Sampling and pixels

Lectures on Digital Photography
Spring 2016

Research at Google

Marc Levoy
Principal Engineer
Google Research

Professor, Emeritus
Computer Science Department
Stanford University

# Why study sampling theory?

- Why do I sometimes get moiré artifacts in my images?

- What is an antialiasing filter?

- How many megapixels is enough?

- How do I compute circle of confusion for depth of field?

- Is Apple's "Retina Display" just hype?

- What do MTF curves in lens reviews mean?

- What does Photoshop do when you downsize/upsize?

- Which is better: more pixels or more bits per pixel?

Outline

- ✦ frequency representations of images
  - filtering, blurring, sharpening
  - MTF as a measure of sharpness in images

- ✦ resolution and human perception
  - the spatial resolution of typical display media
  - the acuity of the human visual system
  - the right way to compute circle of confusion (C)

- ✦ sampling and aliasing
  - aliasing in space and time
  - prefiltering using convolution to avoid aliasing
  - prefiltering and sampling in cameras and Photoshop

- ✦ sampling versus quantization

Frequency representations
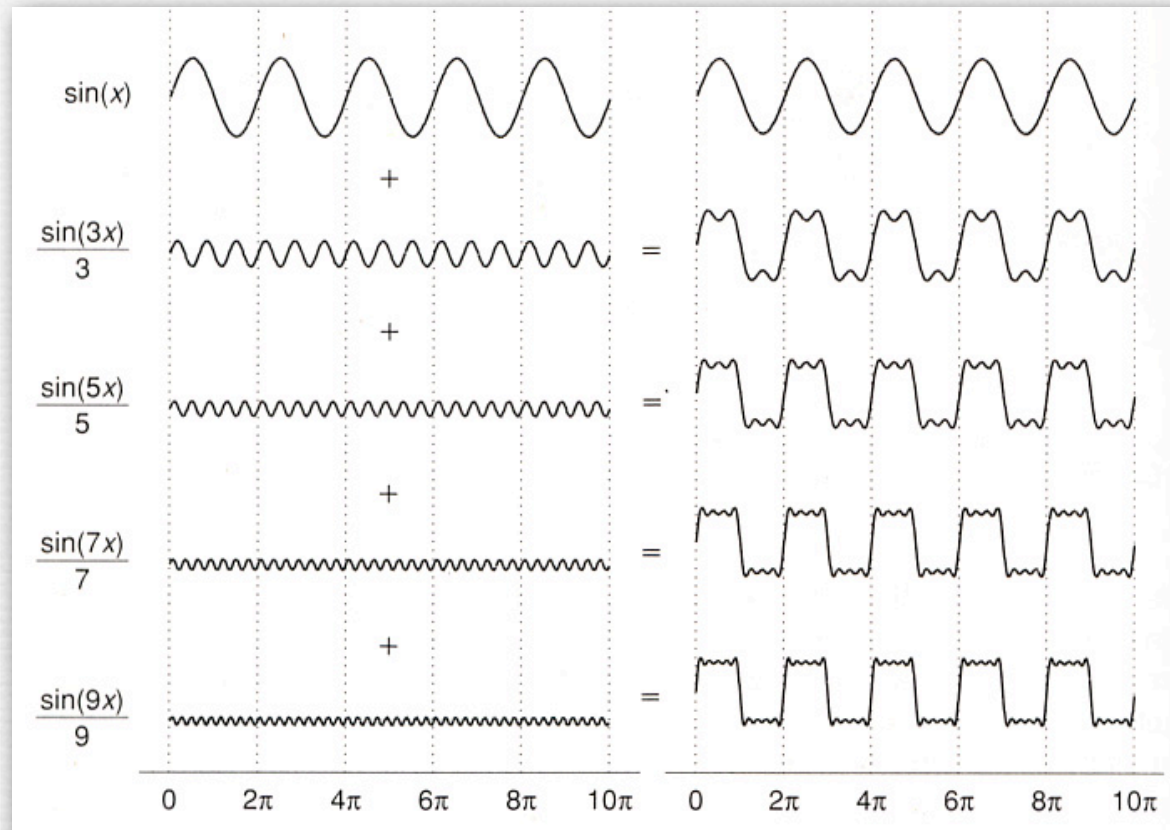
(Foley)

- ✦ a sum of sine waves, each of different wavelength (*frequency*) and height (*amplitude*), can approximate arbitrary functions

- ✦ to adjust horizontal position (*phase*), replace with cosine waves, or use a mixture of sine and cosine waves

Frequency representations

✦ <u>Fourier series:</u> any continuous, integrable, periodic function can be represented as an infinite series of sines and cosines

$$f(x) \;=\; \frac{a_0}{2} + \sum_{n=1}^{\infty} \Big[a_n \cos(nx) + b_n \sin(nx) \Big]$$

✦ a sum of sine waves, each of different wavelength (*frequency*) and height (*amplitude*), can approximate arbitrary functions

✦ to adjust horizontal position (*phase*), replace with cosine waves, or use a mixture of sine and cosine waves

Fourier transforms of images

- θ gives angle of sinusoid
- r gives spatial frequency
- brightness gives amplitude of sinusoid present in image

r

θ
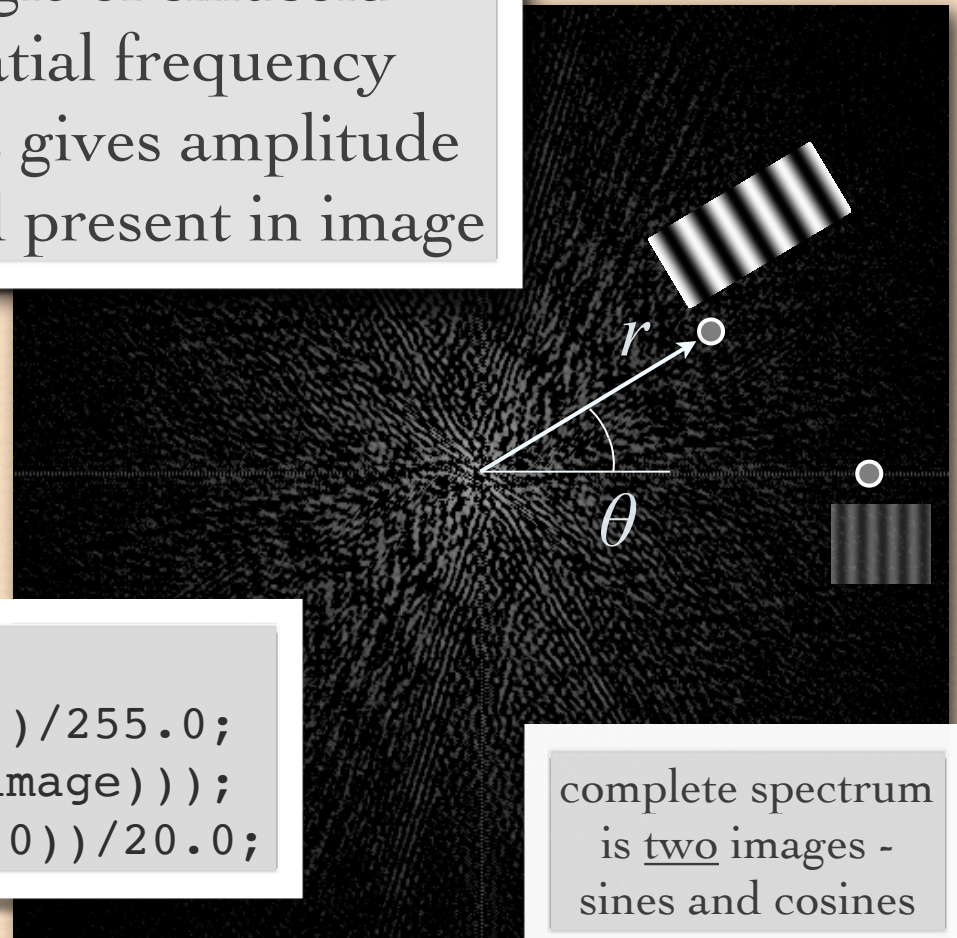
```
% In Matlab:
image = double(imread('flower.tif'))/255.0;
fourier = fftshift(fft2(ifftshift(image)));
fftimage = log(max(real(fourier),0.0))/20.0;
```

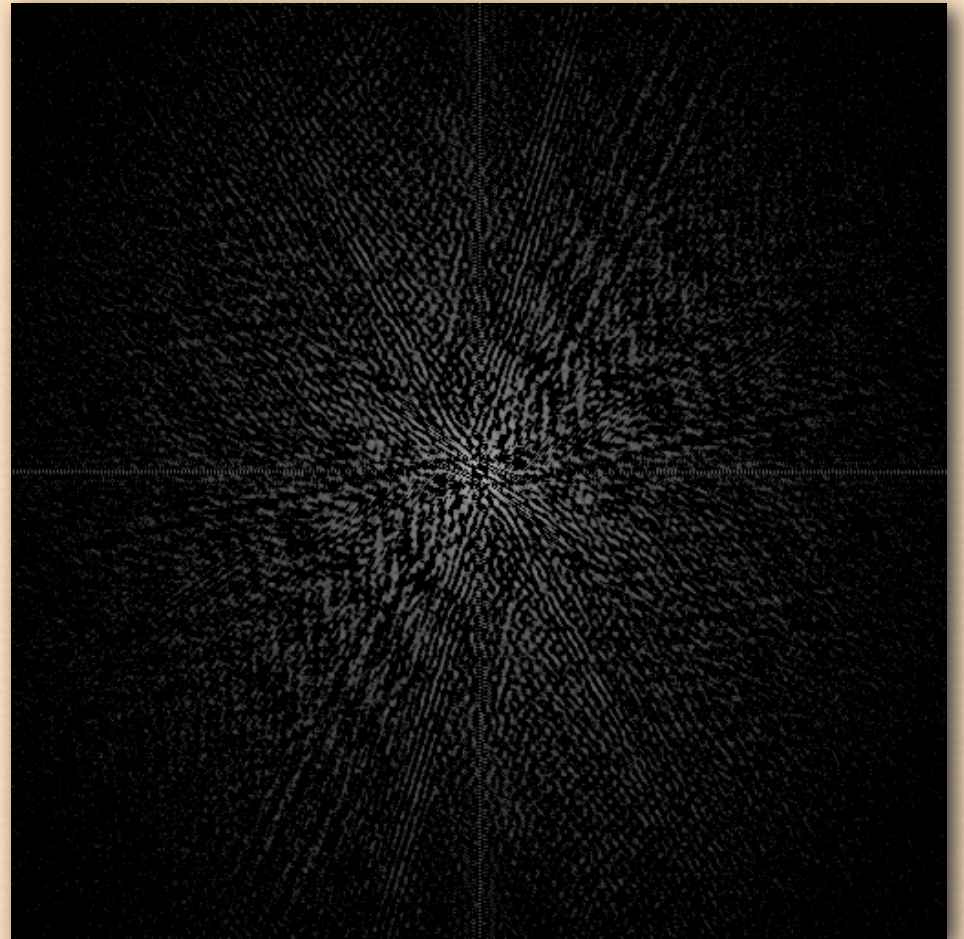complete spectrum is two images - sines and cosines

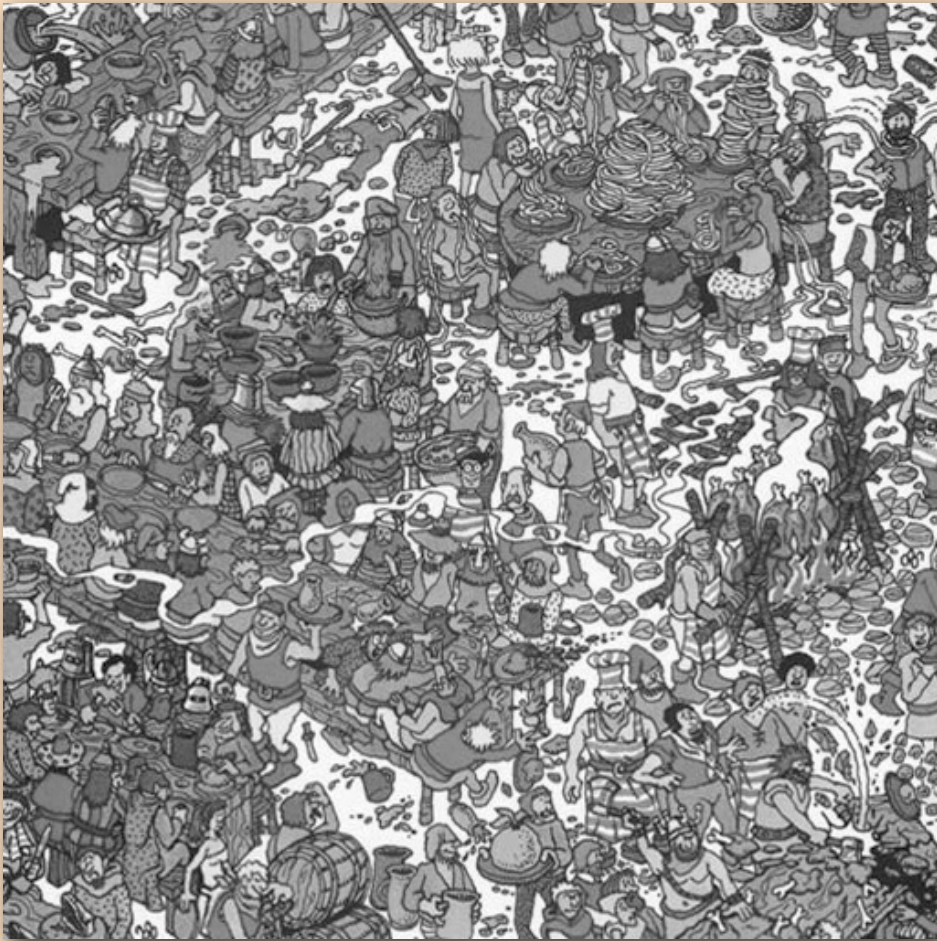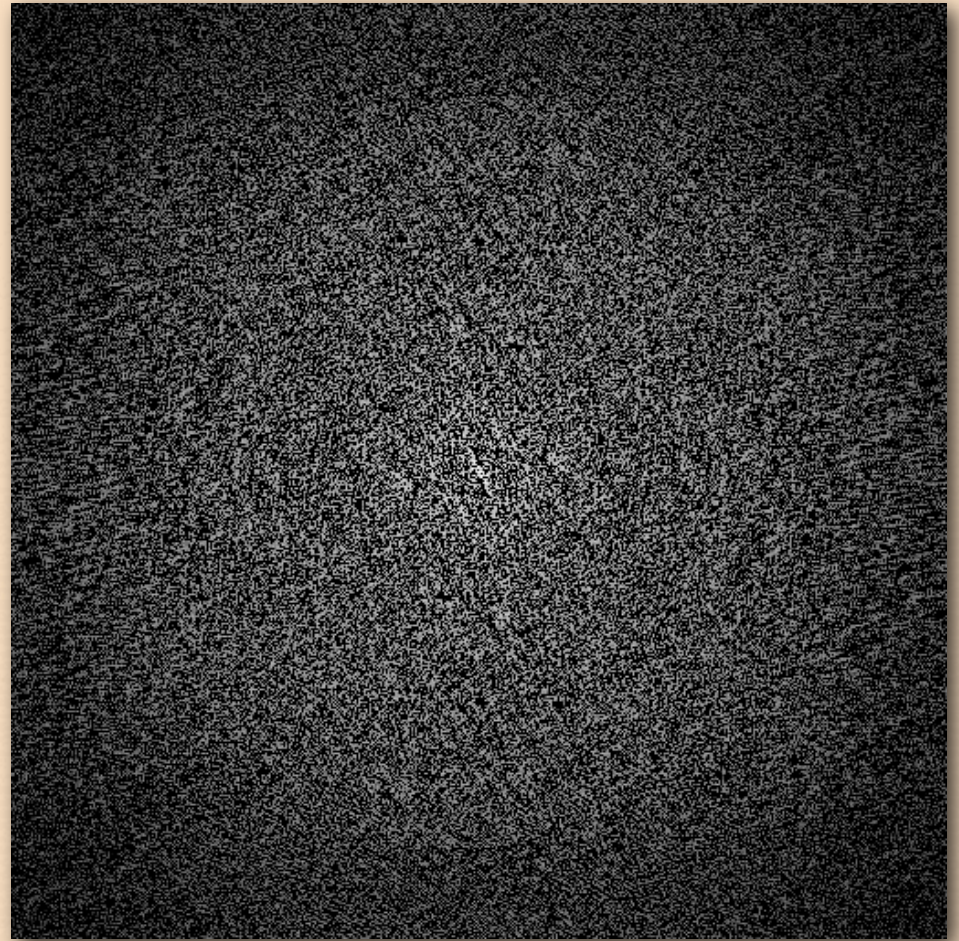image

