Sampling and pixels

CS 178, Spring 2010

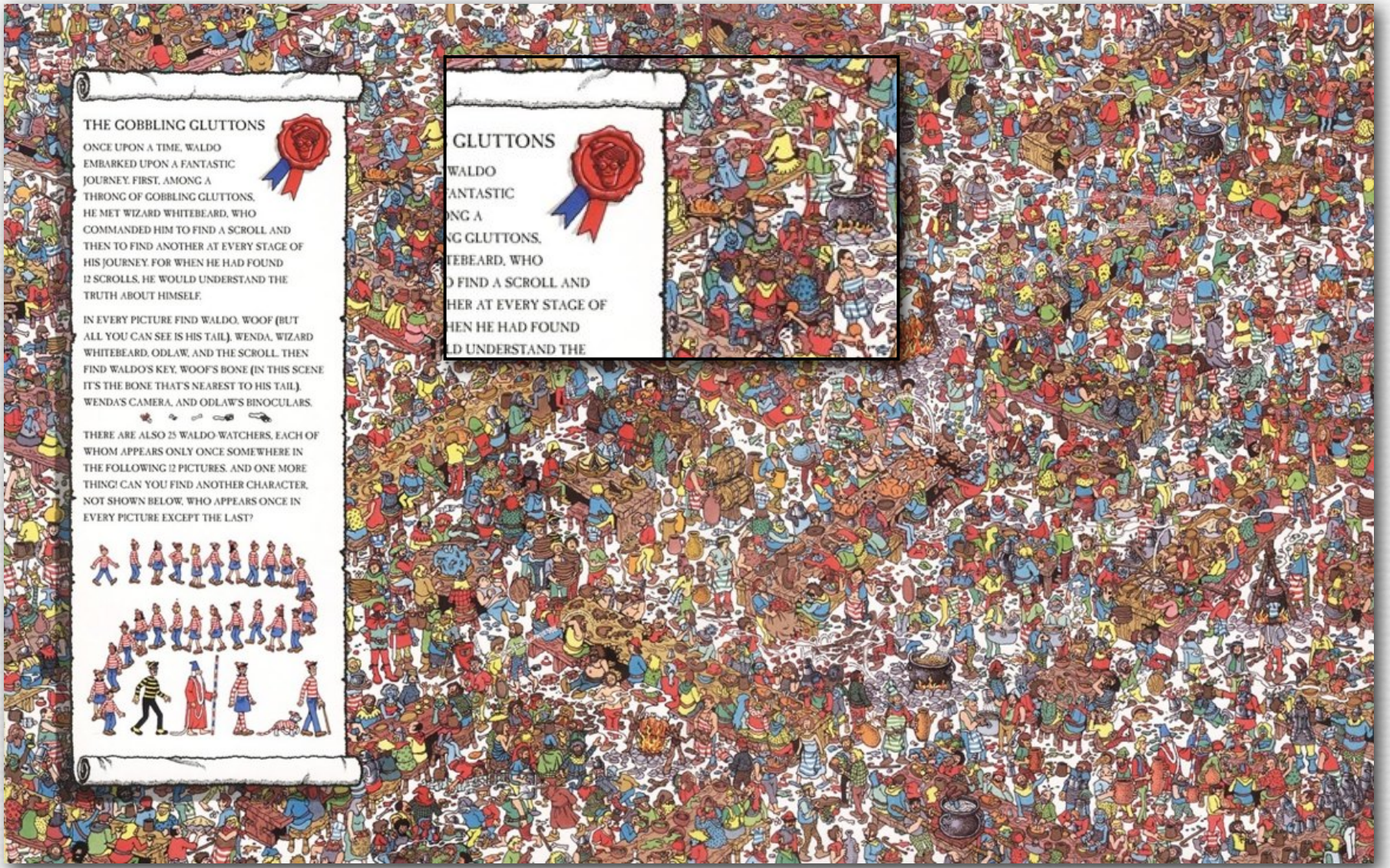
Marc Levoy
Computer Science Department
Stanford University

Outline

- sampling and aliasing
- solutions to aliasing
  - raise the sampling rate
  - prefilter using convolution
- sampling and human perception
  - the "sampling rates" of typical display media
  - the acuity of the human visual system
  - the right way to compute circle of confusion (*C*)
- prefiltering versus postfiltering versus reconstruction
- display and printing technologies

© Marc Levoy

Point sampling a scene



(Classic Media)

(original image is 1976 × 1240 pixels)

© Marc Levoy

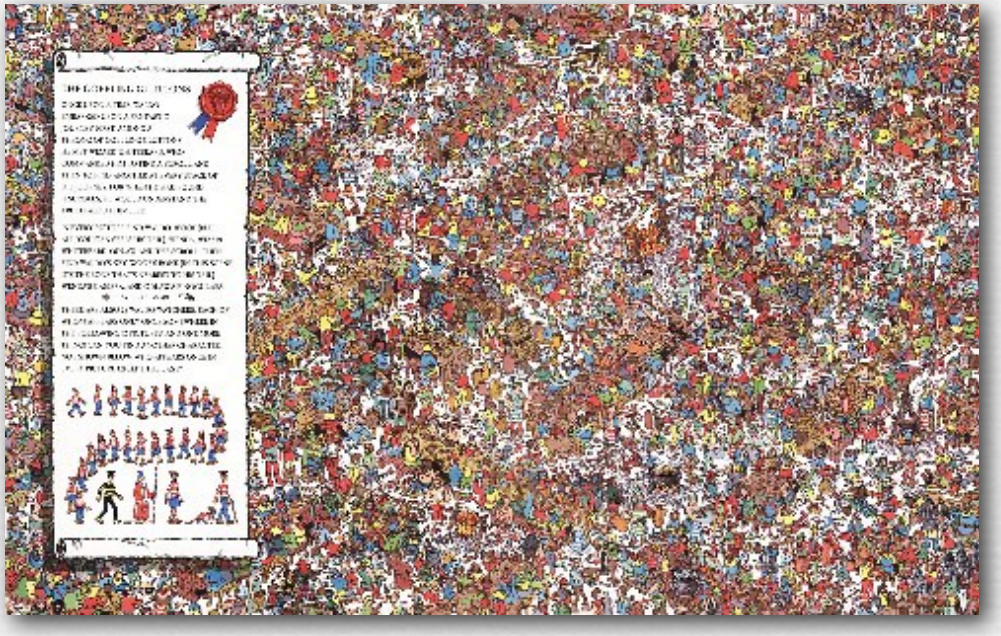
Point sampling a scene

(Classic Media)

(original image is 1976 × 1240 pixels)

© Marc Levoy

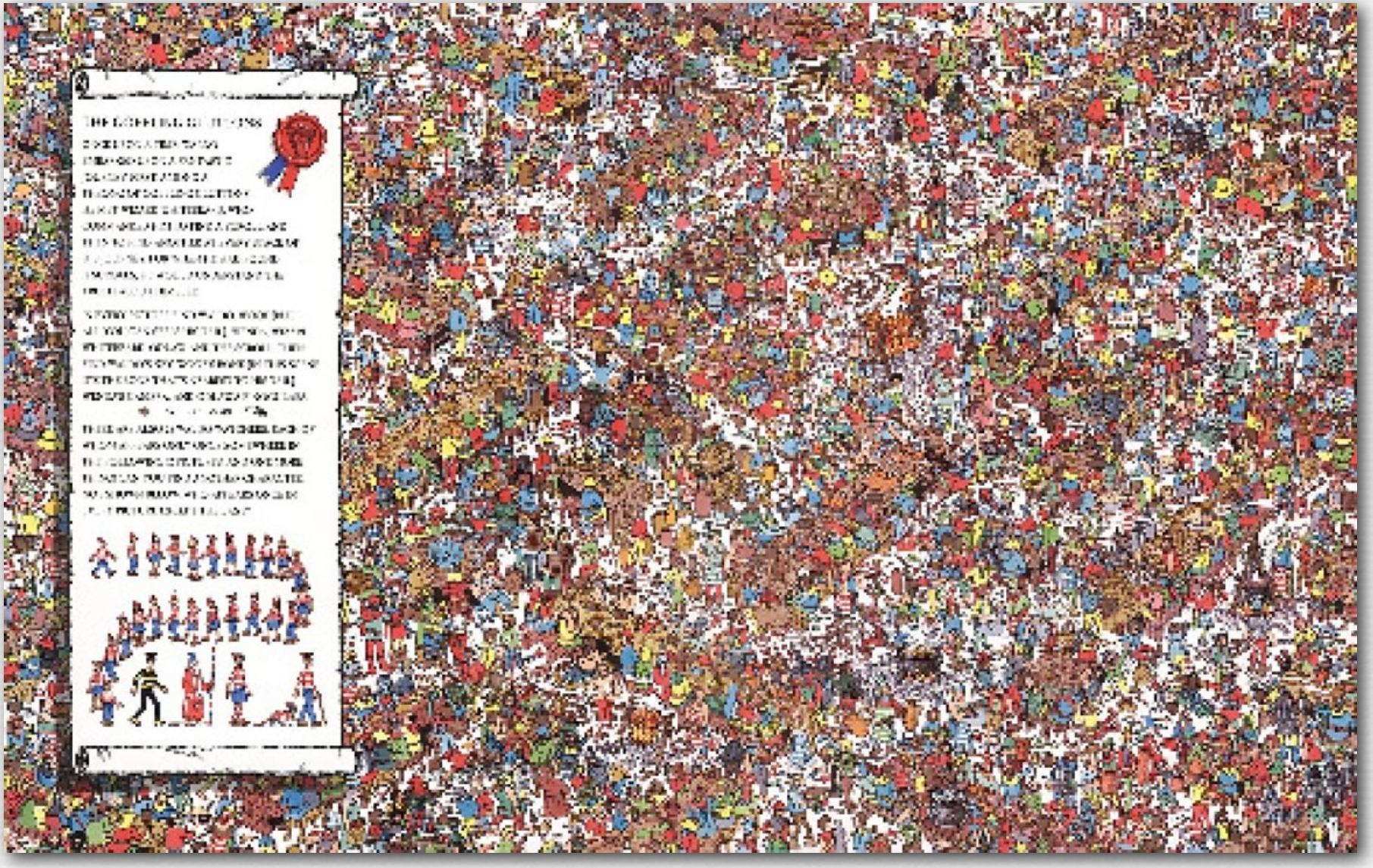
Point sampling a scene



every 4^{th} pixel in x and y

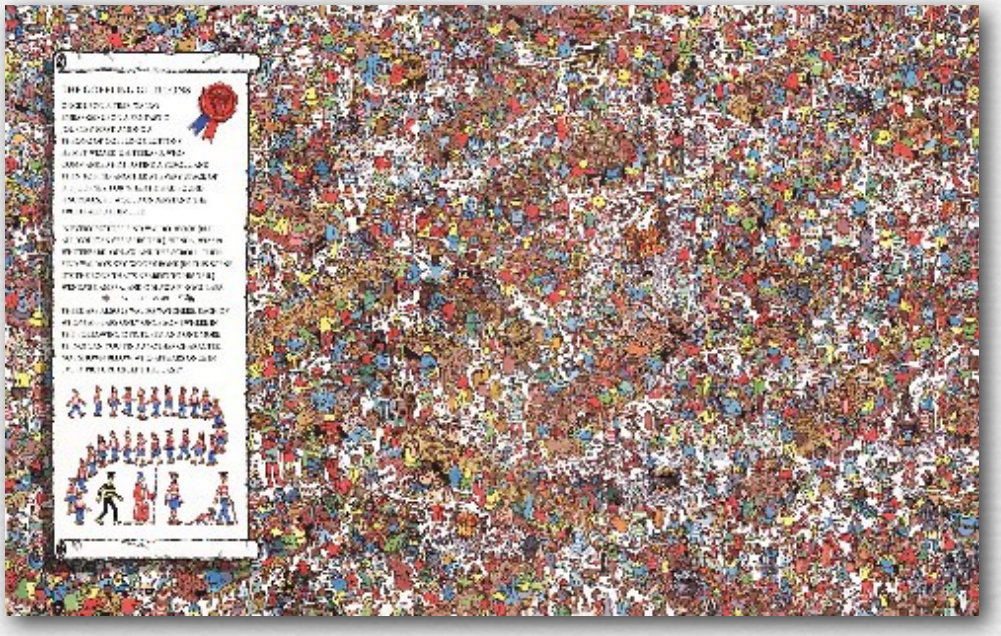
© Marc Levoy

Point sampling a scene



© Marc Levoy

Point sampling a scene



- *point sampling* means sampling at widely separated positions without first *prefiltering* (smoothing)
- in the presence of high frequencies in the original, this often produces *aliasing* (jaggies, moiré)

