i,j=-\infty}^{\infty} f(i, j) I(x - i, y - j)$$

Definition of discrete 2D correlation:

notice the lack of a flip

$$(f * g)(x, y) = \sum_{i,j=-\infty}^{\infty} f(i, j) I(x + i, y + j)$$

- Most of the time won't matter, because our kernels will be symmetric.

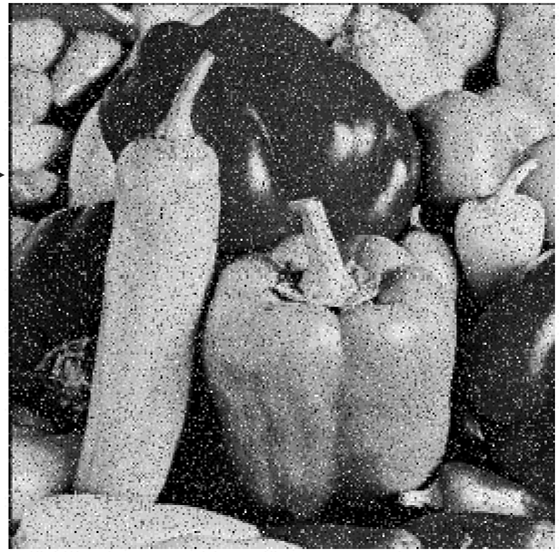
Separability

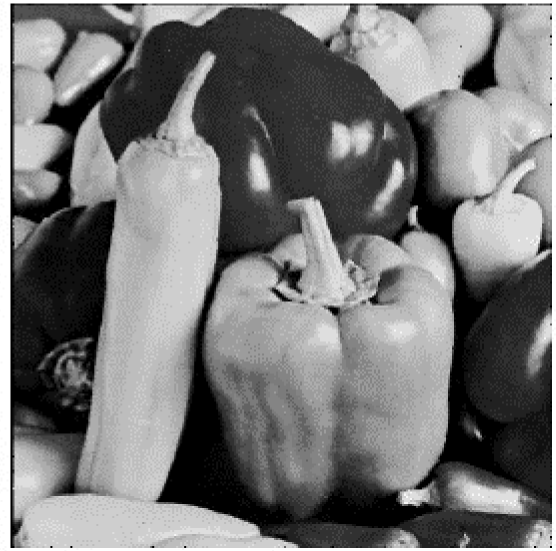
- In some cases, filter is separable, and we can factor into two steps:
  - Convolve all rows with a 1D filter
  - Convolve all columns with a 1D filter

$$\begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & 0 \end{bmatrix} \circ \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 0 \end{bmatrix}$$

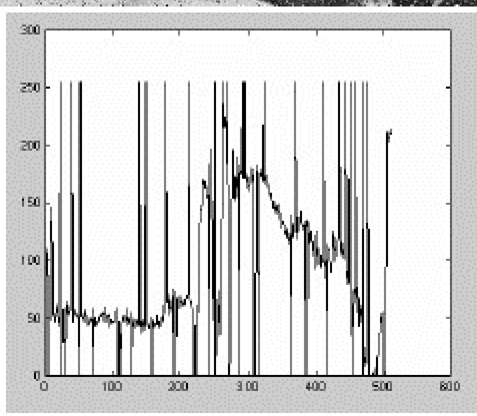
Median filter

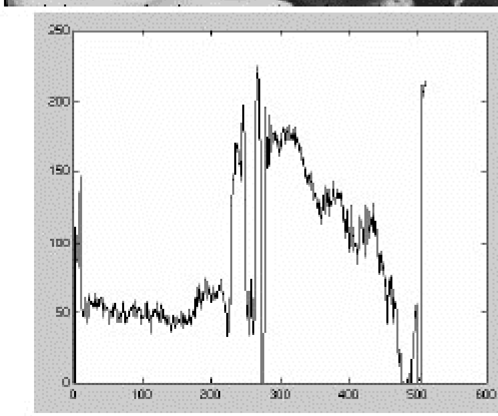
Salt and pepper noise





Median filtered





Plots of a row of the image

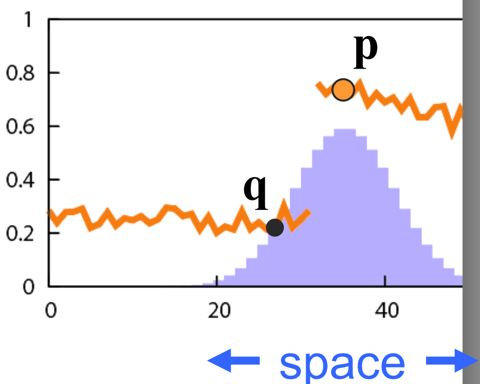
Matlab: output im = medfilt2(im, [h w]);

Objective of bilateral filtering

Smooth texture

Preserve edges

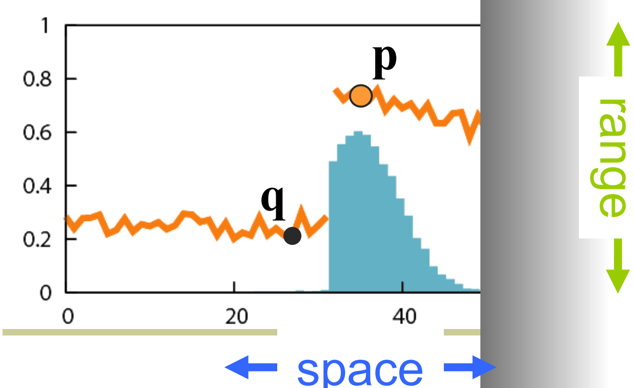
Definition



Gaussian blur

$$I^{\mathrm{b}}_{\mathbf{p}} = \sum_{\mathbf{q}\in\mathcal{S}} \underbrace{\boxed{G_{\sigma_{\mathrm{s}}}(\|\mathbf{p}-\mathbf{q}\|)}}_{\text{space}} I_{\mathbf{q}}$$

- only spatial distance, intensity ignored



Bilateral filter

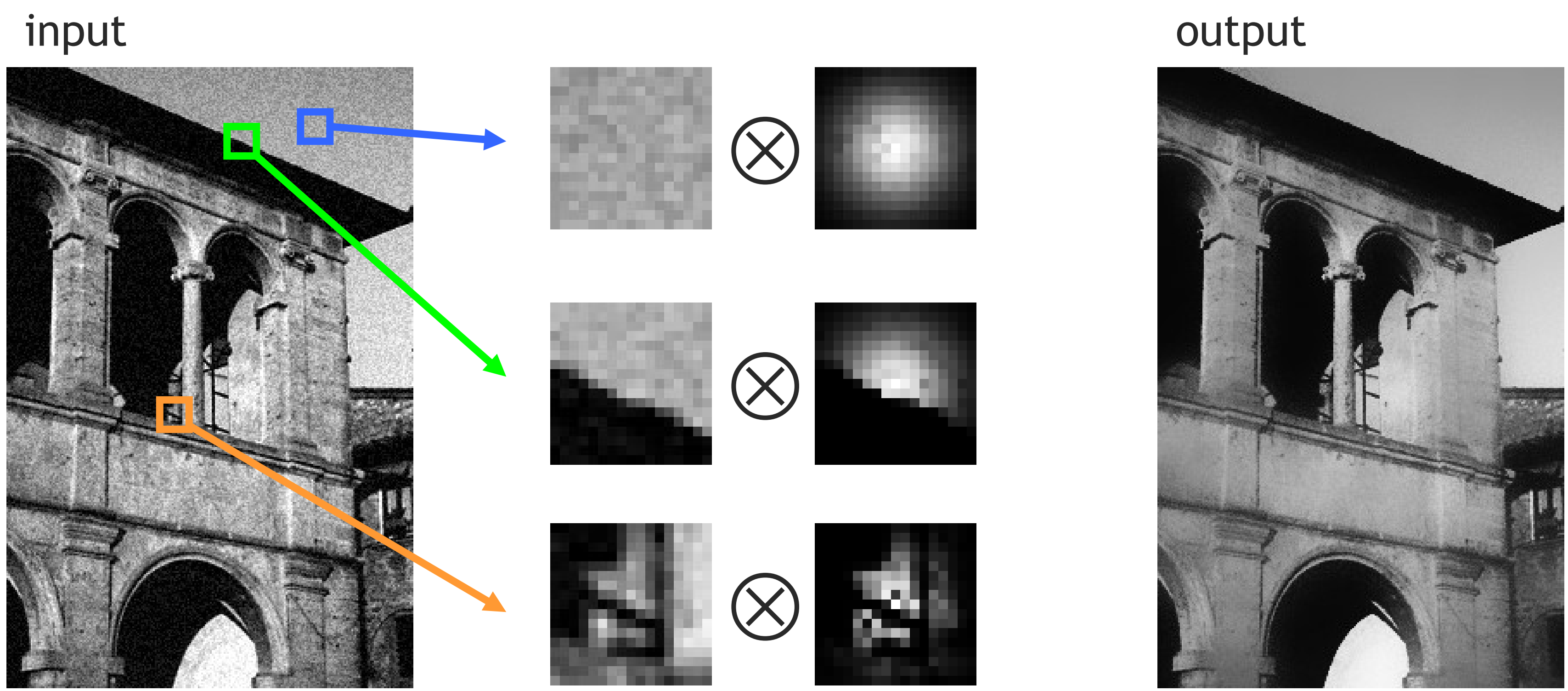
[Aurich 95, Smith 97, Tomasi 98]

$$I^{\mathrm{bf}}_{\mathbf{p}} = \underbrace{\boxed{\frac{1}{W^{\mathrm{bf}}_{\mathbf{p}}}}}_{\text{normalization}} \sum_{\mathbf{q}\in\mathcal{S}} \underbrace{\boxed{G_{\sigma_{\mathrm{s}}}(\|\mathbf{p}-\mathbf{q}\|)}}_{\text{space}} \underbrace{\boxed{G_{\sigma_{\mathrm{r}}}(|I_{\mathbf{p}}-I_{\mathbf{q}}|)}}_{\text{range}} I_{\mathbf{q}}$$

- spatial and range distances
    - weights sum to 1

Example on a Real Image

- Kernels can have complex, spatially varying shapes.



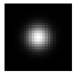
Three views of filtering

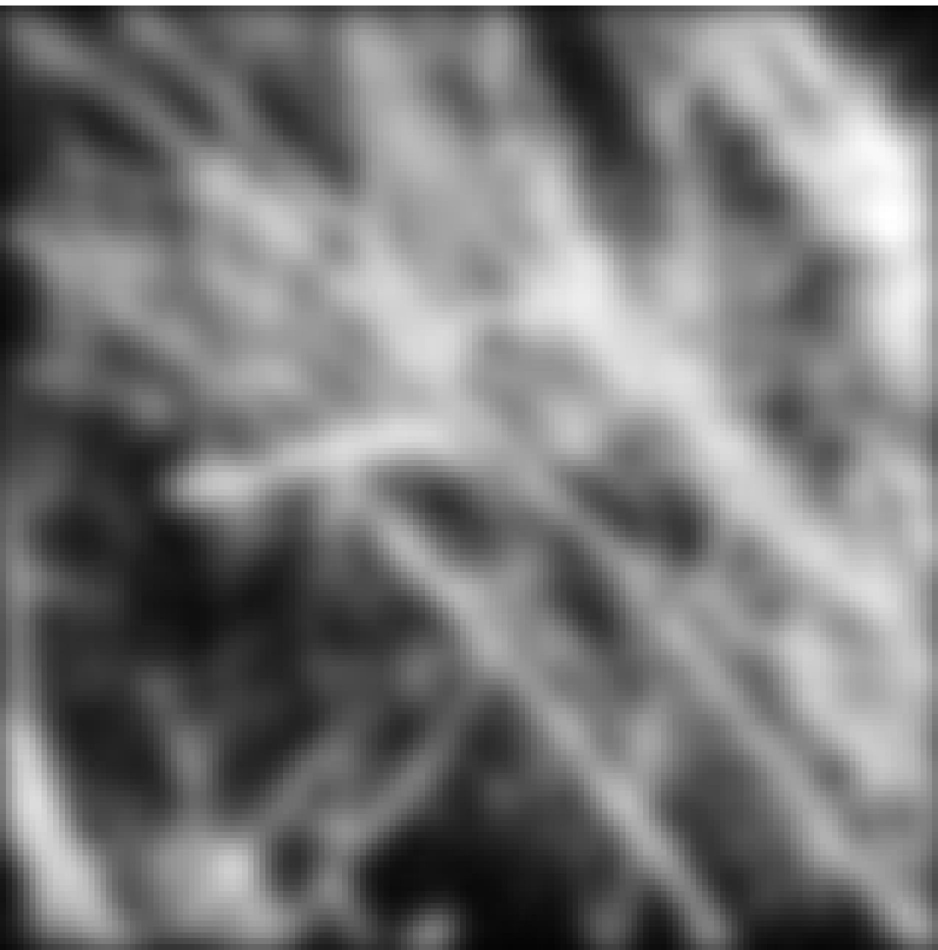
- Image filters in spatial domain
  - Filter is a mathematical operation on values of each patch
  - Smoothing, sharpening, measuring texture
- Image filters in the frequency domain
  - Filtering is a way to modify the frequencies of images
  - Denoising, sampling, image compression
- Templates and Image Pyramids
  - Filtering is a way to match a template to the image
  - Detection, coarse-to-fine registration

Today's Class

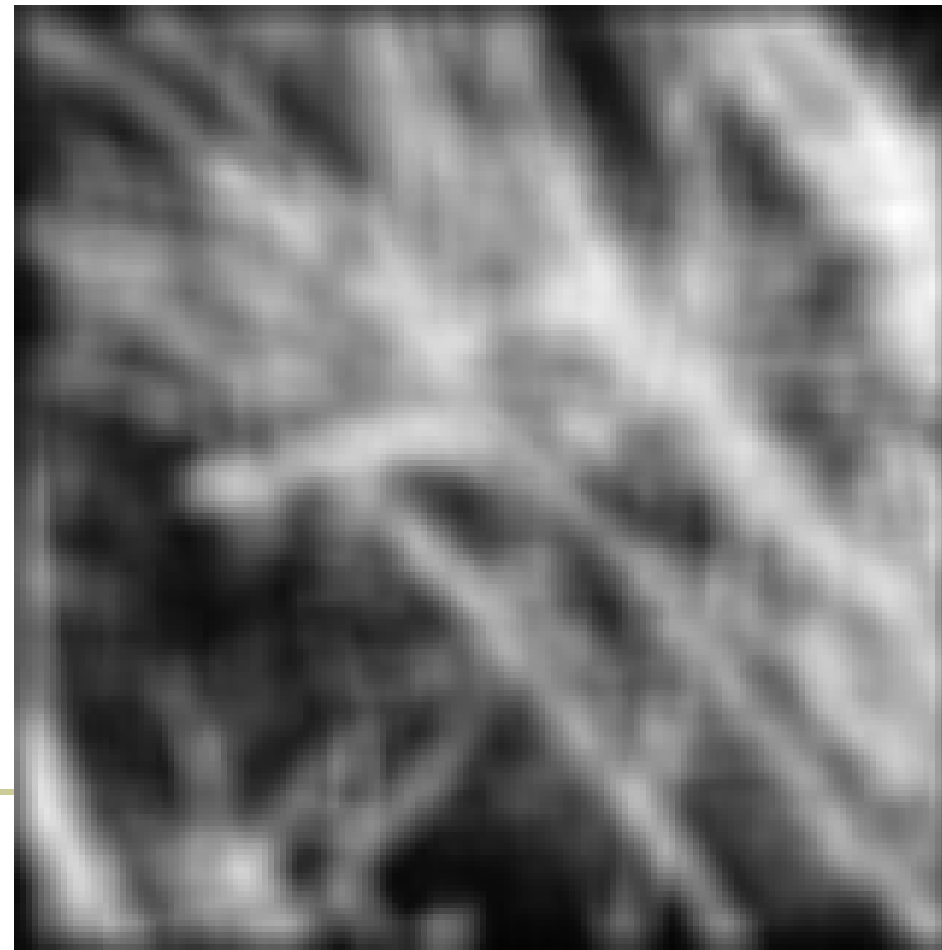
- Fourier transform and frequency domain
  - Frequency view of filtering
- Hybrid Image
- Sampling

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

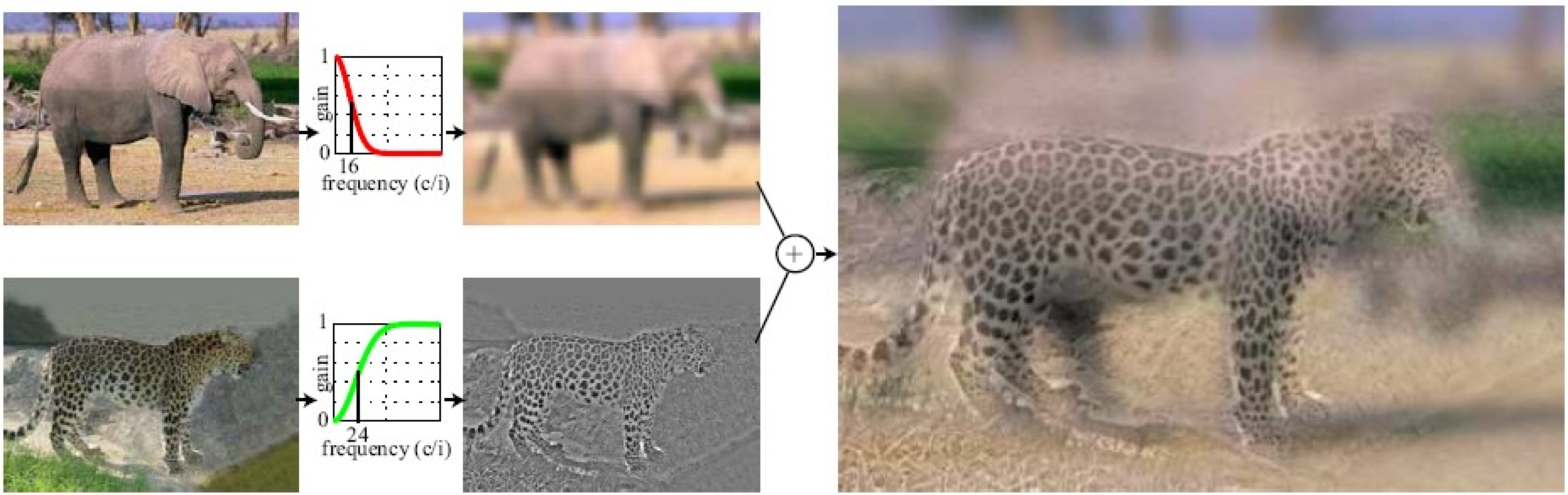
Gaussian 



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/

Jean Baptiste Joseph Fourier (1768-1830)

had crazy idea (1807):

...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.

