CSC589 Introduction to Computer Vision
Lecture 9

Sampling, Gaussian Pyramid

Bei Xiao

What we have learned so far

- Image contrast, histograms, and basic manipulation
- Image Convolution
- Image Filters
- Fourier transform
- Filtering in Fourier transform

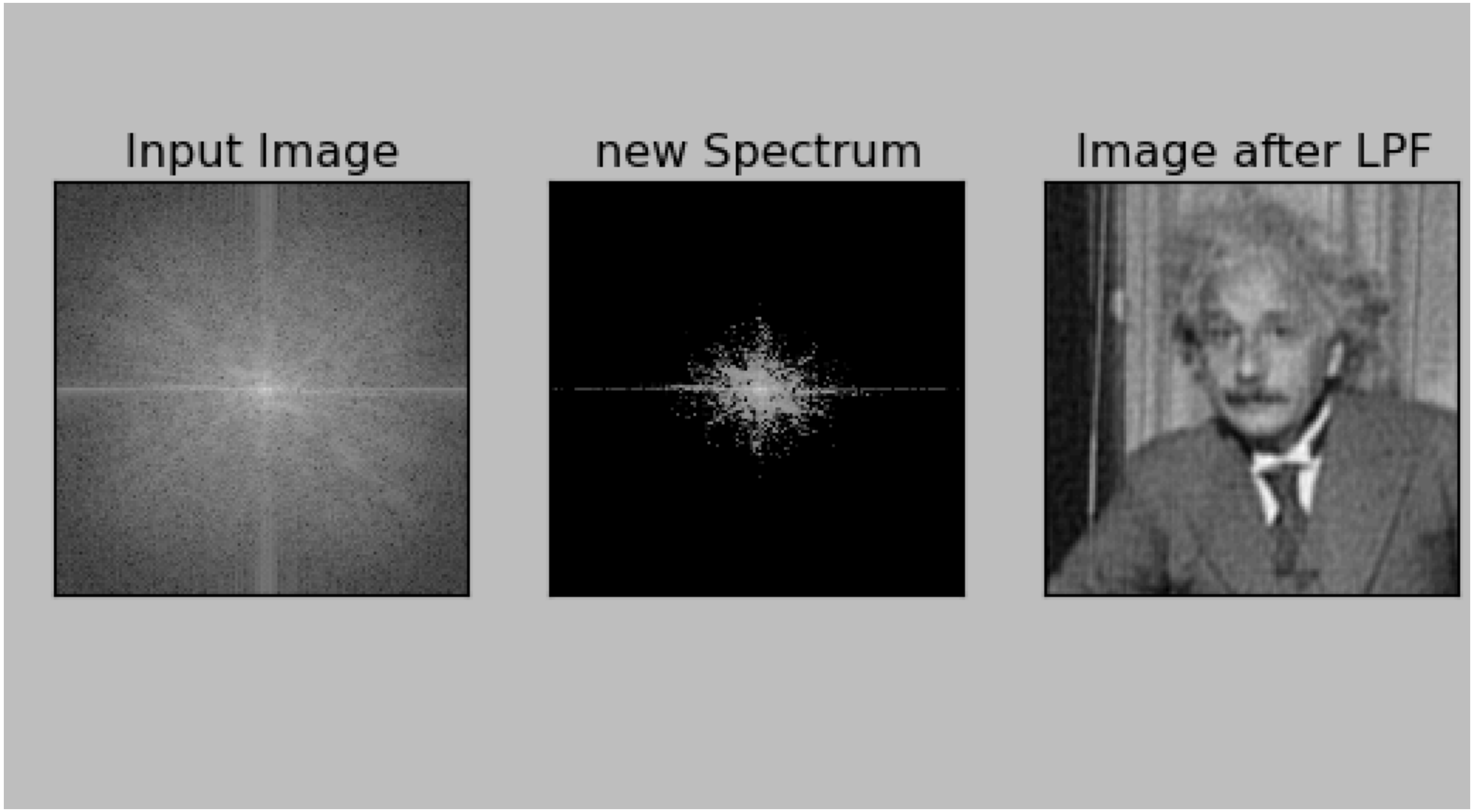
What we want to achieve in this course?

- Basic image processing (extract contours, boundaries, edges).
- Basic understanding of image formation (camera models, projection)
- Basic skills of image synthesizing (textures, panorama images, image hybrid, stereo)
- Basic understanding of image features (HOG, SIFT, optical flow).
- Basic exposure of the state of the art computer vision applications. (multiple object tracking, segmentation, 3D construction, recognition of objects and faces)
- Application of machine learning in CV (Clustering, Bayesian, deep learning intro).
- Basic mathematical concepts behind image processing (Fourier transform, convolution, probability and statistics).
- Brushing up numerical Python skills
- If you have made progress in at least 4 out of the above points, it is a success!

Exercise 1

- Download the FFTAnalysis.py from blackboard.
- Use the einstein.png
- Right now the image removes low frequency (center) of the images.
- Can you modify the code that you are removing the high frequency (low pass) the image? You can do this either by directly modifying the DFT or use a filter.

Exercise 1




```
mask = abs(fshift) < 10000
fshift[mask]= 10
```

Exercise 2

- Download the Cheetha and Zebra images
- DFT both images in Fourier domain and compute magnitude and phase. I have already did this in the helper code PhaseandMagnitude.py
- You can compute phase and magnitude as this:
- magnitude_zebra = 30*np.log(np.abs(fshift))
- phase_zebra = np.angle(fshift)
- Reconstruct the image with Cheetha phase and Zebra magnitude and vice versa. You have to do this yourself!

Inverse Fourier Transform

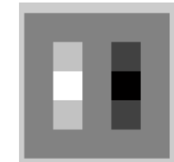
- Forward Fourier:

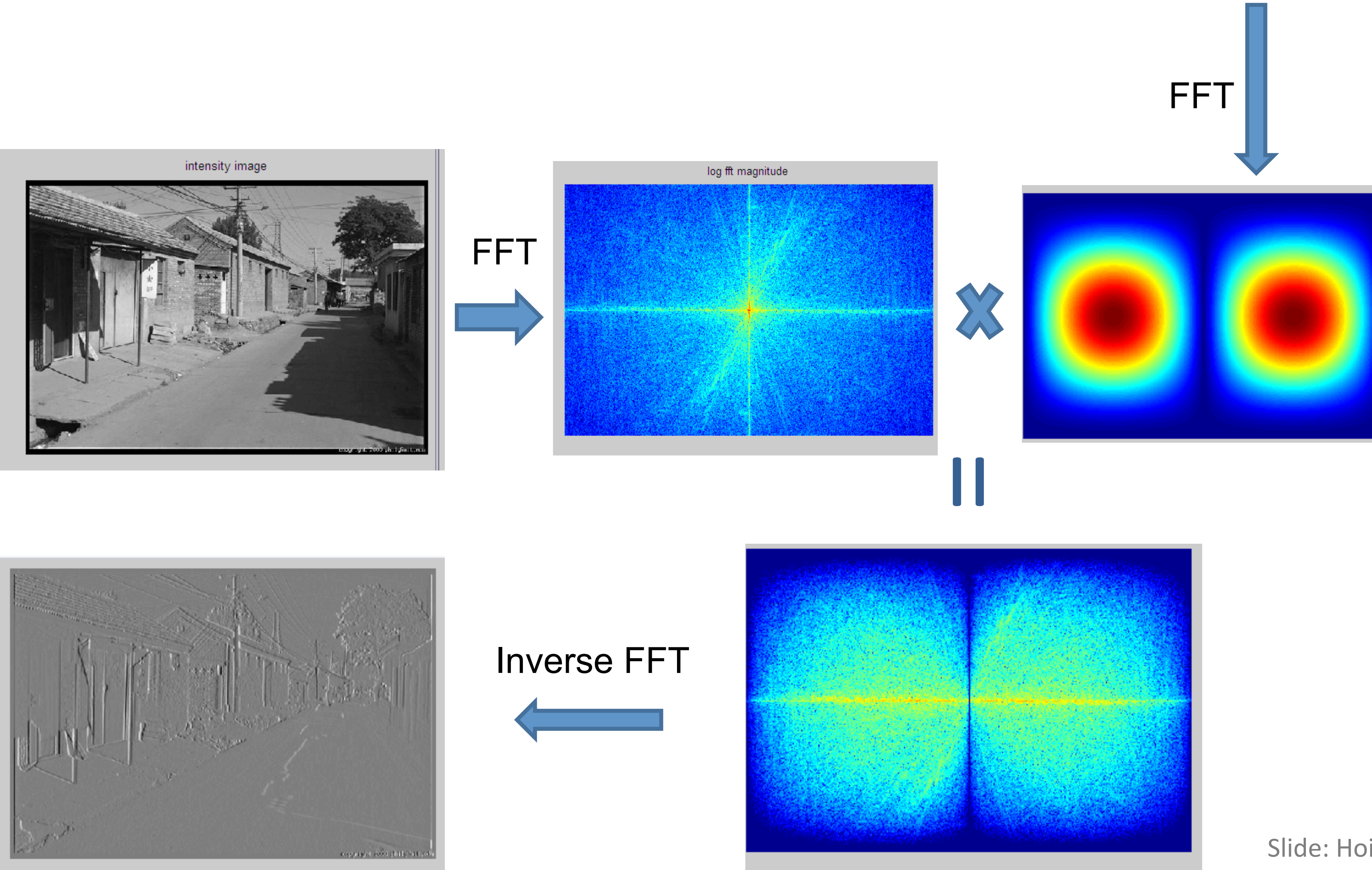
```
img = misc.imread('cheetah.png',flatten=1)
f = np.fft.fft2(img)
fshift = np.fft.fftshift(f)
magnitude_cheetah = np.abs(fshift)
phase_cheetah = np.angle(fshift)
```

- Inverse Fourier:

```
re = magnitude_zebra*np.cos(phase_zebra)
im = magnitude_zebra*np.sin(phase_zebra)
F = re+1j*im
f_ishift = np.fft.ifftshift(F)
img_back = np.fft.ifft2(f_ishift)
img_back = np.abs(img_back)
#img_back= misc.bytescale(img_back)
print img_back.min(), img_back.max()
plt.imshow(np.uint8(img_back), cmap='gray')
```