© Marc Levoy

Point sampling a different scene

(Natasha Gelfand)

© Marc Levoy

Point sampling a different scene

(Natasha Gelfand)

(original image is 1024 × 683 pixels)

© Marc Levoy

Point sampling a different scene

every 4th pixel in *x* and *y*

©Marc Levoy

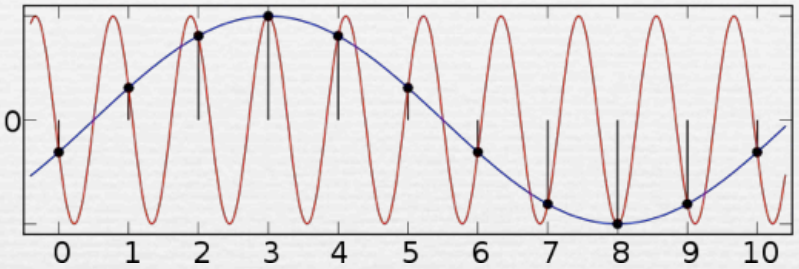
Point sampling a different scene

- sharp boundaries exhibit aliasing, but smooth features don't
- i.e. high frequencies alias easily, but low frequencies don't

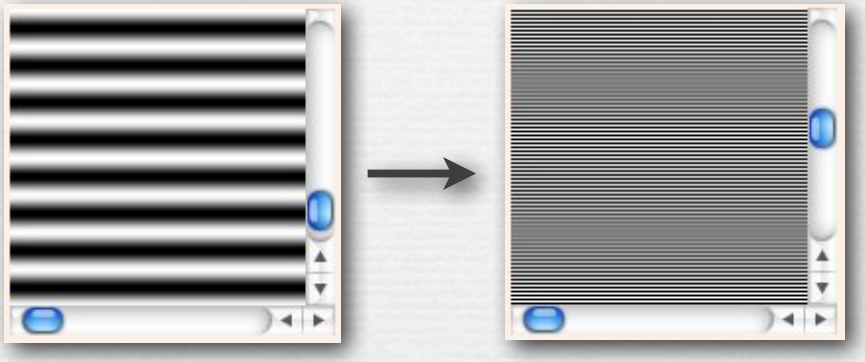
© Marc Levoy

Aliasing of sine waves

abstract function



spatial aliasing in images



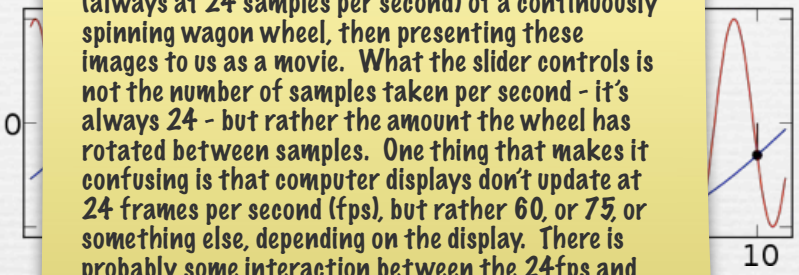
(http://ptolemy.eecs.berkeley.edu/eecs20/week13/moire.html)

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

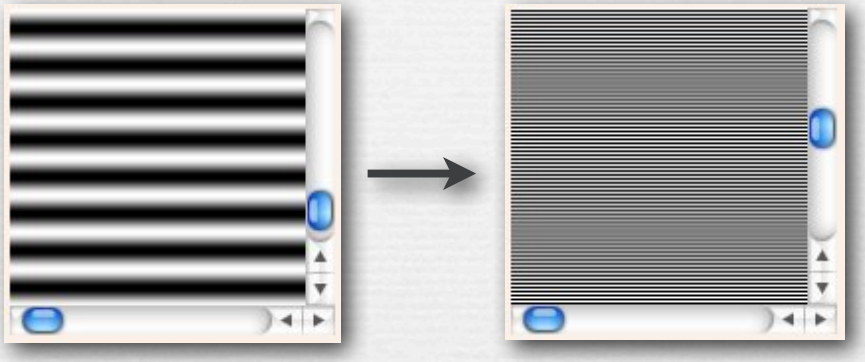
©Marc Levoy

Aliasing of sine waves

The demo of wagon wheels below is fun to play with but a bit confusing. What I believe the author is doing is drawing point samples regularly in time (always at 24 samples per second) of a continuously spinning wagon wheel, then presenting these images to us as a movie. What the slider controls is not the number of samples taken per second - it's always 24 - but rather the amount the wheel has rotated between samples. One thing that makes it confusing is that computer displays don't update at 24 frames per second (fps), but rather 60, or 75, or something else, depending on the display. There is probably some interaction between the 24fps and 60fps that the author is not describing, and we may be seeing. That's why I punted in class and started drawing wagon wheels on the whiteboard.

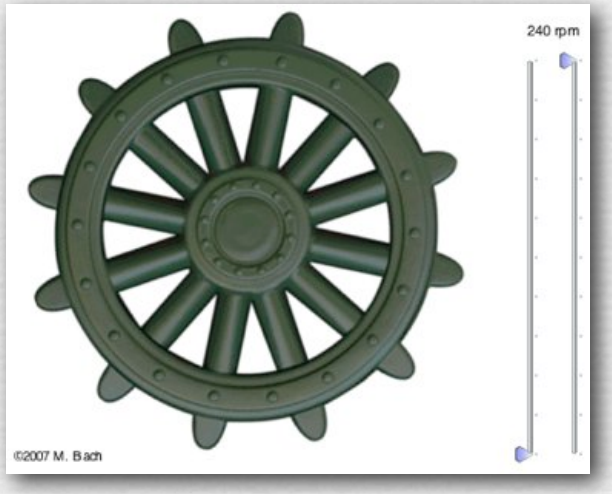


spatial aliasing in images



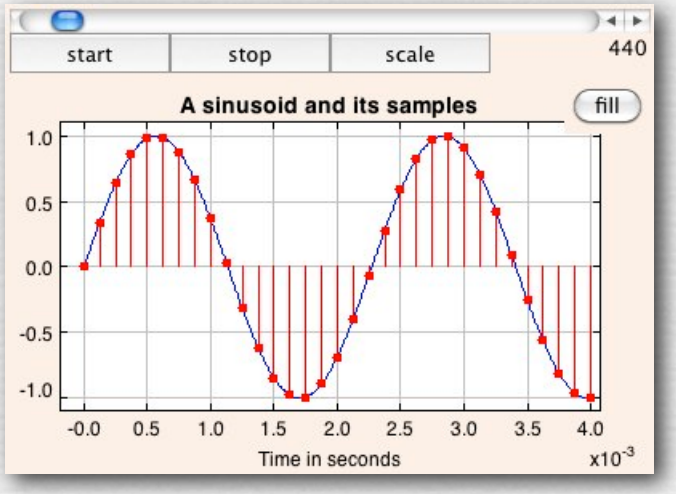
(http://ptolemy.eecs.berkeley.edu/eecs20/week13/moire.html)

temporal aliasing



(http://www.michaelbach.de/ot/mot_wagonWheel/index.html)

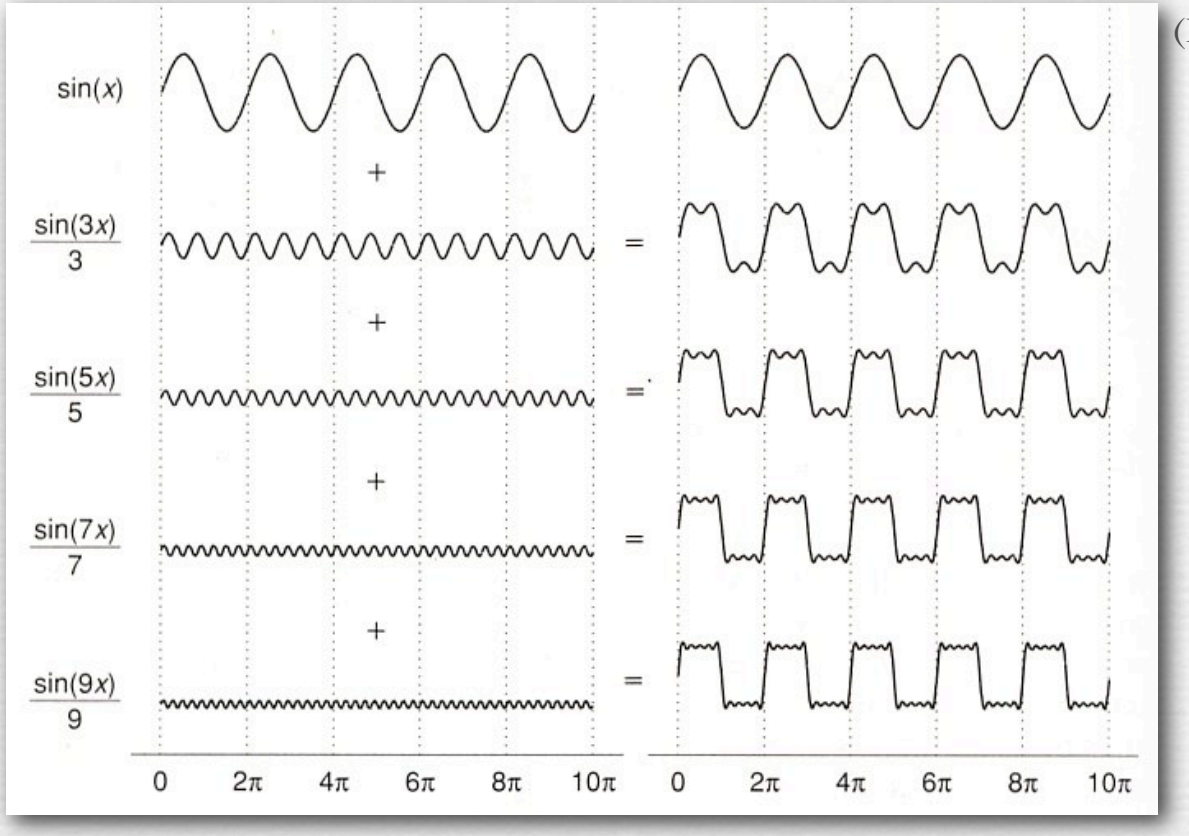
temporal aliasing in audio



(http://ptolemy.eecs.berkeley.edu/eecs20/week13/aliasing.html)

© Marc Levoy

Combinations of sine waves



(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 sines and cosines

© Marc Levoy

Frequency analysis of sampling

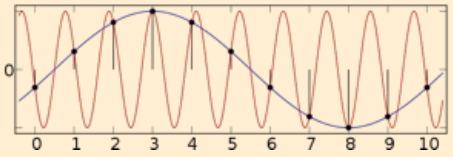
- Fourier series: any continuous, integrable, periodic function can be represented as an infinite series of sines and cosines

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

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

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

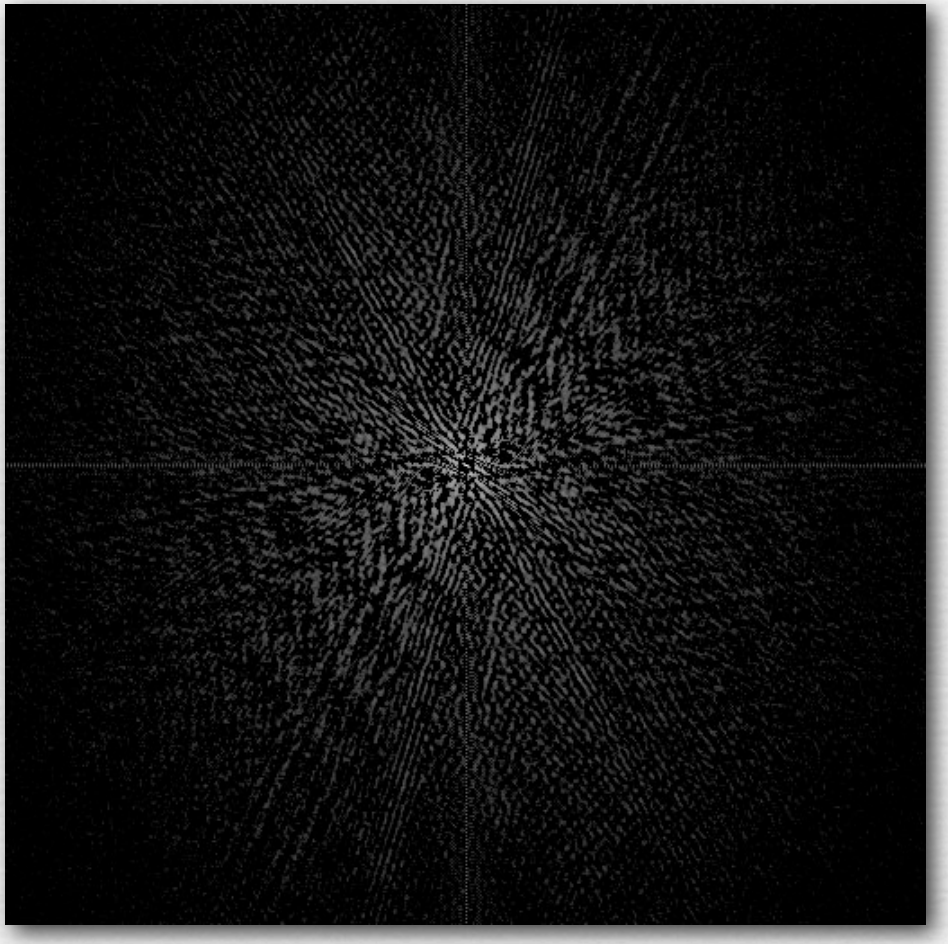
- aliasing: high frequencies masquerading as low frequencies due to insufficient sampling