***Any** univariate function ca*
be rewritten as a weighted
sum of sines and cosines o
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!
  - called Fourier Series
  - there are some subtle restrictions

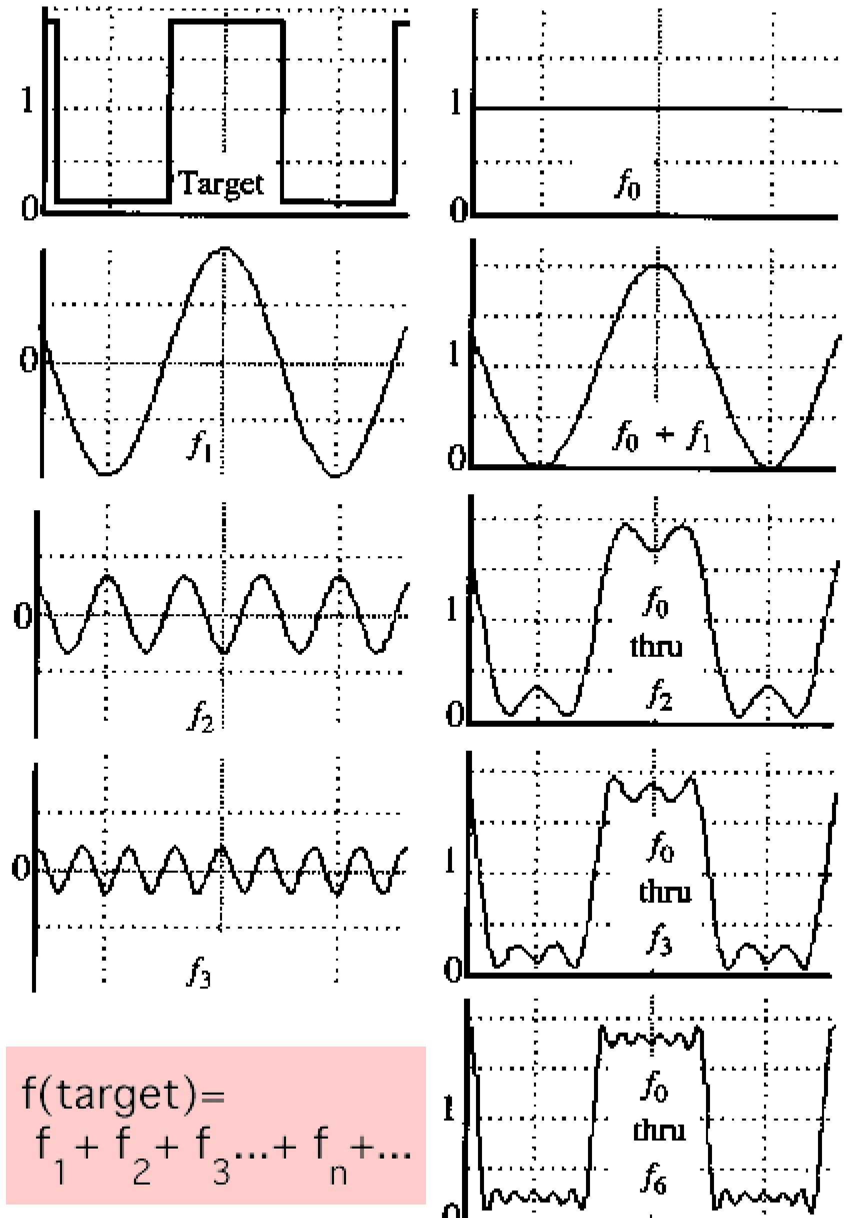
Slides: Efros

A sum of sines

Our building block:

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

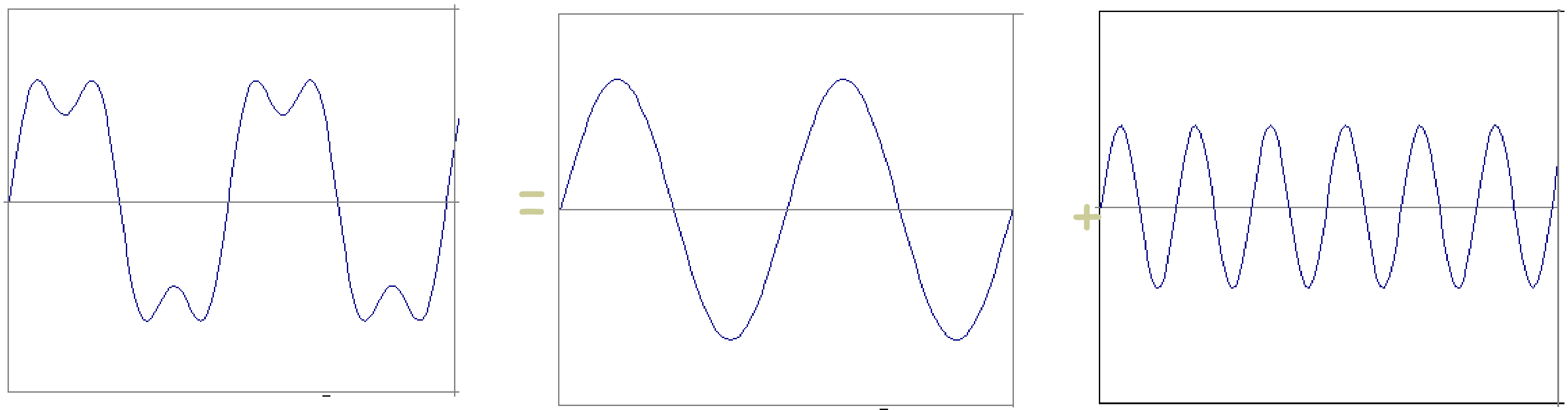
Add enough of them to get any signal *f(x)* you want!

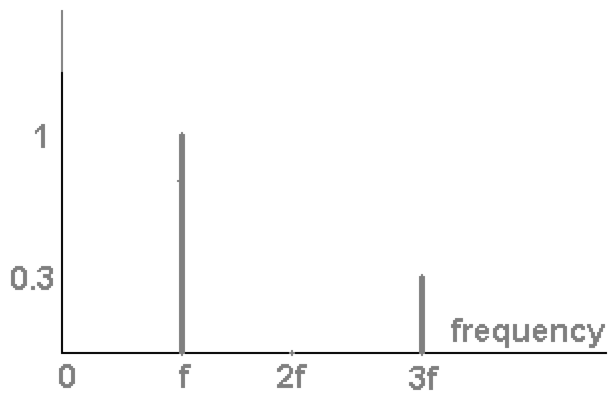


f(target)=
$f_1 + f_2 + f_3 ... + f_n + ...$

Frequency Spectra

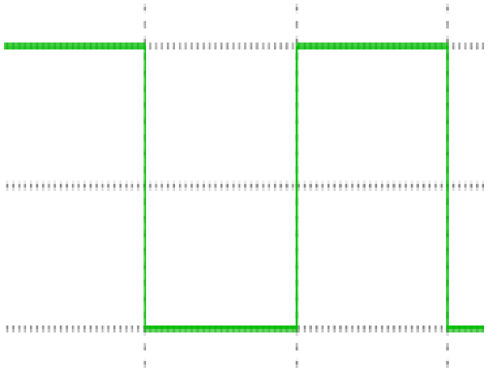
- example : $g(t) = \sin(2\pi f\, t) + (1/3)\sin(2\pi(3f)\, t)$