spectrum

A typical photograph

image

spectrum
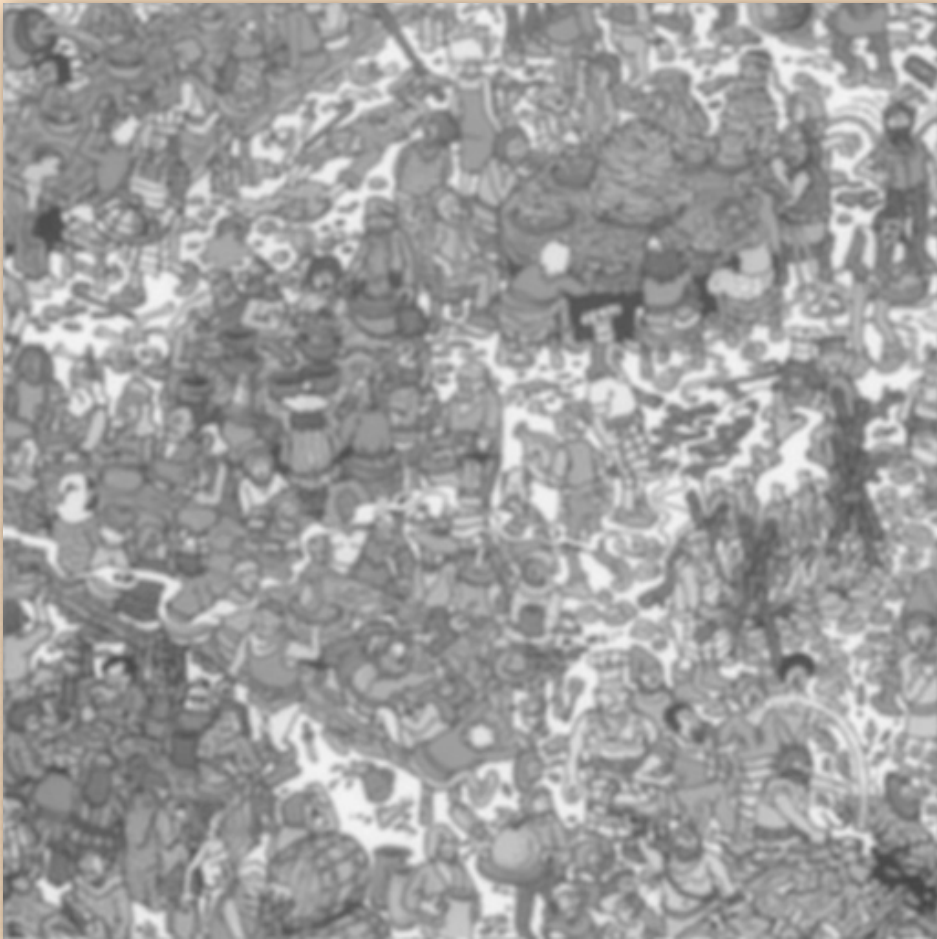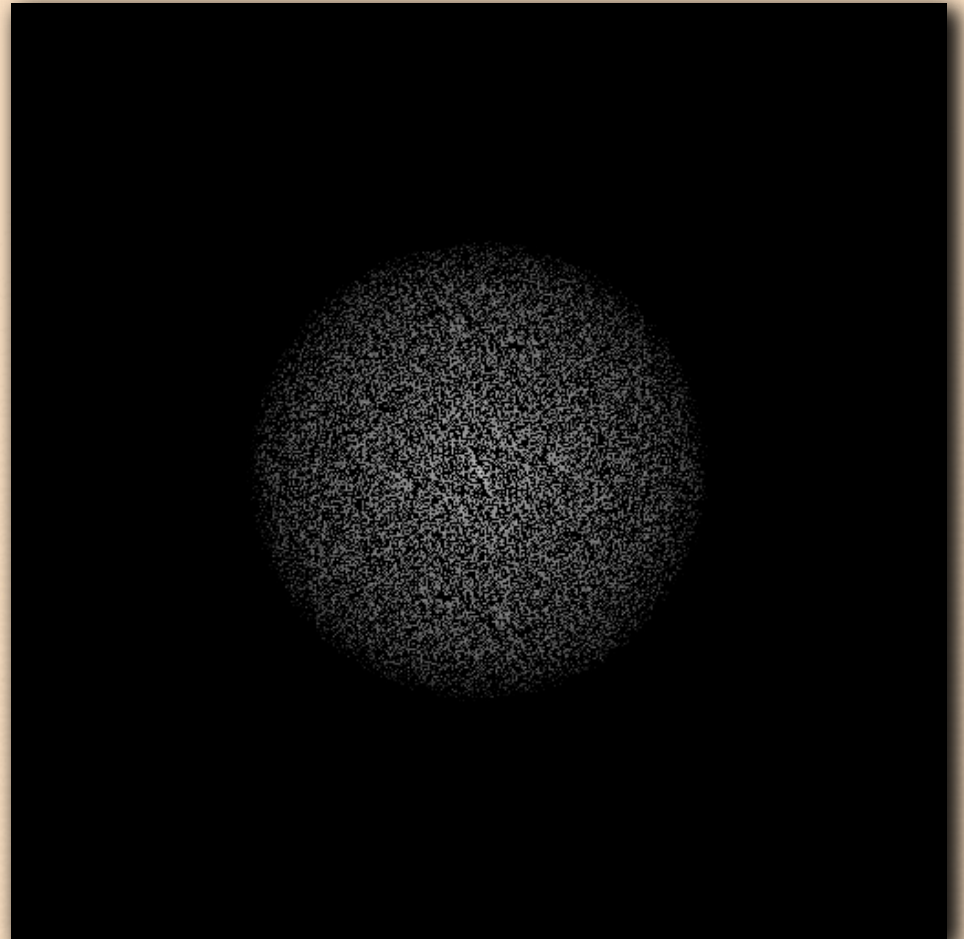
An image with higher frequencies