Filtering in frequency domain

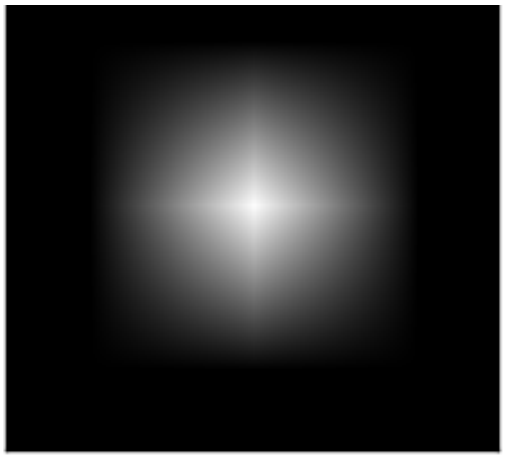




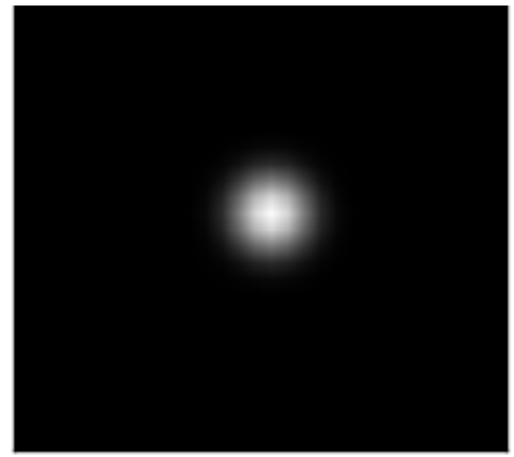
Slide: Hoiem

Filters in Fourier Domain

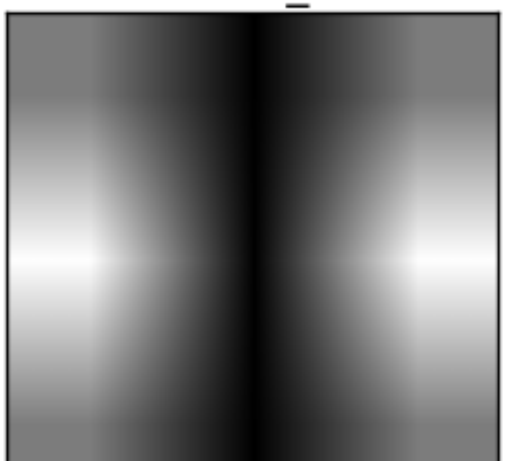
Mean filter



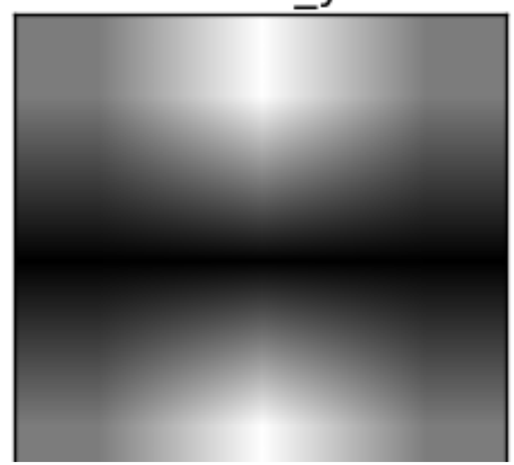
Gaussian Filter



sobel_x



sobel_y

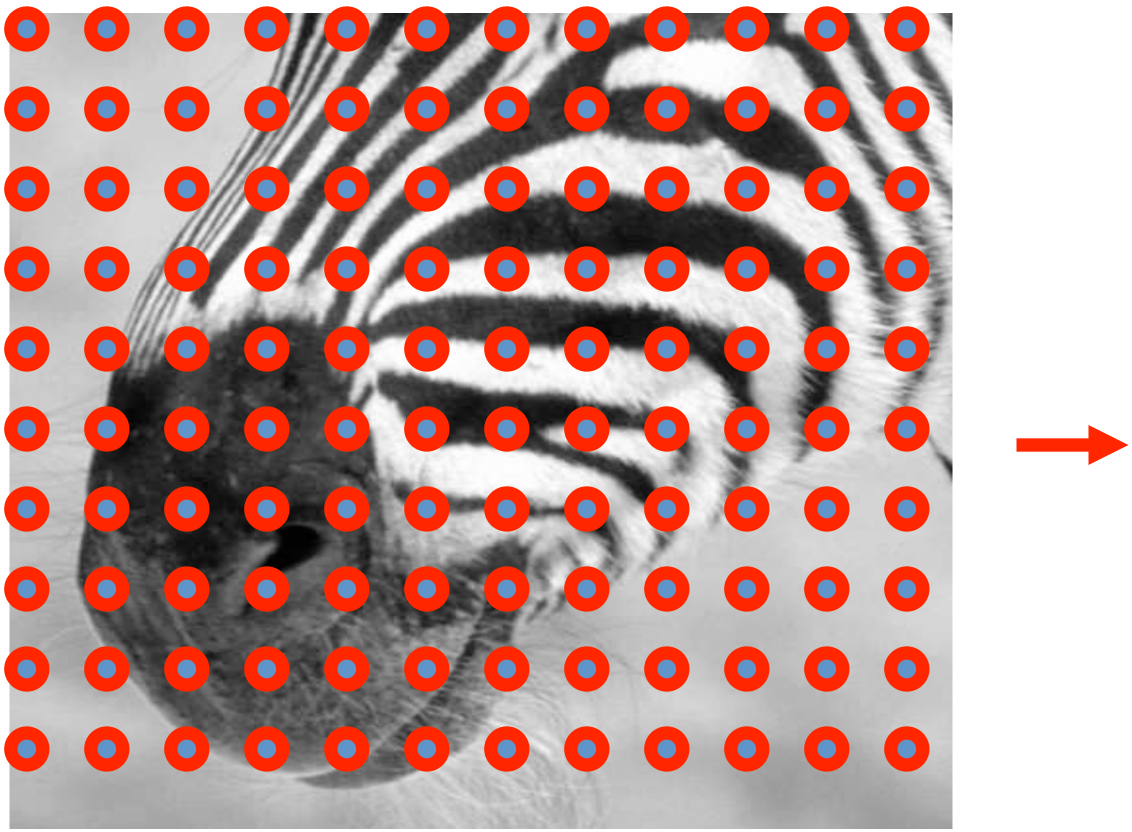


Sampling

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

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

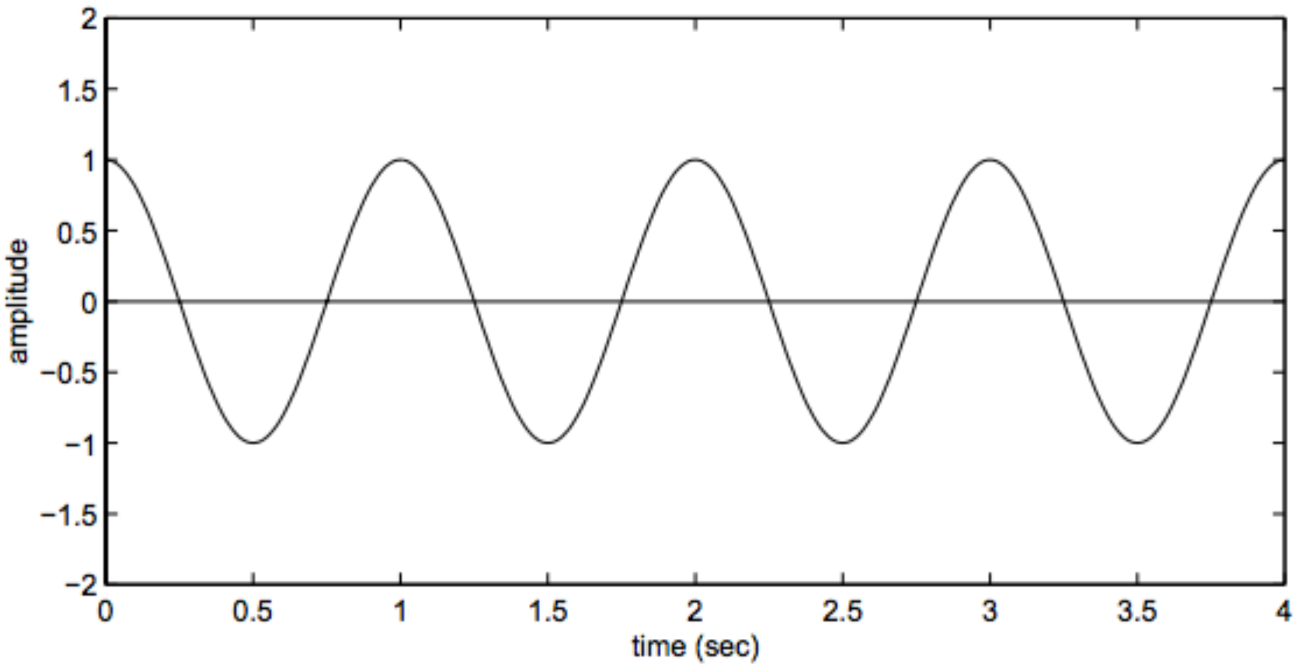
Subsampling by a factor of 2



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

Aliasing problem

- 1D example (sinewave):