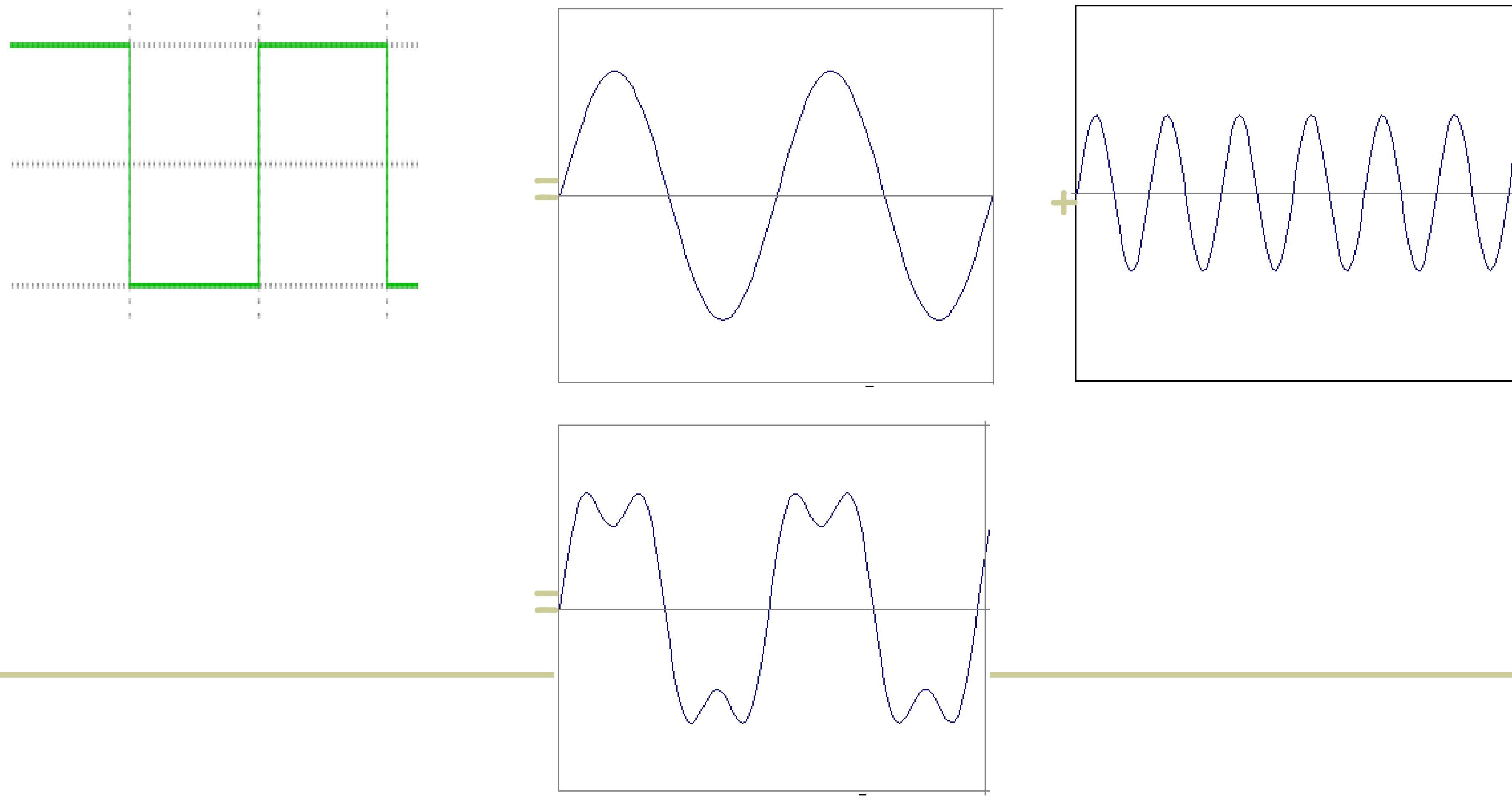


Slides: Efros

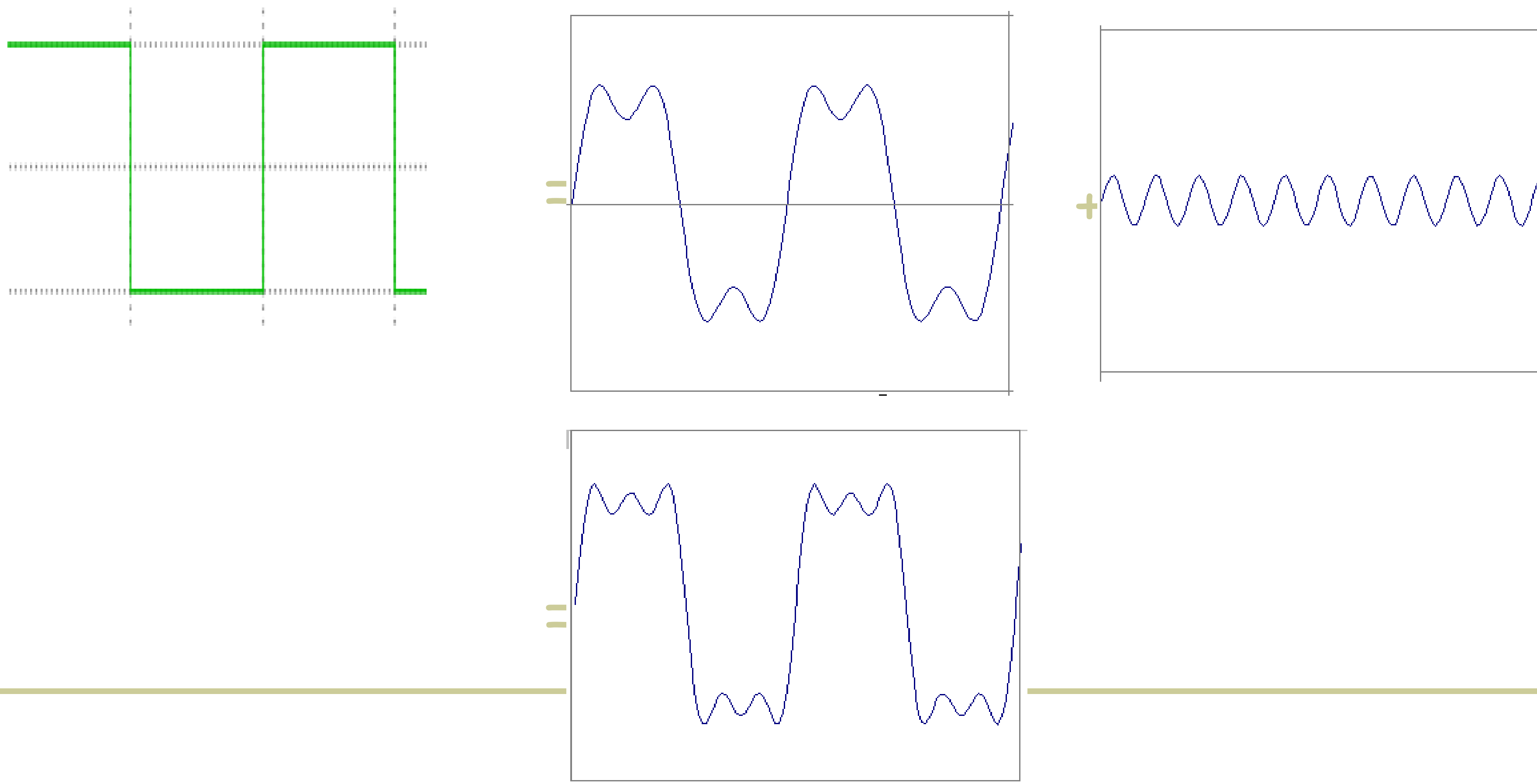
Frequency Spectra



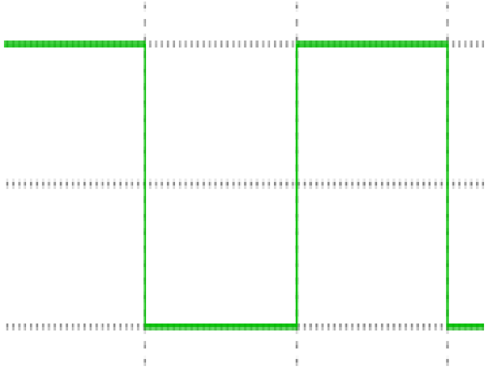
Frequency Spectra

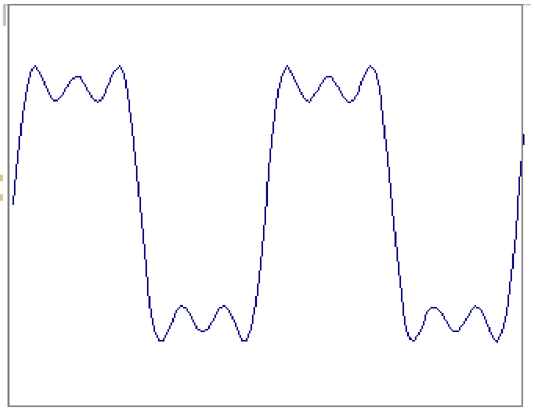


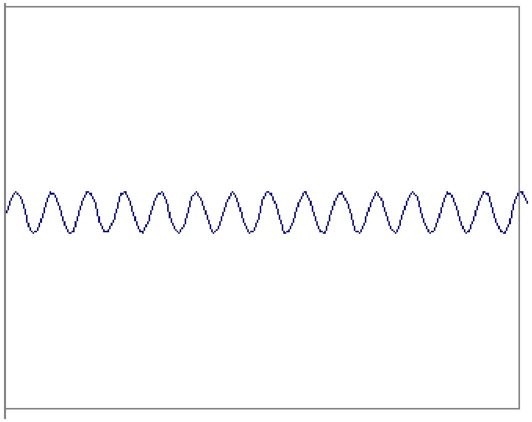
Frequency Spectra

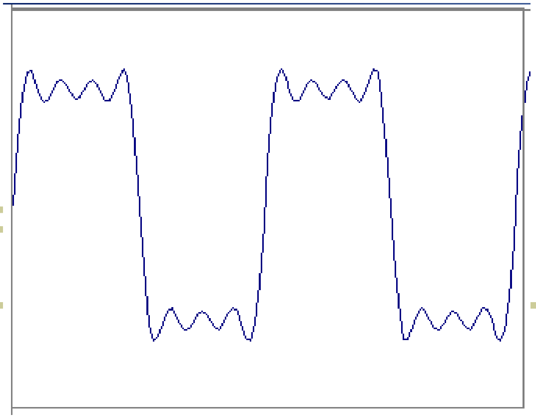


Frequency Spectra

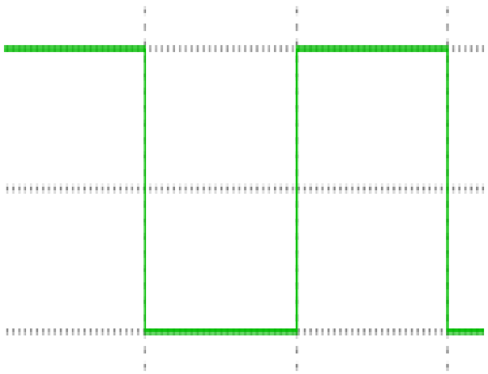


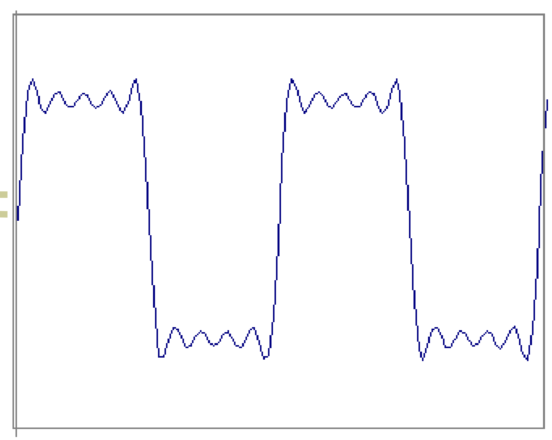


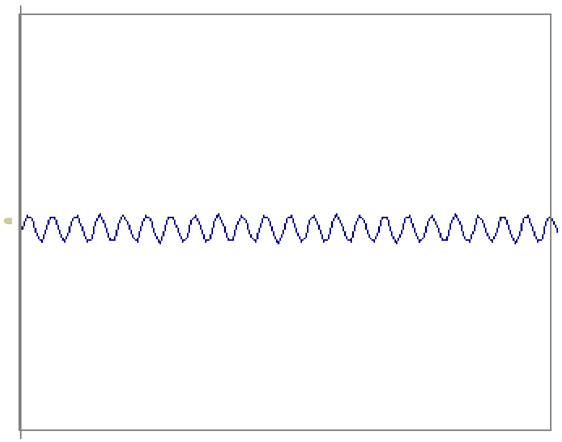


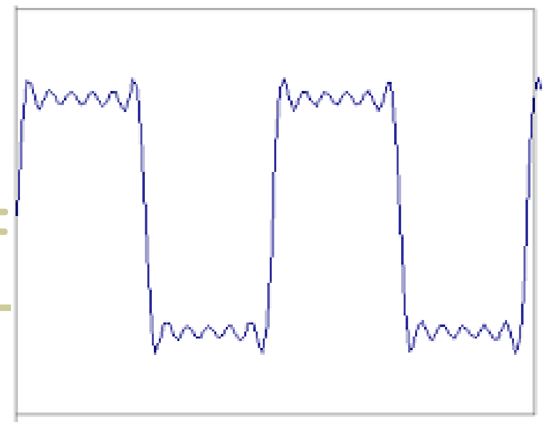


Frequency Spectra

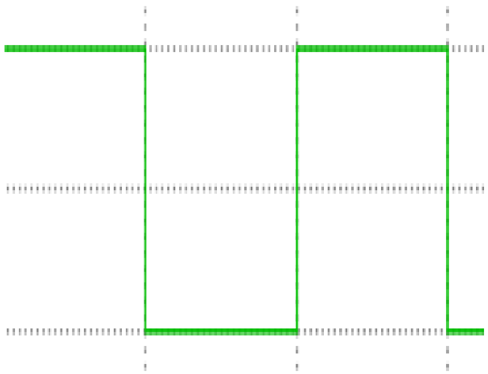


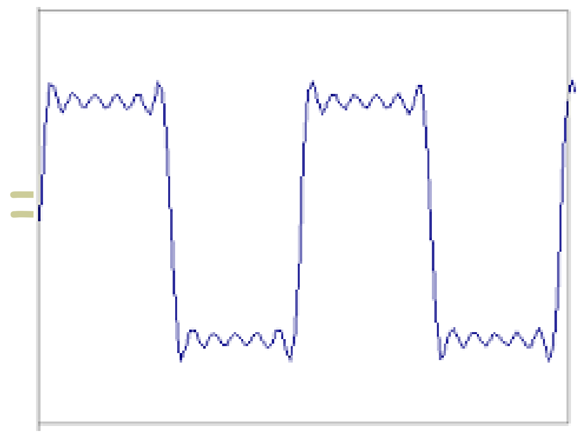


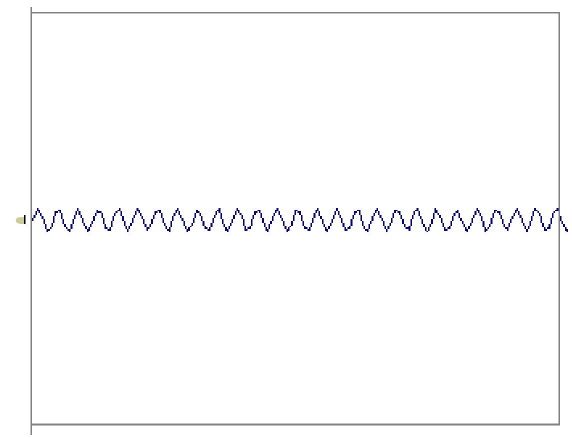


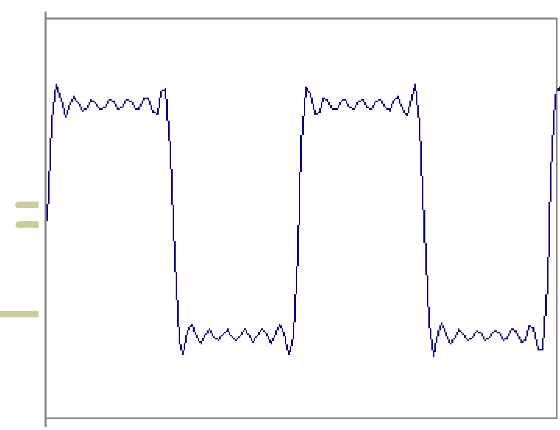


Frequency Spectra

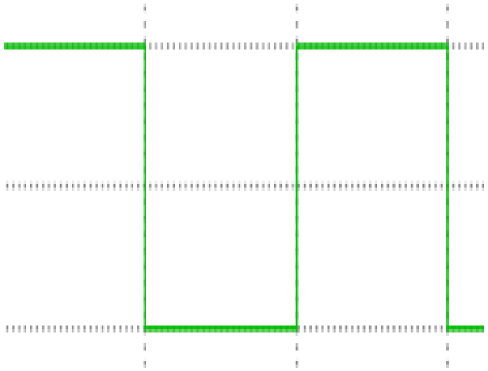




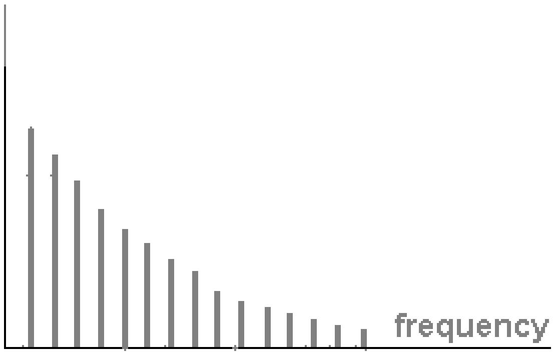




Frequency Spectra

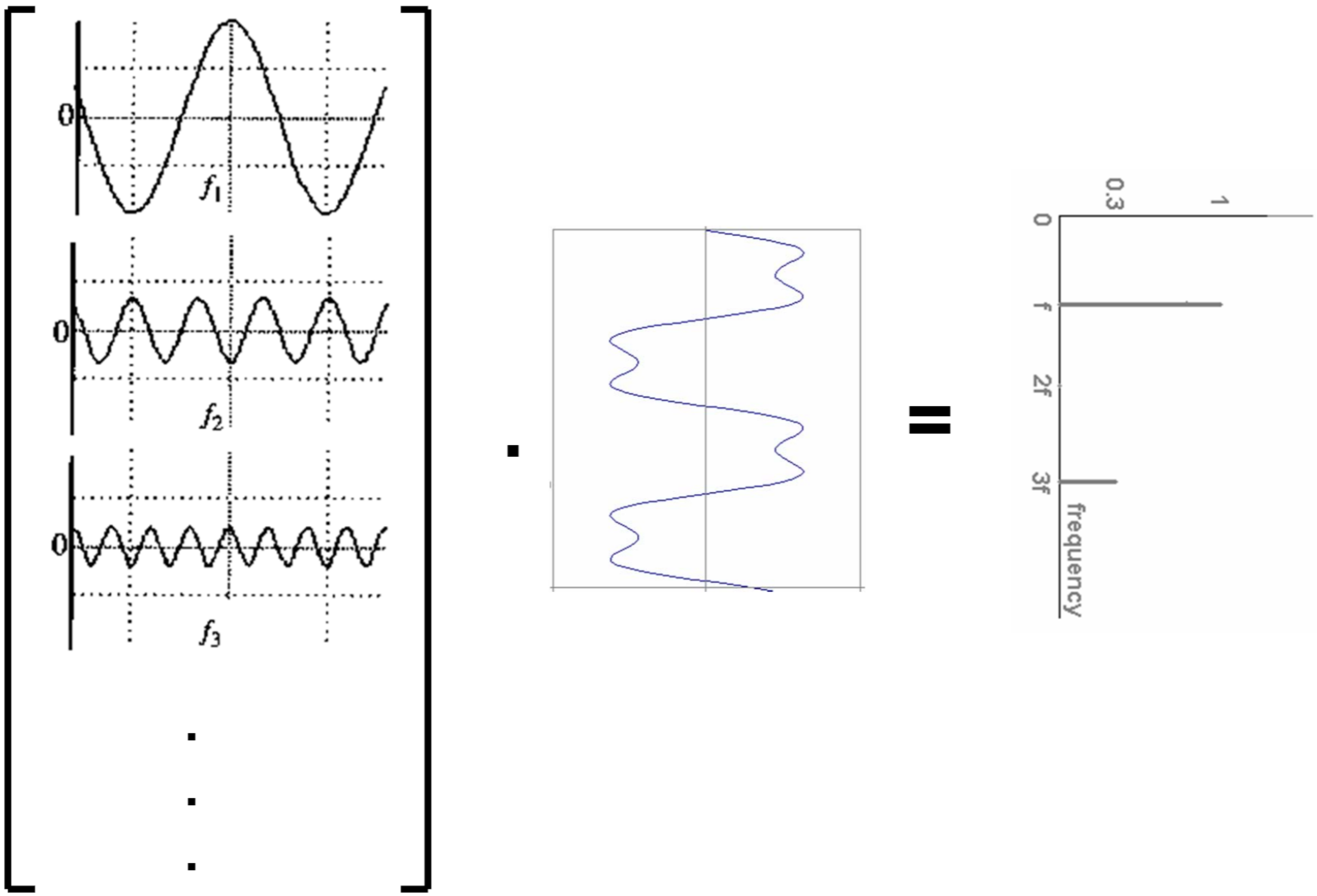


$$= \boxed{A\sum_{k=1}^{\infty}\frac{1}{k}\sin(2\pi kt)}$$



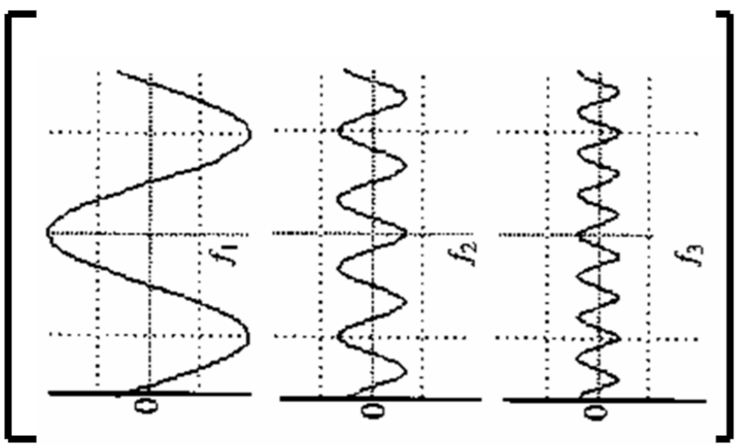
Fourier series: just a change of basis

$$M\ f(x) = F(\omega)$$

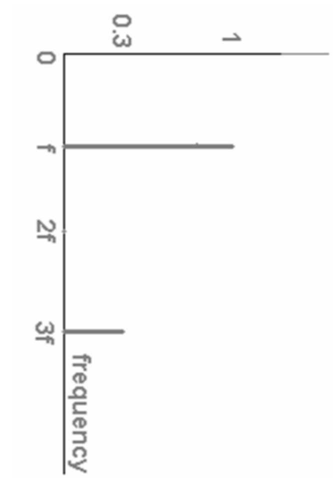


Inverse FT: Just a change of basis

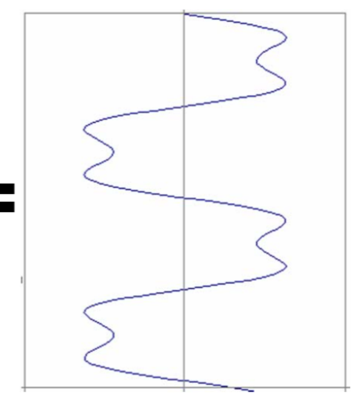
$$M^{-1}\, F(\omega) = f(x)$$



$\cdot$



=

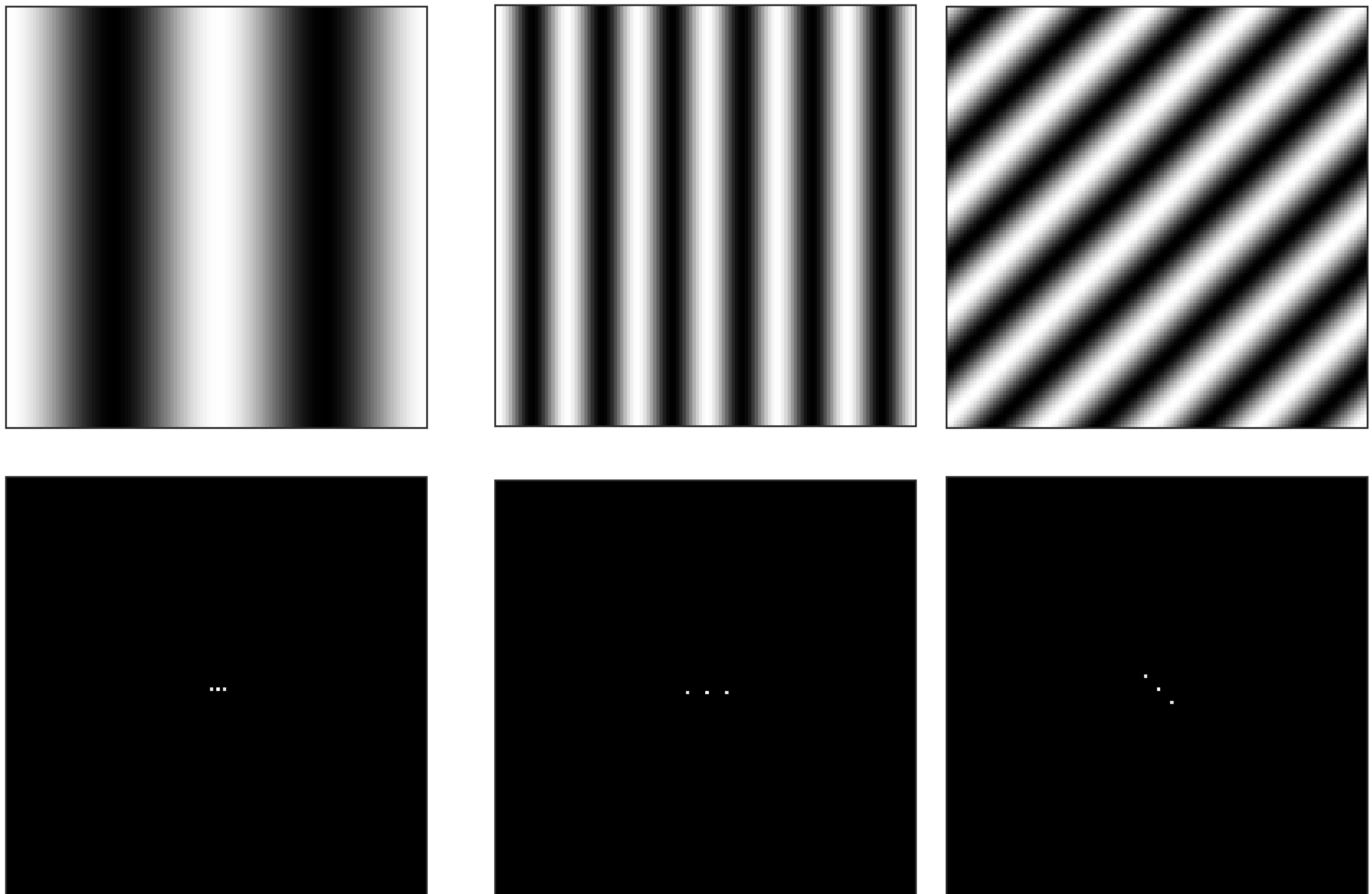


$\vdots$

Fourier analysis in images

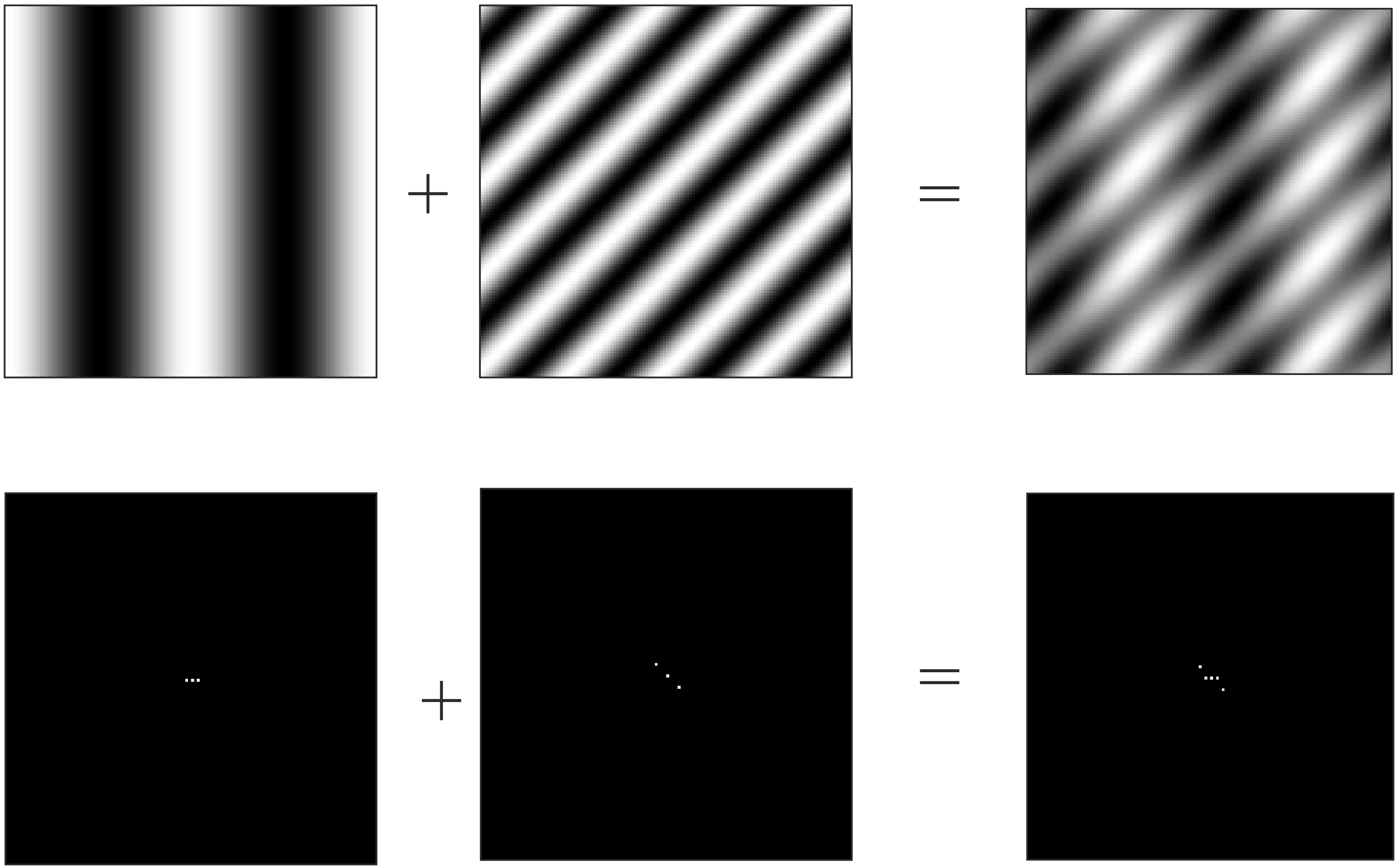
Intensity Image

Fourier Image



http://sharp.bu.edu/~slehar/fourier/fourier.html#filtering

Signals can be composed



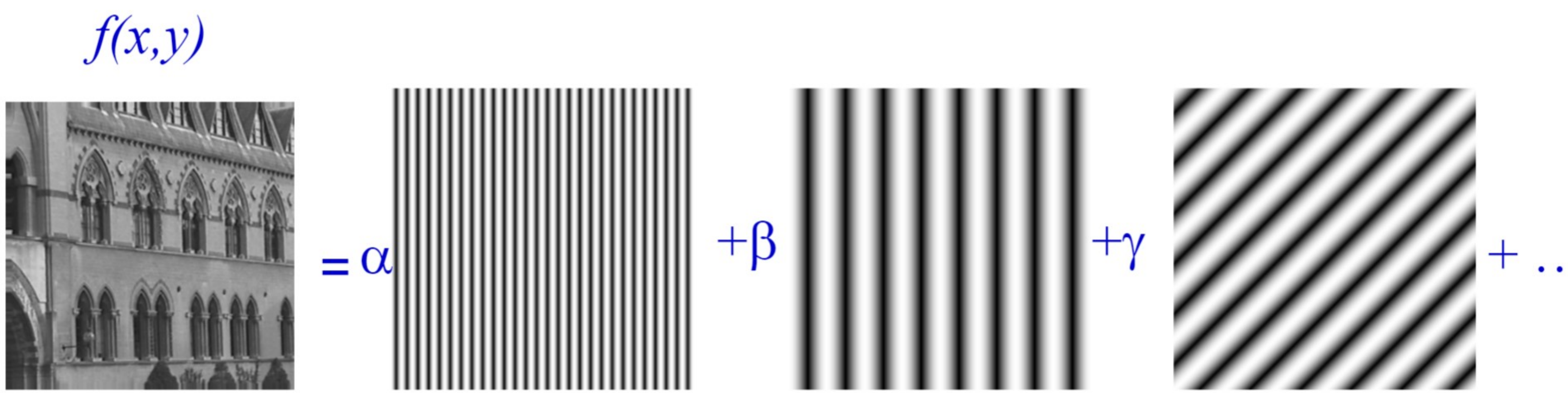
http://sharp.bu.edu/~slehar/fourier/fourier.html#filtering
More: http://www.cs.unm.edu/~brayer/vision/fourier.html