©Marc Levoy

Frequency content of images

image

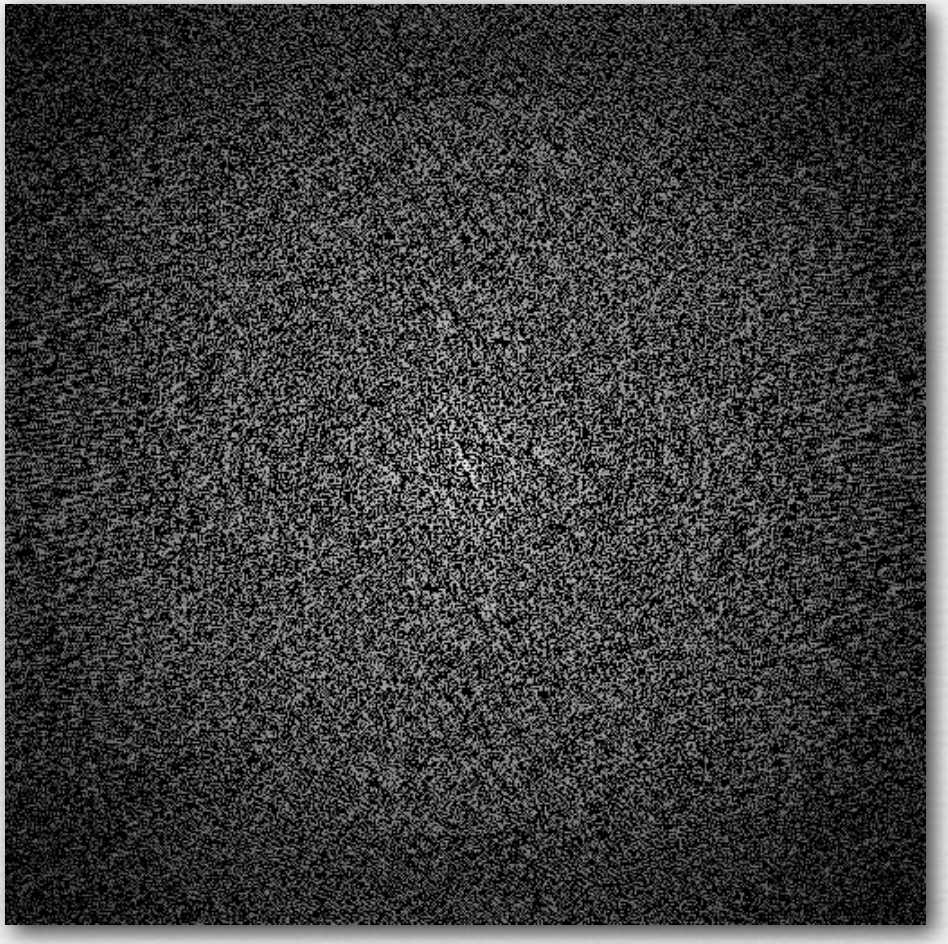


log(real(FFT(image)))

© Marc Levoy

Frequency content of images

image



log(real(FFT(image)))

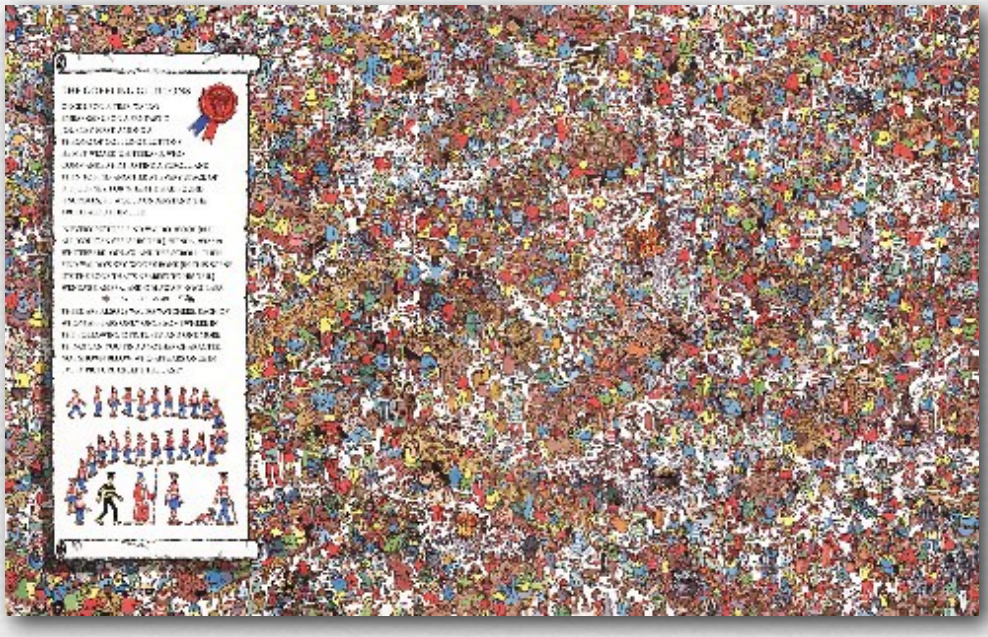
©Marc Levoy

Jaggies are a form of aliasing

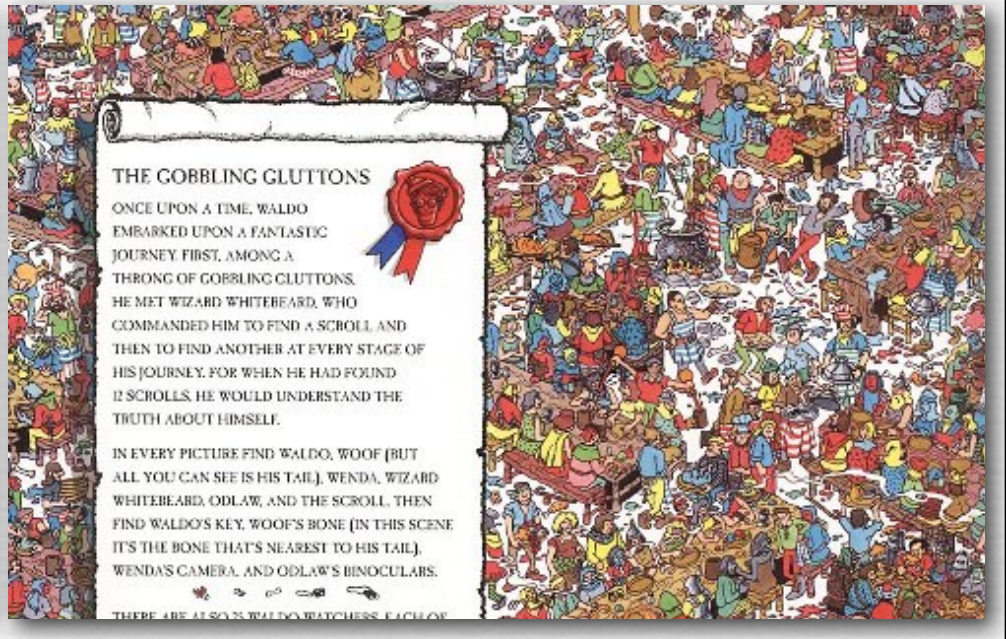
- a discontinuity in color (*step edge*) contains all frequencies
- if point sampled, these always produce aliasing (jaggies)

© Marc Levoy

Solution #1: raise the sampling rate



every 4th pixel in x and y

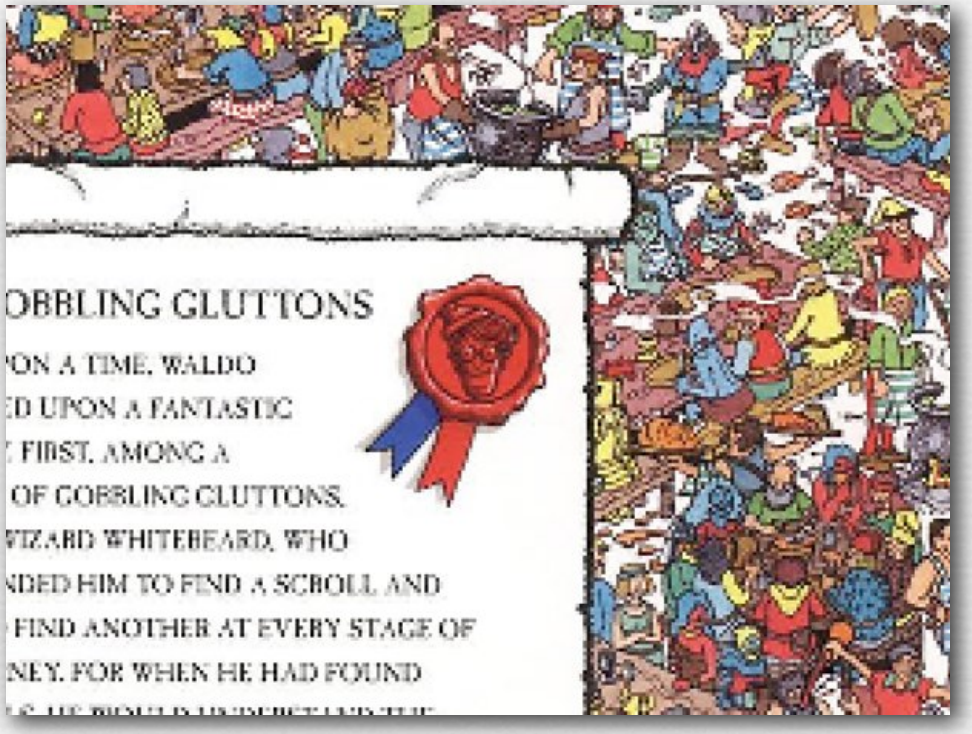


every 2nd pixel in x and y

©Marc Levoy

Solution #1: raise the sampling rate

every 4^{th} pixel in x and y

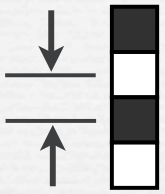


every 2^{nd} pixel in x and y

- raising the sampling rate reduces aliasing
- jaggies in step edges are still present, but they're smaller
- finer sampling may be impractical or expensive

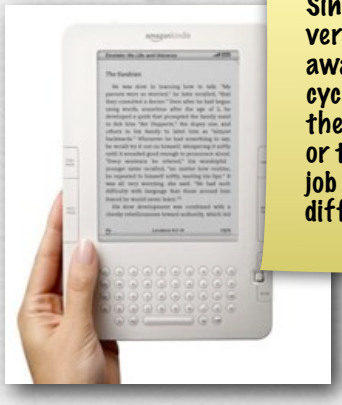
© Marc Levoy

"Sampling rates" of display media

period = Δx  frequency = $1/\Delta x$

- Example #1: Macbook Pro (laptop)
  - 900 pixels on 8" high display
  - Δx = 8"/ 900 pixels = 0.0089"/pixel
  - $1/\Delta x$ = 112 dpi (dots per inch)
- Example #2: Kindle 2
  - 800 pixels on 4.8" high display
  - $1/\Delta x$ = 167 dpi
- Example #3: iPad
  - 768 pixels on 5.8" high display
  - $1/\Delta x$ = 132 dpi