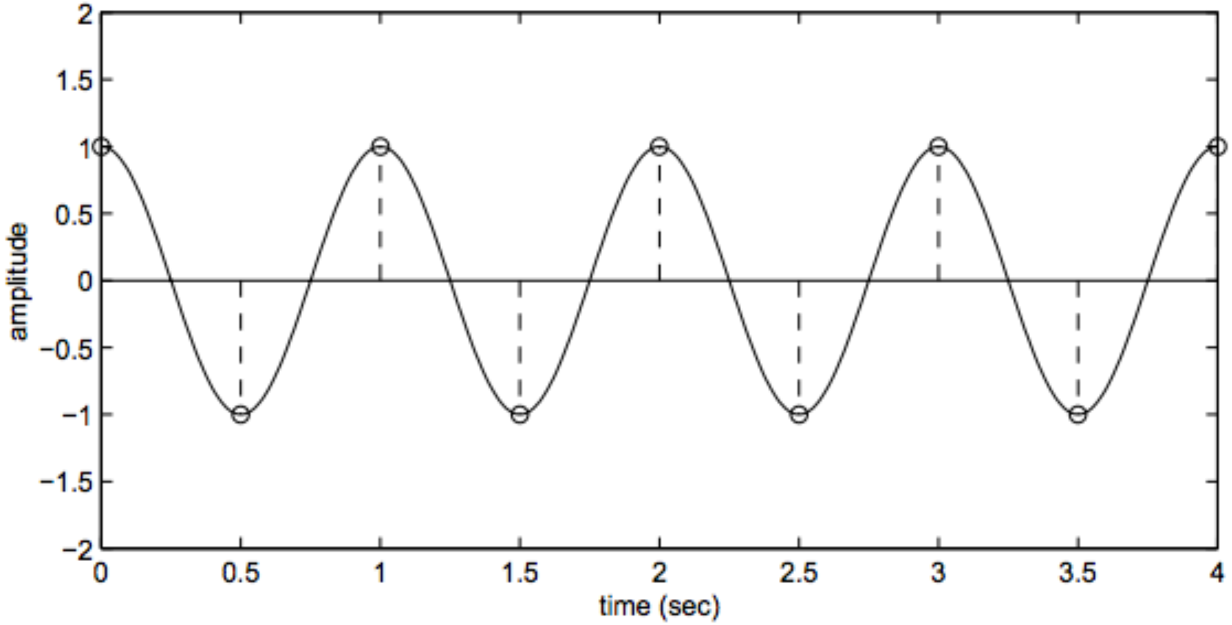


http://redwood.berkeley.edu/bruno/npb261/aliasing.pdf

Aliasing problem

- 1D example (sinewave):

Sample at 2Hz

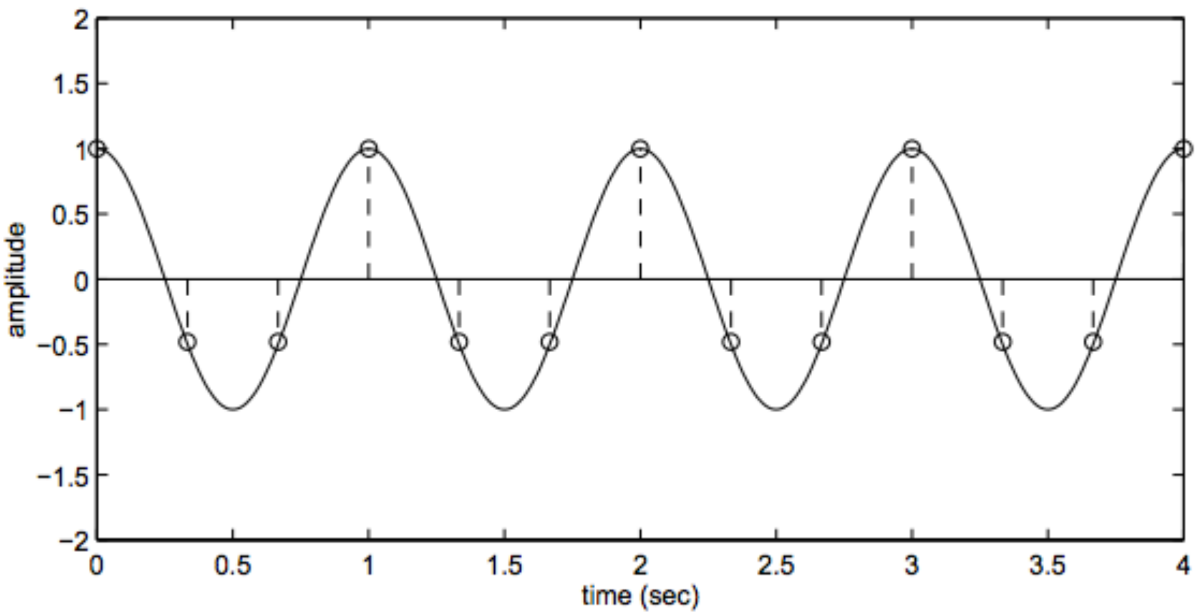


http://redwood.berkeley.edu/bruno/npb261/aliasing.pdf

Aliasing problem

- 1D example (sinewave):

Sample at 3Hz

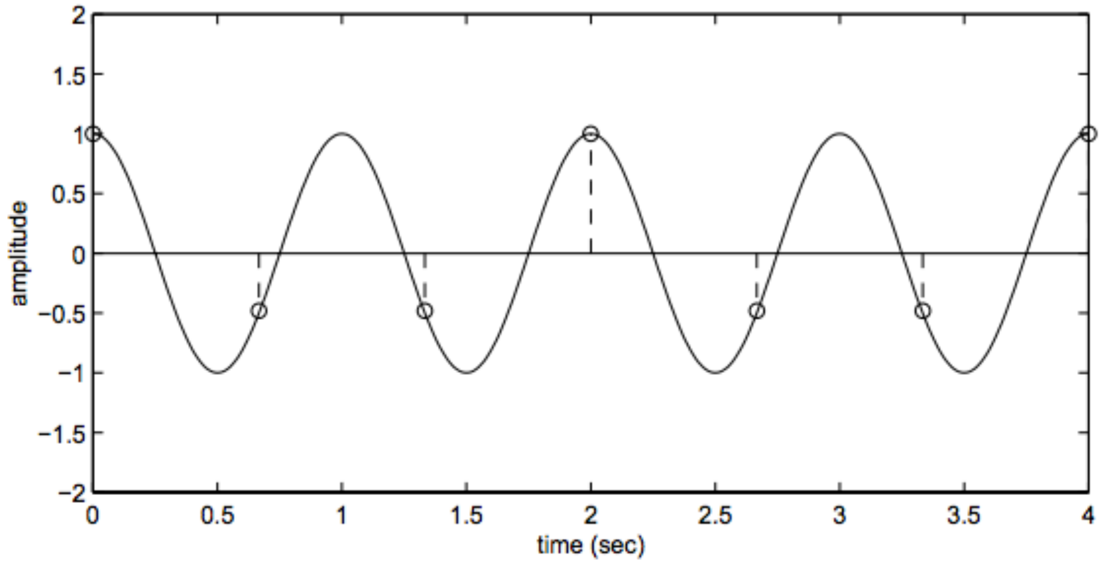


http://redwood.berkeley.edu/bruno/npb261/aliasing.pdf

Aliasing problem

- 1D example (sinewave):

Sample at 1.5 Hz

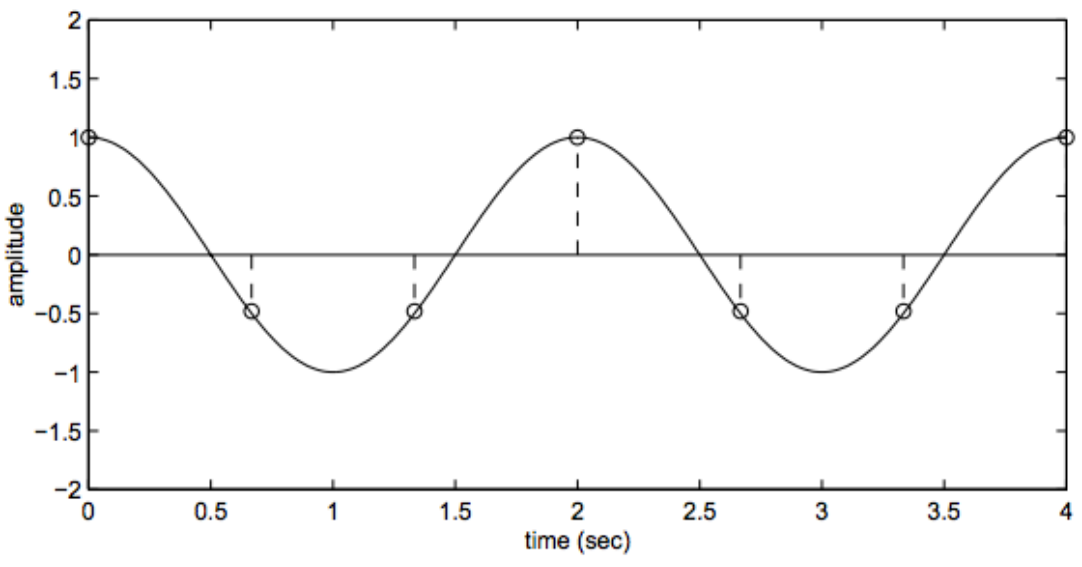


http://redwood.berkeley.edu/bruno/npb261/aliasing.pdf

Aliasing problem

- 1D example (sinewave):