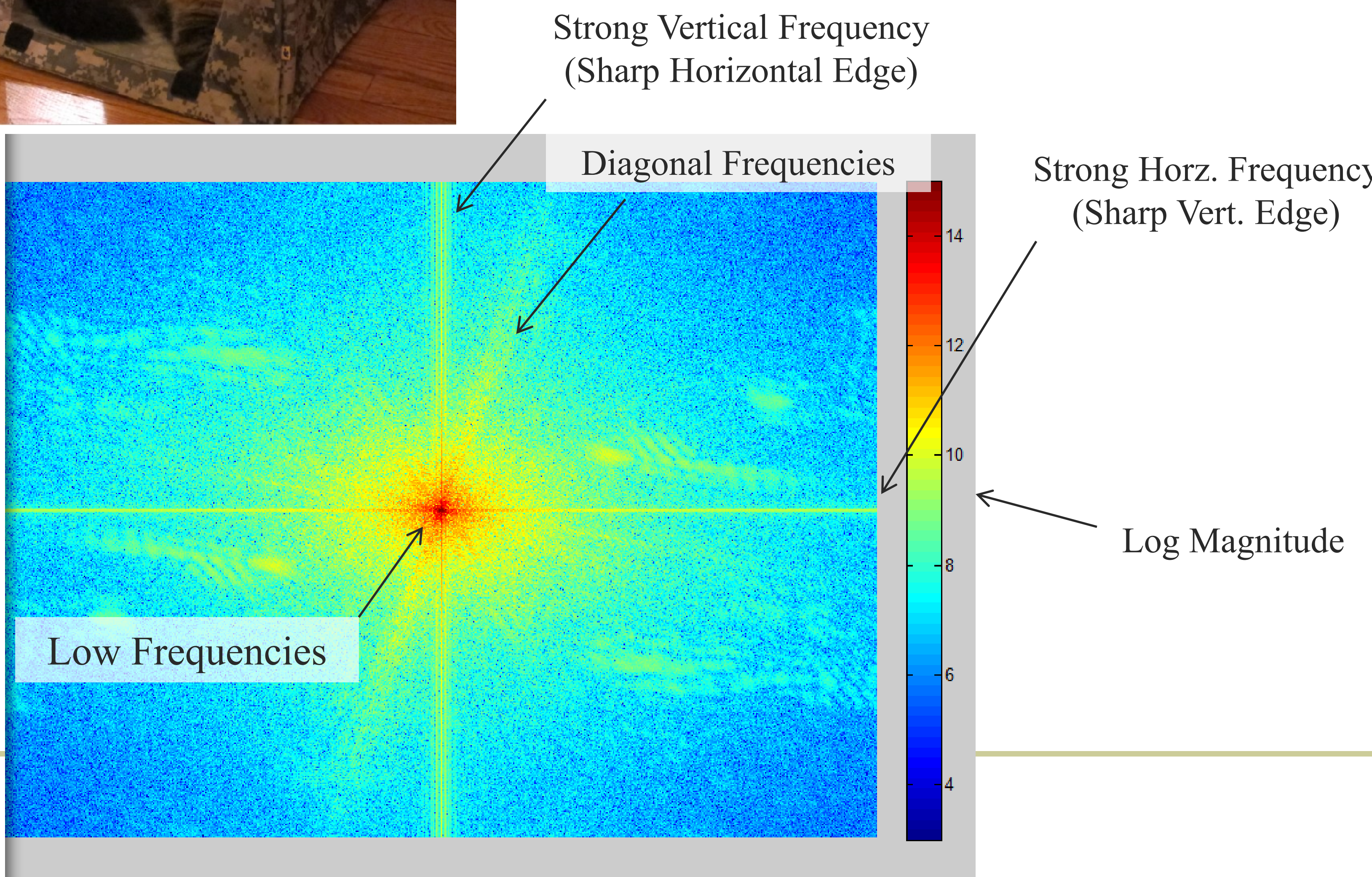
Summary

The spatial function $f(x, y)$

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{j2\pi(ux+vy)} \, du \, dv$$

is decomposed into a weighted sum of 2D orthogonal basis functions in a similar manner to decomposing a vector onto a basis using scalar products.




Strong Vertical Frequency
(Sharp Horizontal Edge)
Diagonal Frequencies
Strong Horz. Frequency
(Sharp Vert. Edge)
Log Magnitude
Low Frequencies
14
12
10
8
6
4

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

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

Fast Fourier Transform (FFT): NlogN

Summary of Fourier Transform

$s(t)$ transforms (continuous-time)

	Continuous frequency	Discrete frequencies
Transform	$S(f) \triangleq \int_{-\infty}^{\infty} s(t)\cdot e^{-i2\pi ft}\,dt$	$\overbrace{\frac{1}{P}\cdot S\left(\frac{k}{P}\right)}^{S[k]} \triangleq \frac{1}{P}\int_{-\infty}^{\infty} s(t)\cdot e^{-i2\pi\frac{k}{P}t}\,dt \equiv \frac{1}{P}\int_P s_P(t)\cdot e^{-i2\pi\frac{k}{P}t}\,dt$
Inverse	$s(t) = \int_{-\infty}^{\infty} S(f)\cdot e^{i2\pi ft}\,df$	$\underbrace{s_P(t) = \sum_{k=-\infty}^{\infty} S[k]\cdot e^{i2\pi\frac{k}{P}t}}_{\text{Poisson summation formula (Fourier series)}}$

$s(nT)$ transforms (discrete-time)

	Continuous frequency	Discrete frequencies
Transform	$\underbrace{\frac{1}{T}S_{\frac{1}{T}}(f) \triangleq \sum_{n=-\infty}^{\infty} s(nT)\cdot e^{-i2\pi fnT}}_{\text{Poisson summation formula (DTFT)}}$	$\overbrace{\frac{1}{T}S_{\frac{1}{T}}\left(\frac{k}{NT}\right)}^{S[k]} \triangleq \sum_{n=-\infty}^{\infty} s(nT)\cdot e^{-i2\pi\frac{kn}{N}}$ $\equiv \underbrace{\sum_{n} s_P(nT)\cdot e^{-i2\pi\frac{kn}{N}}}_{\text{DFT}}$
Inverse	$s(nT) = T\int_{\frac{1}{T}} \frac{1}{T}S_{\frac{1}{T}}(f)\cdot e^{i2\pi fnT}\,df$ $\sum_{n=-\infty}^{\infty} s(nT)\cdot\delta(t-nT) = \underbrace{\int_{-\infty}^{\infty}\frac{1}{T}S_{\frac{1}{T}}(f)\cdot e^{i2\pi ft}\,df}_{\text{inverse Fourier transform}}$	$s_P(nT) = \overbrace{\frac{1}{N}\sum_{k} S[k]\cdot e^{i2\pi\frac{kn}{N}}}^{\text{inverse DFT}}$ $= \frac{1}{P}\sum_{k} S_{\frac{1}{T}}\left(\frac{k}{P}\right)\cdot e^{i2\pi\frac{kn}{N}}$

2-D DFT

变换对公式：1D—>2D推广

$$F(u,v)=\frac{1}{\sqrt{MN}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\exp[-j2\pi(\frac{ux}{M}+\frac{vy}{N})]$$

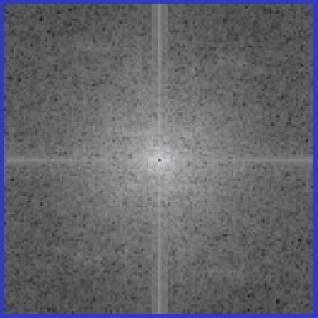
$$f(x,y)=\frac{1}{\sqrt{MN}}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)\exp[j2\pi(\frac{ux}{M}+\frac{vy}{N})]$$

频谱（幅度） $|F(u,v)|=\left[R^2(u,v)+I^2(u,v)\right]^{1/2}$

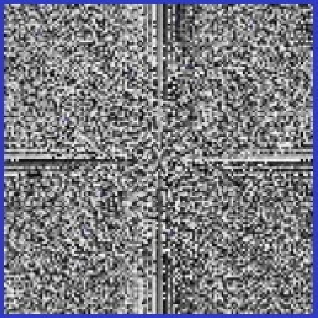
相位角 $\phi(u,v)=\arctan[I(u,v)/R(u,v)]$

功率谱 $P(u,v)=|F(u,v)|^2=R^2(u,v)+I^2(u,v)$

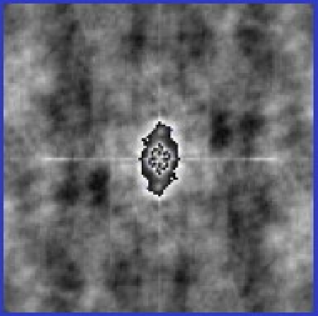
原图像



幅度谱



相位谱



由幅度谱重建

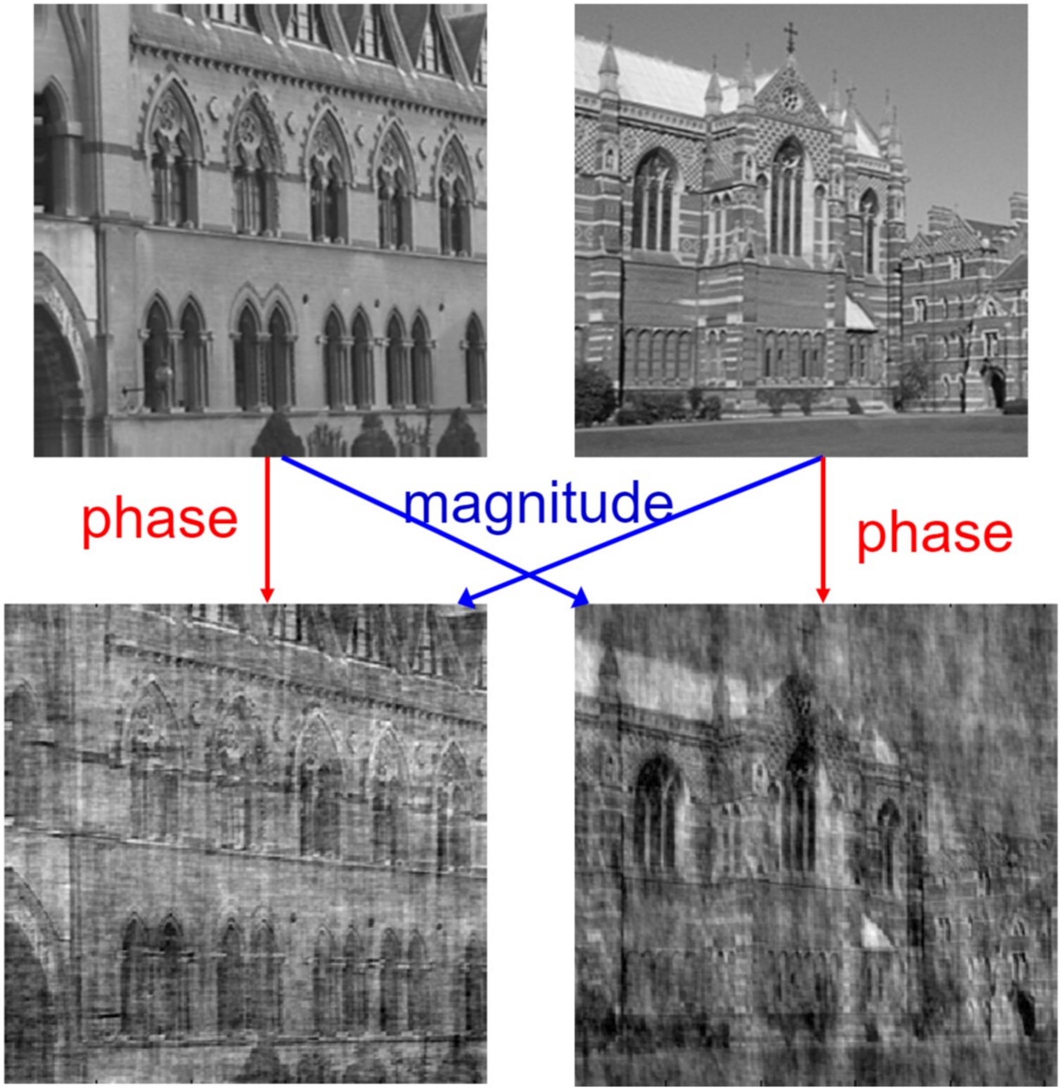
相位谱为0

由相位谱重建

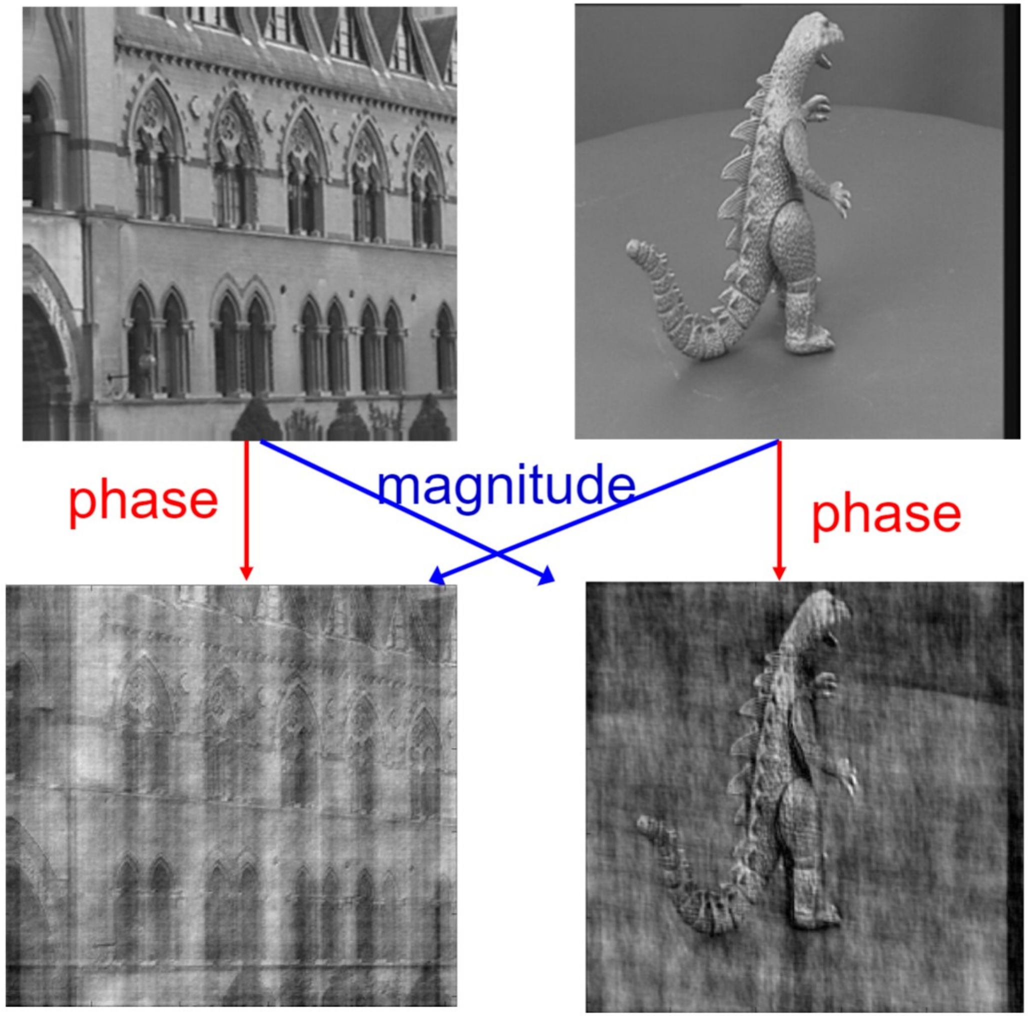
幅度谱为常数

结论：相位谱可能具有更重要的应用

The importance of phase



A second example



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!