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

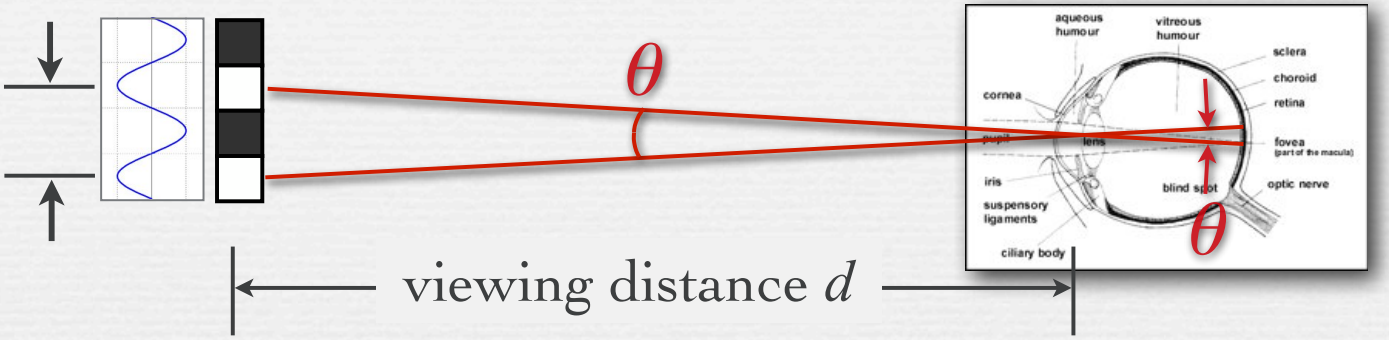


Since the iPad has roughly the same number of pixels vertically as the Kindle, you could hold it further away from you, use a bigger font, and get the same cycles per degree on your retina as the Kindle. Also, the iPad allows more graylevels than the Kindle's 4, or the Kindle 2's 16, so it might be able to do a better job of anti-aliasing (prefiltering) text, but this difference is likely to be small.

© Marc Levoy

Spatial frequency on the retina

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

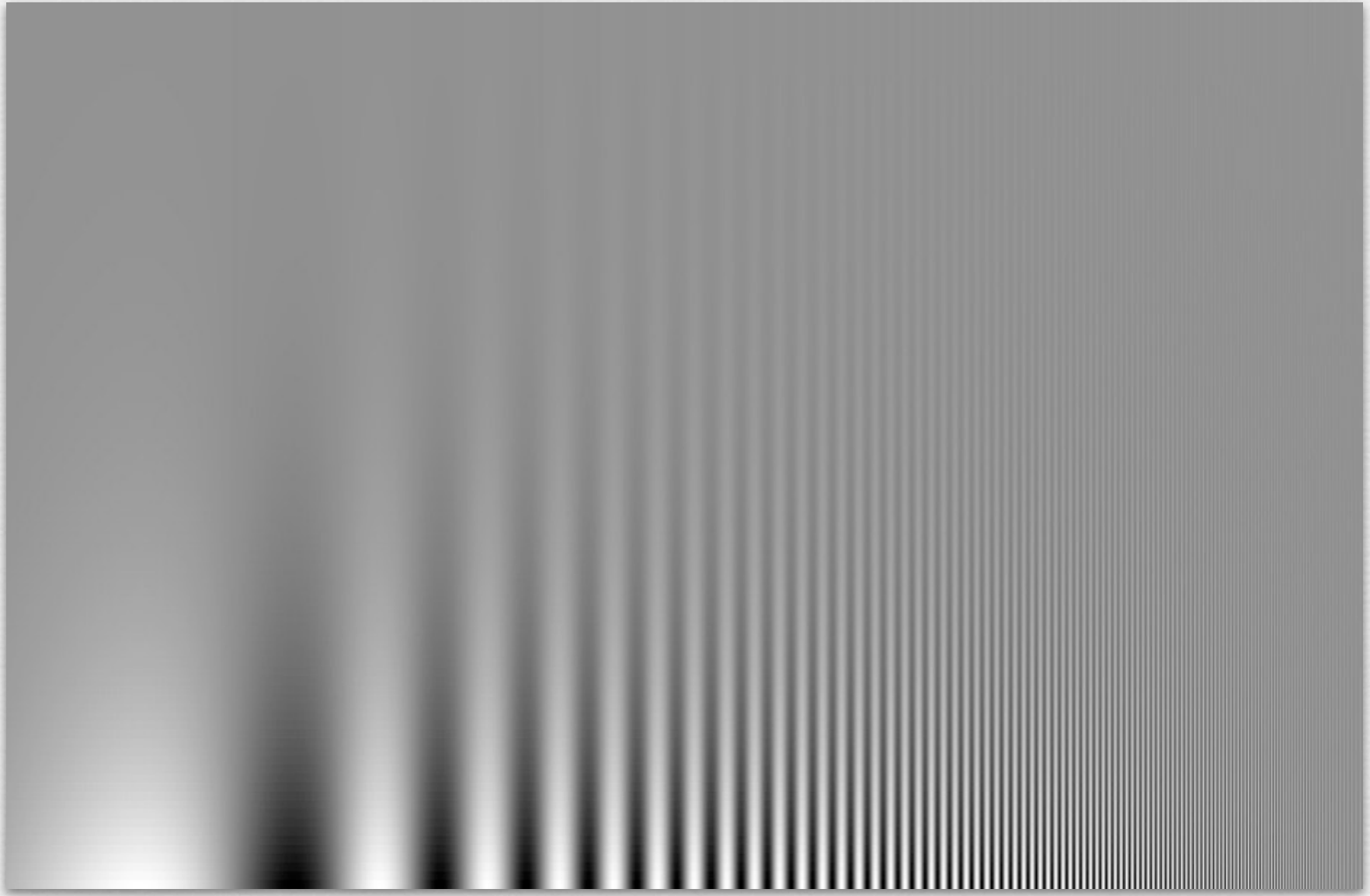


- Example #1: Macbook Pro viewed at $d = 18$”
  - 900 pixels on 8” high display, $p = 2 \times 0.0089$”
  - retinal arc $\theta = 2 \arctan(p / 2d) = 0.057^o$
  - spatial frequency on retina $1/\theta = 17.6$ cycles per degree

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

© Marc Levoy

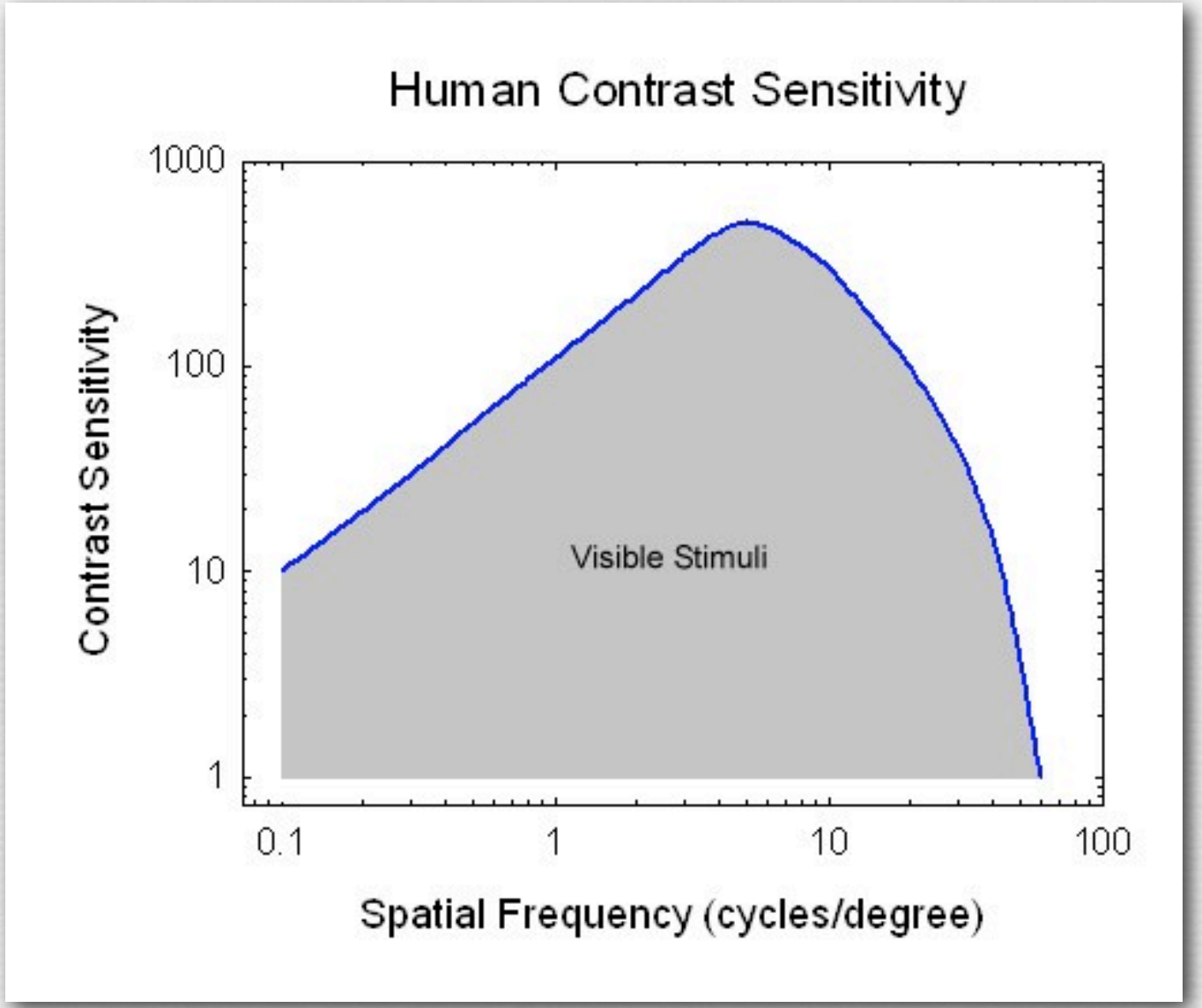
Human spatial sensitivity
(Campbell-Robson Chart)



(neurovision.berkeley.edu)

© Marc Levoy

Human spatial sensitivity



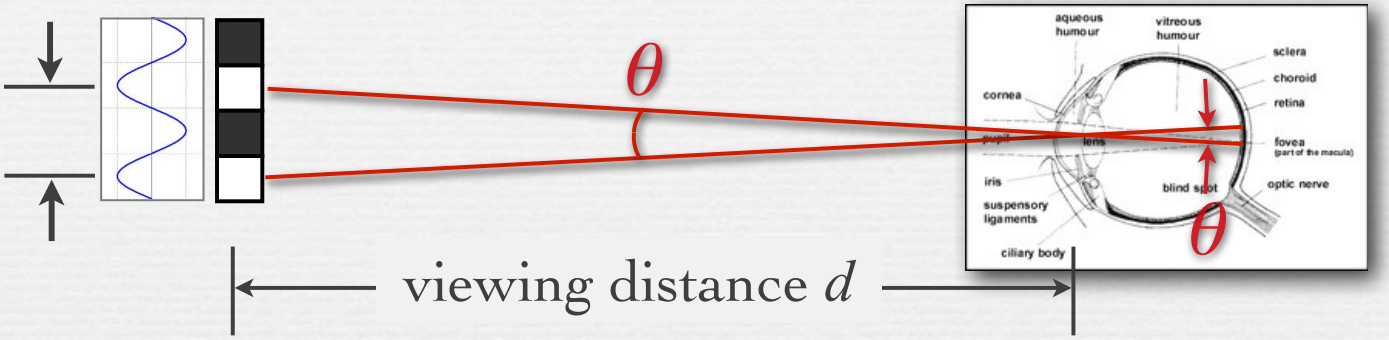
(horizontal axis not comparable to image on previous slide)

(psych.ndsu.nodak.edu)