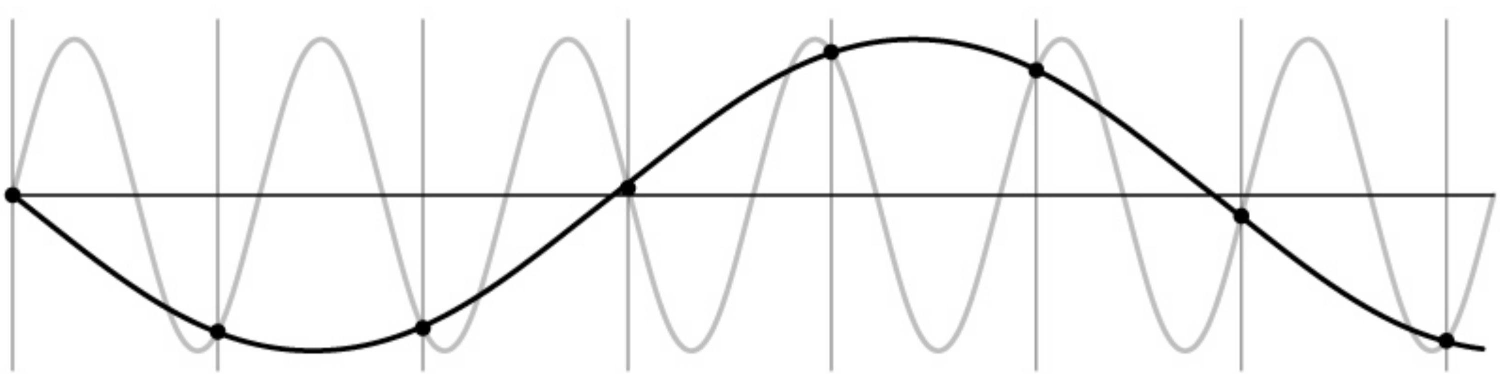
Sample at 1.5 Hz



http://redwood.berkeley.edu/bruno/npb261/aliasing.pdf

Aliasing problem

- 1D example (sinewave):



http://redwood.berkeley.edu/bruno/npb261/aliasing.pdf

Aliasing problem

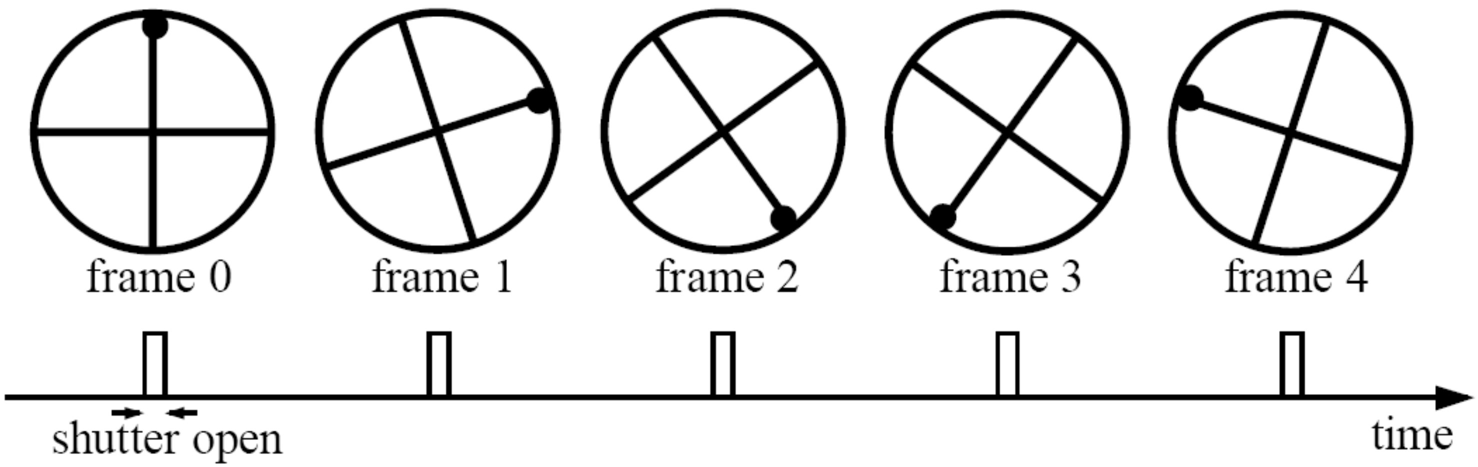
- Sub-sampling may be dangerous….
- Characteristic errors may appear:
  - "Wagon wheels rolling the wrong way in movies"
  - "Checkerboards disintegrate in ray tracing"
  - "Striped shirts look funny on color television"

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):

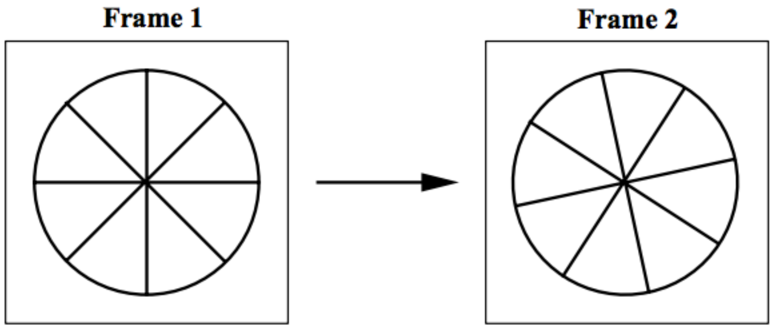


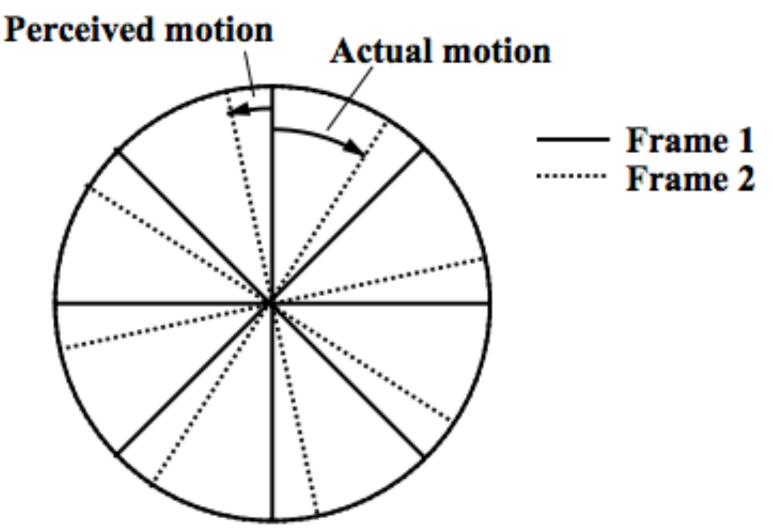
Without dot, wheel appears to be rotating slowly backwards! (counterclockwise)

Visual illusion: http://www.michaelbach.de/ot/mot-wagonWheel/index.html

Slide by Steve Seitz

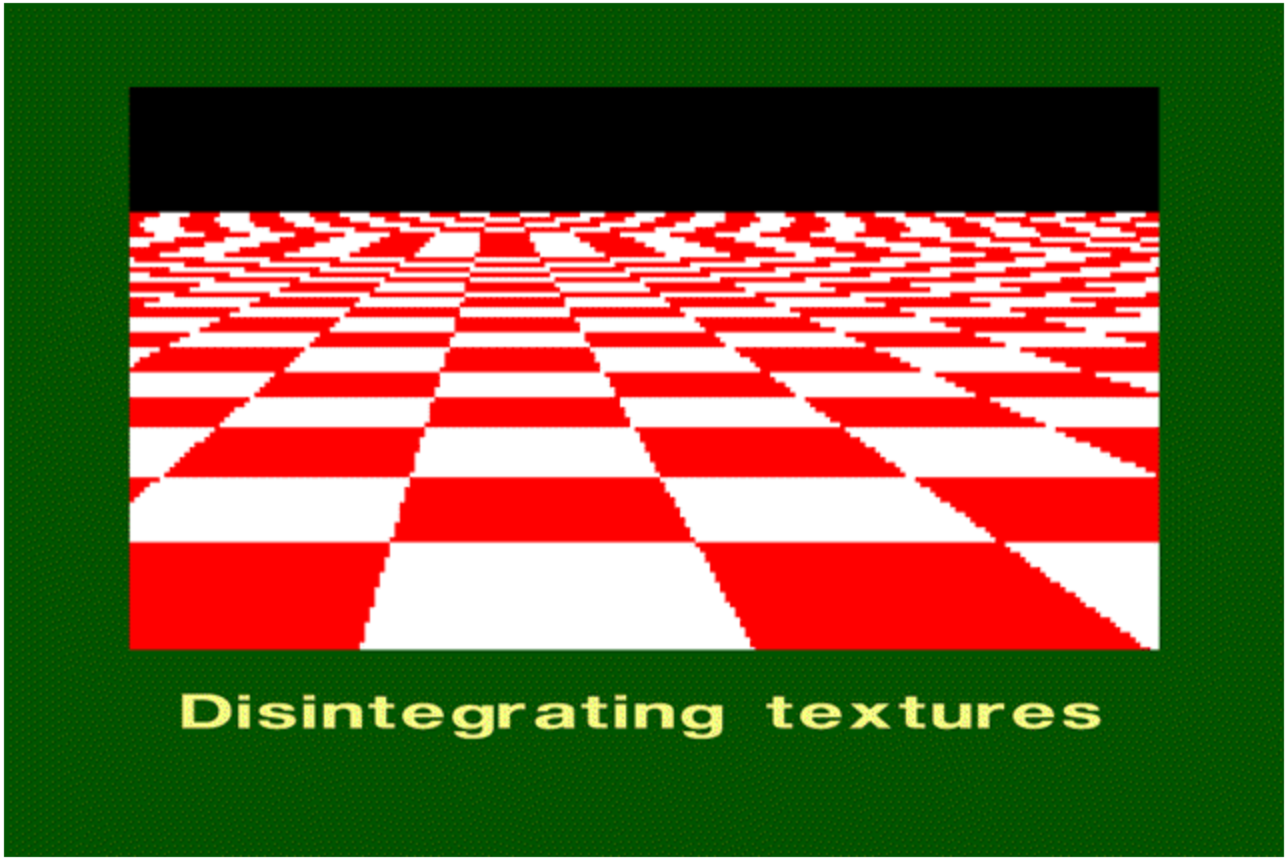
Aliasing in video





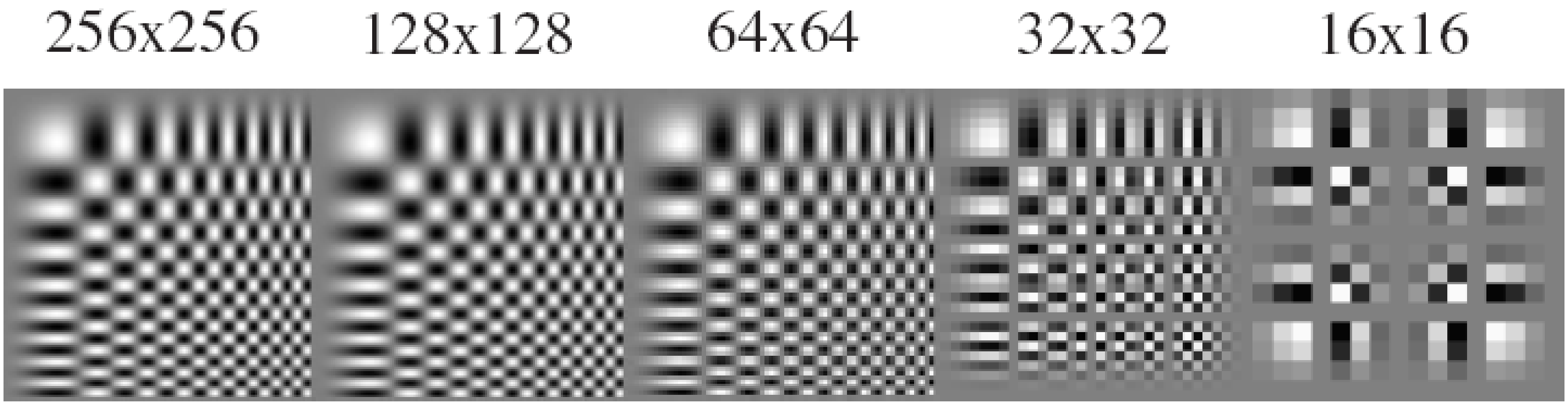
http://redwood.berkeley.edu/bruno/npb261/aliasing.pdf