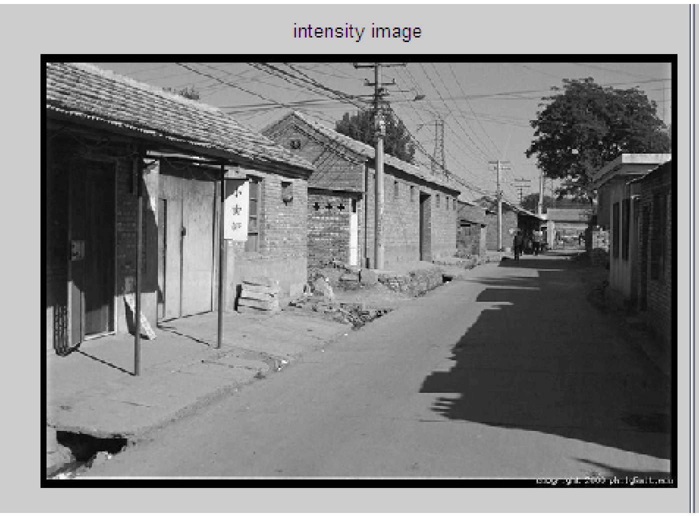
Properties of Fourier Transforms

- 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

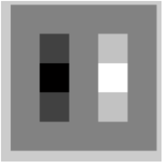
See Szeliski Book (3.4)

Filtering in spatial domain

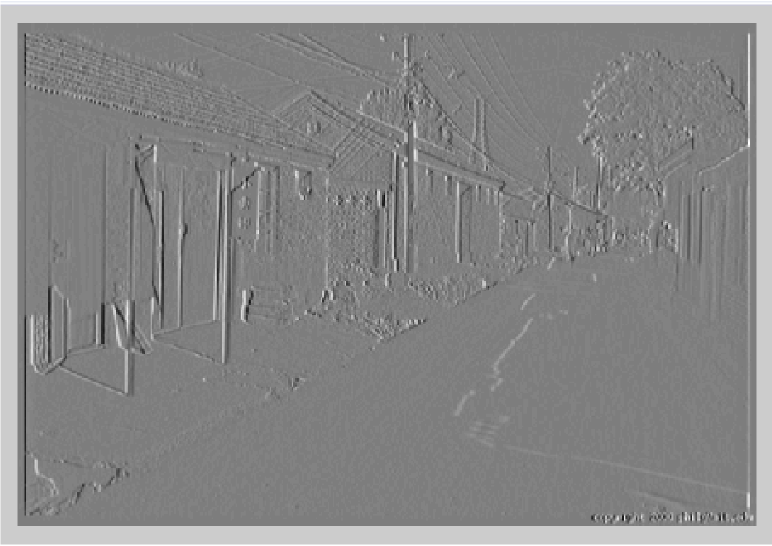
1	0	-1
2	0	-2
1	0	-1



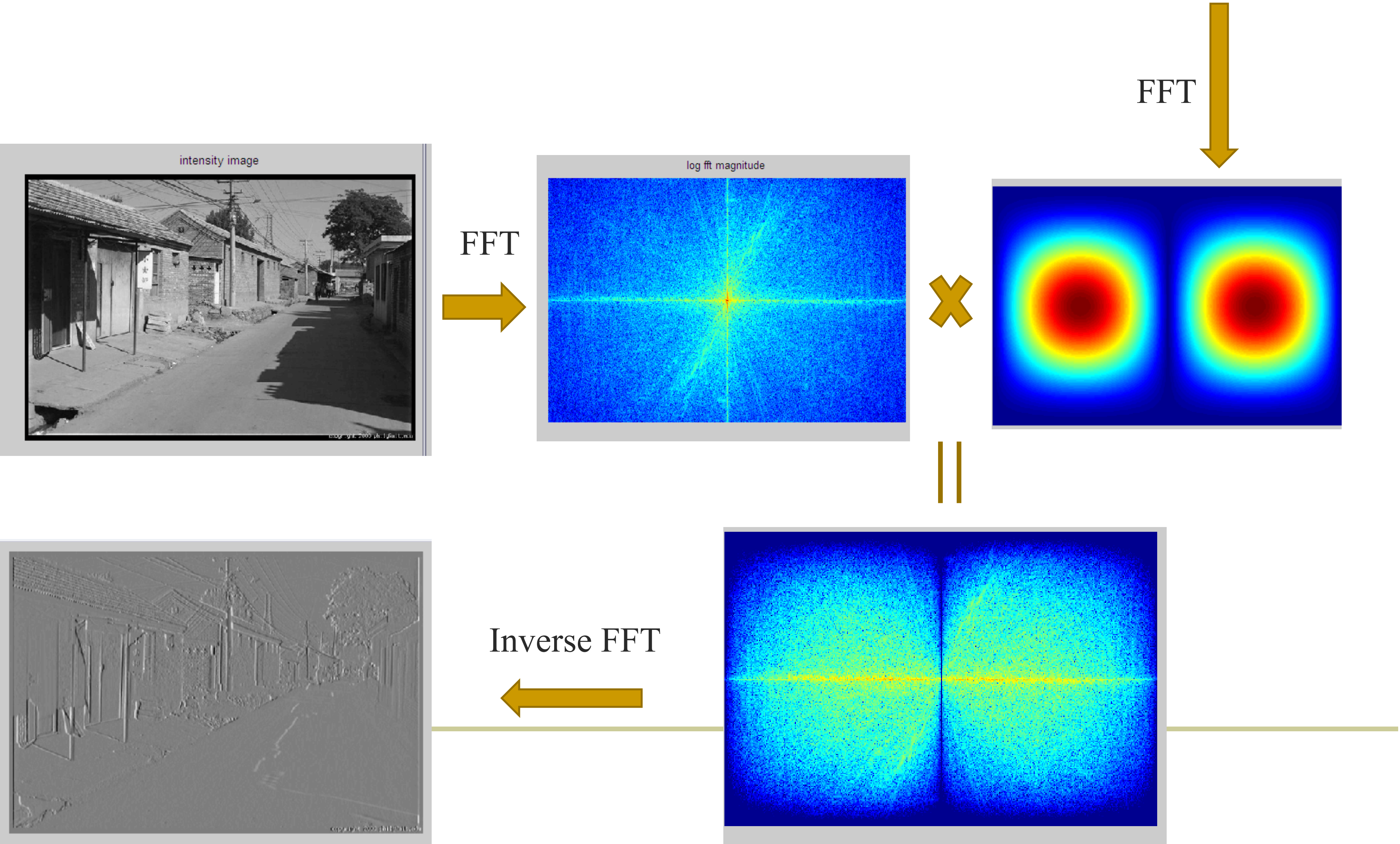
$*$



$=$



Filtering in frequency domain



FFT 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?

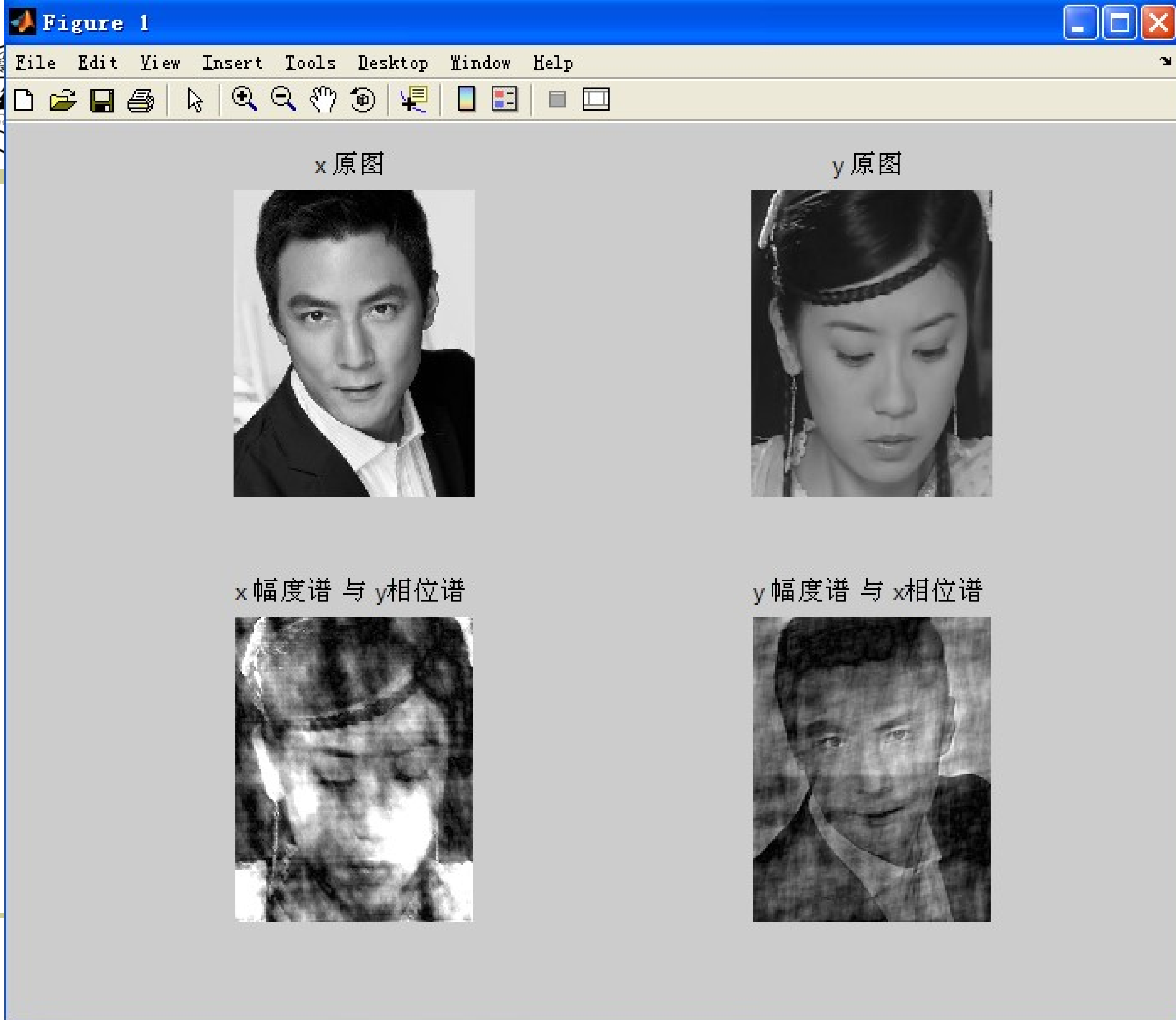
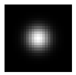


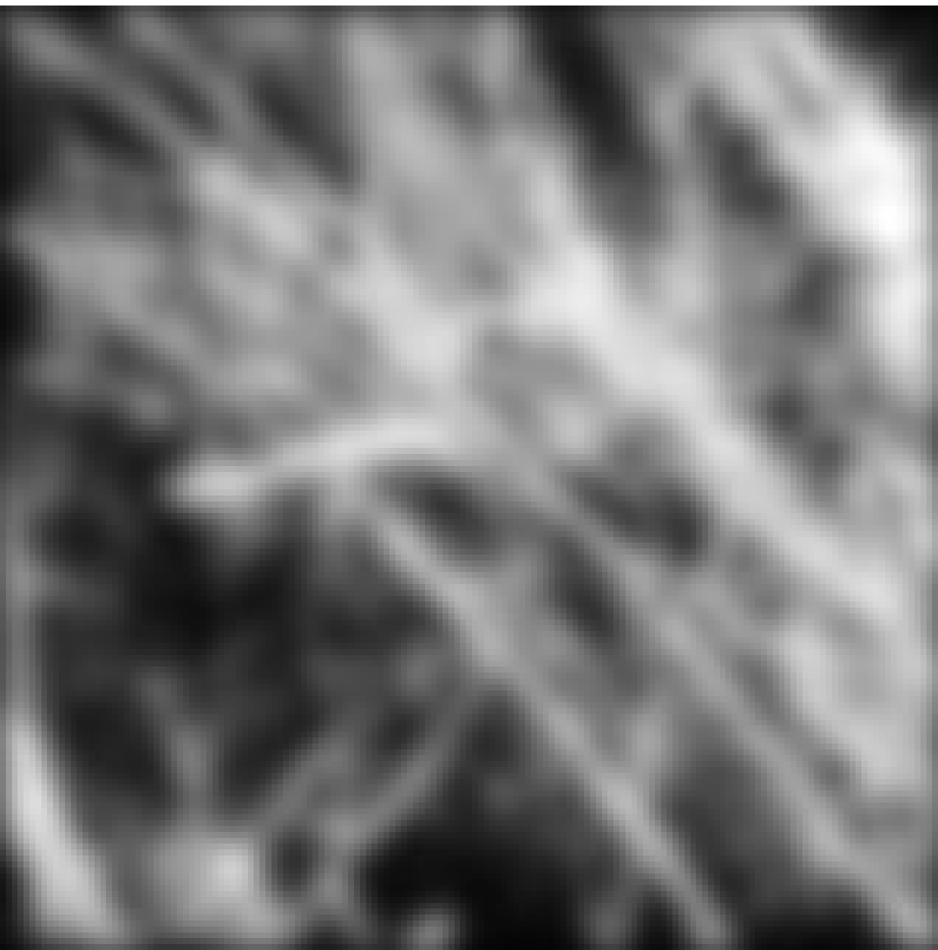
Figure 1
File Edit View Insert Tools Desktop Window Help
x 原图
y 原图
x 幅度谱 与 y相位谱
y 幅度谱 与 x相位谱