© Marc Levoy

Spatial frequency on the retina

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



- Example #1: Macbook Pro viewed at *d = 18”*
  - 900 pixels on 8” high display, so *p = 2 × 0.0089”*
  - retinal arc *θ = 2 arctan (p / 2d) = 0.057°*
  - spatial frequency on retina *1/θ = 17.6* cycles per degree

not nearly as high as human acuity

© Marc Levoy

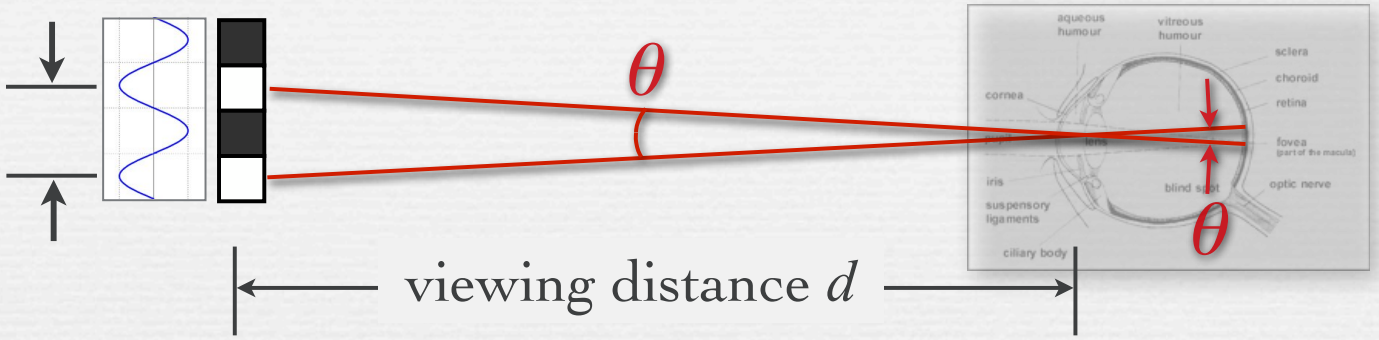
(Graham Flint)

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



- Example #1: Macbook Pro viewed at *d = 18”*
  - 900 pixels on 8” high display, *p = 2 × 0.0089”*
  - retinal arc *θ = 2 arctan (p / 2d) = 0.057°*
  - spatial frequency on retina *1/θ = 17.6* cycles per degree

- Example #2: 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

© Marc Levoy

Retinal sampling rate

- the human retina consists of discrete sensing cells
- therefore, the retinal mosaic performs sampling
- sampling theory says $f_{sampling} > 2 \times f_{cutoff}$
- if observed human cutoff is 50 cycles per degree, then sampling must be > 100 cycles per degree
- this agrees with observed retinal cell spacing!

(Cornsweet)

spacing between L,M cone cells is 1μ ≈ 30 arc-seconds

© 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

- Example: photographic print from viewed at 12”
  - max human acuity on retina $1/\theta \approx 50$ cycles per degree
  - minimum detectable retinal arc $\theta = 30$ arc-seconds
  - minimum feature size $p = 2 \times 12” \times \tan(\theta / 2) = 0.0043”$ *(0.1mm)*

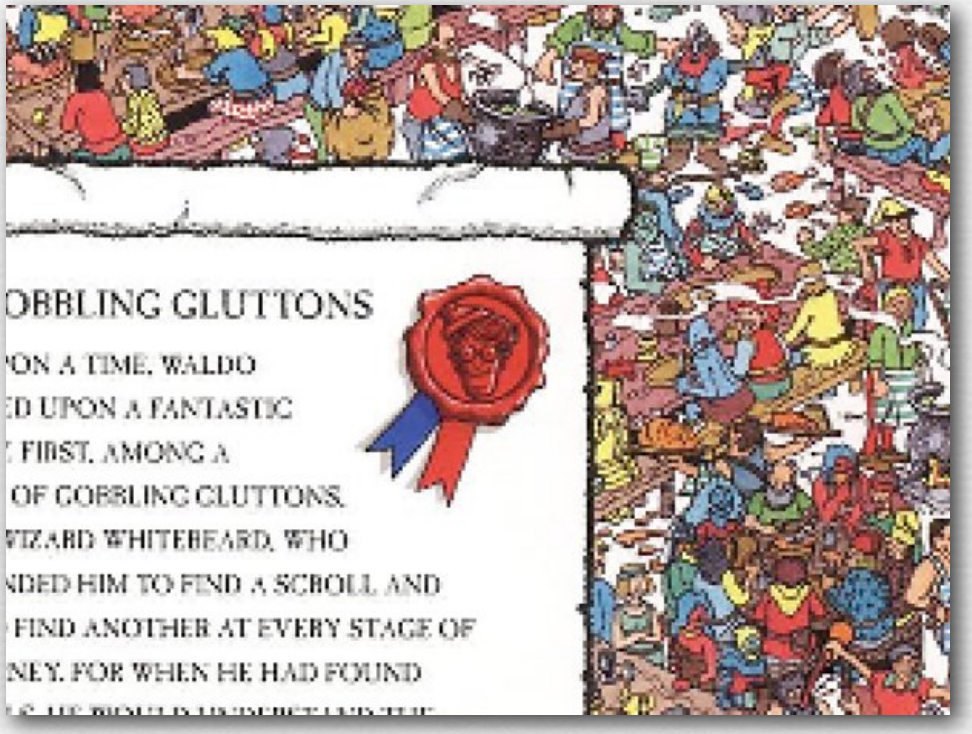
I've fixed the arithmetic. So if you printed your Canon 5D II image at 5x7" and viewed it at 12", your depth of field (the range of scene features that will appear in-focus to you) is about 2.5x greater than the figures we were computing last week, when we assumed that C is one camera pixel wide.

- assume 5” × 7” print and Canon 5D II (5616 × 3744 pixels)
  - 5” / 3744 pixels = 0.0017”/pixel *(0.04mm)*
  - therefore, circle of confusion can be 2.5 pixels wide before it's blurry
  - $C = 6.4\mu$ per pixel × 2.5 pixels = 16μ

© Marc Levoy

Solution #1: raise the sampling rate

every 4^{th} pixel in x and y



every 2^{nd} pixel in x and y

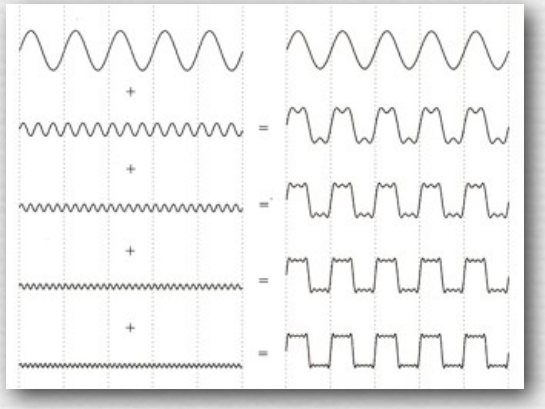
- raising the sampling rate reduces aliasing
- jaggies in step edges are still present, but they're smaller
- finer sampling may be impractical or expensive

© Marc Levoy

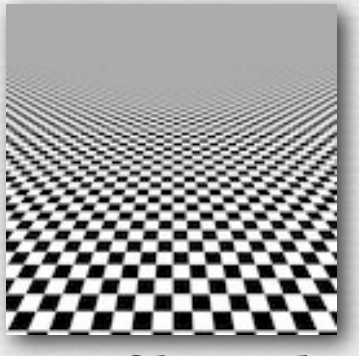
Solution #2: pre-filter the input

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

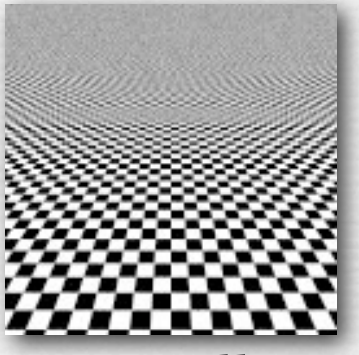
$$f_{sampling} > 2 \times f_{cutoff}$$



unfiltered



prefiltered



partially pre-filtered

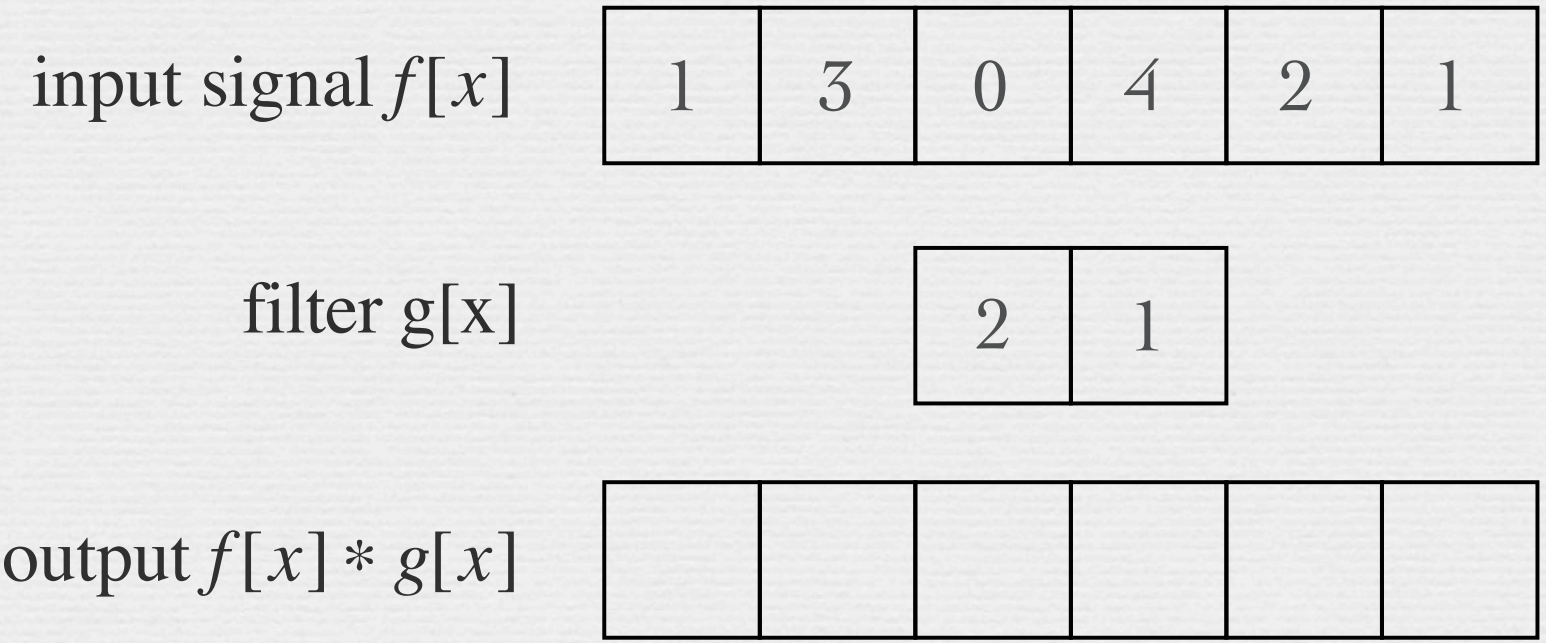
- method: *convolve* the input function by a *filter* function, which smooths it, then perform point sampling as before

©Marc Levoy

Discrete convolution in 1D

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

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



©Marc Levoy

Discrete convolution in 1D

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

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

Note about flipping added 5/3/10 after an alert student noticed the error. This flipping occurs because g is indexed as [x-k] in the formula, not [k].

input signal $f[x]$

1	3	0	4	2	1

filter g[x]