Aliasing in graphics



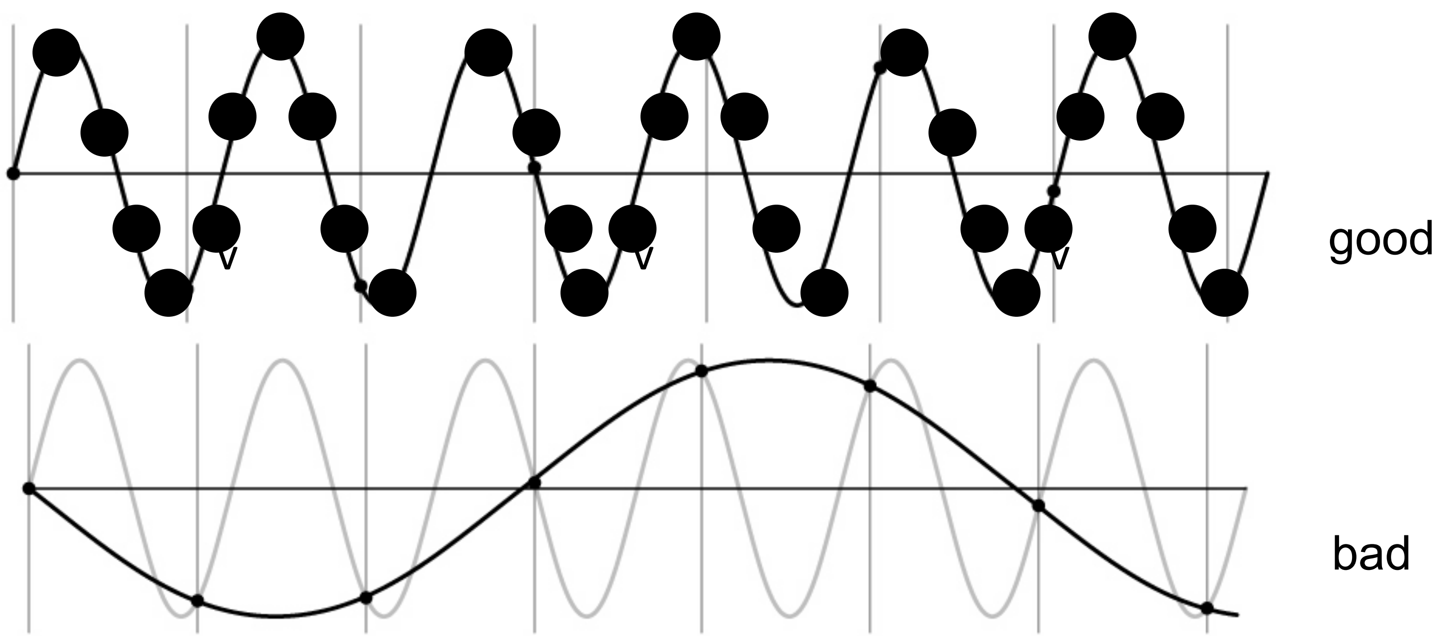
Source: A. Efros

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

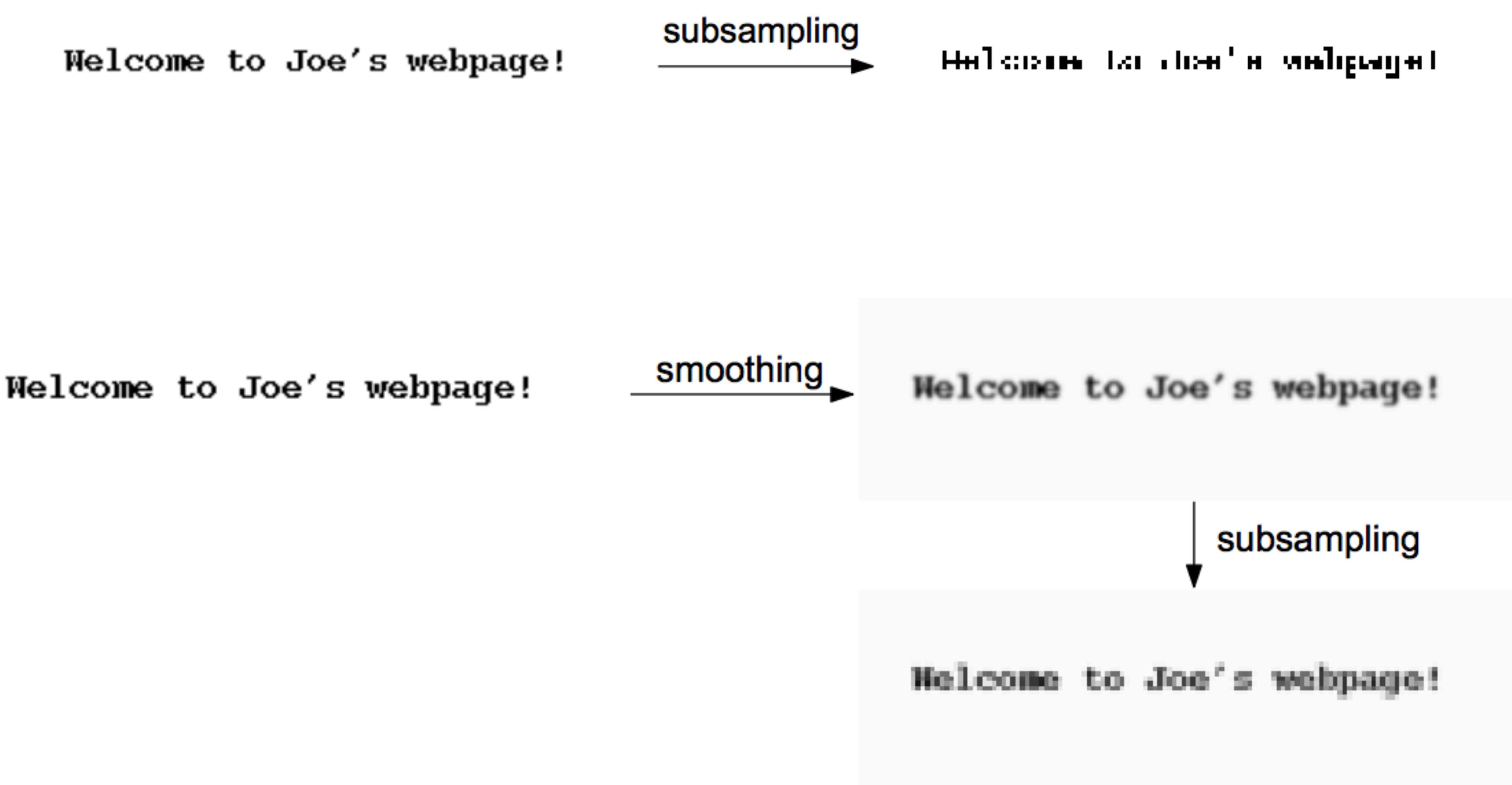
1. Start with image(h, w)
2. Apply low-pass filter

   im_blur = ndimage. filters_ gaussian_filter (image, 7)