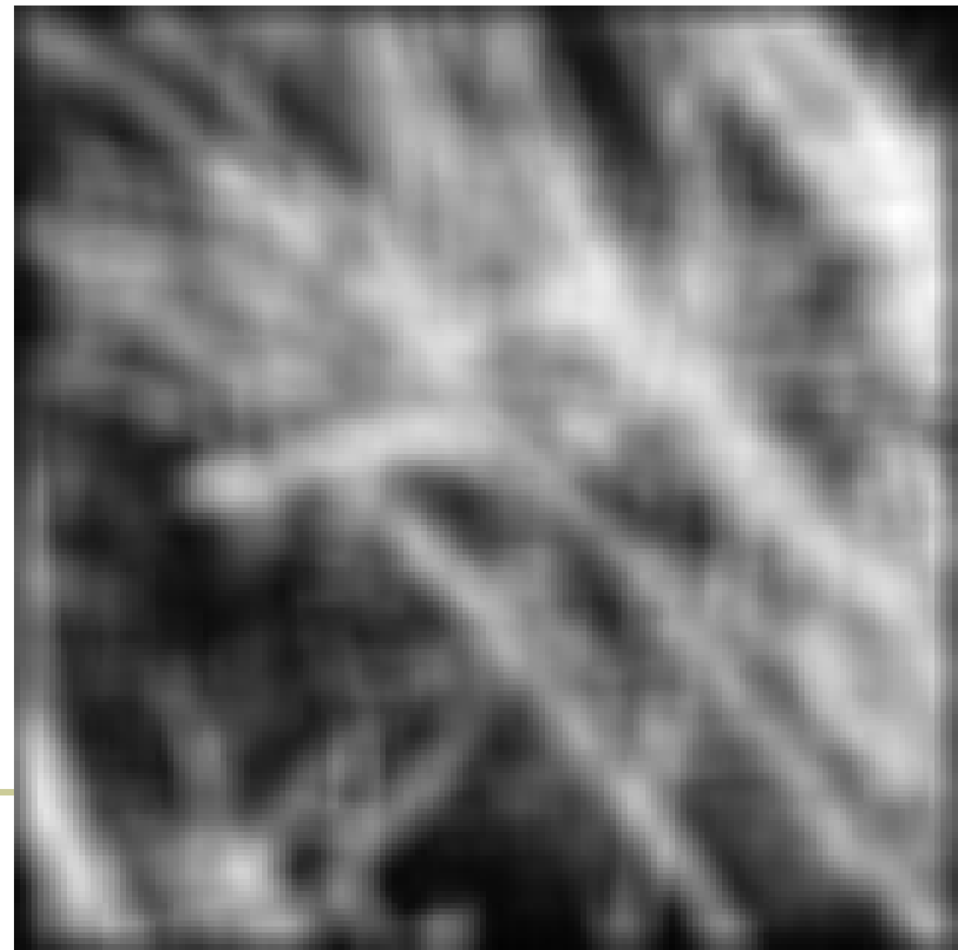
Filtering

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

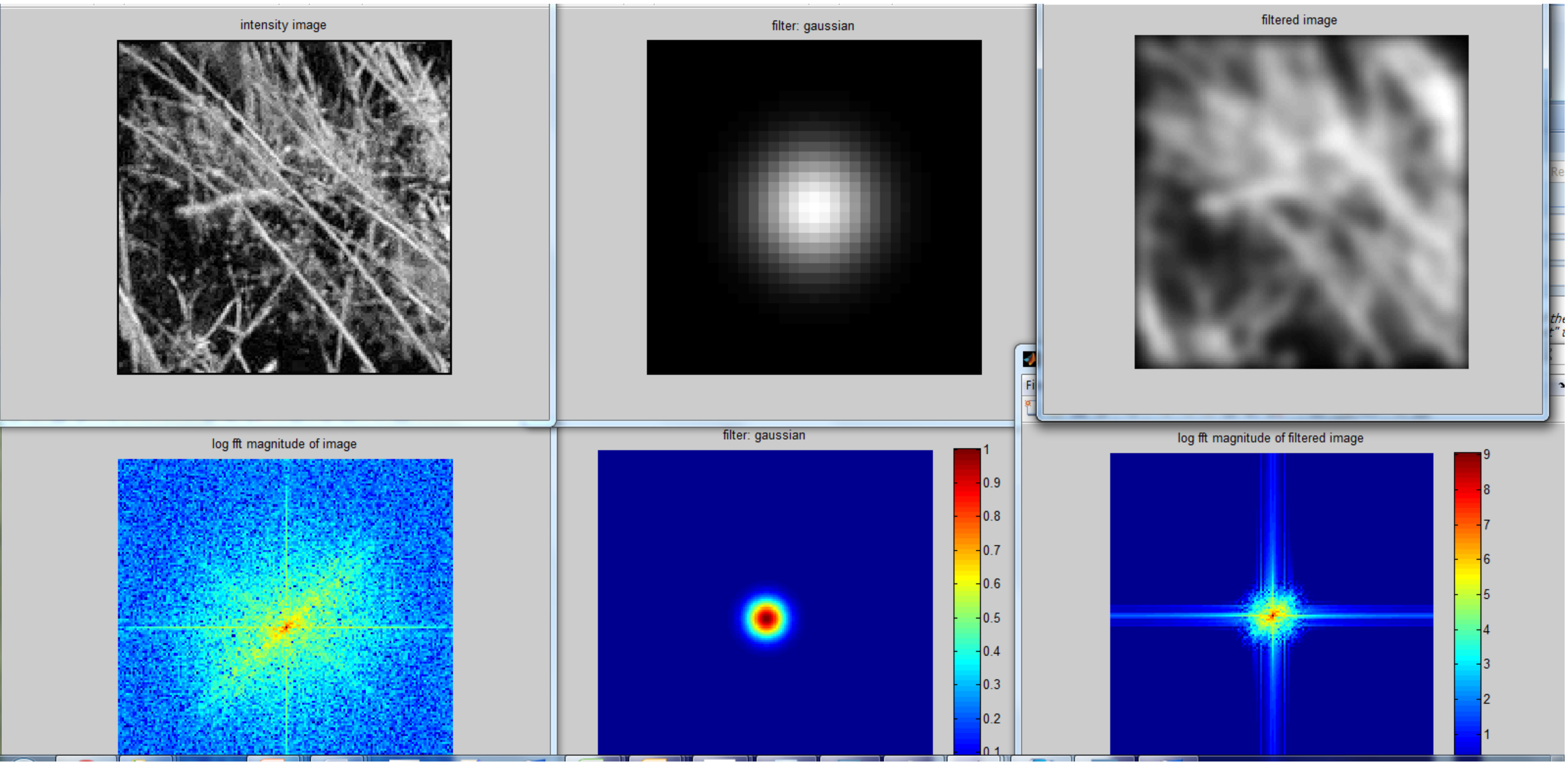
Gaussian 

Box filter 

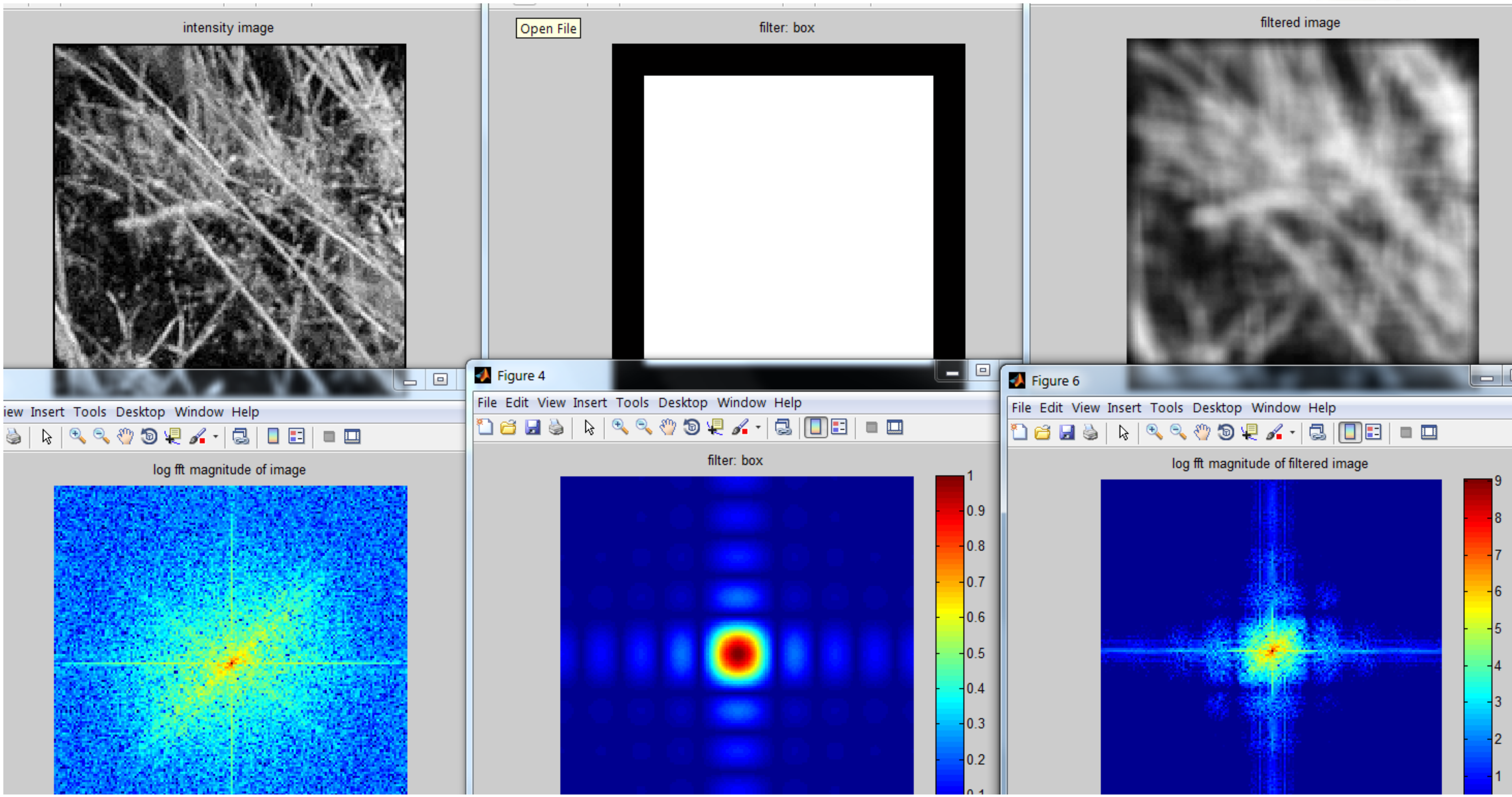




Gaussian Filter



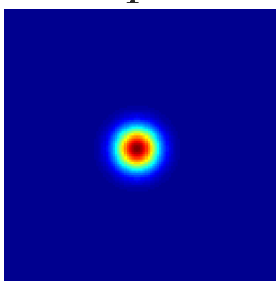
Box Filter


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

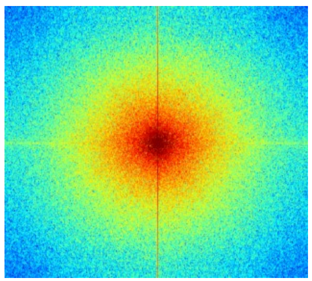
Question

Match the spatial domain image to the Fourier magnitude image

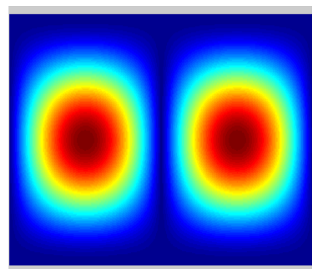
1



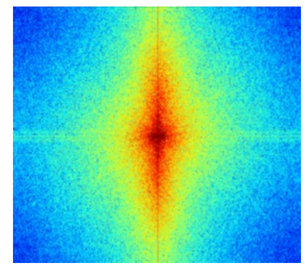
2



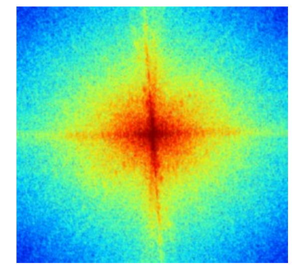
3



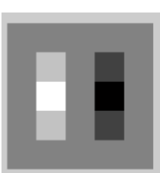
4



5

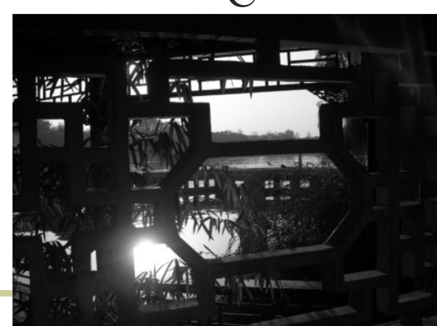


A

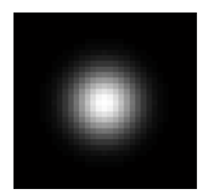


B

C



D



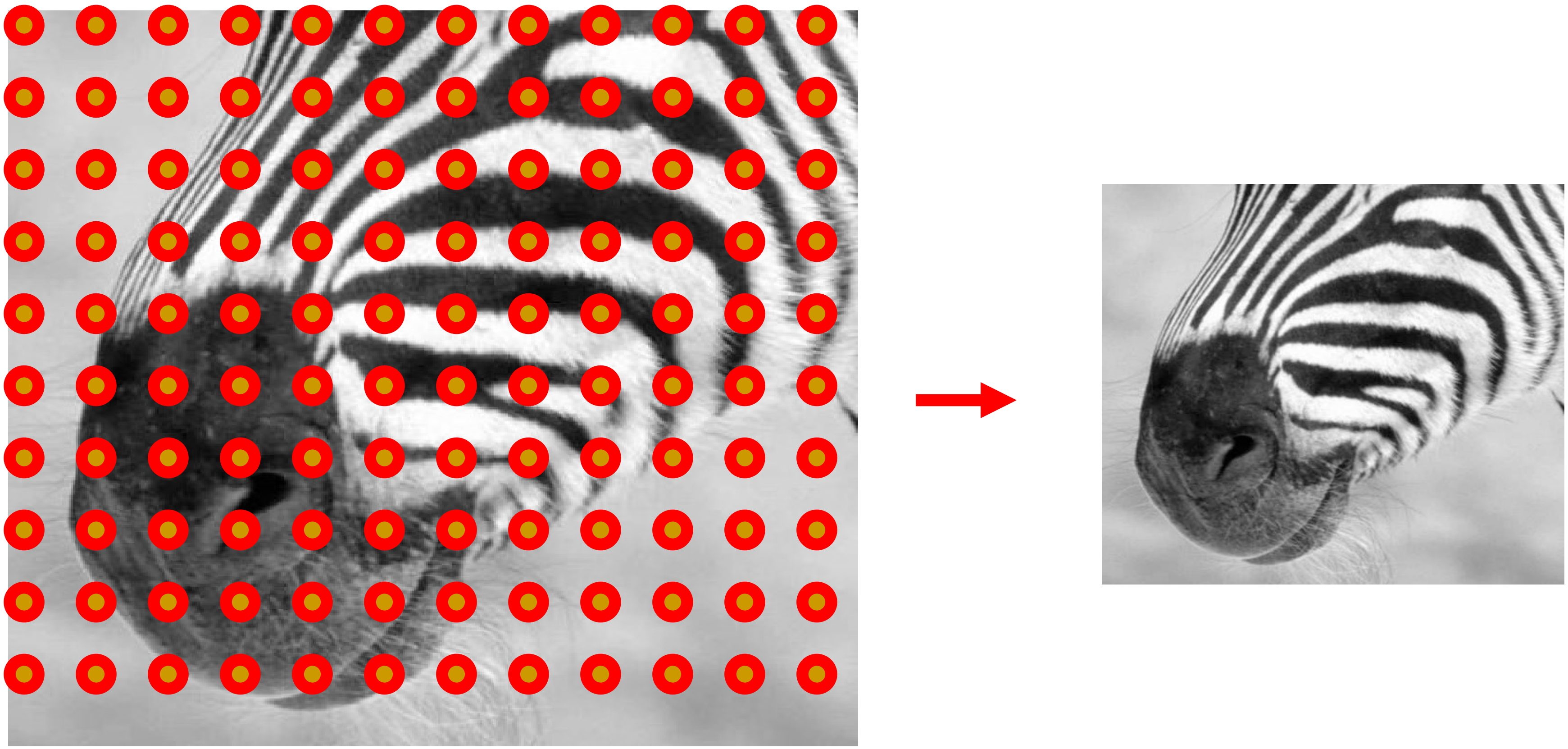
E

Sampling

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

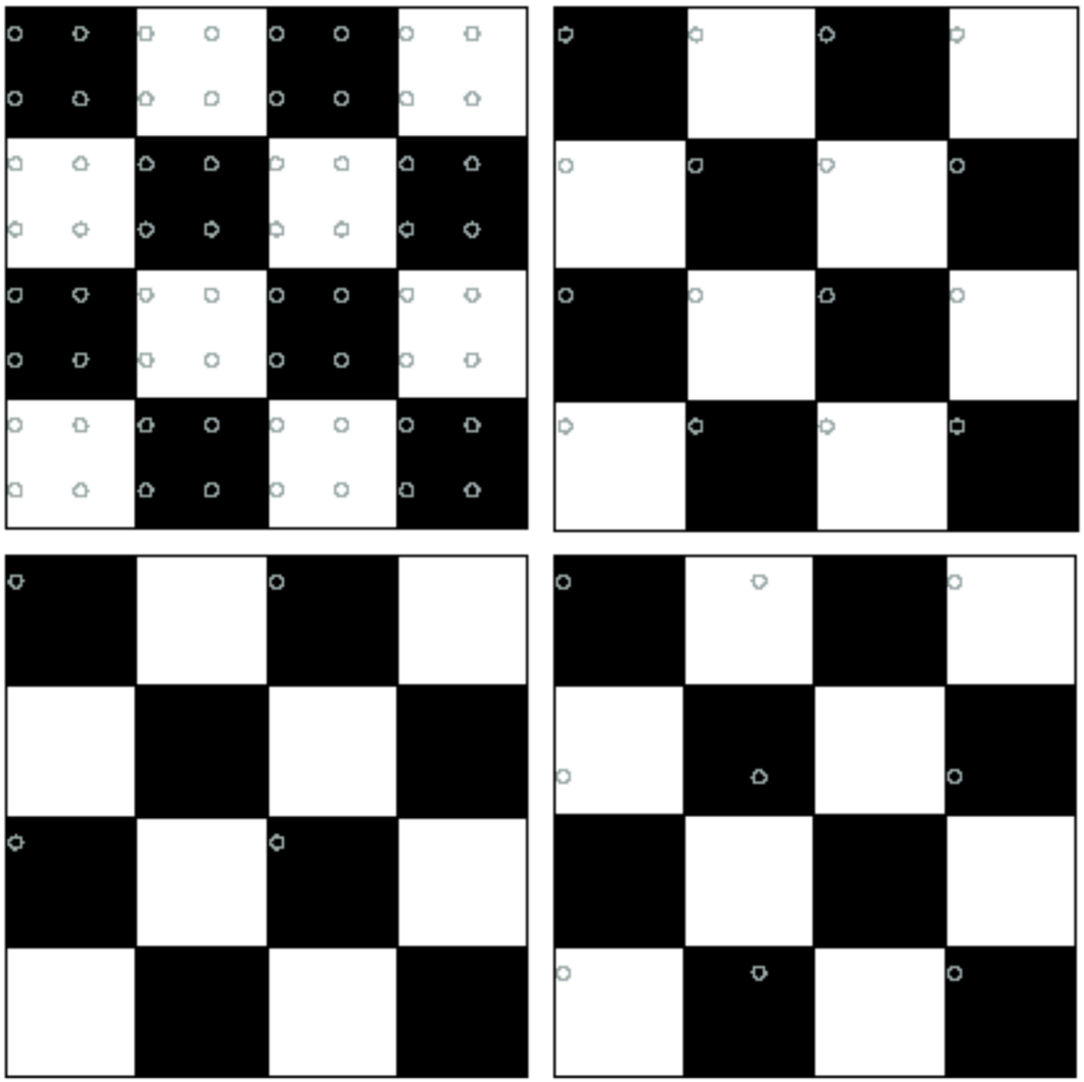
Image: http://www.flickr.com/photos/igorms/136916757/

Subsampling by a factor of 2



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

How should we go about sampling



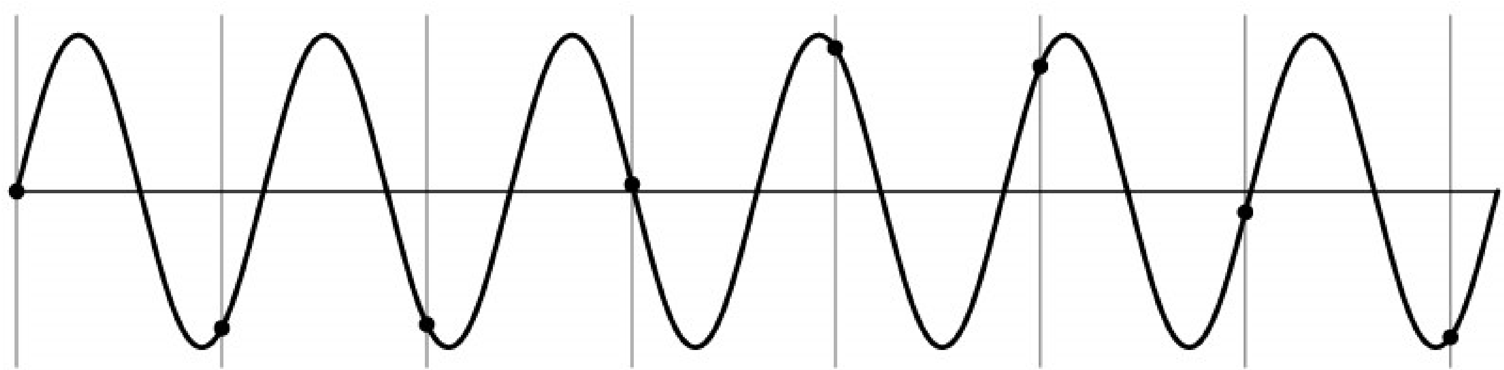
Let's resample the checkerboard by taking one sample at each circle.

In the top left board, the new representation is reasonable. Top right also yields a reasonable representation.

Bottom left is all black (dubious) and bottom right has checks that are too big.

How should we go about sampling

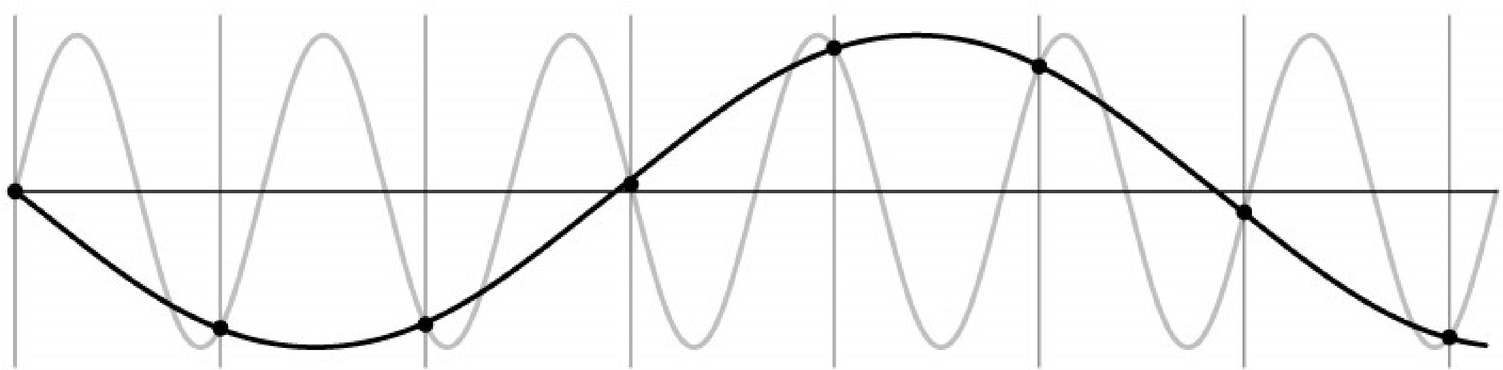
- 1D example (sinewave):



Source: S. Marschner

How should we go about sampling

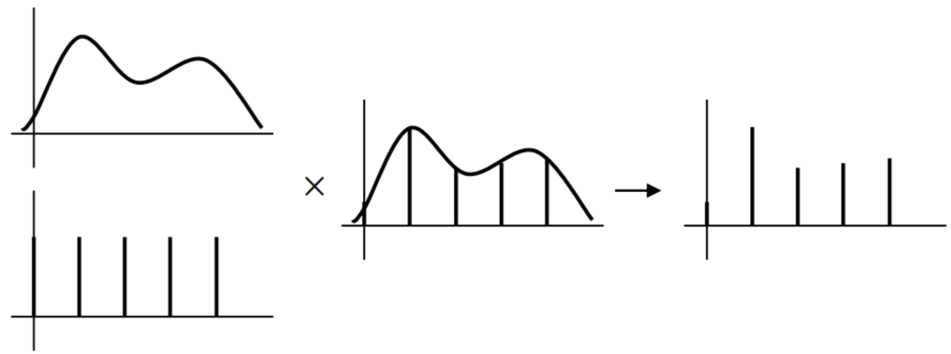
- 1D example (sinewave):



Source: S. Marschner

Fourier Interpretation: Sampling

- Sampling in the spatial domain is like multiplying with a spike function.

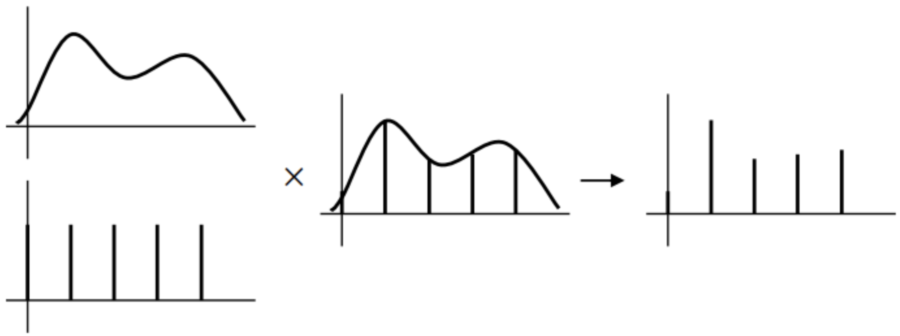


- Sampling in the frequency domain is like...

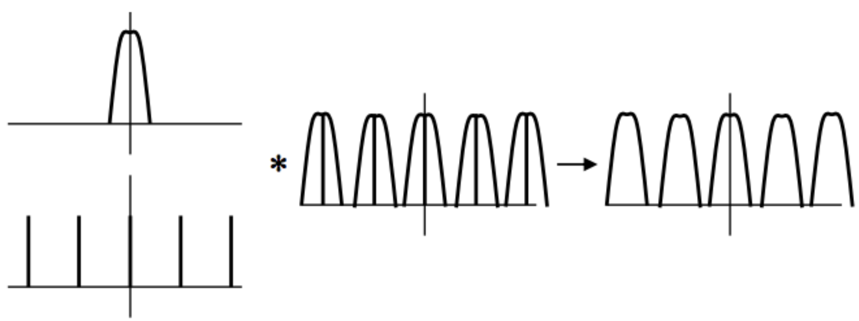
?