1	2

notice that the filter gets flipped when applied

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

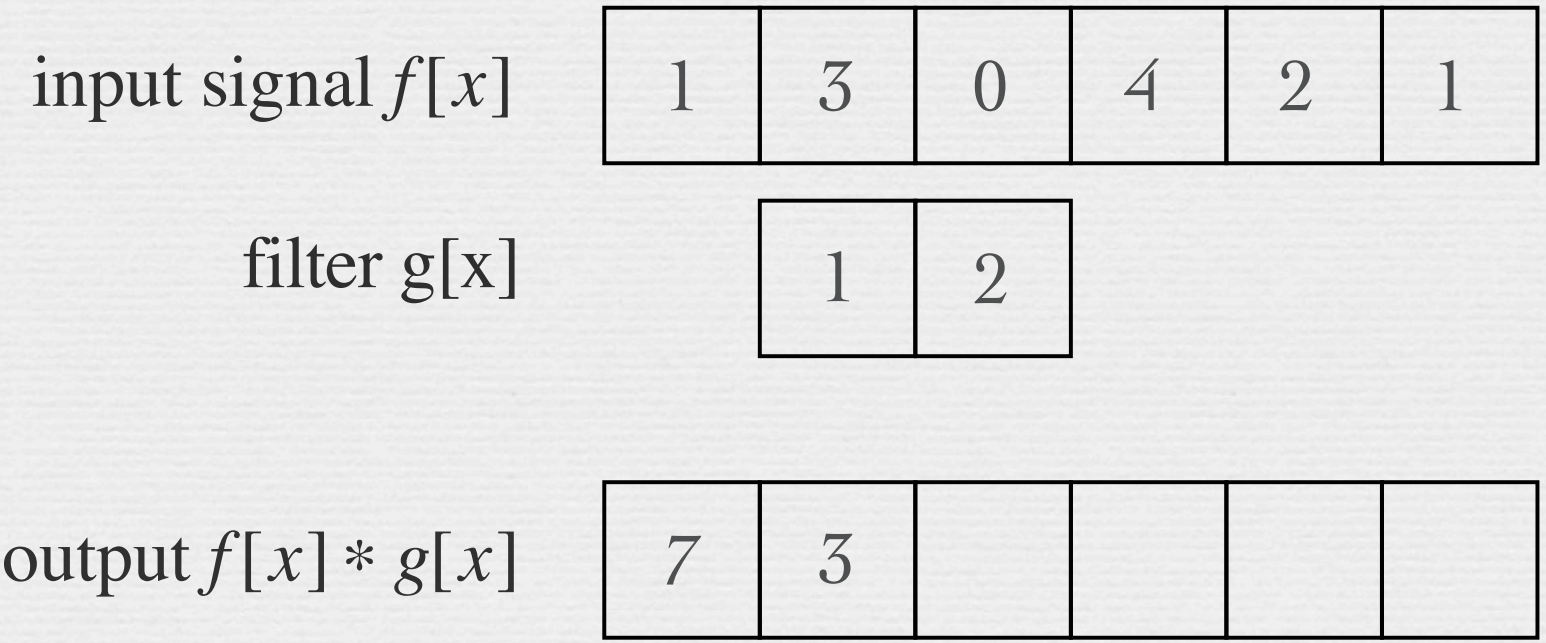
7					

©Marc Levoy

Discrete convolution in 1D

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

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

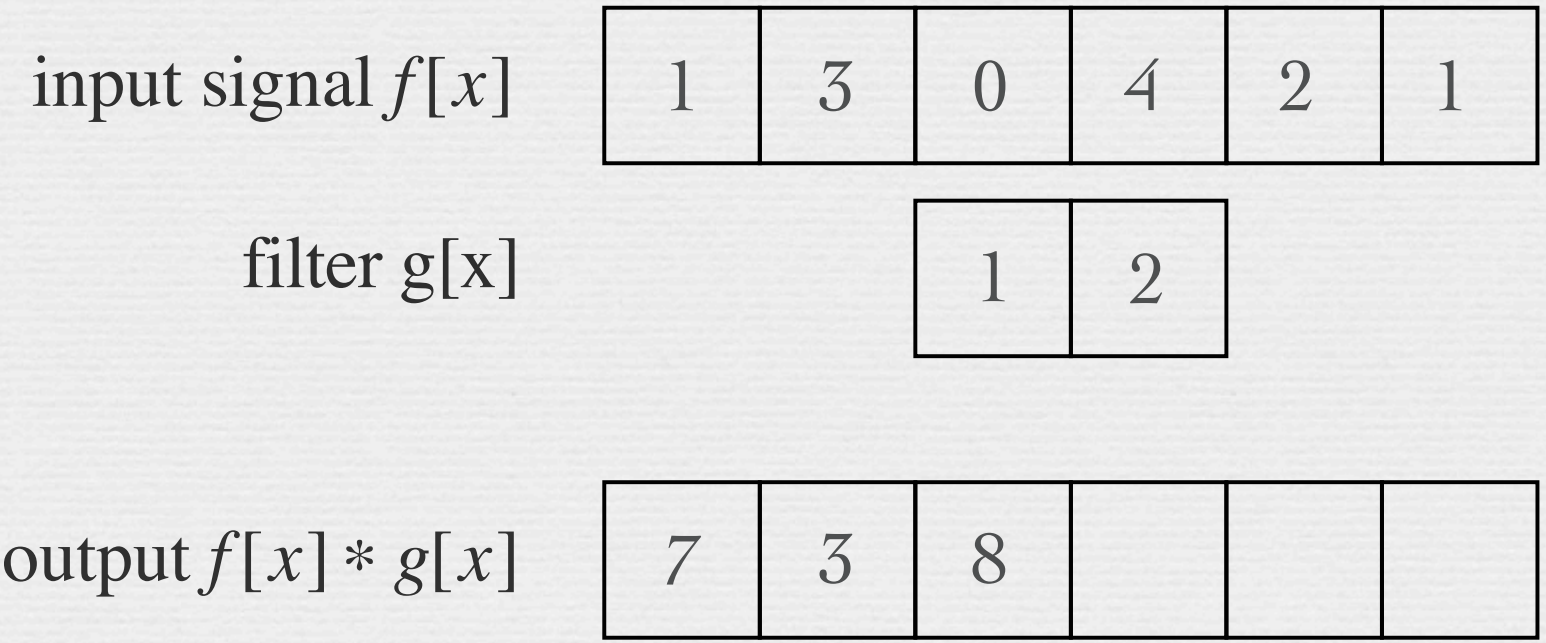


© Marc Levoy

Discrete convolution in 1D

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

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



© Marc Levoy

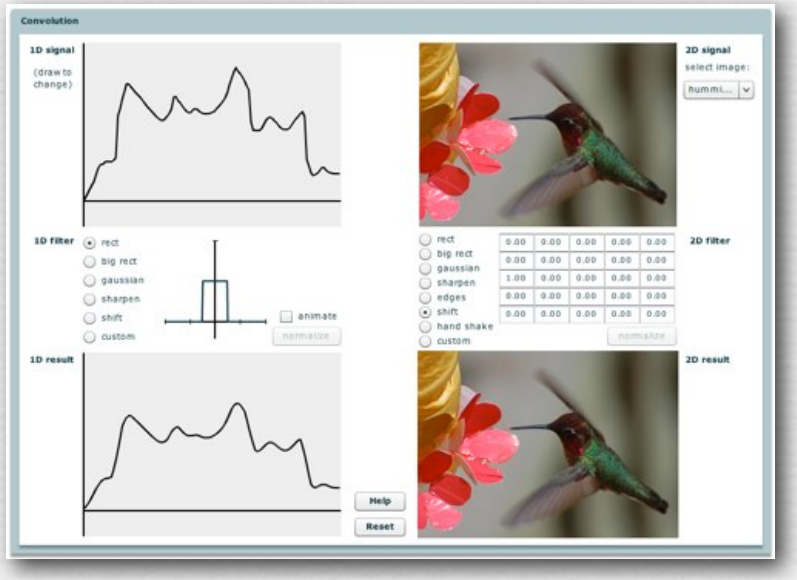
More convolution formulae

- 1D discrete: defined only on the integers

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

- 1D continuous: defined on the real line

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

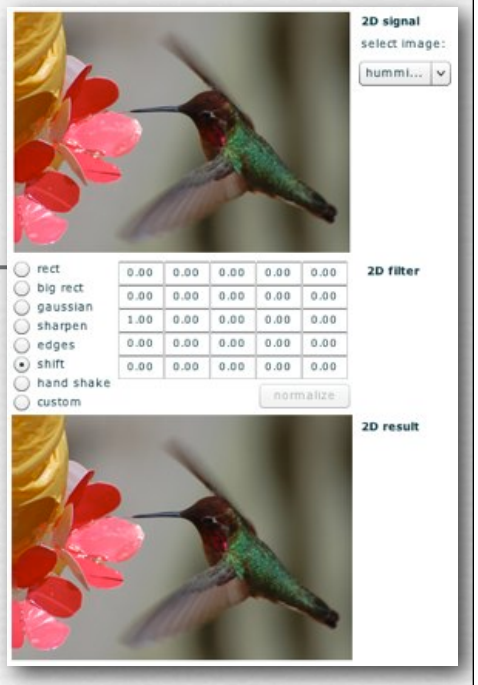


(FLASH DEMO)

http://graphics.stanford.edu/courses/cs178/applets/convolution.html

© Marc Levoy

More convolution formulae



- 1D discrete: defined only on the integers

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

- 1D continuous: defined on the real line

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

- 2D discrete: defined on the x, y integer grid

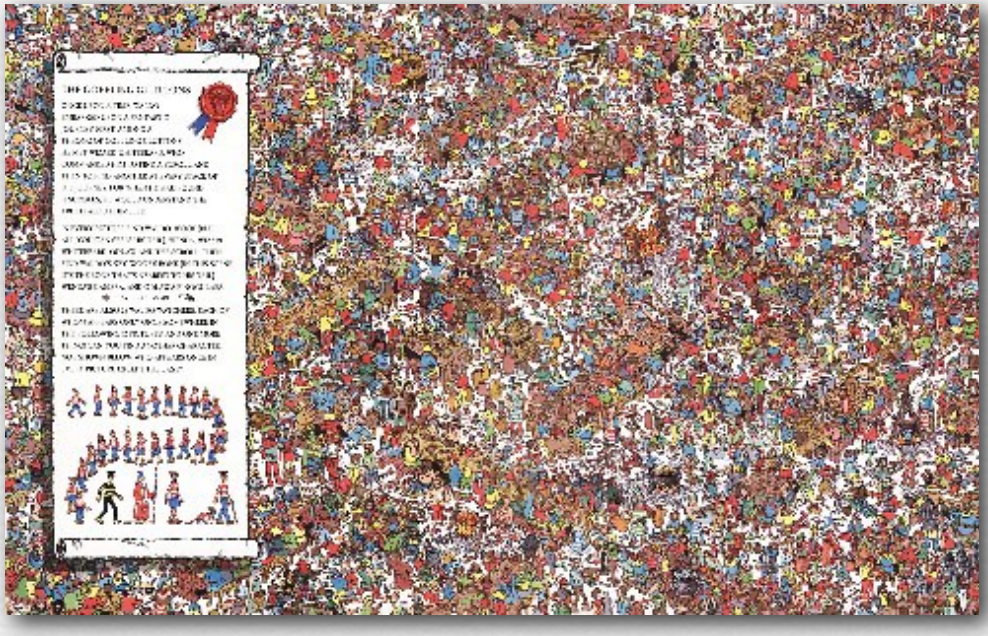
$$f[x,y] * g[x,y] = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f[i,j] \cdot g[x-i, y-j]$$

- 2D continuous: defined on the x,y plane

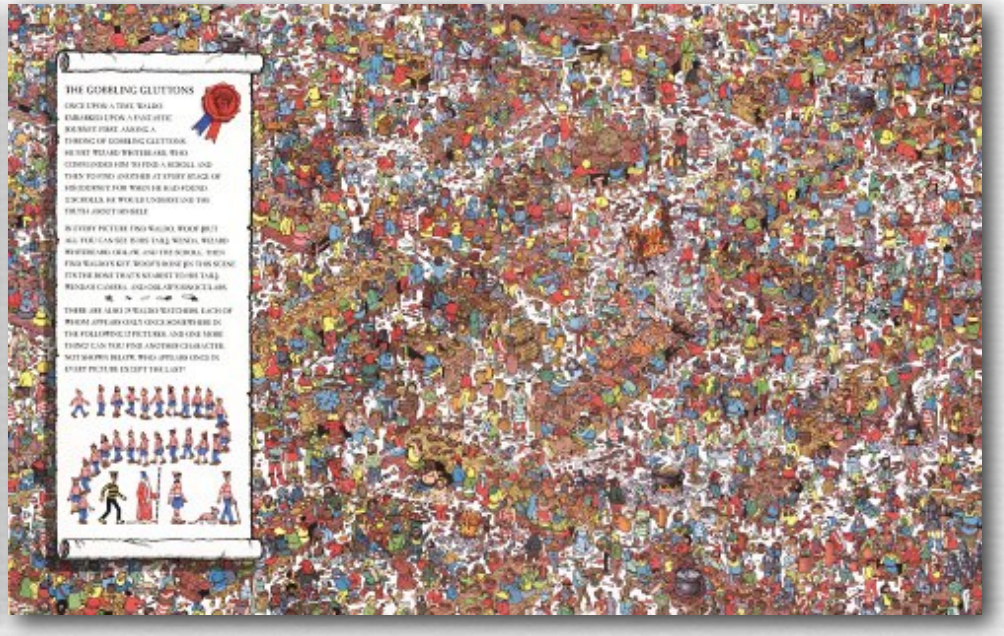
$$f(x,y) * g(x,y) = \int_{\tau_1=-\infty}^{\infty} \int_{\tau_2=-\infty}^{\infty} f(\tau_1,\tau_2) \cdot g(x-\tau_1, y-\tau_2)\, d\tau_1\, d\tau_2$$