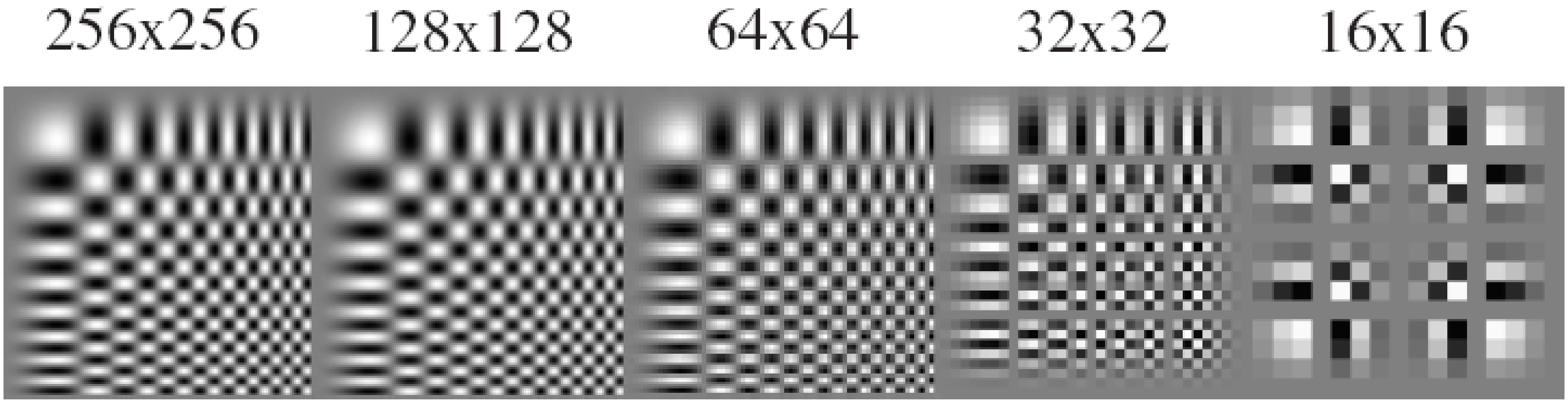
3. Sample every other pixel

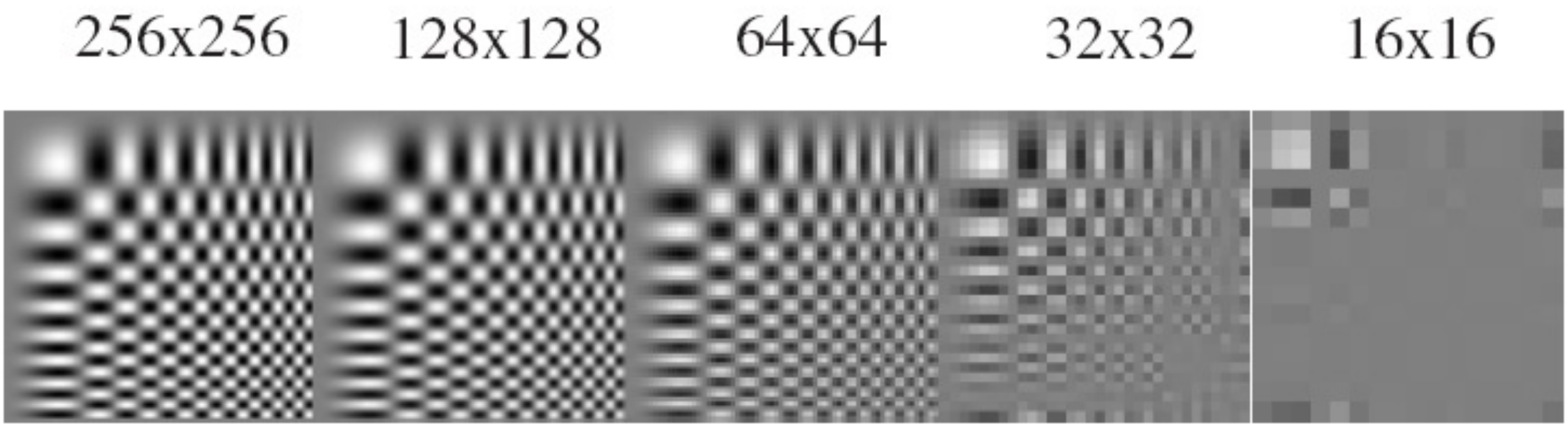
   im_small = im_blur[::2; ::2];

Text images


Welcome to Joe's webpage!
subsampling
smoothing
Welcome to Joe's webpage!
Welcome to Joe's webpage!
subsampling
Welcome to Joe's webpage!
Welcome to Joe's webpage!

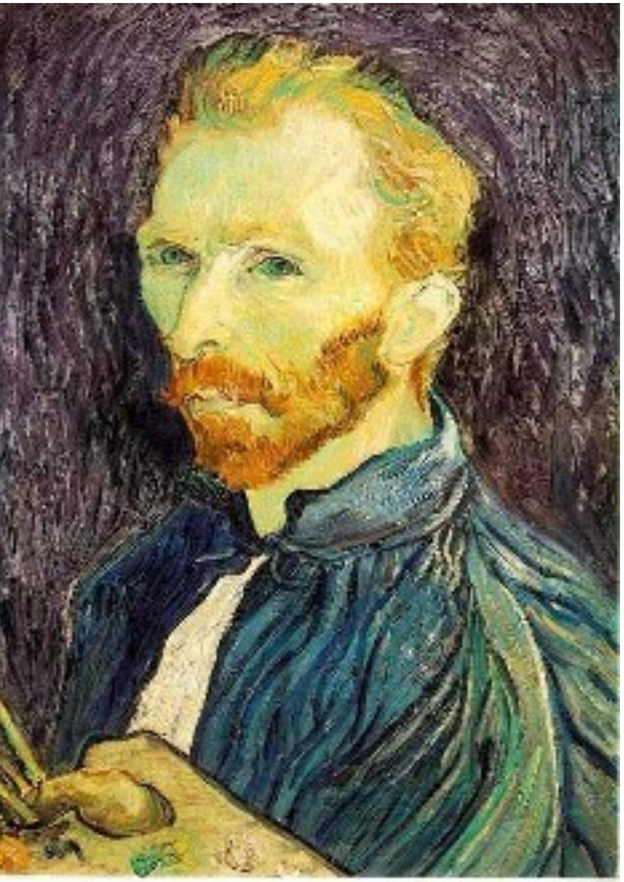
Anti-aliasing





Forsyth and Ponce 2002

Subsampling without pre-filtering



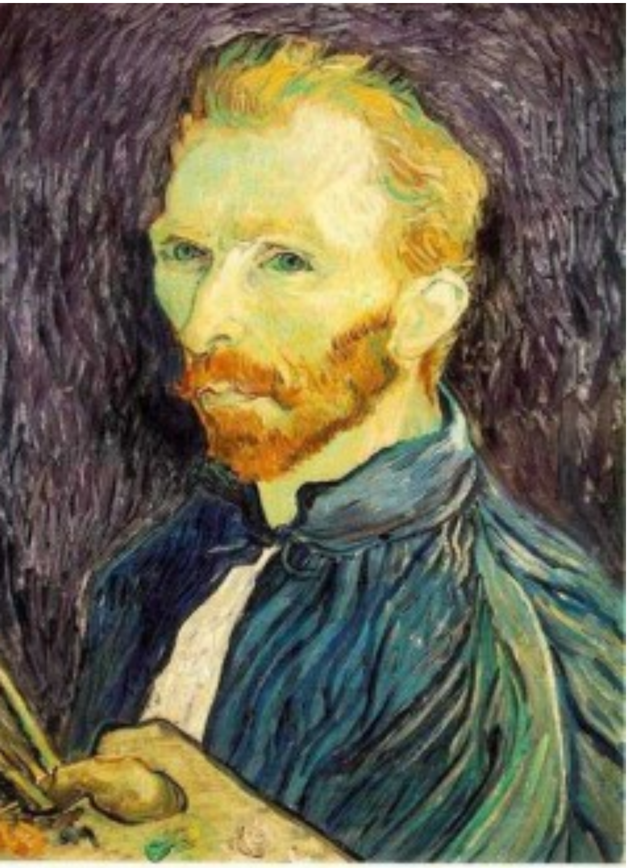
1/2

1/4 (2x zoom)

1/8 (4x zoom)