Fourier Interpretation: Sampling

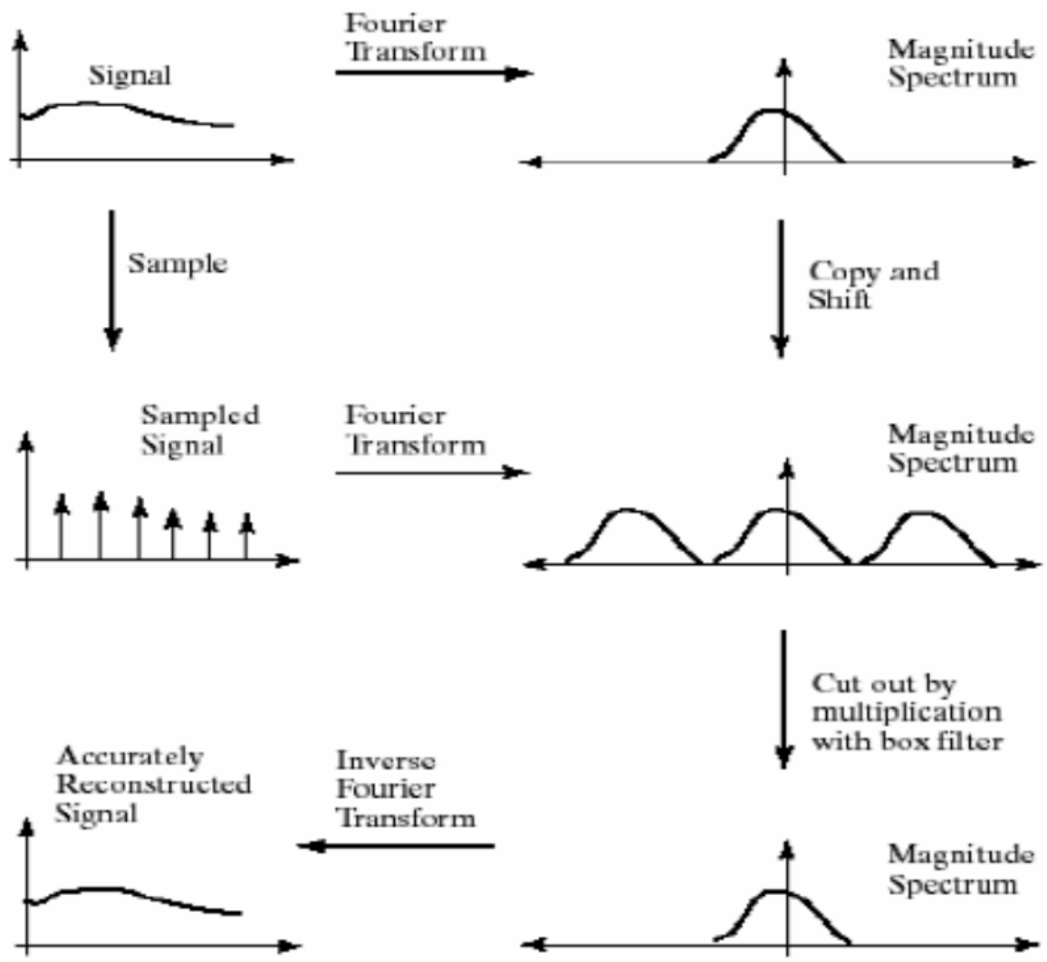
- Sampling in the spatial domain is like multiplying with a spike function.



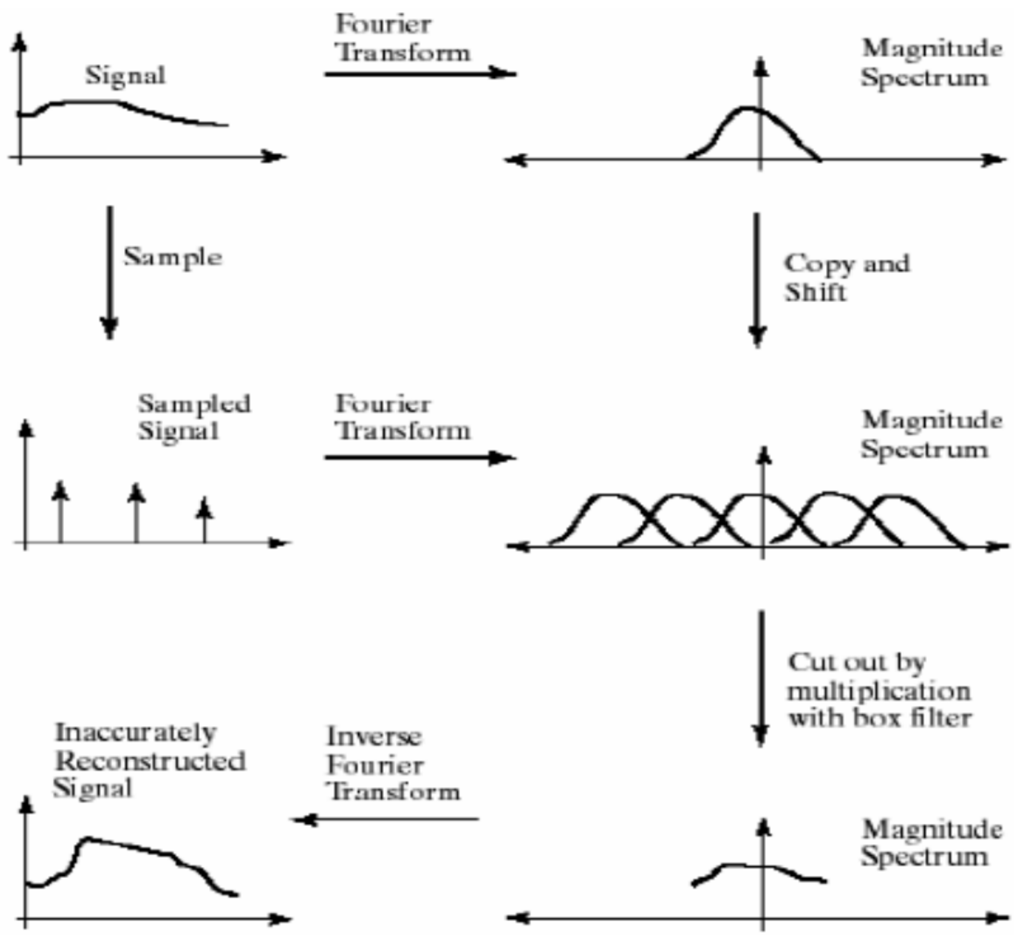
- Sampling in the frequency domain is like convolving with a spike function.



Fourier Interpretation: Sampling



Fourier Interpretation: Sampling



Aliasing problem

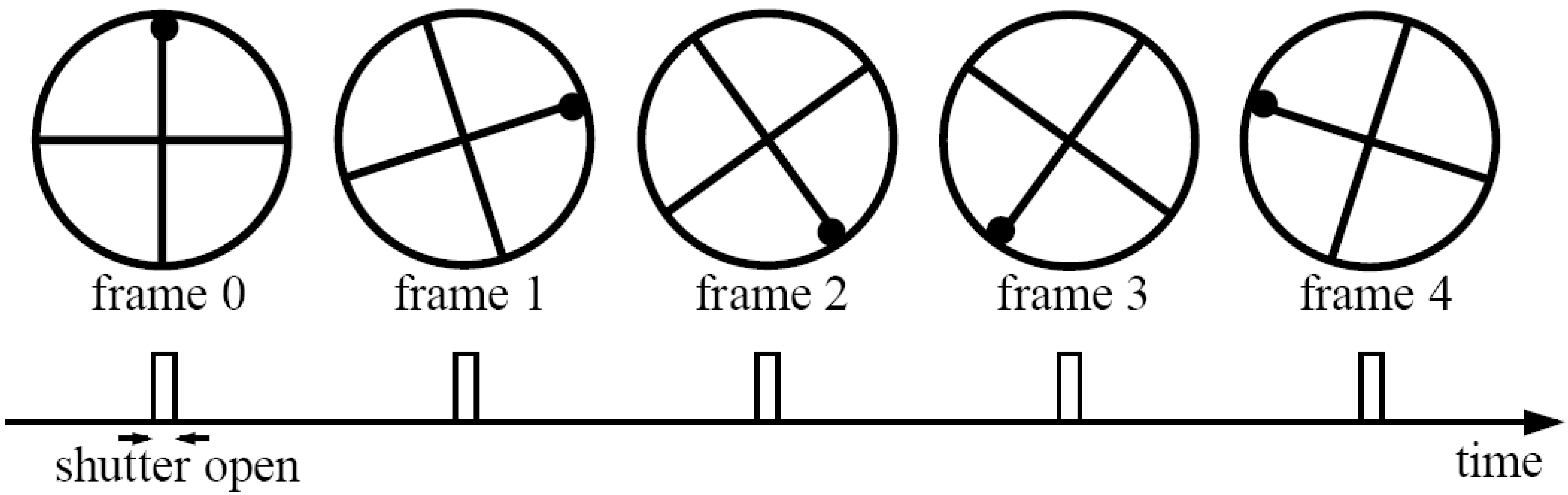
- Sub-sampling may be dangerous….
- Characteristic errors may appear:
  - "Wagon wheels rolling the wrong way in movies"
  - "Checkerboards disintegrate in ray tracing"
  - "Striped shirts look funny on color television"

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.

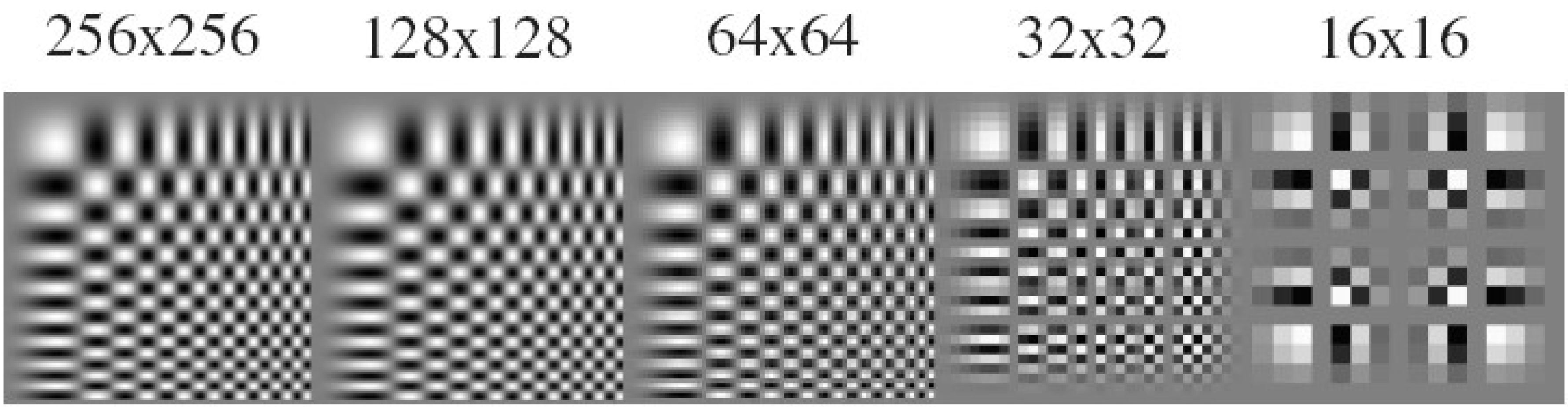
If camera shutter is only open for a fraction of a frame time (frame time = 1/30 sec. for video, 1/24 sec. for film):



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

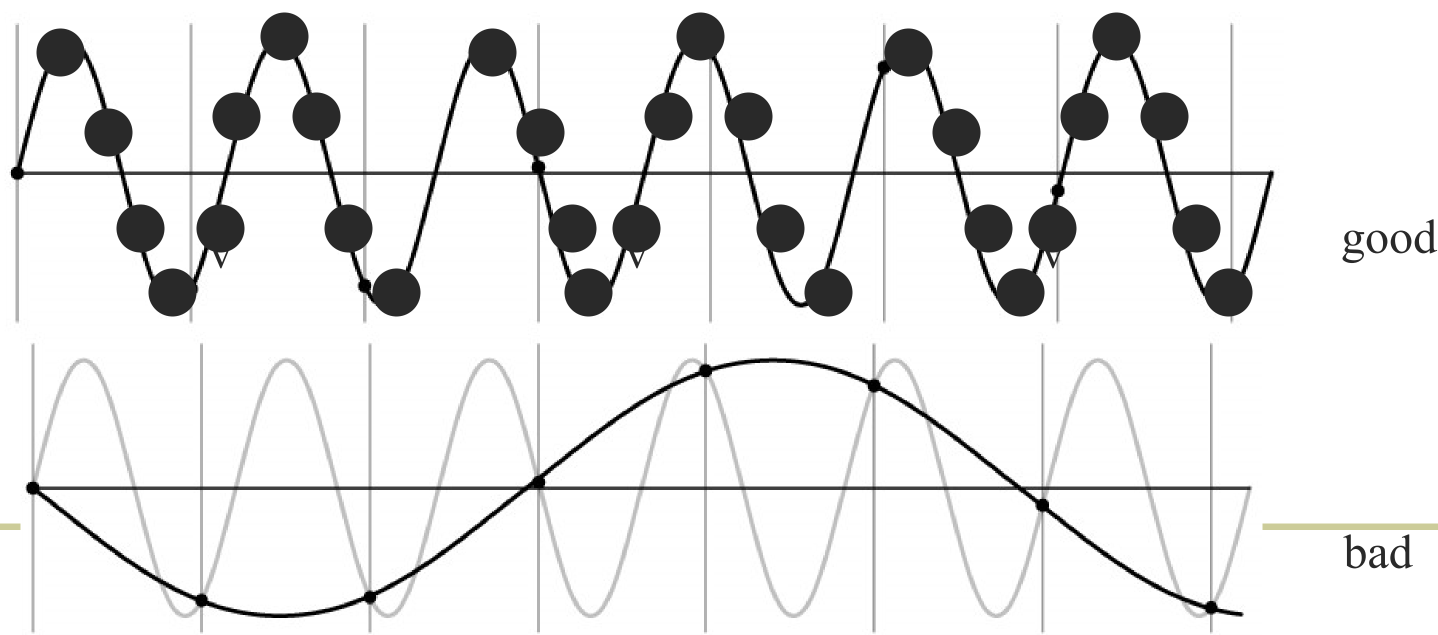
Slide by Steve Seitz

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

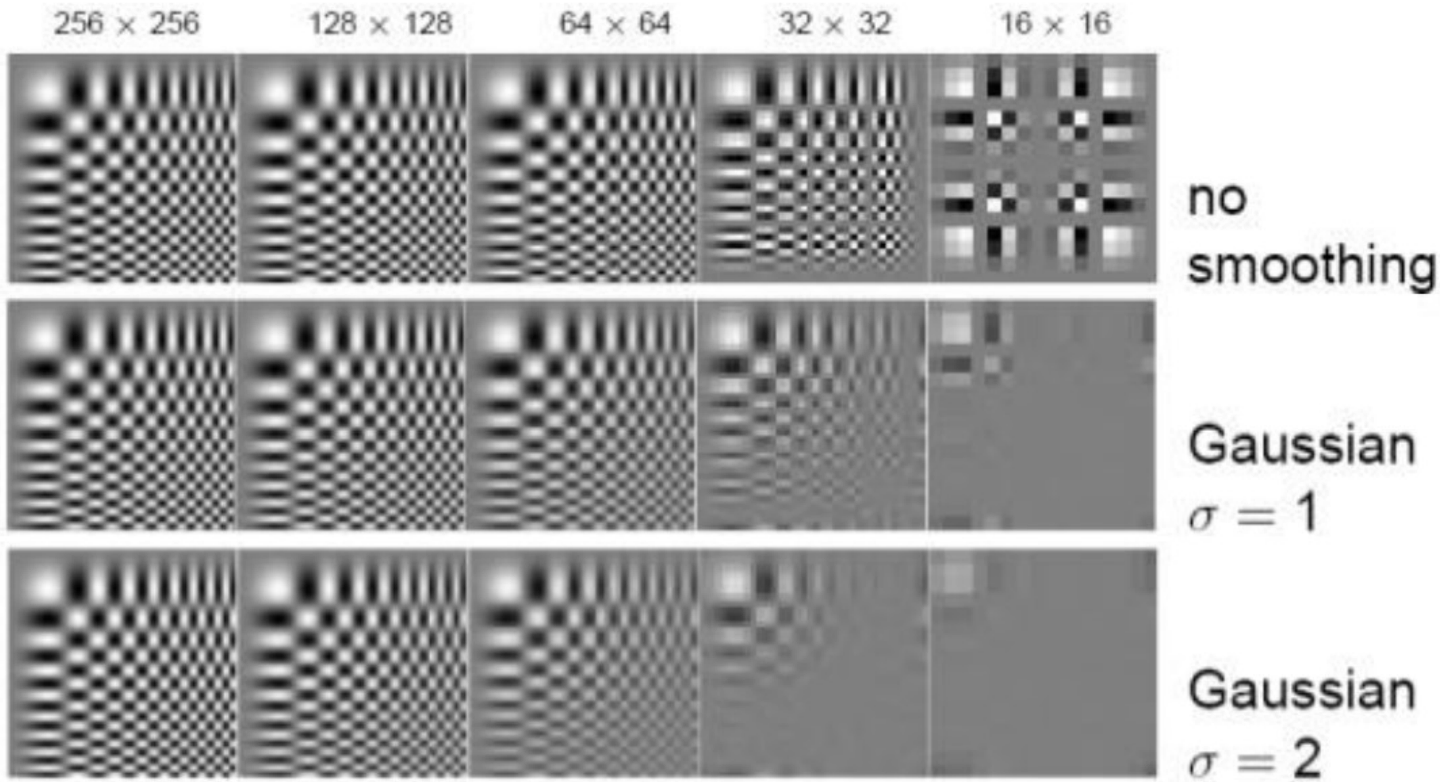
Algorithm for downsampling by factor of 2

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);

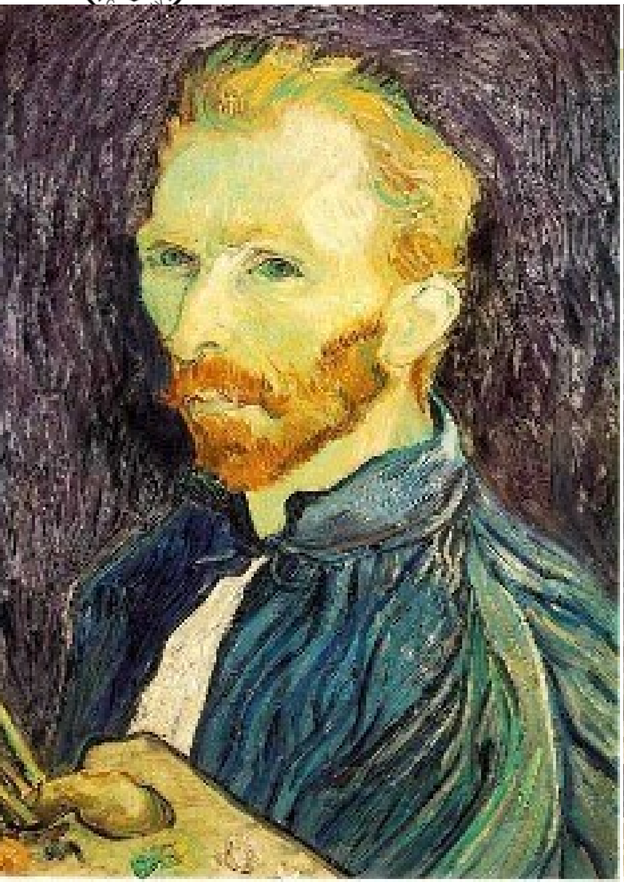
Anti-aliasing



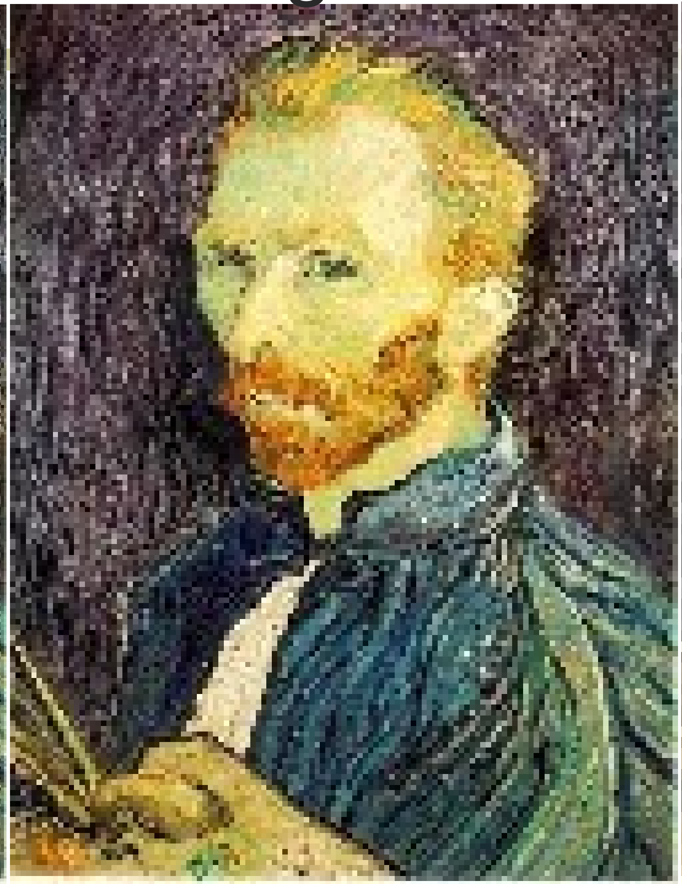
Note: We cannot recover the high frequencies, but we can avoid artifacts by smoothing before resampling.