image

spectrum

Blurring in the Fourier domain
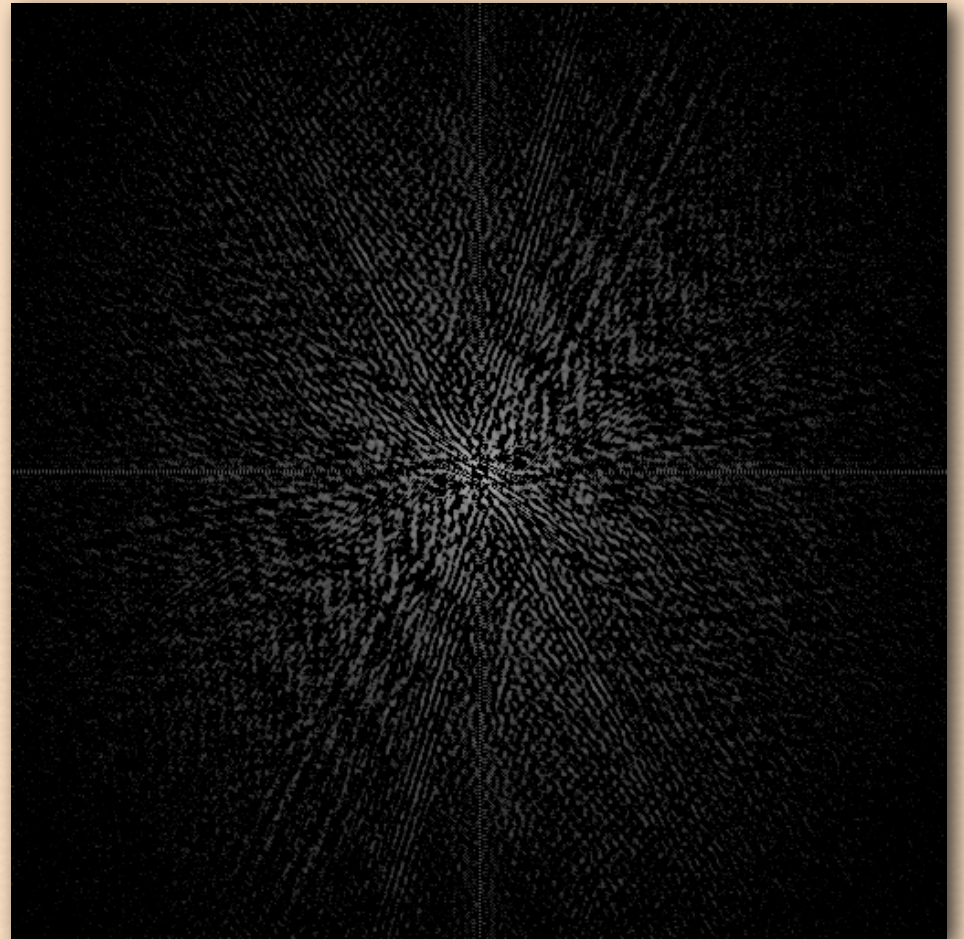

image

spectrum

Original flower
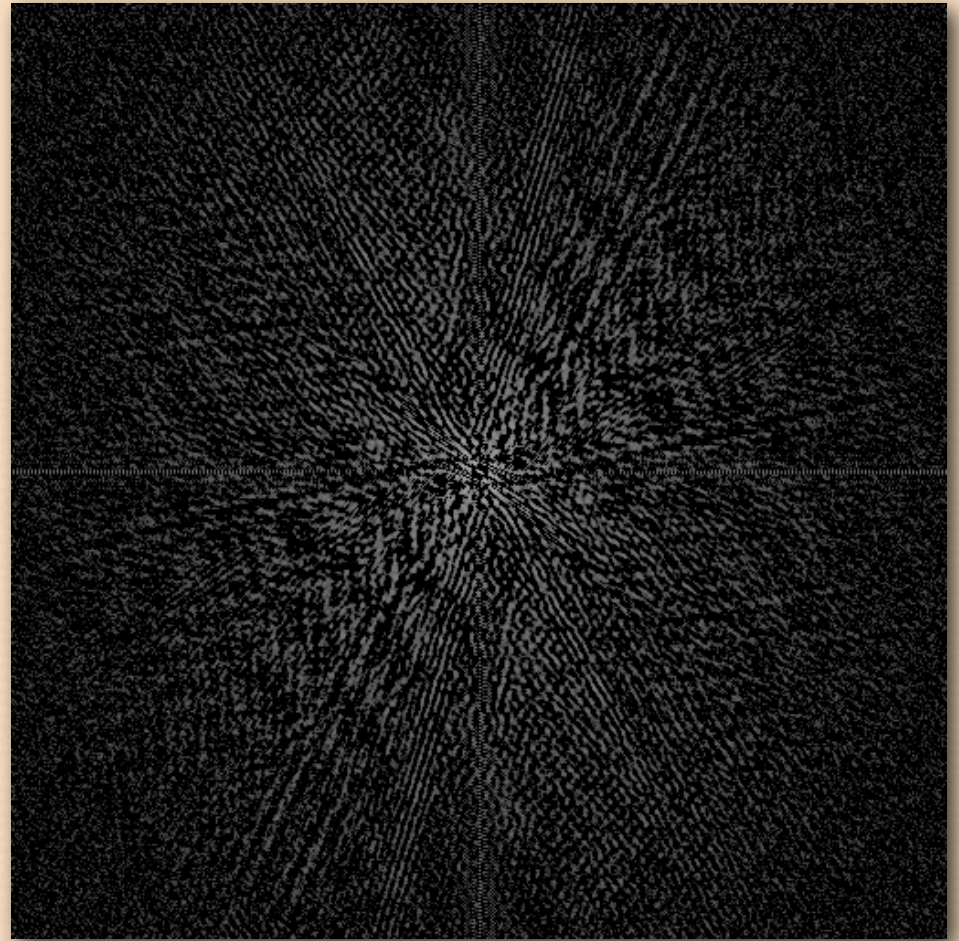

image

spectrum

Sharpening in the Fourier domain
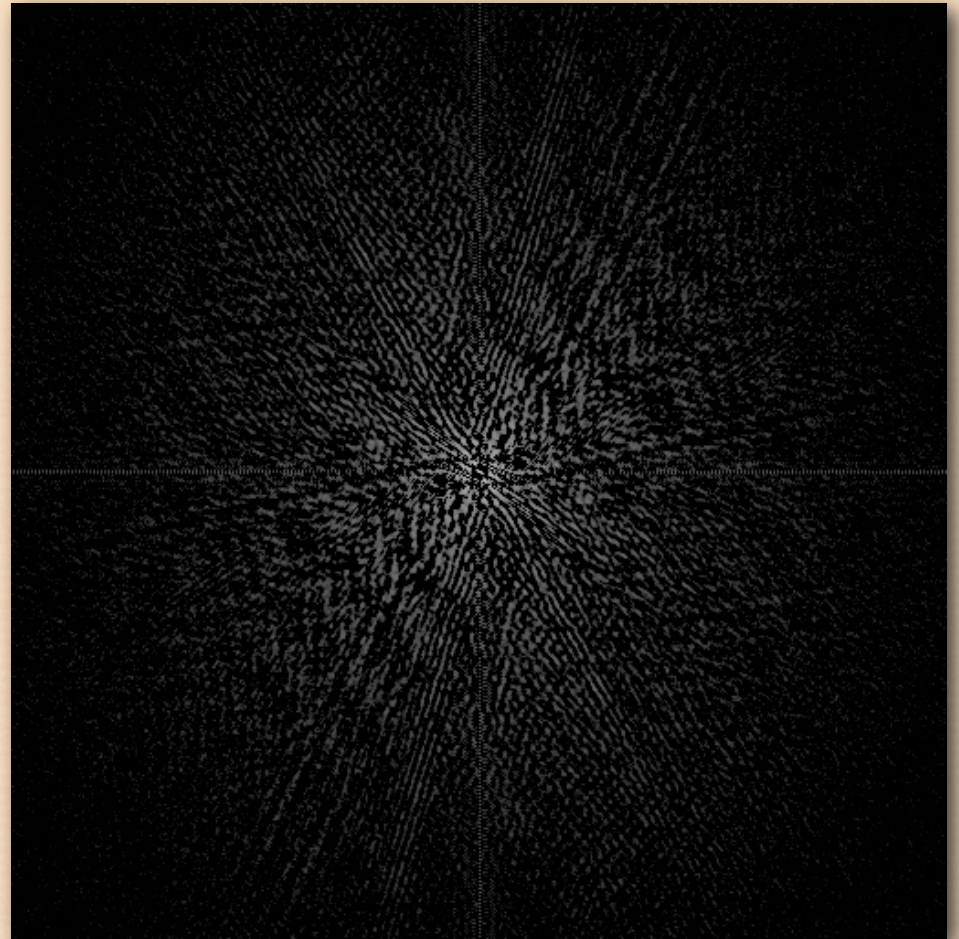

image

spectrum

Q. What does <u>this</u> filtering operation do?
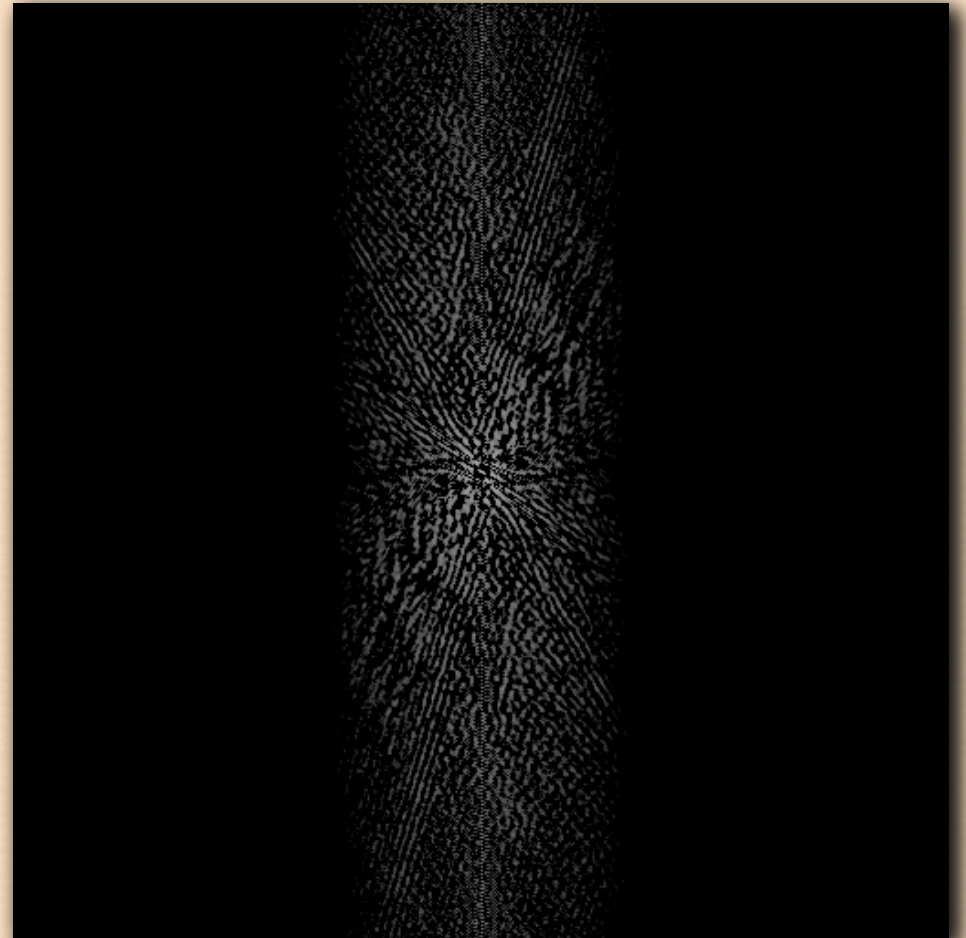

image

spectrum

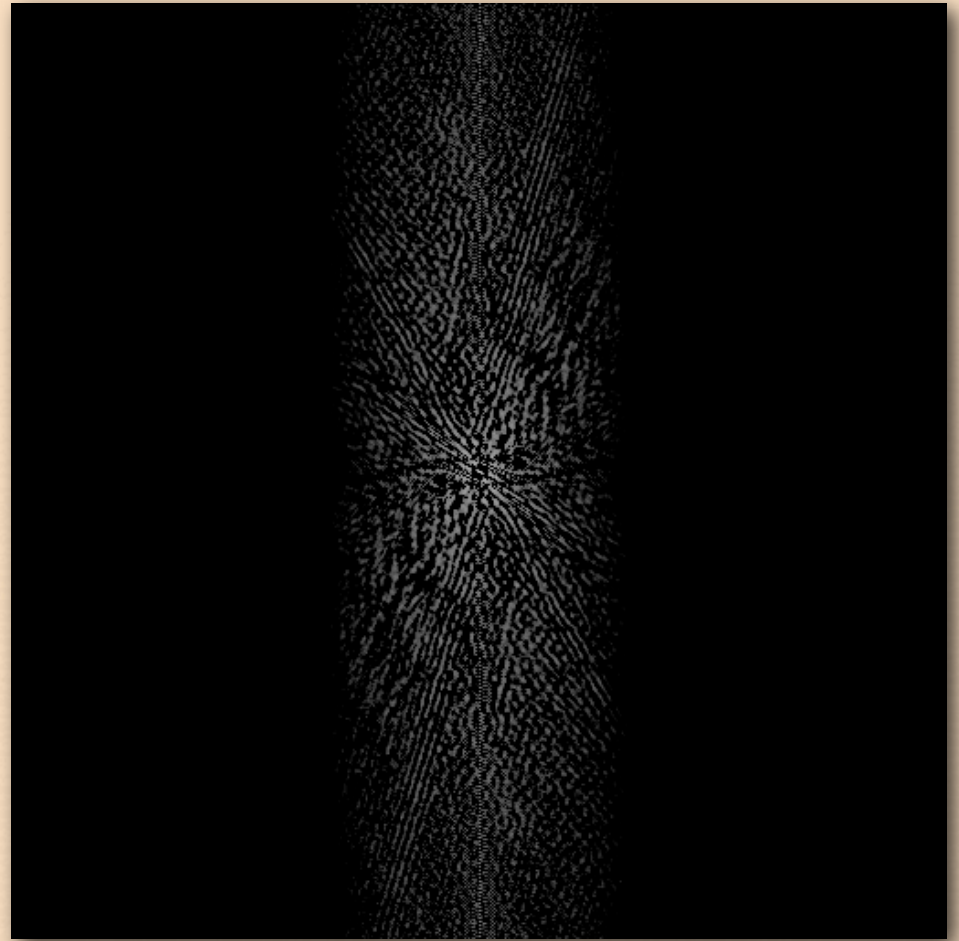# Q. What does <u>this</u> filtering operation do?

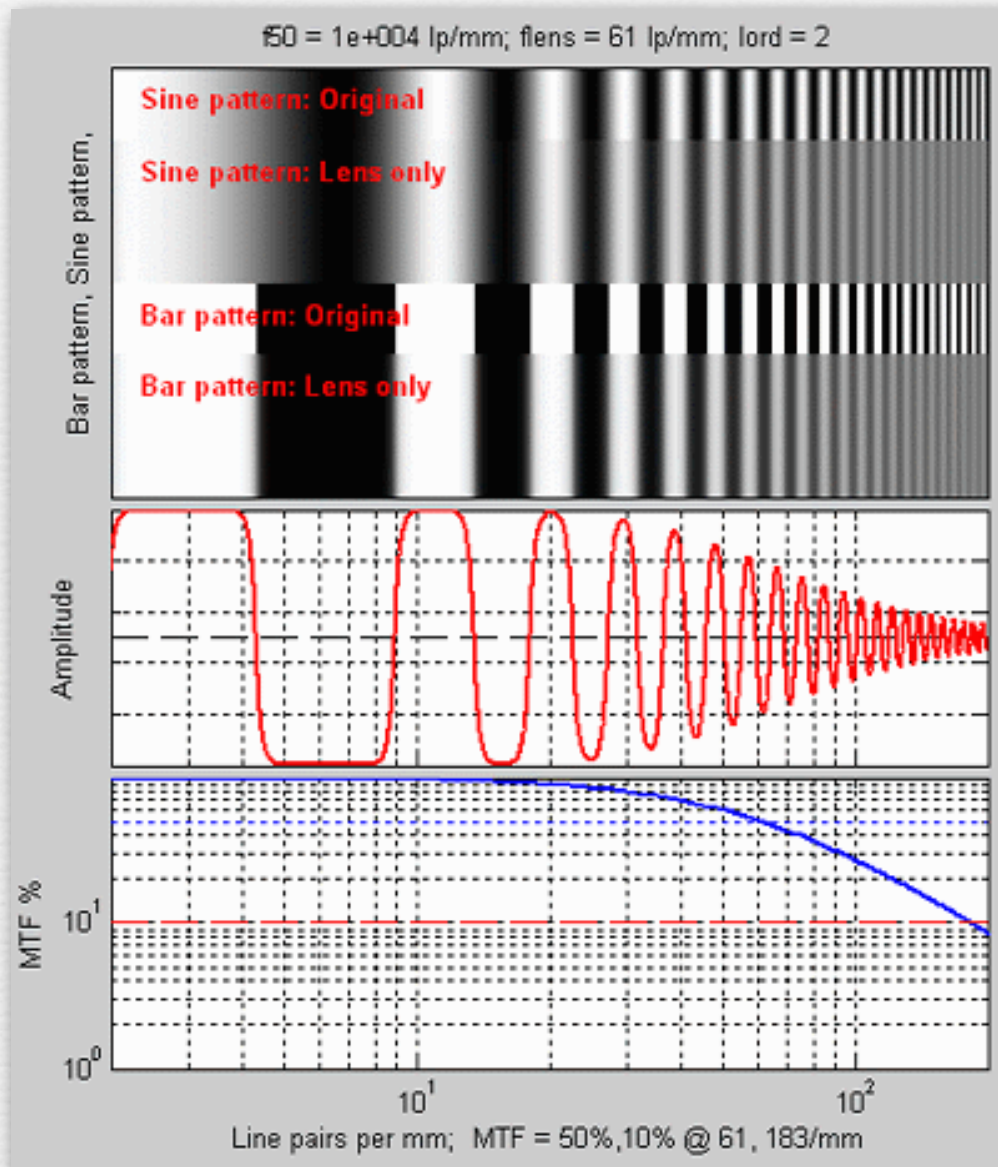
image

spectrum

Blurring in *x*, sharpening in *y*

image

spectrum

argh, astigmatism!

Describing sharpness in images:
the modulation transfer function (MTF)

f50 = 1e+004 lp/mm; flens = 61 lp/mm; lord = 2

Sine pattern: Original
Sine pattern: Lens only
Bar pattern: Original
Bar pattern: Lens only

Bar pattern, Sine pattern,
Amplitude
MTF %

Line pairs per mm; MTF = 50%,10% @ 61, 183/mm

- ✦ the amount of each spatial frequency that can be reproduced by an optical system

- ✦ loss may be due to misfocus, aberrations, diffraction, manufacturing defects, nose smudges, etc.

- ✦ MTF is contrast at each frequency relative to original signal

(imatest.com)

© Marc Levoy

Two different MTF curves

✦ in one curve, contrast stays high, but drops off at a relatively low resolution

✦ in the other curve, higher-resolution features are preserved, but contrast is lower throughout
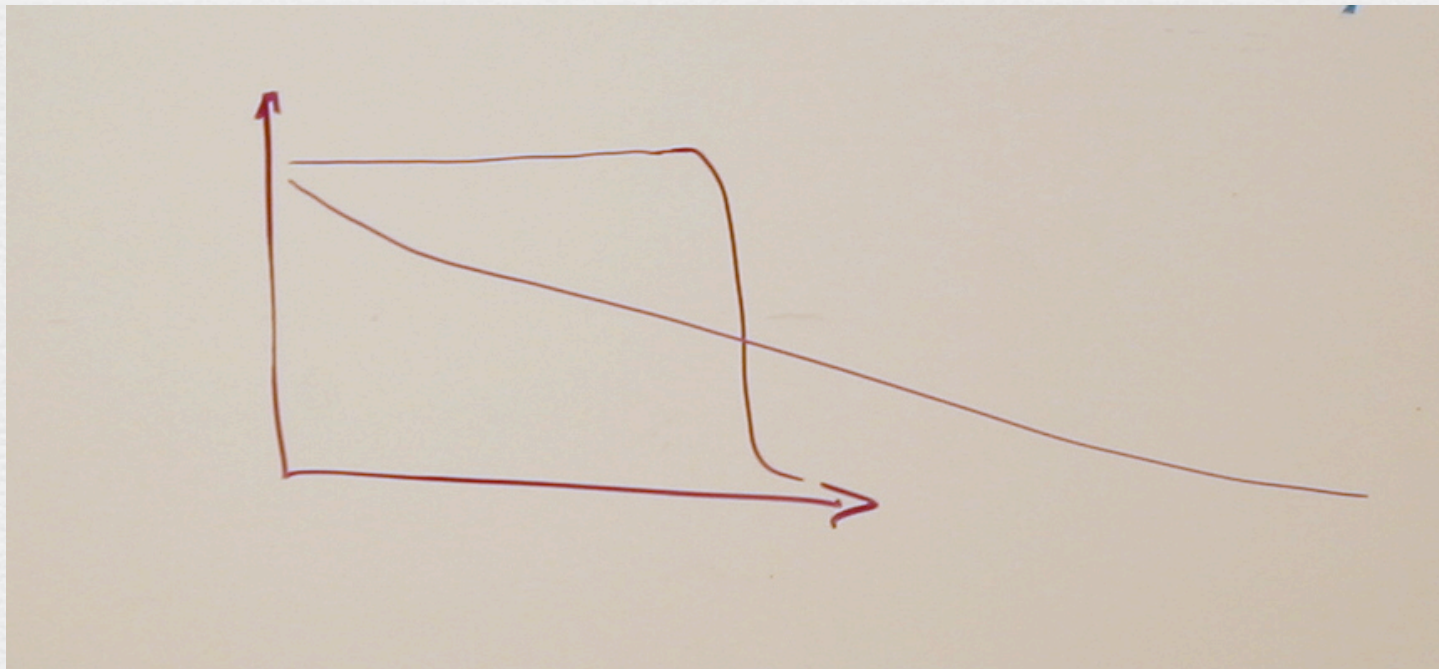
Sharpness versus contrast