©Marc Levoy

Prefiltering reduces aliasing



point sampled



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

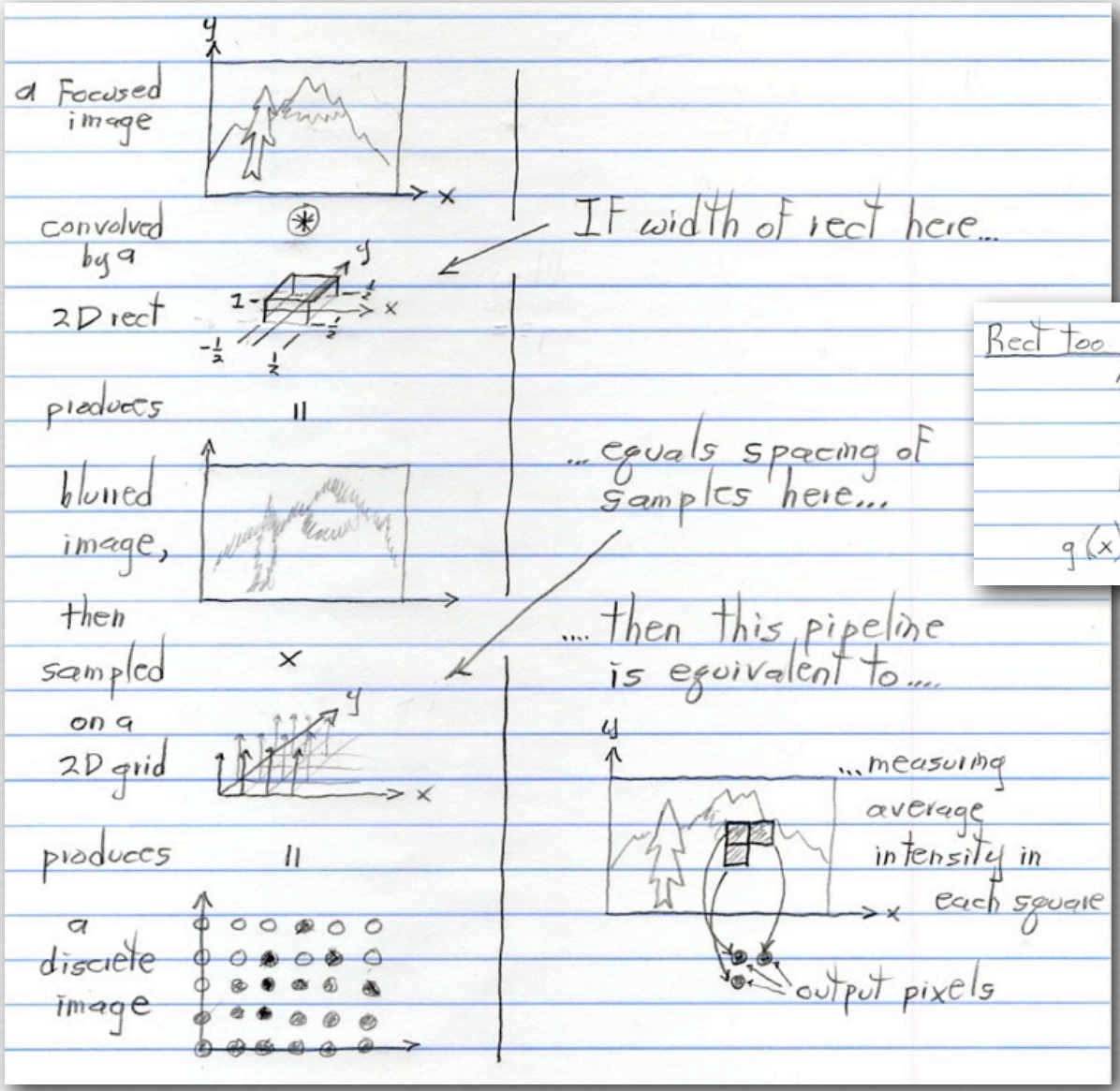
© Marc Levoy

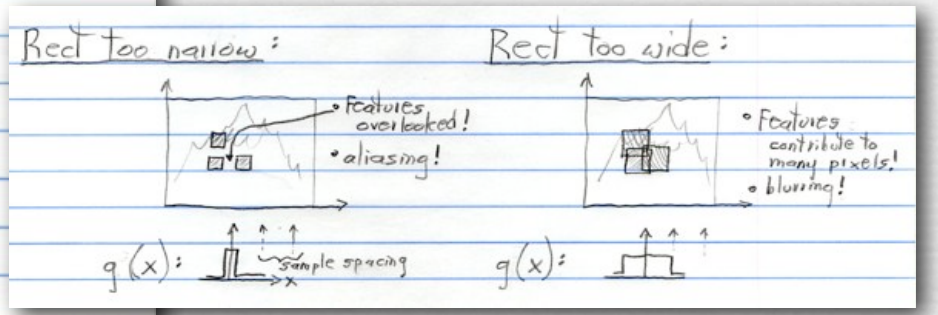
Prefiltering & sampling in photography

- 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 width of the rect is typically equal to the spacing between sample positions
  - narrower leaves aliasing; wider produces excessive blur

© Marc Levoy

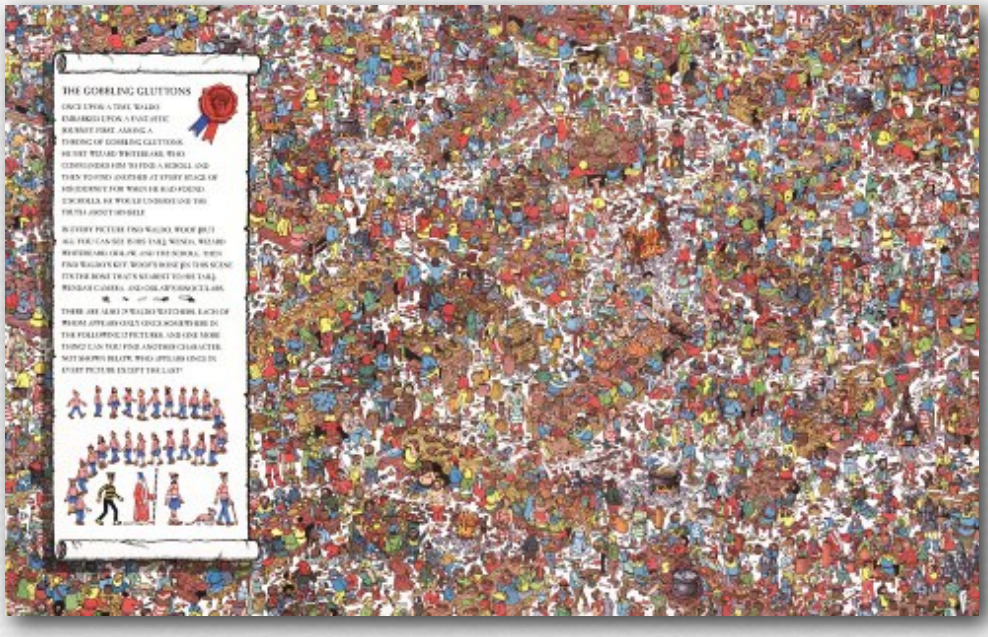
Prefiltering & sampling in photography (contents of whiteboard)





© Marc Levoy

Prefiltering versus postfiltering



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

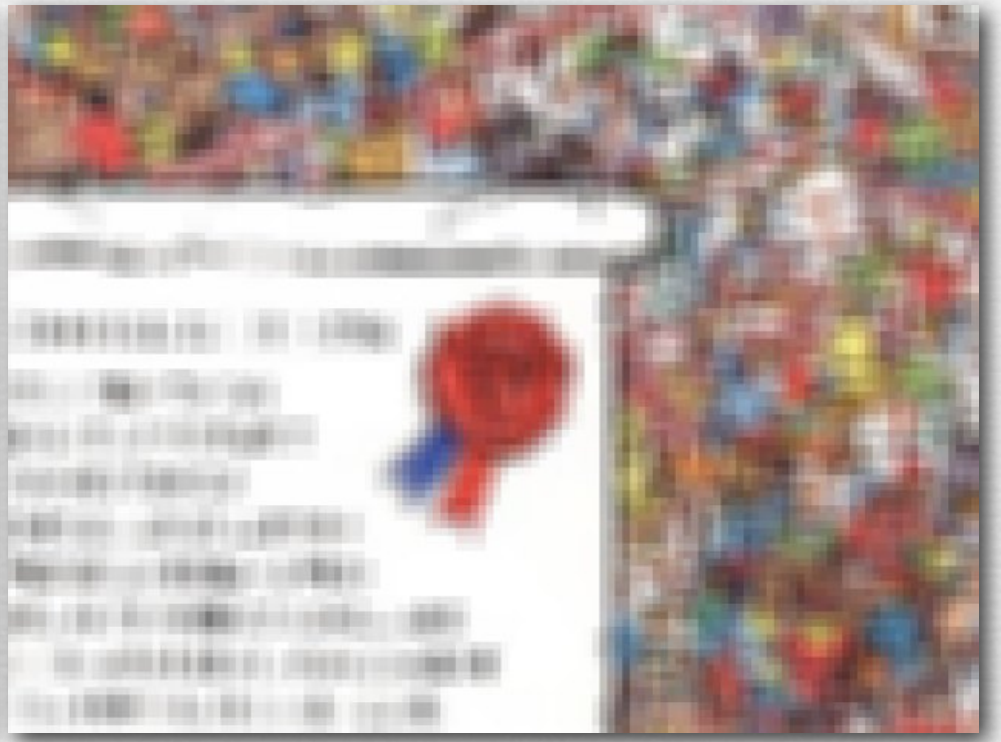
point sampled every 4th pixel,
then convolved by 4×4 pixel rect

- *postfiltering* means filtering by convolution after sampling
- if a camera's sensitive areas don't fill the space between pixels, can you fix the resulting aliasing by postfiltering the sampled image?

©Marc Levoy

Prefiltering versus postfiltering

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



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

- aliasing is high frequencies masquerading as low frequencies;
  it cannot be removed by postfiltering without destroying the image

© Marc Levoy

"Filtering" in cartoon animation

prefilter

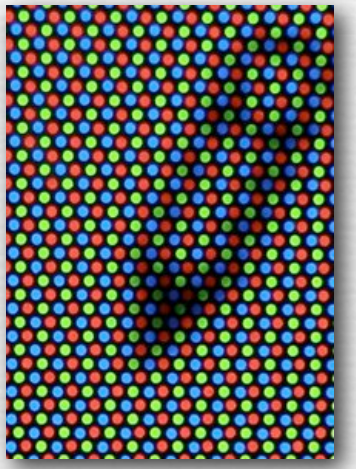
postfilter

© Marc Levoy

Reconstruction for display

- *reconstruction* is the conversion of a discrete (i.e. sampled) image into a continuous image by *interpolation* between the available samples
  - interpolation can be modeled as (another) convolution

- for CRTs and LCDs the reconstruction filter is typically a rect or Gaussian spot of width equal or a bit larger than the sample spacing
  - narrower leaves *reconstruction error*, a.k.a. rastering