Slide by Steve Seitz

Subsampling with Gaussian pre-filtering



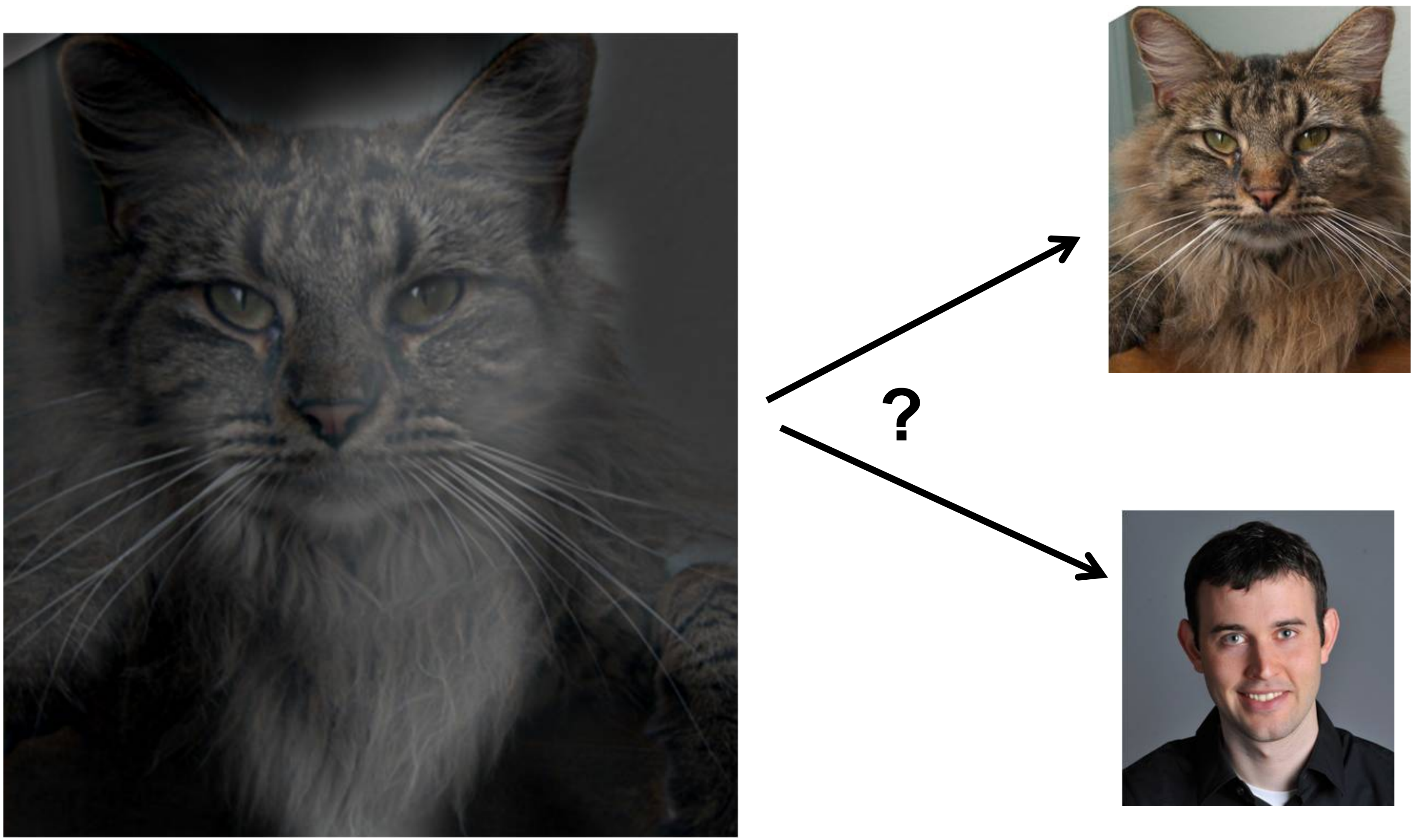
Gaussian 1/2

G 1/4

G 1/8

Slide by Steve Seitz

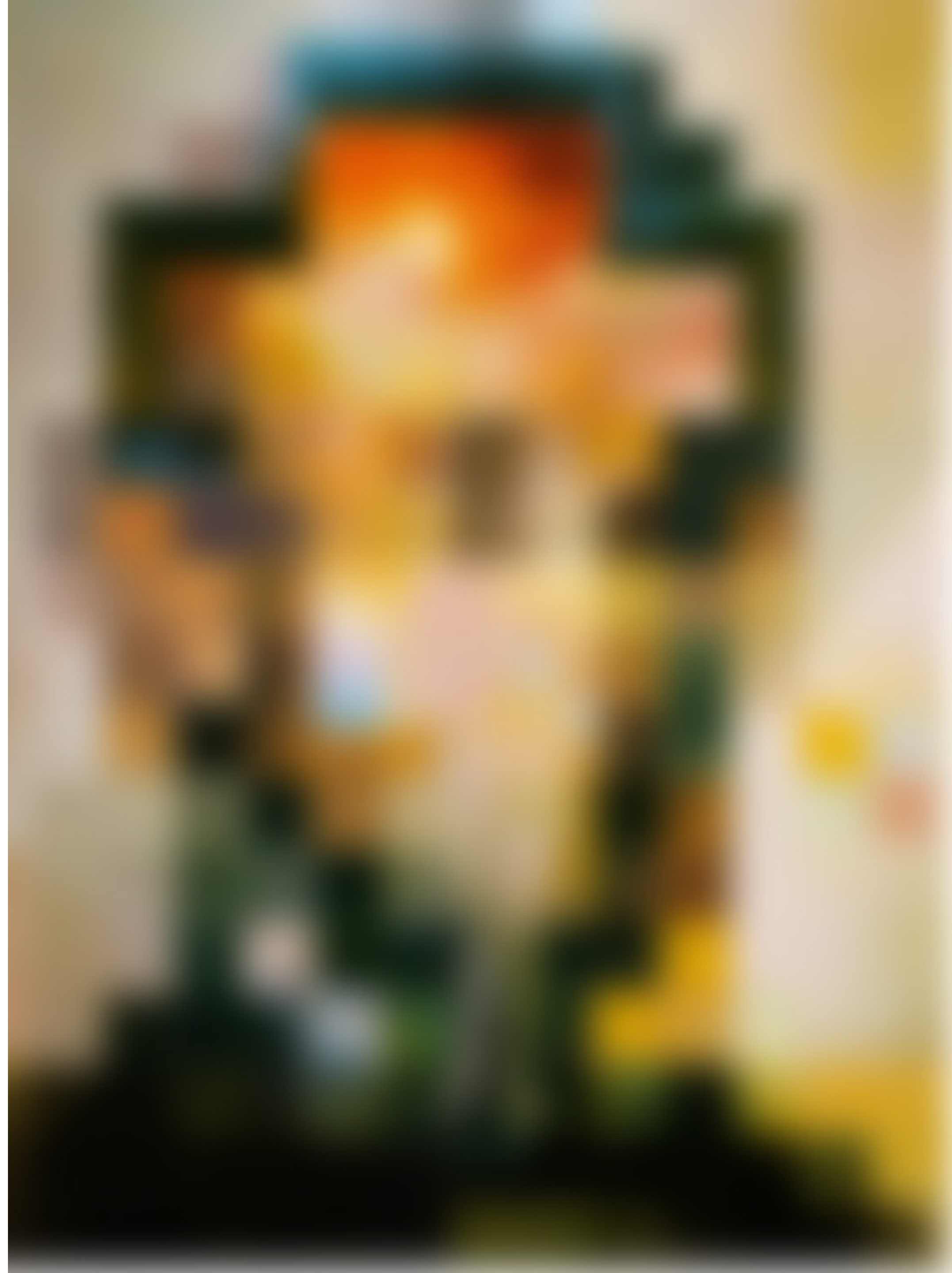
Why do we get different, distance-dependent interpretations of hybrid images?

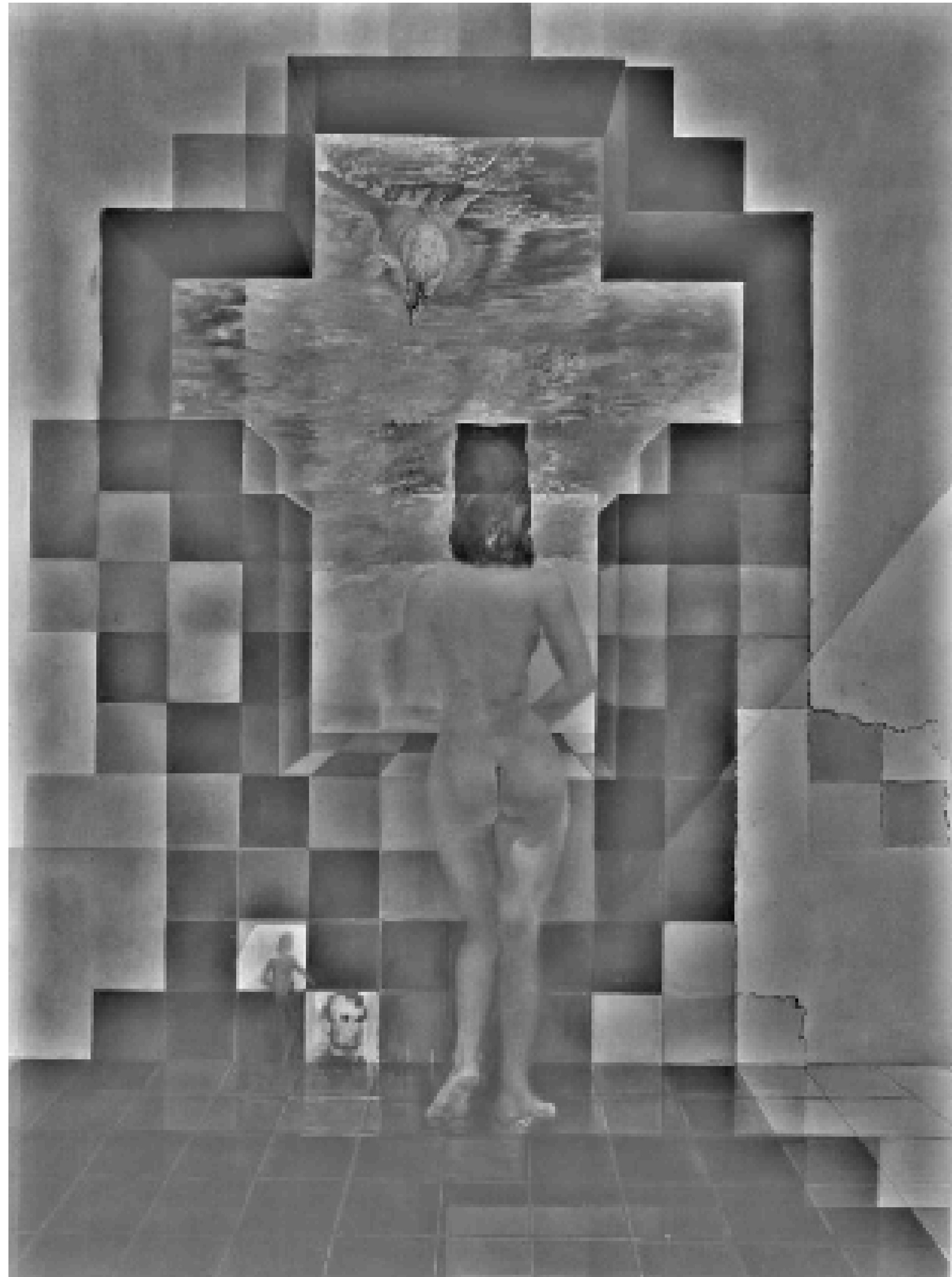


Salvador Dali invented Hybrid Images?

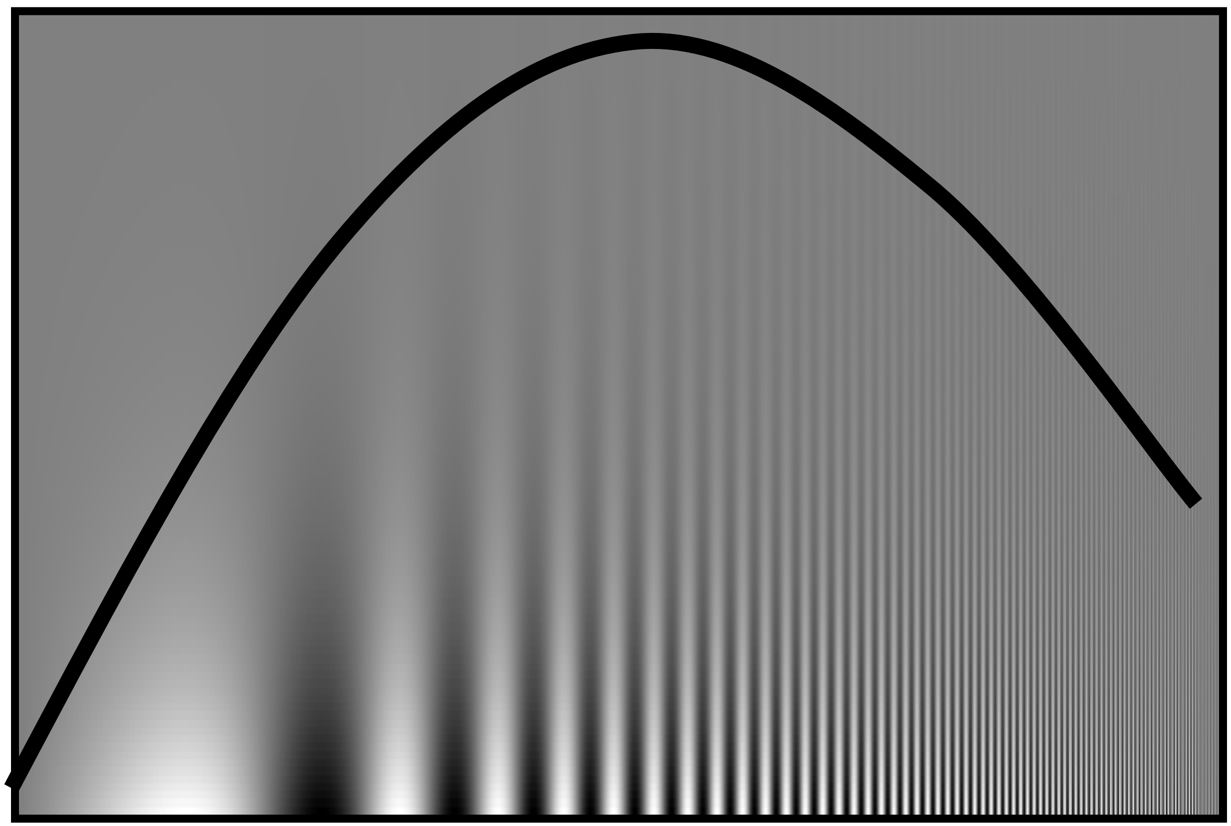
Salvador Dali

"Gala Contemplating the Mediterranean Sea, which at 30 meters becomes the portrait of Abraham Lincoln", 1976