f50 = 1e+004 lp/mm; flens = 61 lp/mm; lord = 2

Sine pattern: Original

Sine pattern: Lens only

Bar pattern: Original

Bar pattern: Lens only

Line pairs per mm; MTF = 50%,10% @ 61, 183/mm

(imatest.com)
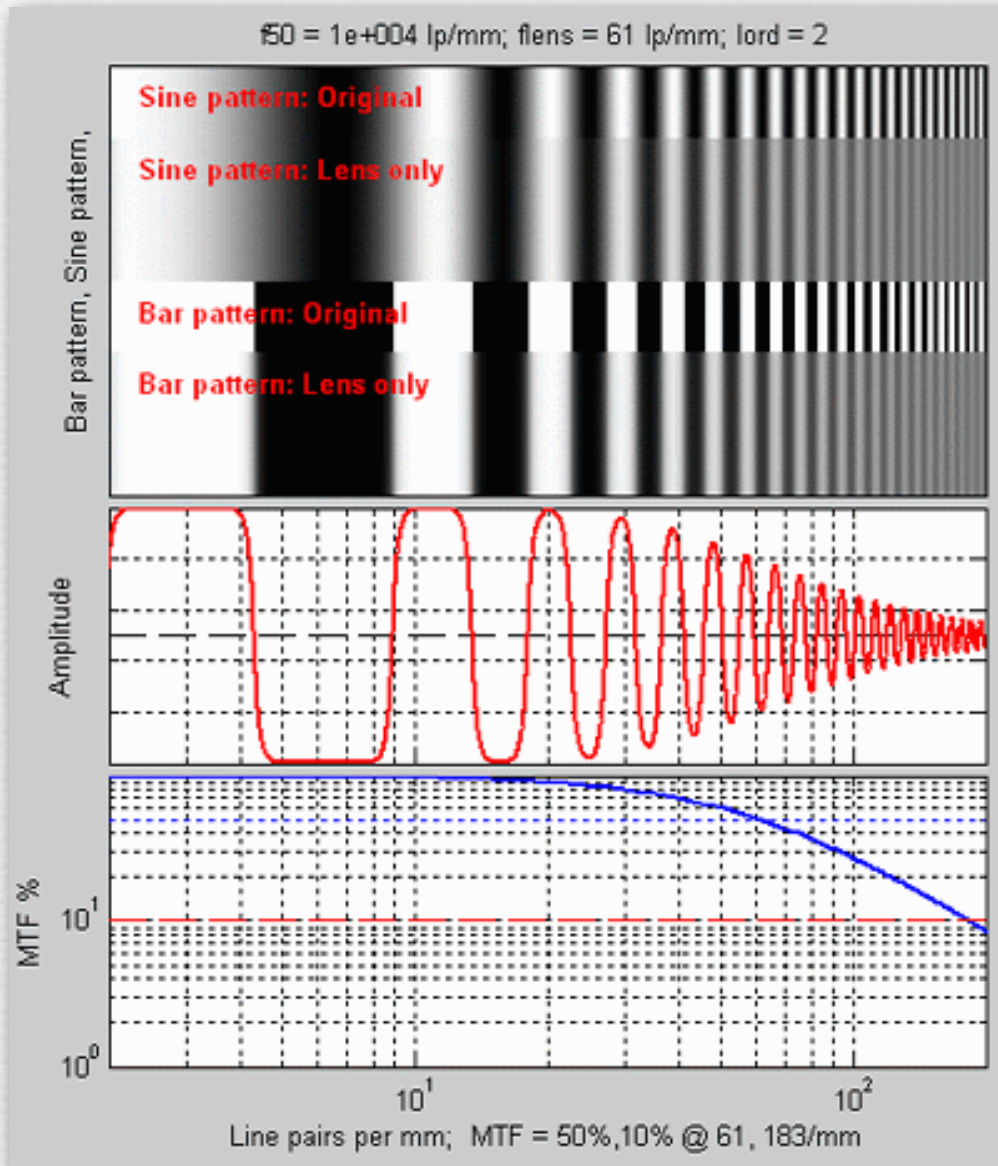
(cambridgeincolour.com)   © Marc Levoy

Restoring images

This slide contained an incorrect set of images when I showed it in class. I have replaced the images here.

poor resolution

(cambridgeincolour.com)

original

poor contrast

In practice, since blurring and loss of contrast both involve attenuation of high frequencies (although to different extents), and restoration involves boosting those attenuated frequencies, then both sharpening and restoration of contrast should produce both some amount of ringing and and some amount of noise enhancement. In these slides the ringing is evident, but not the noise enhancement. We would have to blow the images up to see the noise.

degraded

restored

Recap

✦ any image can be equivalently represented by its Fourier transform, a.k.a. frequency or spectral representation
  - weighted sum of sine and cosine component images
  - each having a frequency, amplitude, and orientation in the plane

✦ filtering, for example blurring or sharpening, can be implemented by amplifying or attenuating selected frequencies
  - i.e. modifying the contrast of selected sine or cosine components relative to others, while maintaining same average over all components
  - attenuating high frequencies ≈ *low-pass-filtering* ≈ blurring
  - attenuating low frequencies ≈ *high-pass filtering* ≈ sharpening

✦ MTF measures preservation of frequencies by an optical system
  - subjective image quality depends on both sharpness and contrast
  - both can be restored, but at a price (in ringing or noise)

Questions?

Spatial resolution of display media

$$\text{pitch} = \Delta x \qquad \text{density} = 1/\Delta x$$

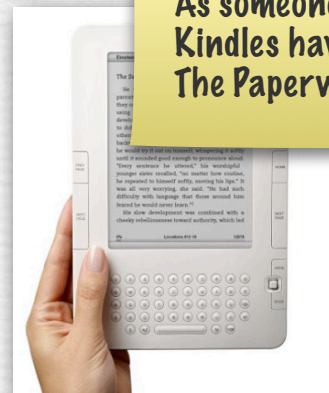- ✦ <u>Example #1</u>: Macbook Air (laptop)
  - 900 pixels on 7" high display
  - $\Delta x = 7"/ 900 \text{ pixels} = 0.0078"/\text{pixel}$
  - $1/\Delta x = 129 \text{ dpi}$ (dots per inch)

Line printers are 300 dpi. This is why we don't like reading on laptops.

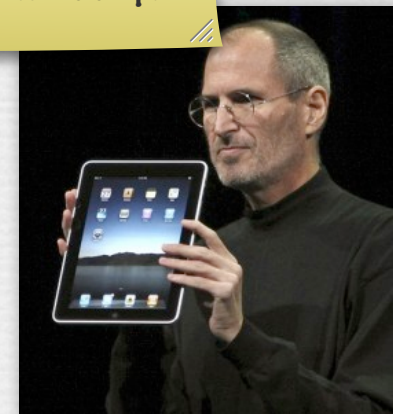- ✦ <u>Example #2</u>: Kindle 2
  - 800 pixels on 4.8" high display
  - $1/\Delta x = 167 \text{ dpi}$

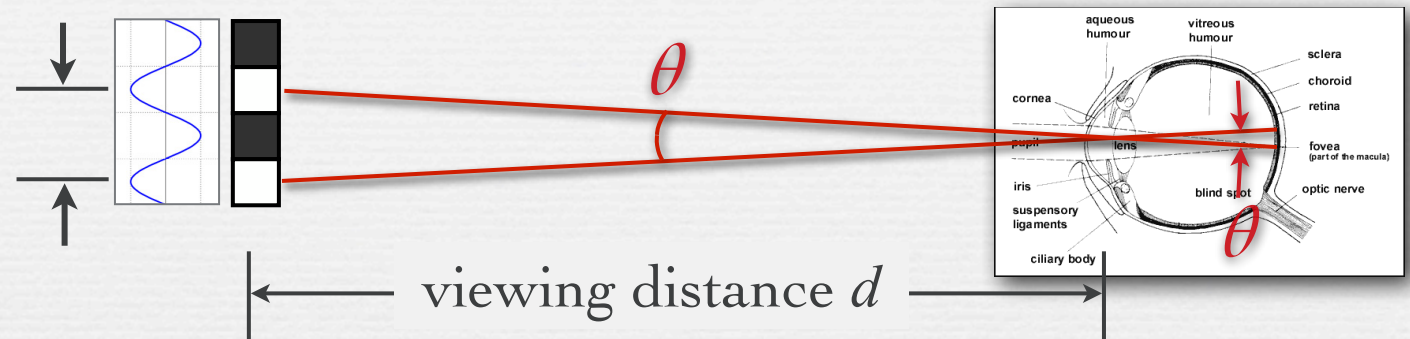As someone pointed out, Kindles have improved. The Paperwhite is 300 dpi.

- ✦ <u>Example #3</u>: iPad ~~iPad3~~ iPad3
  - ~~10~~24 pixels on 7.8" high display 2048
  - $1/\Delta x = 13\cancel{2}$ dpi 263

Spatial frequency on the retina

assume the minimum period p of a sine wave is a black-white pixel pair ("line pair")

θ

viewing distance d

✦ <u>Example #1</u>: Macbook Air viewed at $d = 18$"
- 900 pixels on 7" high display, $p = 2 \times 0.0078$"
- retinal arc $\theta = 2 \arctan(p / 2d) = 0.05°$
- spatial frequency on retina $1/\theta = 20$ cycles per degree

Q. What is the acuity of the human visual system?

Human spatial sensitivity
(Campbell-Robson Chart)

(neurovision.berkeley.edu)
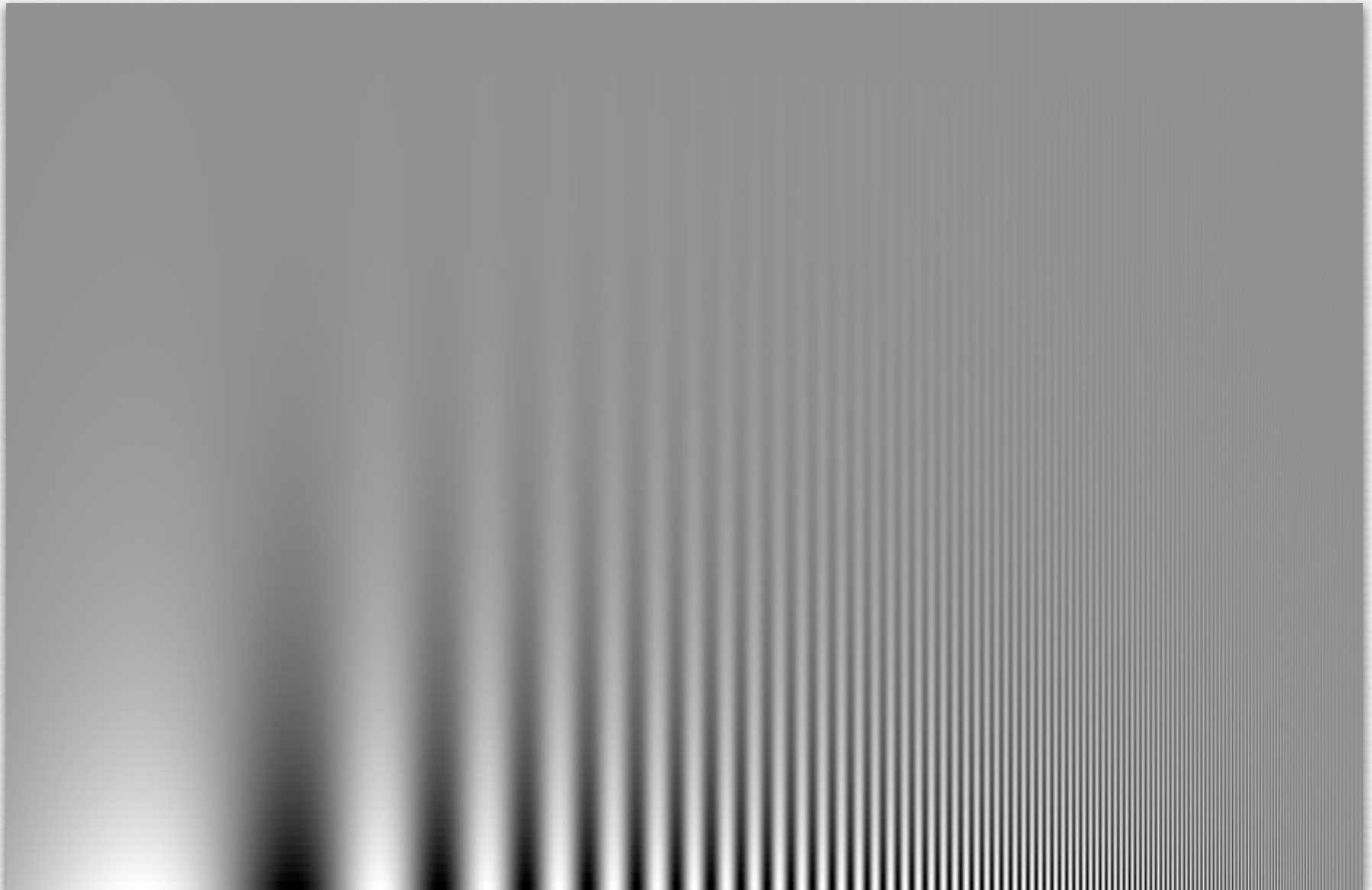
Human spatial sensitivity

(horizontal axis not comparable to image on previous slide)

cutoff is at about 50 cycles per degree

(psych.ndsu.nodak.edu)