Forsyth and Ponce 2002

Subsampling without pre-filtering



1/2

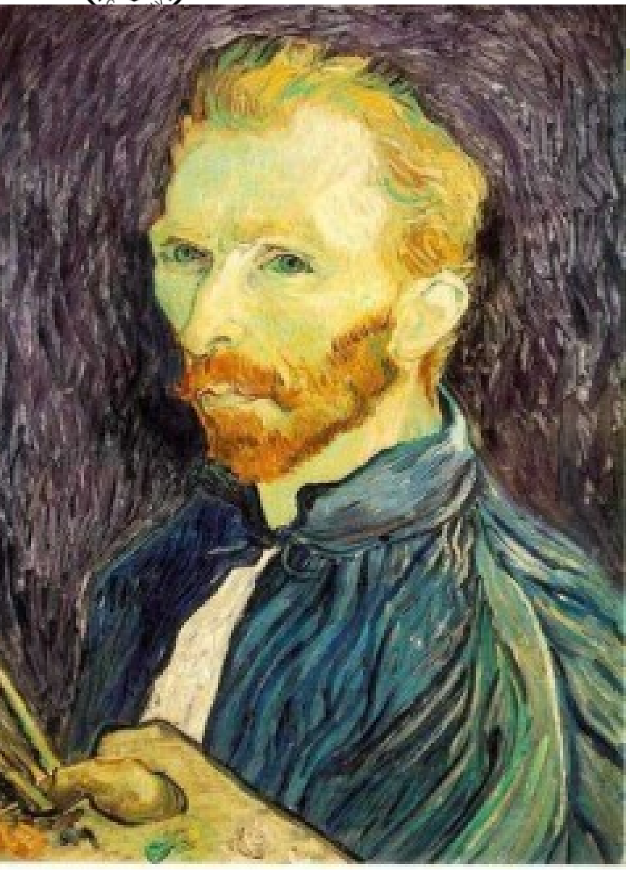


1/4 (2x zoom)

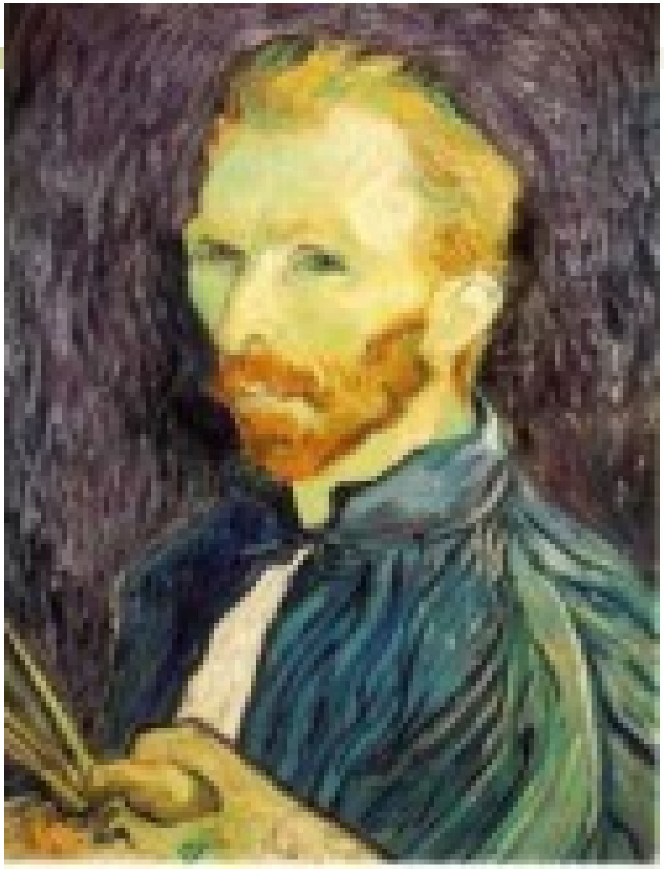
1/8 (4x zoom)

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?

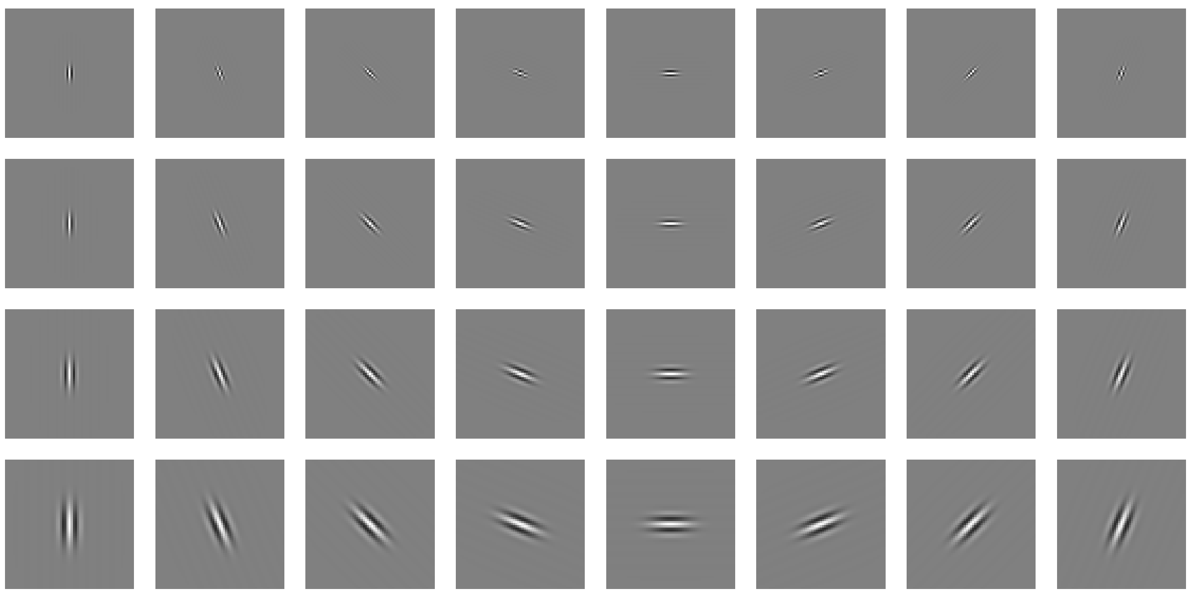
Image: http://www.flickr.com/photos/igorms/136916757/

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

?

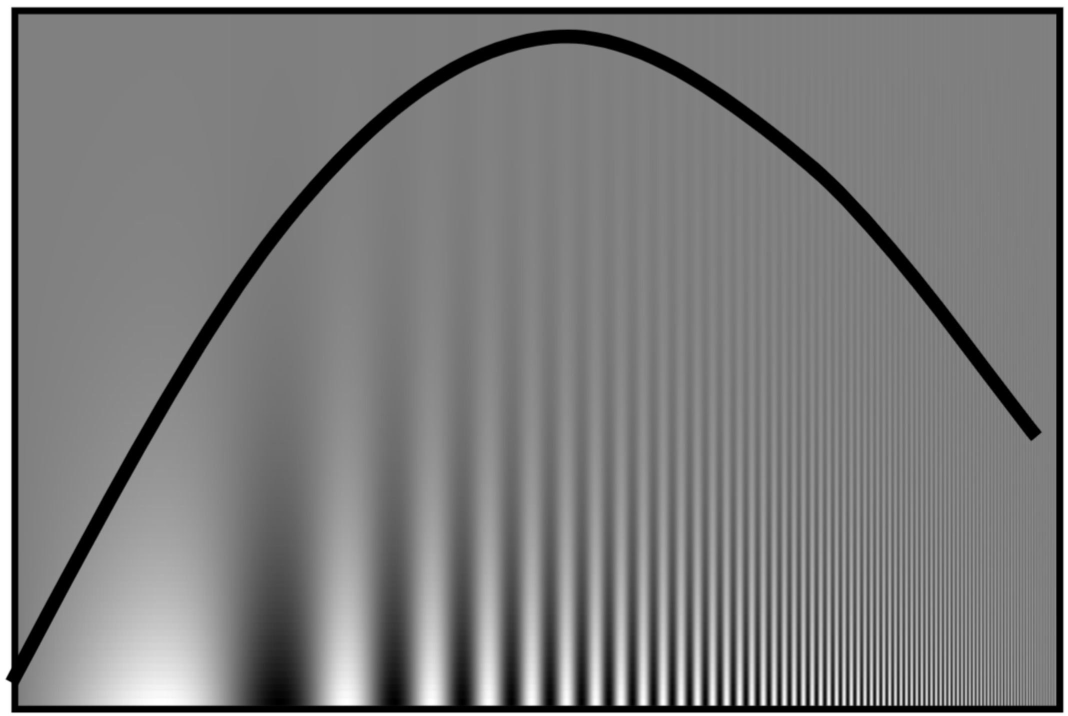
Clues from Human Perception

- Early processing in humans filters for various orientations and scales of frequency
- Perceptual cues in the mid-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

Frequency Domain and Perception

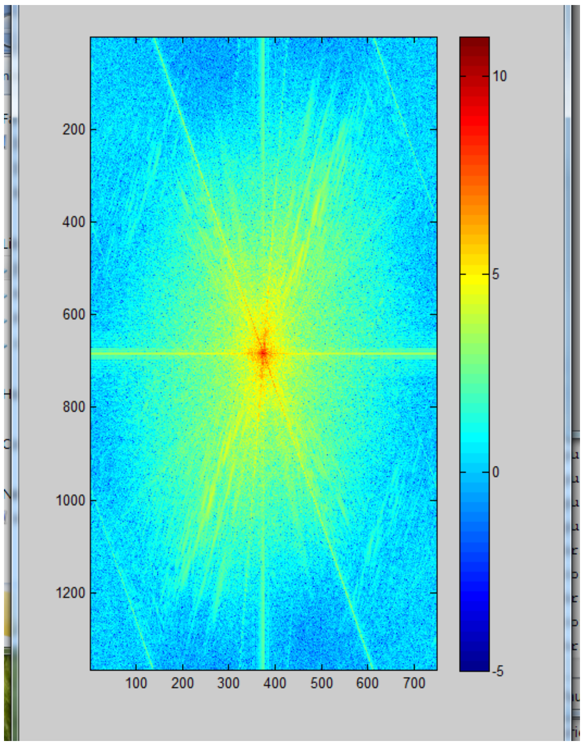


Campbell-Robson contrast sensitivity curve

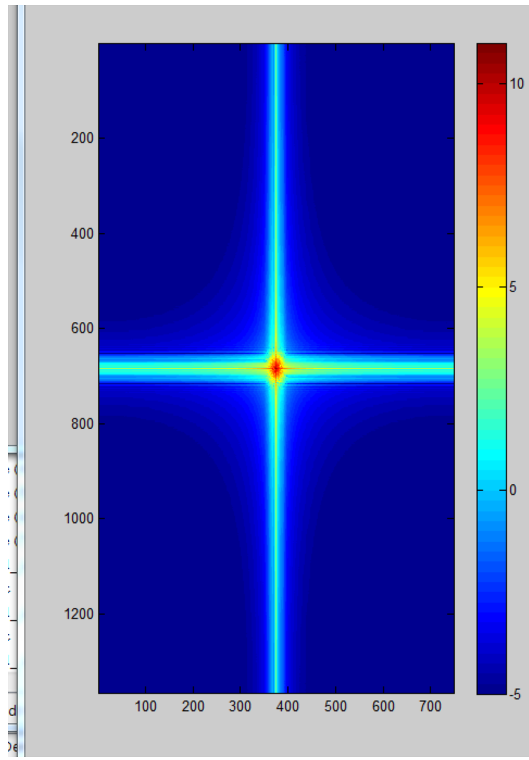
slide: A. Efros

Hybrid Image in FFT

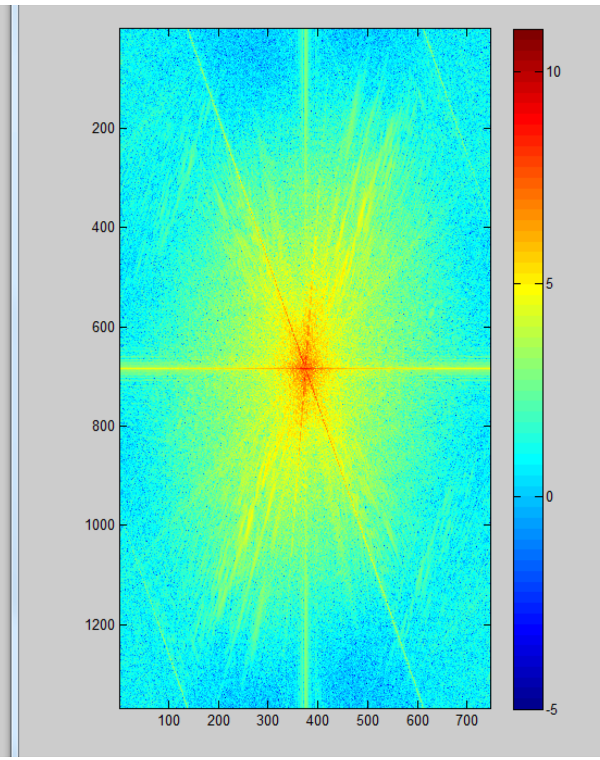
Hybrid Image



Low-passed Image

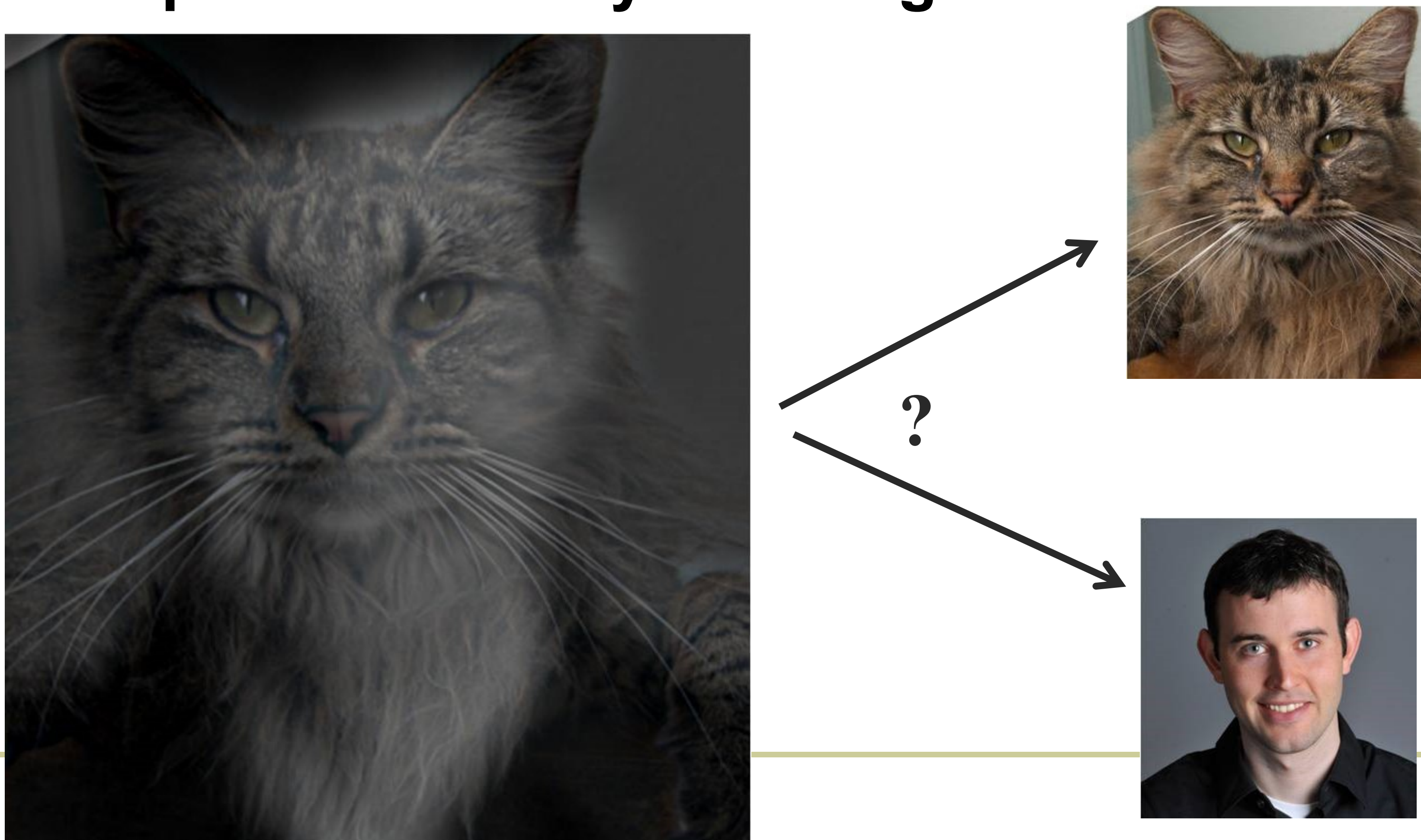


High-passed Image

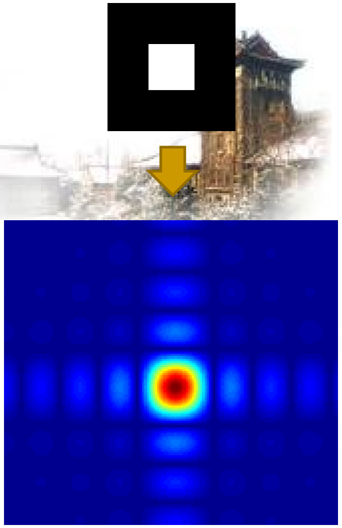


Perception

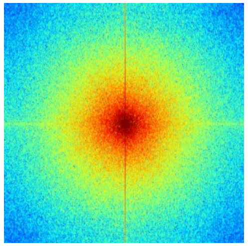
Why do we get different, distance-dependent interpretations of hybrid images?



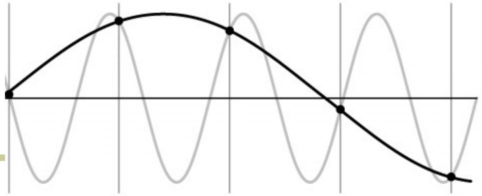
Things to Remember



- 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)



- Remember to low-pass before sampling