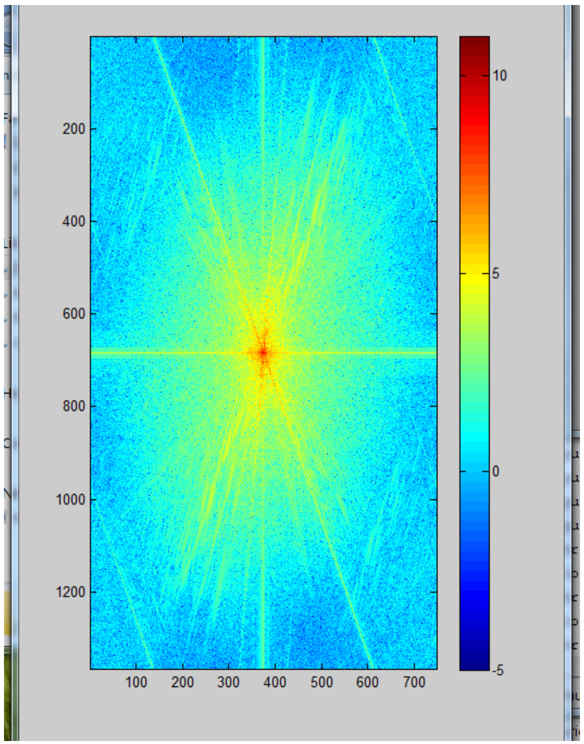


Campbell-Robson contrast sensitivity curve

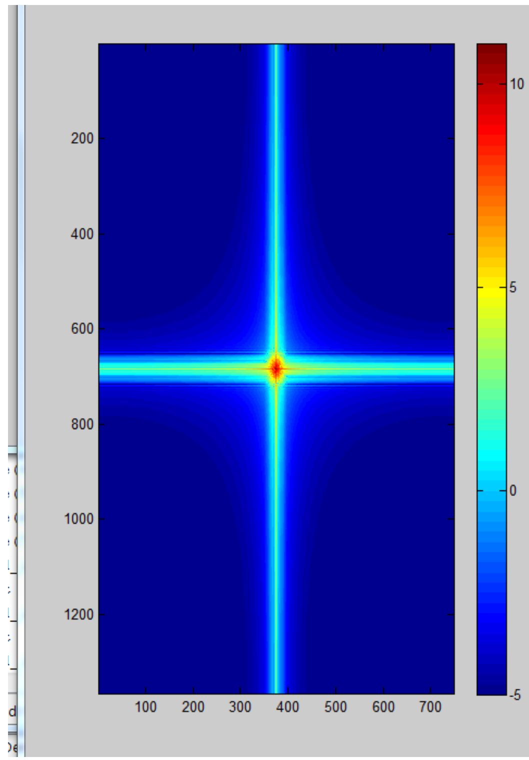


Hybrid Image in FFT

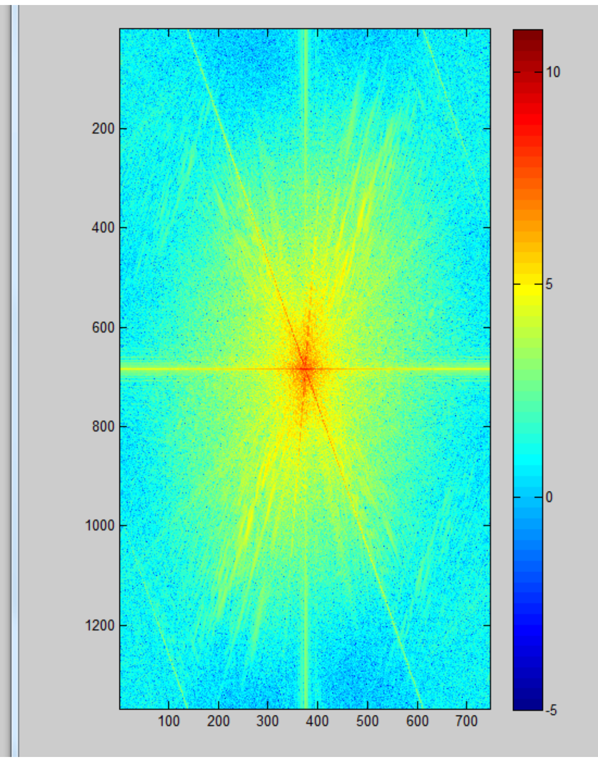
Hybrid Image



Low-passed Image



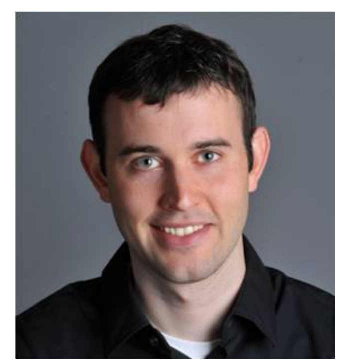
High-passed Image



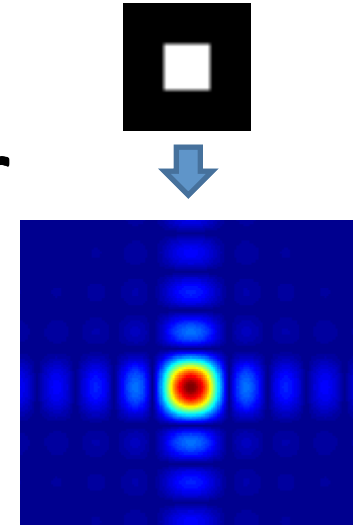
Perception

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

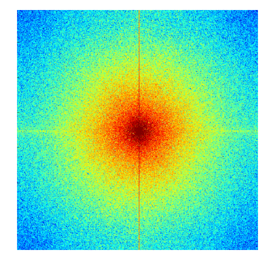
?



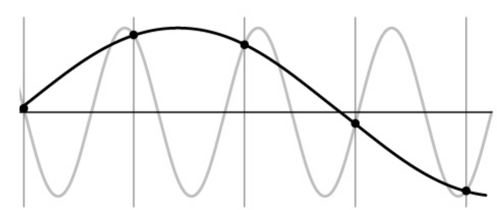
Things to Remember



- 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)



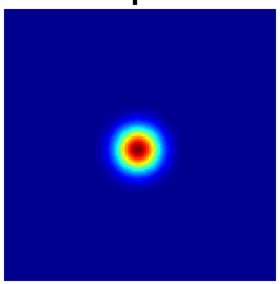
- Images are mostly smooth
  - Basis for compression
- Remember to low-pass before sampling



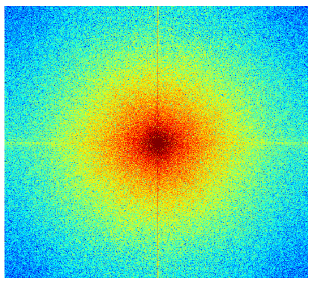
Practice question

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

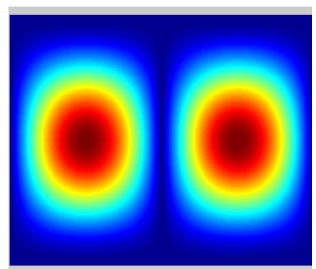
1



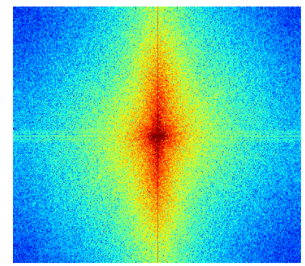
2



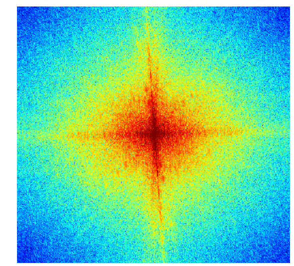
3



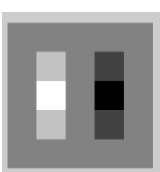
4



5

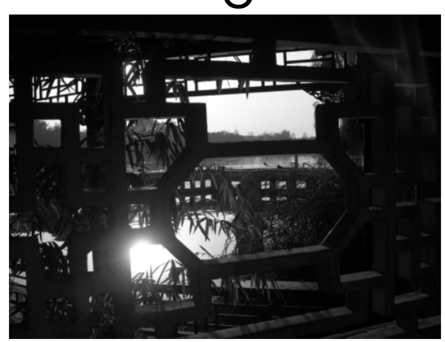


A

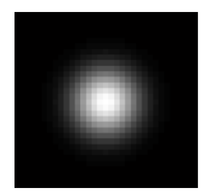


B

C



D



E

Take home reading

- Aliasing
  http://redwood.berkeley.edu/bruno/npb261/aliasing.pdf
- Fourier Transform: Chapter 3.4

- Next class: Template Matching, Gaussian Pyramid, Filter banks and Texture