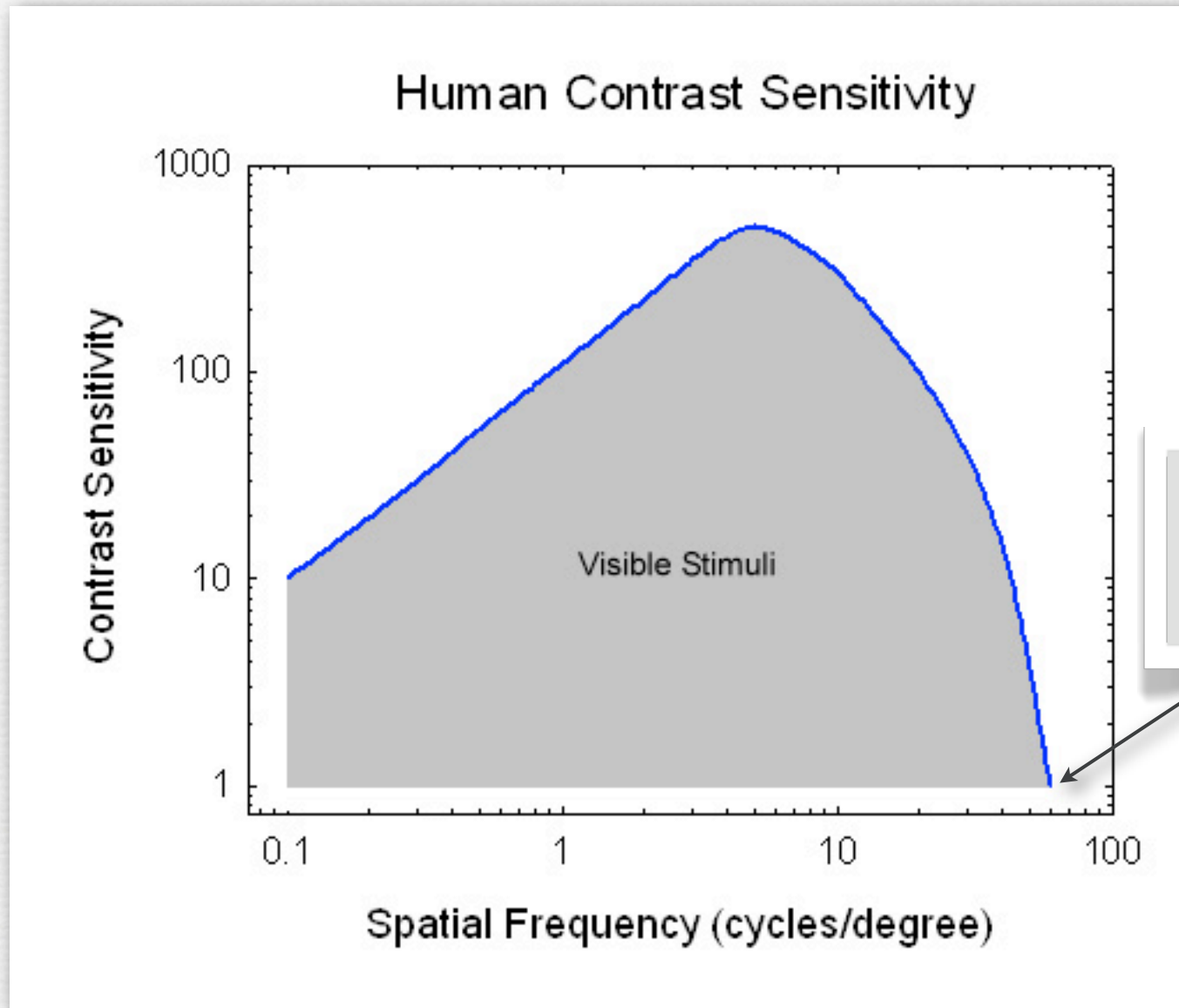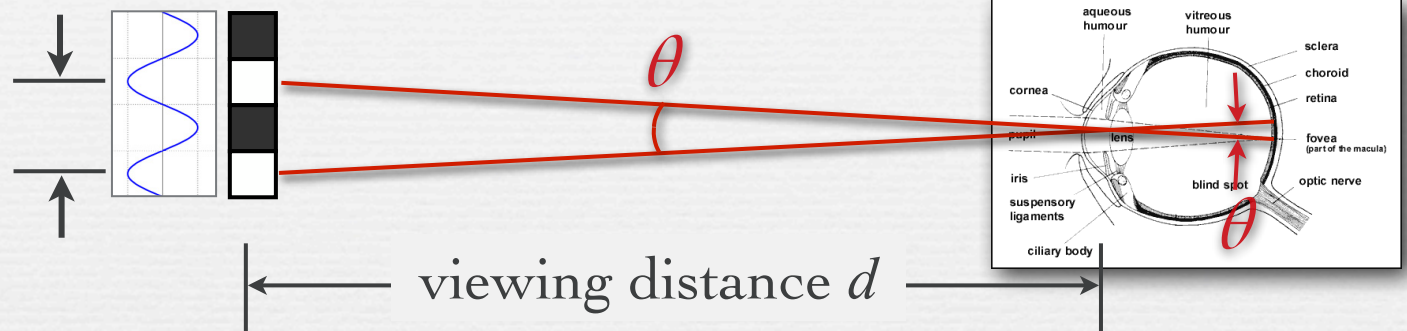# Spatial frequency on the retina

assume the minimum period p of a sine wave is a black-white pixel pair

θ

viewing distance d

✦ Example #1: Macbook Air viewed at $d = 18''$
  • 900 pixels on $7''$ high display, so $p = 2 \times 0.0078''$
  • retinal arc $\theta = 2 \arctan(p / 2d) = 0.05°$
  • spatial frequency on retina $1/\theta = 20$ cycles per degree
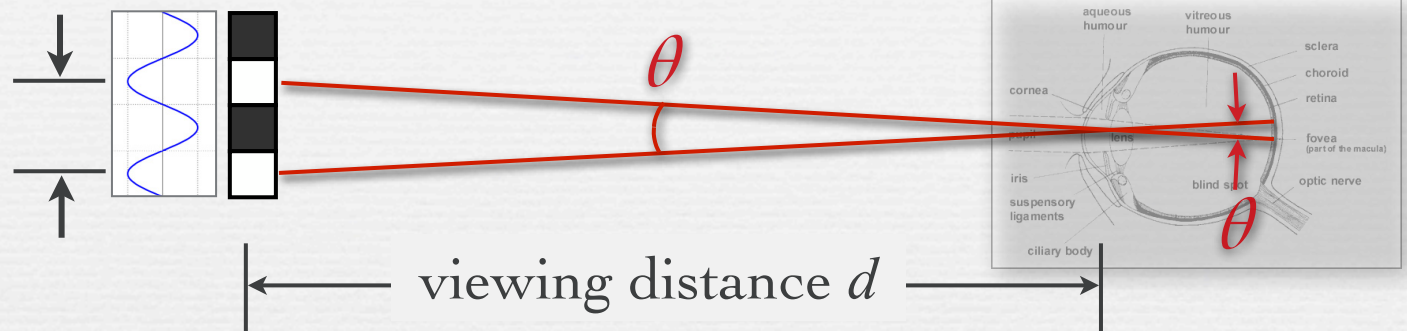
not nearly as high as human acuity

(Graham Flint)

Balboa Park, San Diego

(original is 40K × 20K pixels, Gates Hall print is 72" × 36")

Spatial frequency on the retina

assume the minimum
period *p* of a sine
wave is a black-white
pixel pair

θ

viewing distance *d*

- ✦ <u>Example #1</u>: Macbook Air viewed at *d = 18"*
  - 900 pixels on 7" high display, *p = 2 × 0.0078"*
  - retinal arc *θ = 2 arctan (p / 2d) = 0.05°*
  - spatial frequency on retina *1/θ = 20* cycles per degree

- ✦ <u>Example #2</u>: gigapixel photo viewed at *d = 48"*
  - 20,000 pixels on 36" high print, *p = 2 × 0.0018"*
  - spatial frequency on retina *1/θ = 232* cycles per degree

much finer than
human acuity

Spatial frequency on the retina

assume the minimum
period p of a sine
wave is a black-white
pixel pair

θ

viewing distance d

θ
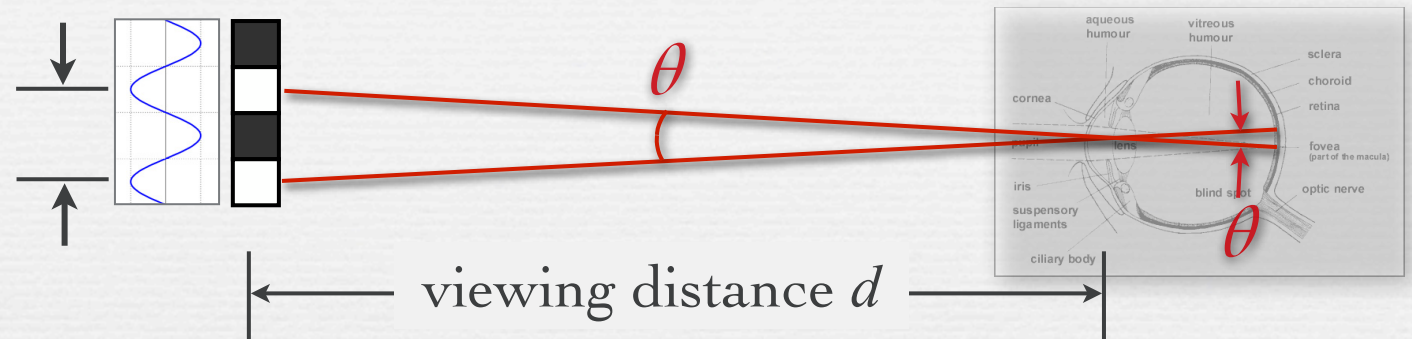
✦ <u>Example #2</u>: gigapixel photo viewed at $d = 48$"
  - 20,000 pixels on 36" high print, $p = 2 \times 0.0018$"
  - spatial frequency on retina $1/\theta = 232$ cycles per degree

✦ <u>Example #3</u>: iPad Pro
  - 2048 pixels on 7.8" high display, $p = 2 \times 0.0038$"
  - $1/\Delta x = 263$ dpi
  - if viewed at 18", spatial frequency on retina $= 41$ cycles per degree

almost retina
resolution

© Marc Levoy

Human acuity & circle of confusion

- ✦ the maximum allowable circle of confusion (C) in a photograph can be computed from human spatial acuity projected onto the intended display medium
  - depends on viewing distance

  As someone pointed out, a good rule of thumb is to assume that the viewing distance is 3 x the picture height. Obviously this will depend on aspect ratio.

- ✦ <u>Example</u>: photographic print viewed at 12"
  - max human acuity on retina $1/\theta \approx 50$ cycles per degree
  - minimum detectable retinal arc $\theta \approx 0.02^o$
  - minimum feature size $p = 2 \times 12" \times tan\,(\theta\,/\,2) = 0.0042"$ (0.1mm)

- ✦ assume $5" \times 7"$ print and Canon 5D III (5660 × 3840 pixels)
  - 5" / 3840 pixels = 0.0013"/pixel (0.03mm)
  - therefore, blur can be 3.2 pixels wide before print appears blurry
  - $C = 6.3\mu$ per pixel on camera × 3.2 pixels = 20.2μ

Recap

✦ spatial resolution of display media is measured by
  - pitch (distance between dots or pixels) or density (dots per inch)

✦ effect on human observers is measured by
  - retinal angle (degrees of arc) or frequency (cycles per degree)
  - depends on viewing distance

✦ human spatial acuity is about 50 cycles per degree
  - depends on contrast
  - convert back to pitch to obtain circle of confusion for depth of field, and this conversion depends on viewing distance
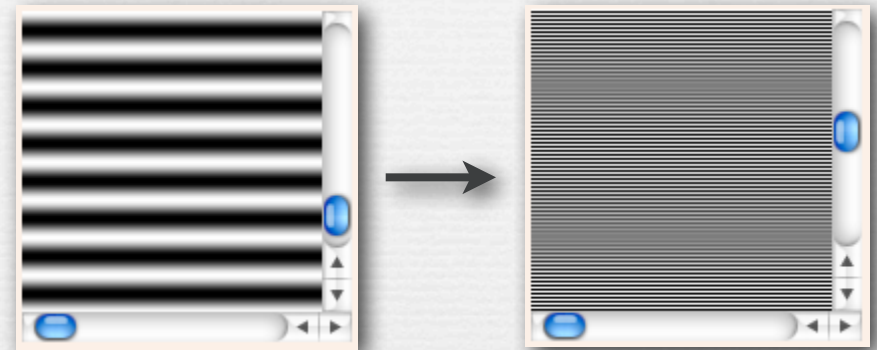
Questions?

Sampling and aliasing
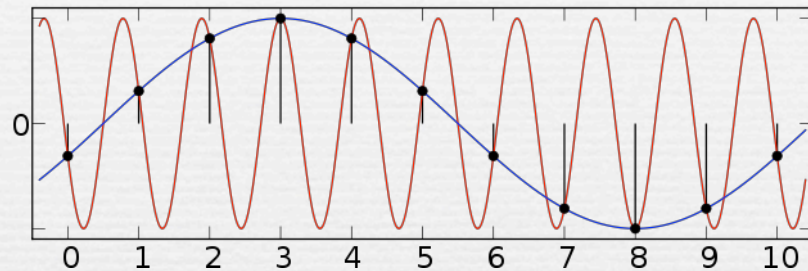
abstract function
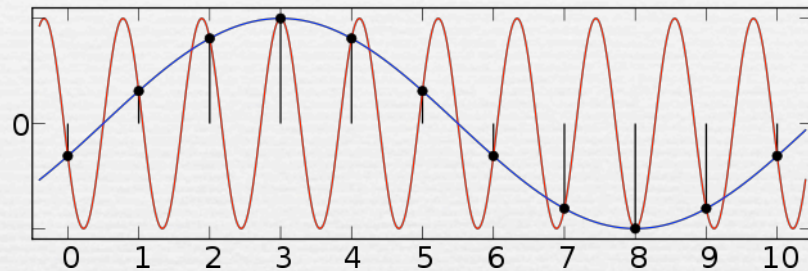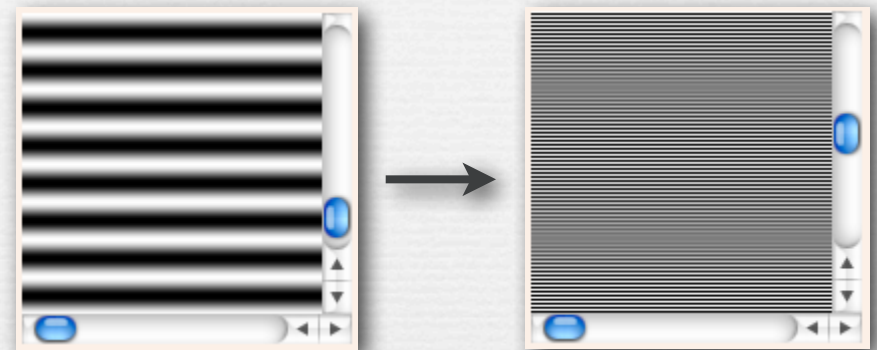

spatial aliasing in images

✦ aliasing is high frequencies masquerading as low frequencies due to insufficiently closely spaced samples

Sampling and aliasing
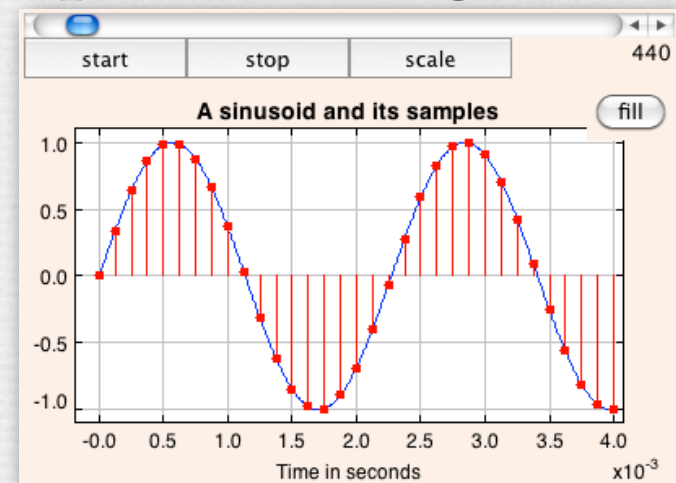
abstract function

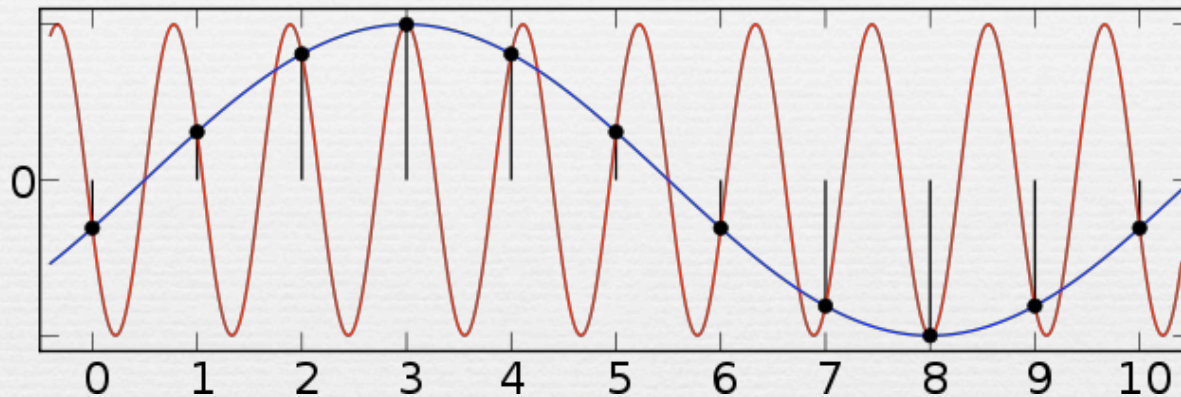## spatial aliasing in images

temporal aliasing

240 rpm

©2007 M. Bach

temporal aliasing in audio

start   stop   scale   440

A sinusoid and its samples   fill

Time in seconds   x10^{-3}

Fourier analysis of aliasing

- ✦ <u>Nyquist-Shannon sampling theorem:</u> a function having frequencies no higher than *n* can be completely determined by samples spaced *1 / 2n* apart

$$f_{sampling} > 2 \times f_{cutoff}$$

Retinal sampling rate

✦ the human retina consists of discrete sensing cells

✦ therefore, the retina performs sampling

✦ sampling theory says $f_{sampling} > 2 \times f_{cutoff}$

✦ if observed human cutoff is 50 cycles per degree,
then its sampling rate must be > 100 samples per degree

✦ this agrees with observed retinal cell spacing!
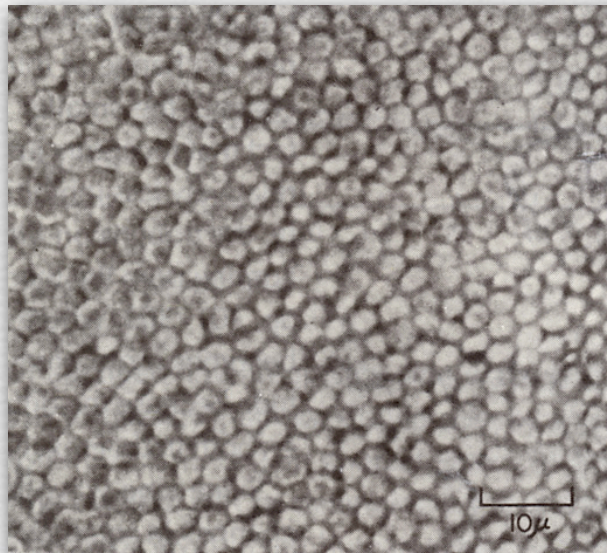
(Cornsweet)

spacing between L,M cone cells is 1μ ≈ 30 arc-seconds (1/120°)

Vernier acuity (a.k.a. hyperacuity)

✦ we can detect jaggies as small as 5 seconds of retinal arc (1/720°)

- even though our cells are spaced 30 seconds apart

- to make such jaggies invisible, the iPad Pro display would need to be 15K pixels vertically !!

I didn't show this slide during lecture, but I discussed it, so I might as well include the slide here.

Aliasing in photography

✦ a lens creates a focused image on the sensor

✦ suppose the sensor measured this image at points on a 2D grid, but ignored the imagery between points?
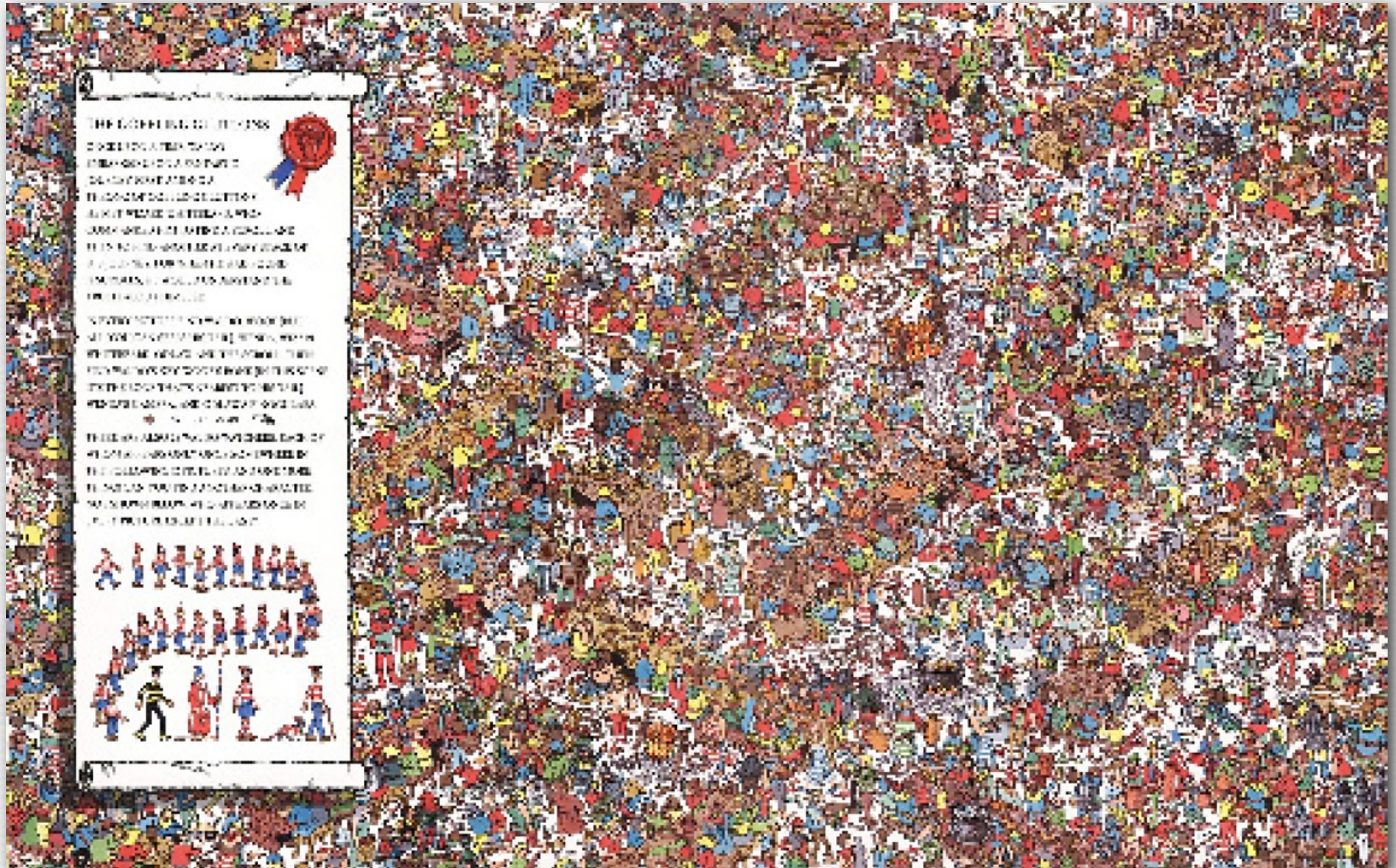 - a.k.a. *point sampling*

Simulation of point sampling
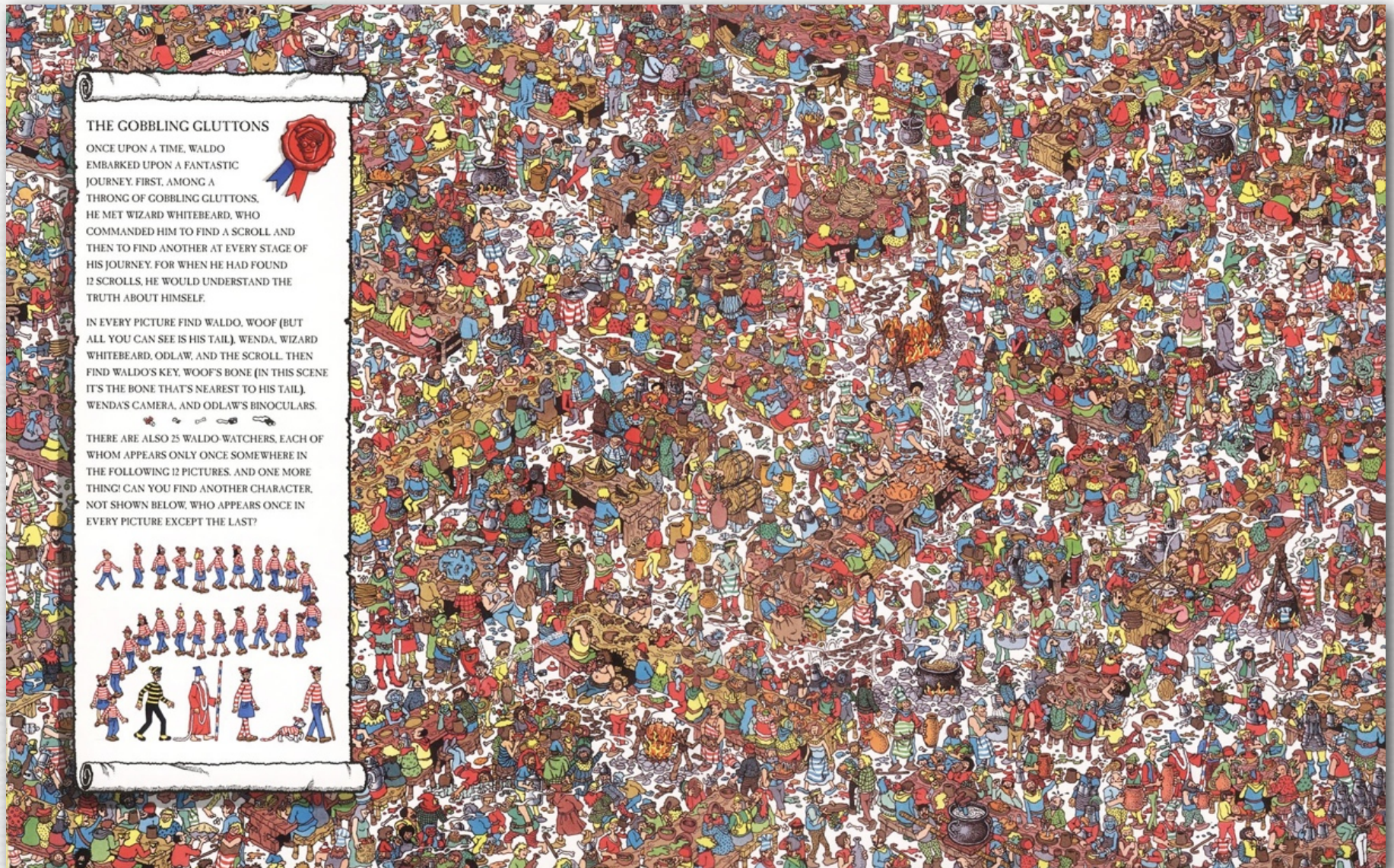
digital image, 1976 x 1240 pixels
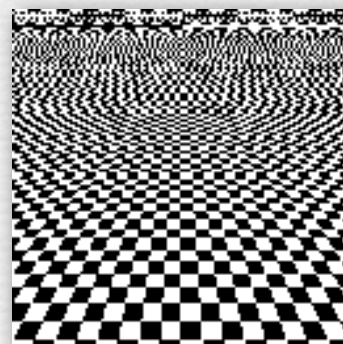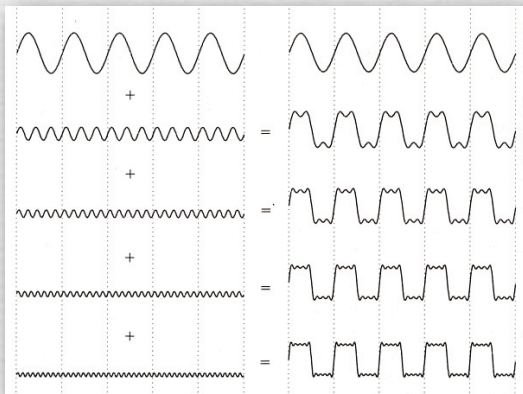
Simulation of point sampling

every 4th pixel in *x* and *y,* then upsized using pixel replication

Prefiltering to avoid aliasing
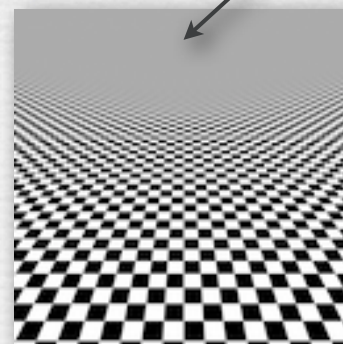
- ✦ before sampling, remove (or at least attenuate) sine waves of frequency greater than half the sampling rate

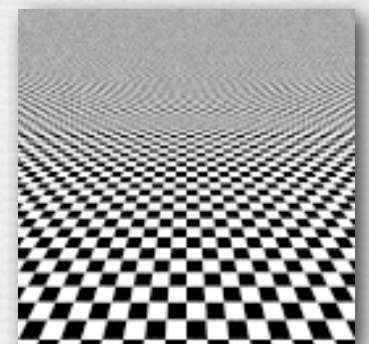$$f_{cutoff} \; < \; \frac{1}{2} f_{sampling}$$

replace removed waves with their average intensity (gray in this case)

unfiltered   prefiltered   partially pre-filtered

© Marc Levoy

Methods for prefiltering

✦ <u>method #1</u>: frequency domain

    1. convert image to frequency domain

    2. remove frequencies above f_{cutoff} (replace with gray)

    3. convert back to spatial domain

    4. perform point sampling as before

    • conversions are slow

    • not clear how to apply this method to images as they enter a camera

✦ <u>method #2</u>: spatial domain

    1. blur image using *convolution*

    2. perform point sampling as before

    • direct and faster

    • equivalent to method #1 (proof is beyond scope of this course)

Convolution in 1D

✦ replace each input value with a weighted sum of itself and its neighbors, with weights given by a filter function

$$f[x] * g[x] \quad = \quad \sum_{k=-\infty}^{\infty} f[k] \cdot g[x-k]$$

input signal $f[x]$

1	3	0	4	2	1

filter g[x]

2	1

output $f[x] * g[x]$

Convolution in 1D

✦ replace each input value with a weighted sum of itself and its neighbors, with weights given by a filter function

$$f[x] * g[x] \ = \ \sum_{k=-\infty}^{\infty} f[k] \cdot g[x-k]$$

input signal $f[x]$

1	3	0	4	2	1

1	2

notice that the filter gets flipped when applied

output $f[x] * g[x]$

7					

Convolution in 1D

✦ replace each input value with a weighted sum of itself and its neighbors, with weights given by a filter function

$$f[x] * g[x] = \sum_{k=-\infty}^{\infty} f[k] \cdot g[x-k]$$

input signal $f[x]$

1	3	0	4	2	1

1	2

output $f[x] * g[x]$

7	3				

Convolution in 1D

✦ replace each input value with a weighted sum of itself and its neighbors, with weights given by a filter function

$$f[x] * g[x] \;=\; \sum_{k=-\infty}^{\infty} f[k] \cdot g[x-k]$$

input signal $f[x]$

1	3	0	4	2	1

1	2

output $f[x] * g[x]$

7	3	8			

More convolution formulae

- ✦ <u>1D discrete</u>: defined only on the integers

$$f[x] * g[x] \quad = \quad \sum_{k=-\infty}^{\infty} f[k] \cdot g[x-k]$$

- ✦ <u>1D continuous</u>: defined on the real line

$$f(x) * g(x) \quad = \quad \int_{-\infty}^{\infty} f(\tau) \cdot g(x-\tau)\, d\tau$$

(FLASH DEMO)

https://sites.google.com/a/google.com/
digital-photography/applets/spatial-
convolution

More convolution formulae

- ✦ <u>1D discrete</u>: defined only on the integers

$$f[x] * g[x] \;=\; \sum_{k=-\infty}^{\infty} f[k] \cdot g[x-k]$$

- ✦ <u>1D continuous</u>: defined on the real line

$$f(x) * g(x) \;=\; \int_{-\infty}^{\infty} f(\tau) \cdot g(x-\tau)\, d\tau$$

- ✦ <u>2D discrete</u>: defined on the x, y integer grid

$$f[x,y] * g[x,y] \;=\; \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f[i,j] \cdot g[x-i, y-j]$$
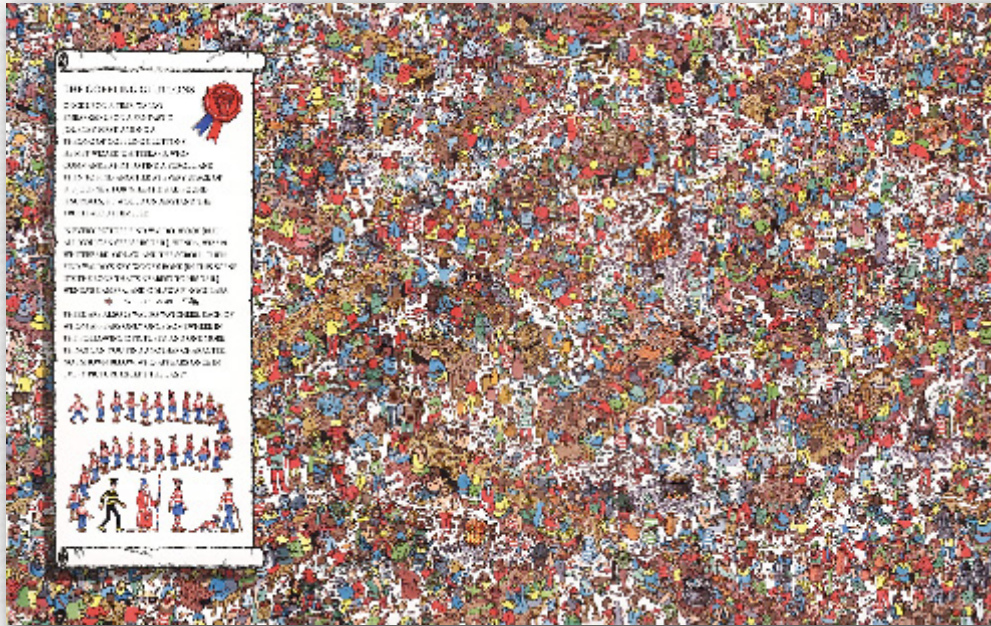
- ✦ <u>2D continuous</u>: defined on the x,y plane

$$f(x,y) * g(x,y) \;=\; \int_{\tau_1=-\infty}^{\infty} \int_{\tau_2=-\infty}^{\infty} f(\tau_1, \tau_2) \cdot g(x-\tau_1, y-\tau_2)\, d\tau_1\, d\tau_2$$
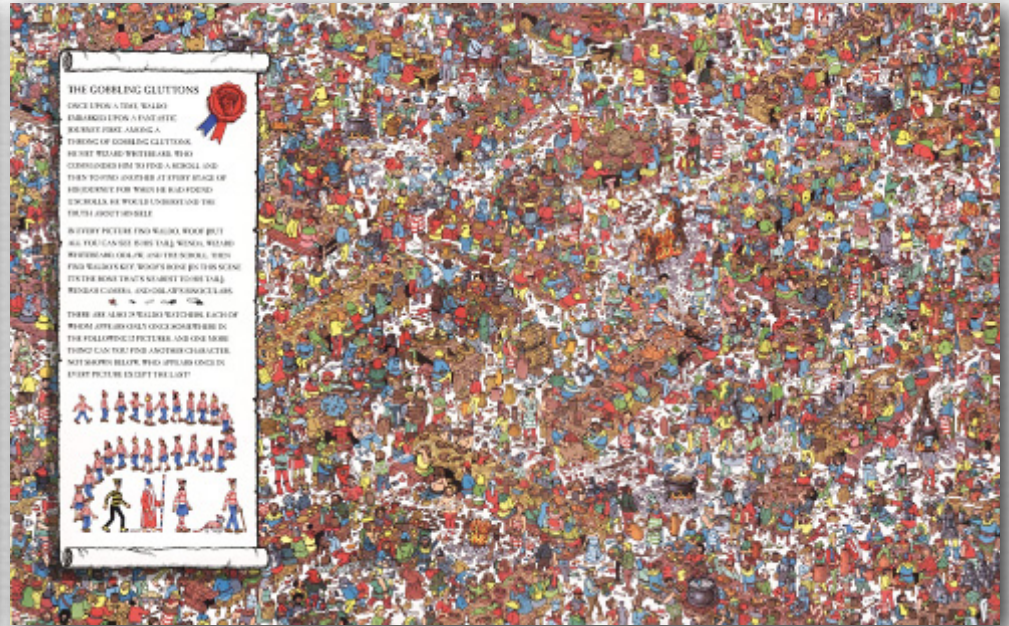
To simulate astigmatism, select "big rect", then negate the leftmost two columns and rightmost two columns (i.e. -0.04) and change the middle column to +0.36. I muffed this demo during lecture.

Prefiltering reduces aliasing

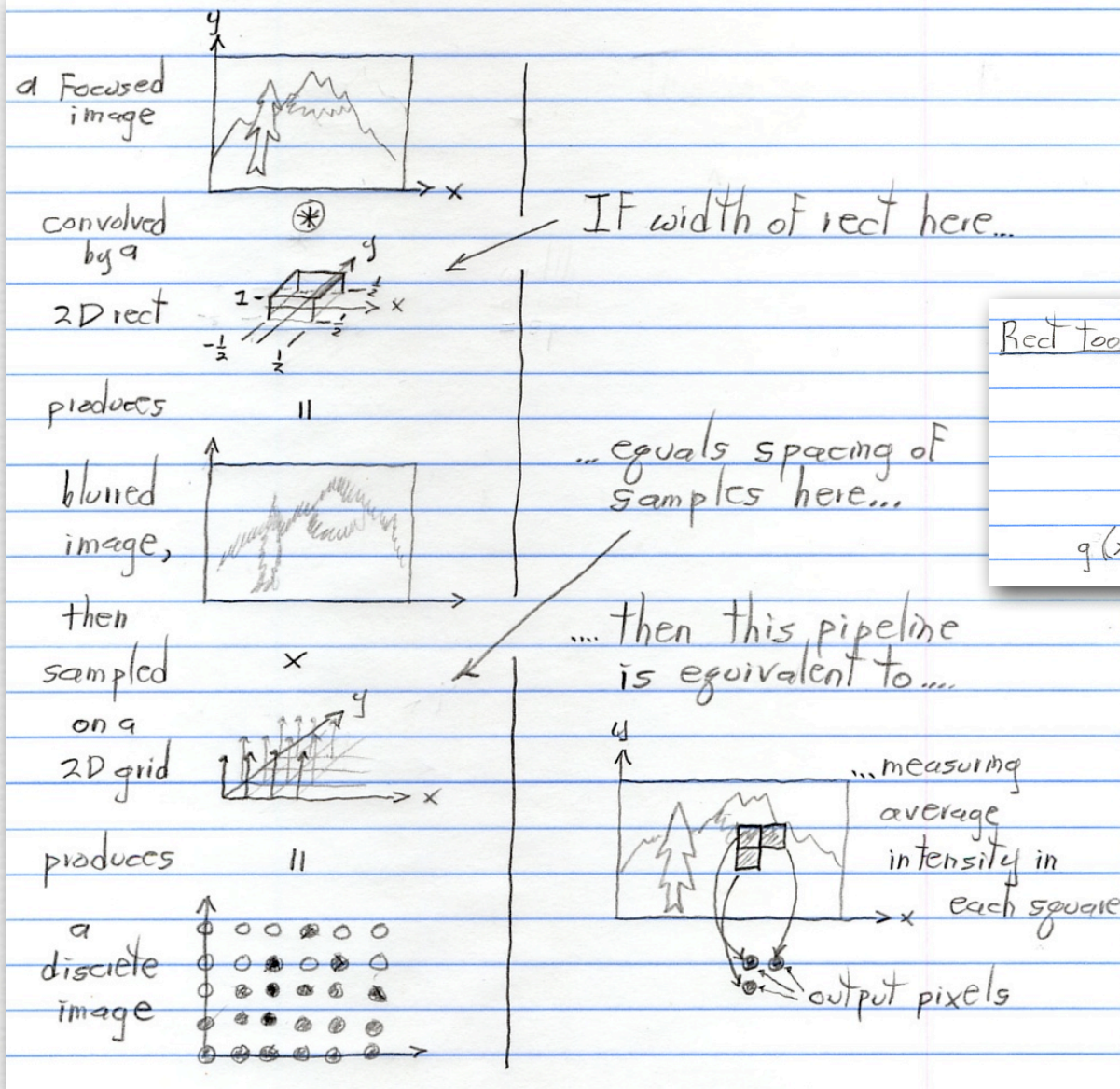every 4th pixel in x and y

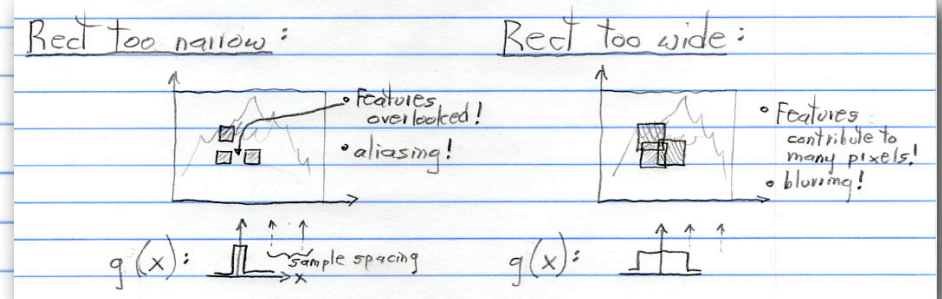convolved by 4×4 pixel rect, then sampled every 4th pixel

Prefiltering & sampling in photography

✦ photography consists of convolving the focused image by a 2D rect filter, then sampling on a 2D grid
- each point on this grid is called a *pixel*

✦ if convolution is followed by sampling, you only need to compute the convolution at the sample positions
- for a rect filter of width equal to the sample spacing, this is equivalent to measuring the average intensity of the focused image in a grid of abutting squares
- this is exactly what a digital camera does

✦ the rect width <u>should</u> roughly match the pixel spacing
- much narrower would leave aliasing in the image
- much wider would produce excessive blurring in the image

Prefiltering & sampling in photography (contents of whiteboard)

Upsizing/downsizing in Photoshop

✦ *resampling* is the conversion of a discrete image into a second discrete image having more or fewer samples
  1. interpolate between samples using convolution
  2. if downsizing, blur to remove high frequencies
  3. point sample at the new rate

  • these steps can be simplified into a single discrete convolution

As I mentioned briefly during lecture, steps 1 and 2 are both convolutions, and convolution is associative. Thus, (f ⊛ r) ⊛ g ≡ f ⊛ (r ⊛ g), where f is the input image, r is the reconstruction filter (see next slide), and g is a blurring filter (such as are shown in the online convolution applet). This equation says that the two filters r and g can be convolved with each other, thereby producing a single filter, sometimes called the resampling filter ρ (rho), which is larger in non-zero extent than either r or g, and that can be applied to the input image, followed by point sampling at the new rate.

Interpolation via convolution (contents of whiteboard)
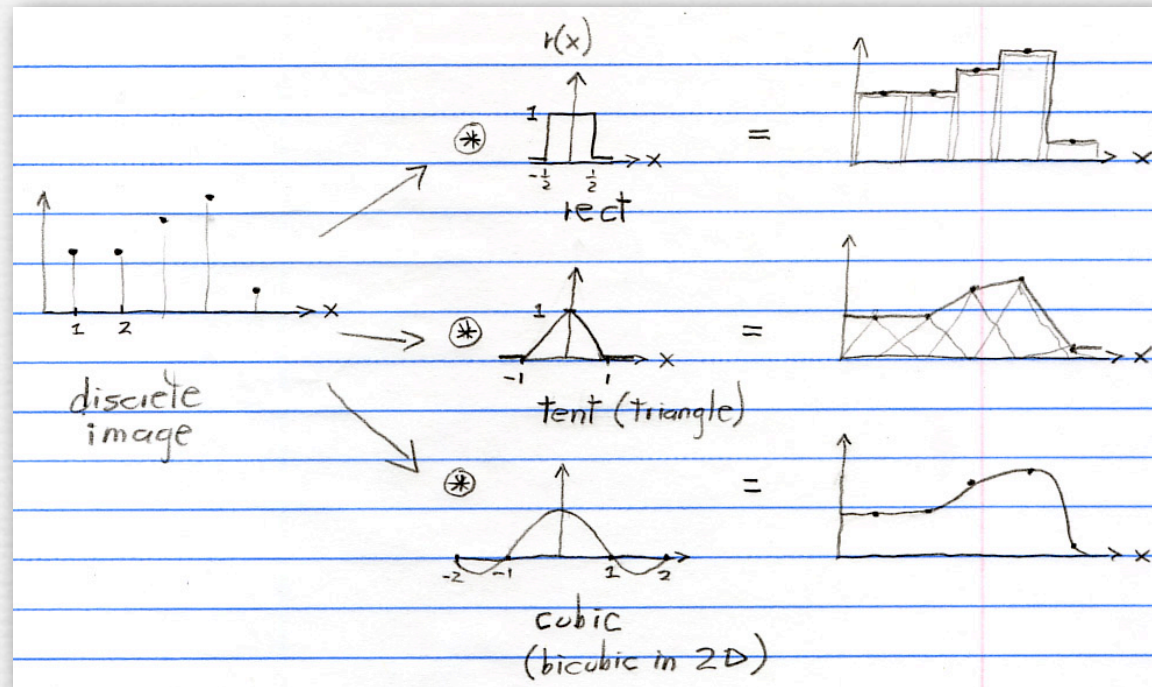
- ✦ if the input is a discrete (i.e. sampled) function, then convolution can be treated as placing a vertically-scaled copy of the filter *r(x)* at each sample position as shown, summing the results, and dividing by the area under the filter (1.0 in the cases shown)

- ✦ the effect is to interpolate between the samples, hence reconstructing a continuous function from the discrete function
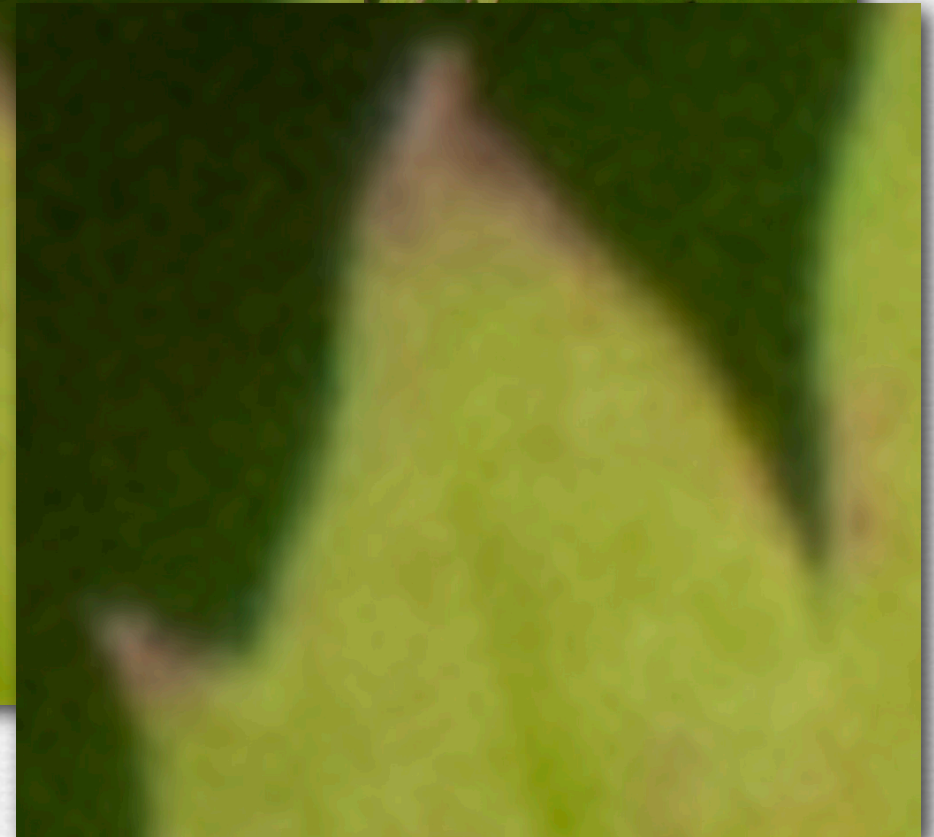
Upsizing by 16:1
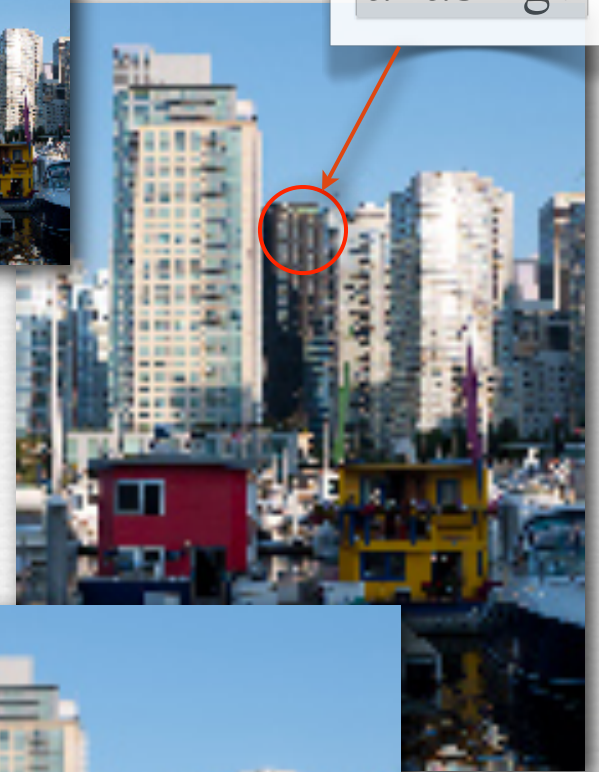
nearest neighbor
(a.k.a. pixel replication)

bilinear

bicubic

Downsizing by 1:6
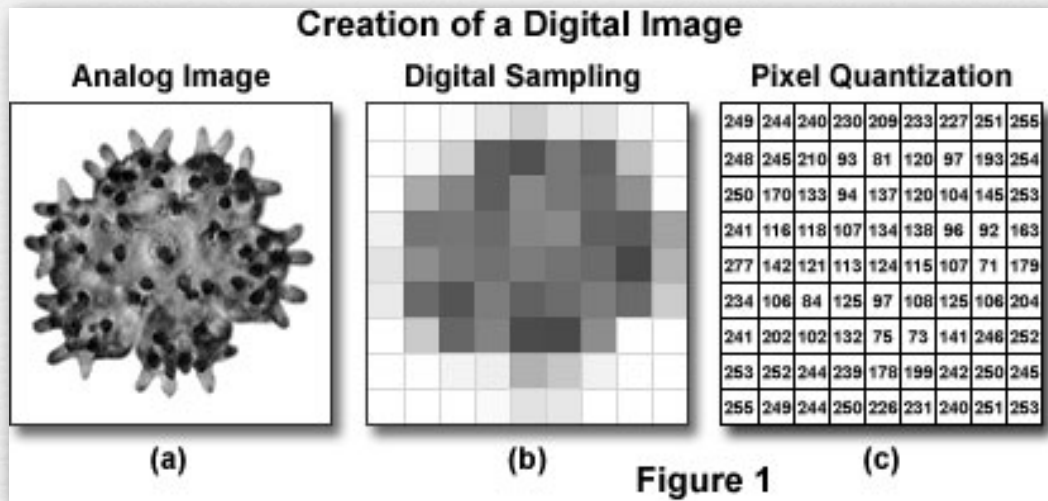
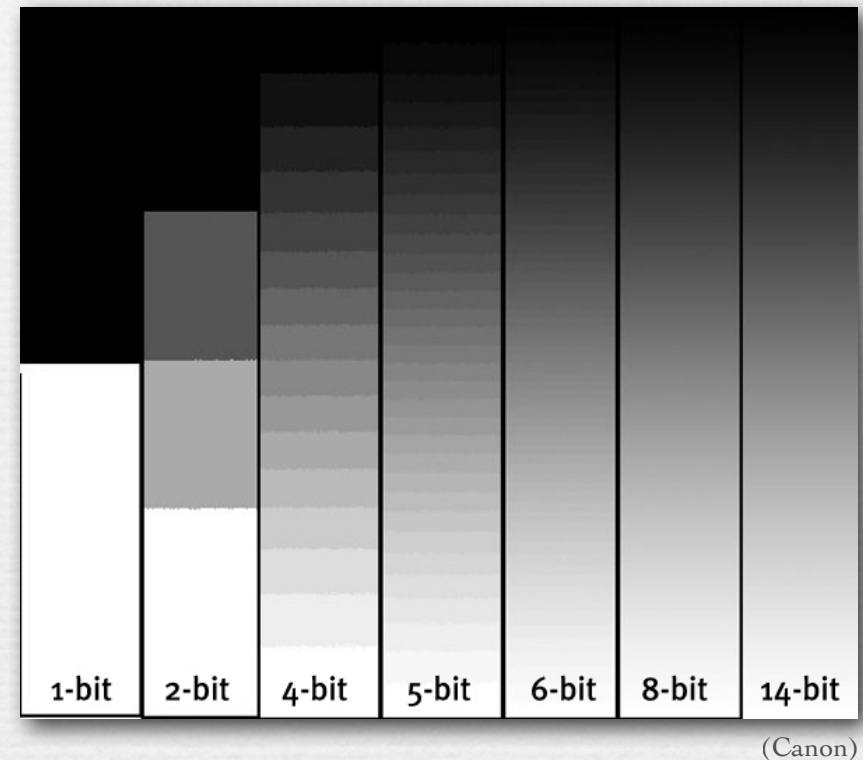aliasing!

nearest neighbor
(point sampling)

bicubic

Recap

✦ *aliasing* is high frequencies masquerading as low frequencies due to insufficiently closely spaced samples

✦ reduce aliasing by prefiltering the input before sampling
   - implement by multiplication in the frequency domain
   - or convolution in the spatial domain
   - in the spatial domain, the prefilter is denoted *g(x)*

✦ in digital photography:
   - *g(x)* is a pixel-sized rect, thus averaging intensity over areas
   - if the rect is too small, aliasing occurs; solve with antialiasing filter

Questions?

Sampling versus quantization

Creation of a Digital Image

Analog Image — Digital Sampling — Pixel Quantization

(a) (b) (c) Figure 1

(http://learn.hamamatsu.com/articles/digitalimagebasics.html)

1-bit 2-bit 4-bit 5-bit 6-bit 8-bit 14-bit

(Canon)

- ✦ an image is a function $\vec{f}(\vec{x})$
  - typically $(\vec{x}) = (x, y)$ and $\vec{f} = (R, G, B)$
- ✦ we sample the domain $(\vec{x})$ of this function as pixels
- ✦ we quantize the range $\vec{f}$ of this function as intensity levels

Example

Canon 1D III,
300mm, f/3.2

Example

© Marc Levoy

Example

Example

Example

59

Example

60

© Marc Levoy

Dithering

Dithering

62

© Marc Levoy

Dithering versus halftoning

- *dithering* for display (on a screen)
  - palette of a few hundred colors (uniform or adaptive)
  - flip some pixels in each neighborhood to the next available color in the palette to approximate intermediate colors when viewed from a distance

- *halftoning* for printing (on paper)
  - palette of only 3 or 4 colors (primaries)
  - print each primary as a grid of dots, superimposed but slightly offset from the other primaries, and vary dot size locally to approximate intermediate colors

- both techniques are applicable to full-color or black and white imagery

- both trade off spatial resolution to obtain more colors, hence to avoid quantization (contouring)
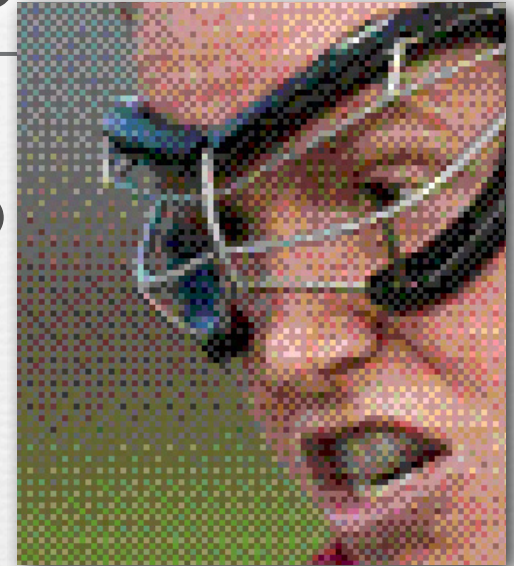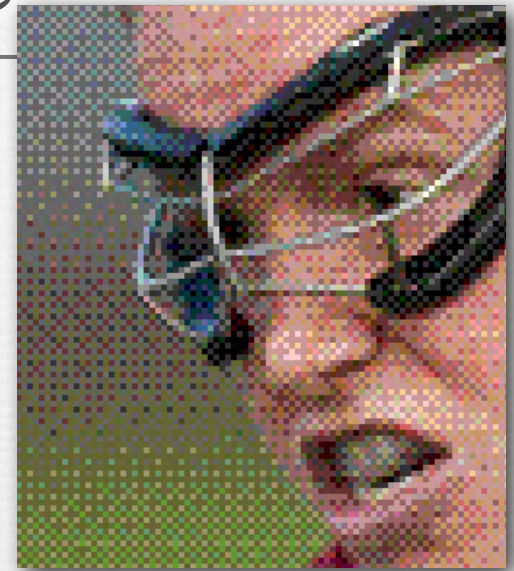

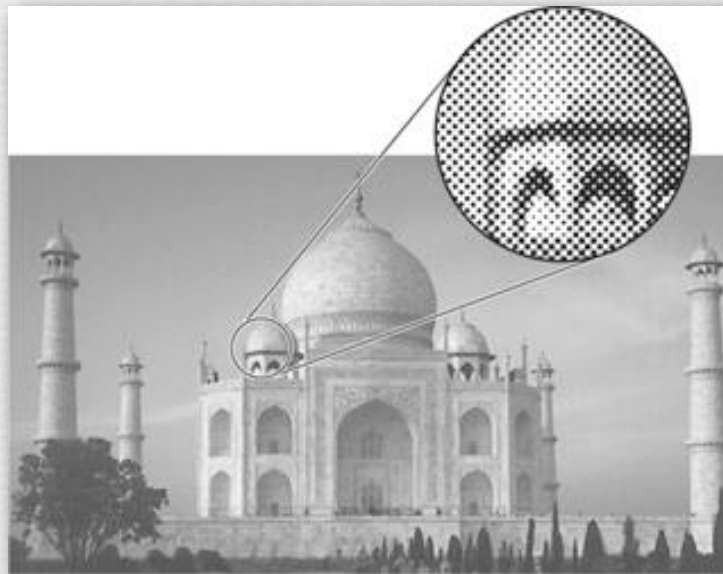(wikimedia)

Dithering versus halftoning

binary dithering

grayscale dithering

color dithering

(see http://bisqwit.iki.fi/jutut/colorquant/ for more examples)

grayscale halftoning

color halftoning

© Marc Levoy

Recap

- ✦ *sampling* describes where in its domain you measure a function
    - for uniformly spaced samples, you can specify a *sampling rate*
    - if the sampling rate is too low, you might suffer from *aliasing*
    - you can reduce aliasing by *prefiltering*

- ✦ *quantization* describes how you represent these measurements
    - for uniformly spaced levels, you can specify a *bit depth*
    - if the bit depth is too low, you might suffer from *contouring*
    - you can reduce contouring by *dithering* (if displaying the image on a screen) or *halftoning* (if printing it on paper)

Questions?

Slide credits

✦ Pat Hanrahan

✦   Cornsweet, T.N., *Visual Perception,* Kluwer Academic Press, 1970.