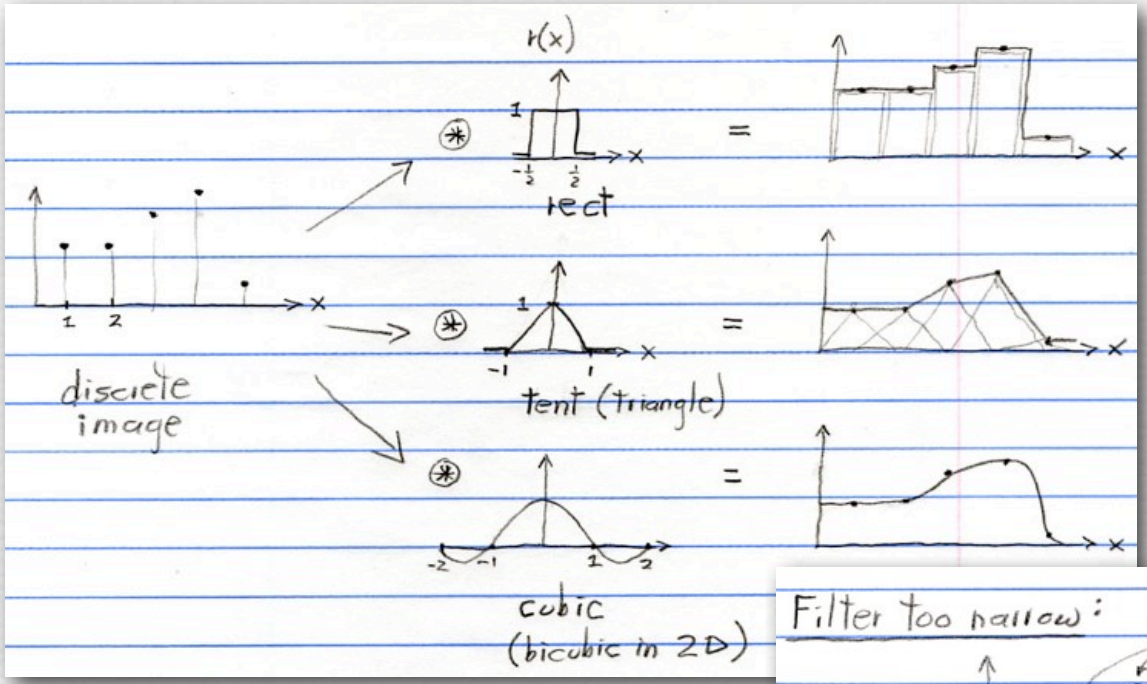


(wikipedia)

- in computer graphics and digital photography, we don't worry about reconstruction; we hope the display screen or printer does it for us
  - for very close samples, blurring in the human eye helps

©Marc Levoy

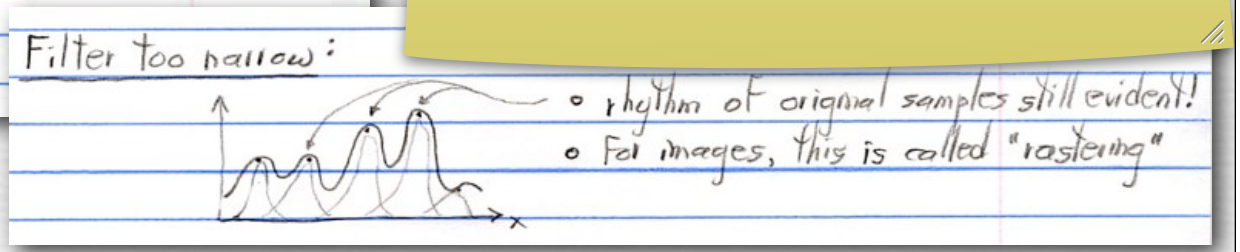
Interpolation via convolutic (contents of whiteboard)



My presentation of the prefilter -> sampling -> reconstruction pipeline was geared towards capturing a real-world scene (a continuous function) using a digital camera, thereby producing a discretely sampled image, then reconstructing the continuous image (as a percept headed to our brain) by reconstruction (a.k.a. interpolation) on the surface of a computer display or in our eye.

In response to a question in class, I mentioned that a similar pipeline is used implicitly in Photoshop when images are resized. In that case, one starts with a discretely sampled image, then reconstructs a continuous image using an interpolation filter, then upsizes, downsizes, or warps that continuous iamge, then prefilters, then re-samples, producing a new discretely sampled image, possibly having a much different number of pixels. In practice, all these steps in this reconstruction -> warp -> prefilter -> resample pipeline can be folded together into a single carefully constructed summation, due to the algebraic properties of convolution and sampling.

I did not describe this "resampling pipeline", or the resulting summation, in detail, because I consider it beyond the scope of this course. If you're interested in learning more about image resampling, look at the "Image resampling pipeline" section of the October 7 notes from my 2008 version of CS 248, at http://graphics.stanford.edu/courses/cs248-08/samp/samp2.html, and the "Texture rendering is image resampling" section from the October 23 notes.



- if the input is a discrete (i.e. sampled) function, then convolution can be treated as placing an 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

© Marc Levoy

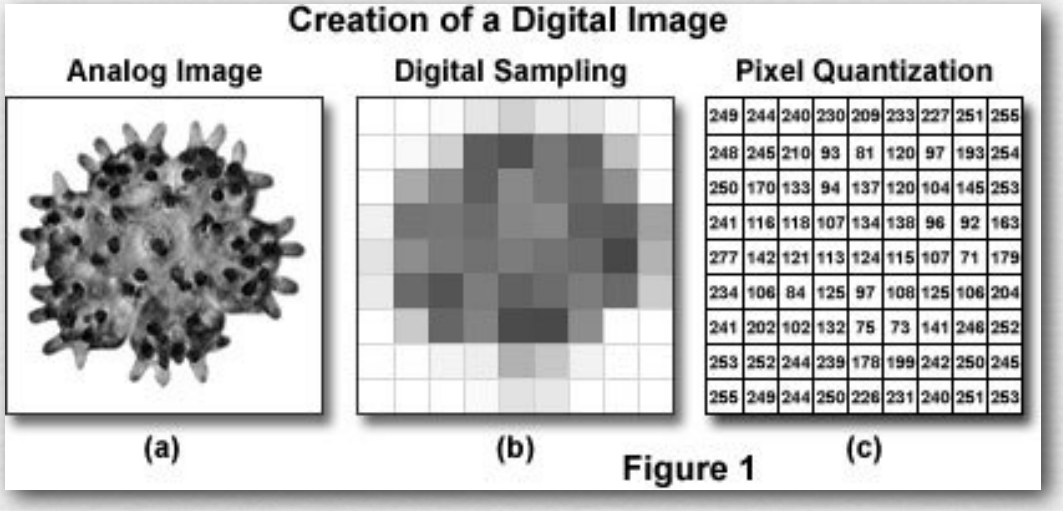
Recap

- *aliasing* is high frequencies masquerading as low frequencies due to insufficient sampling; reduce by:
  - raising the sampling rate, or
  - prefiltering the input before sampling
  - prefiltering can be modeled as convolution by a filter *g(x)*
- *rastering* is a residual of the sampling frequency due to inadequate reconstruction (a.k.a. interpolation)
  - reconstruction can be modeled as convolution by a filter *r(x)*
  - Don't confuse prefiltering v. reconstruction, or aliasing v. rastering!
- in digital photography:
  - the prefilter is a pixel-sized rect, thus averaging intensity over areas
  - the reconstruction filter is a pixel-sized rect or Gaussian on an LCD

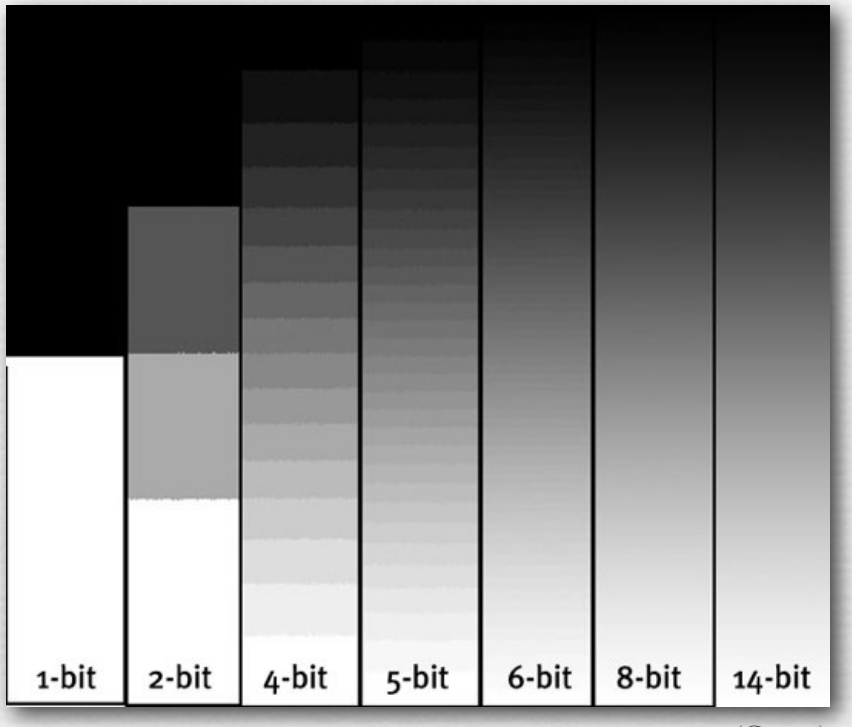
Questions?

© Marc Levoy

Sampling versus quantization



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



(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

© Marc Levoy

Example

8 bits × R,G,B =
24 bits per pixel

Canon 1D III,
300mm, f/3.2

© Marc Levoy

Example

8 bits × R,G,B =
24 bits per pixel

© Marc Levoy

Example

6 bits × R,G,B =
18 bits per pixel

© Marc Levoy

Example

5 bits × R,G,B =
15 bits per pixel

©Marc Levoy

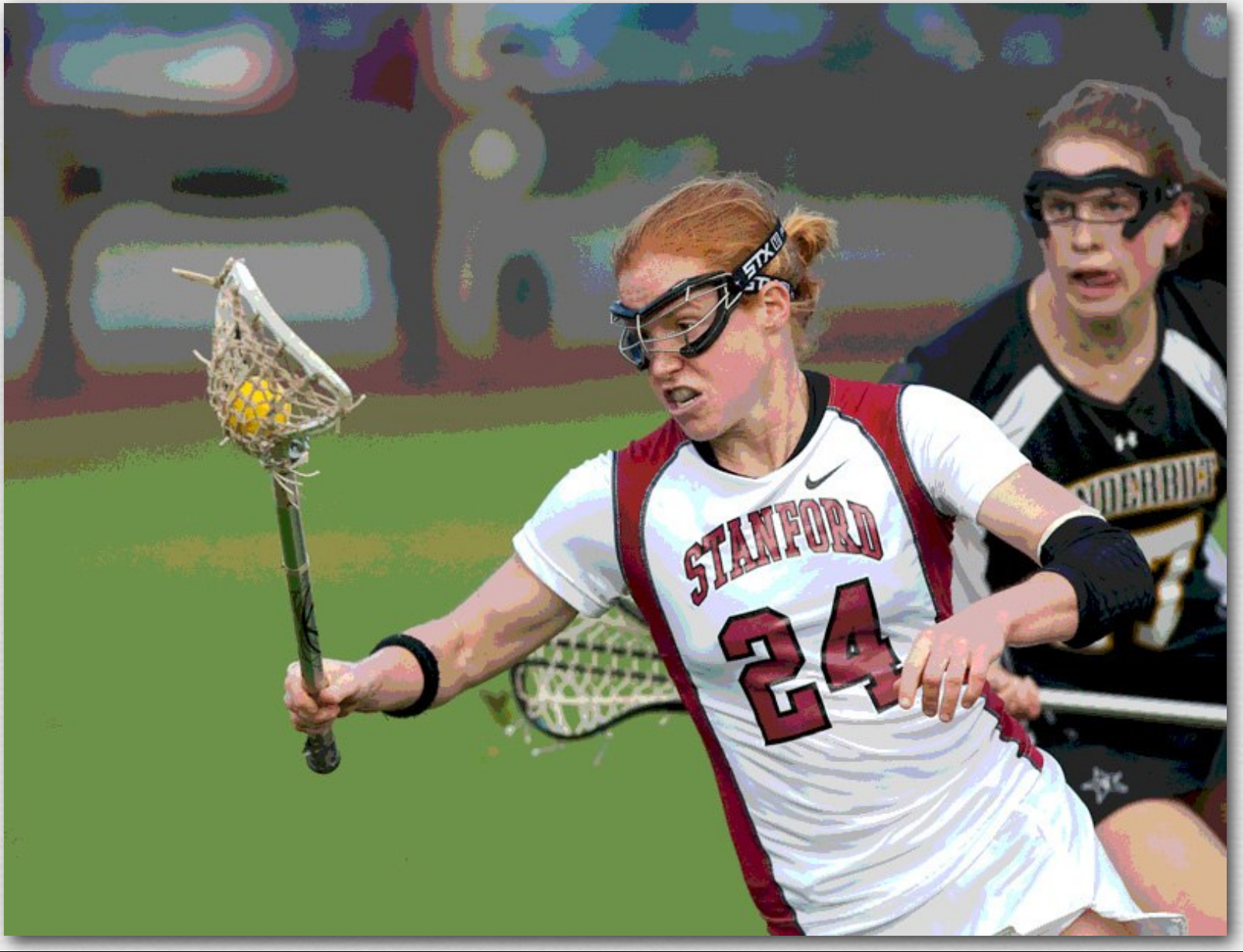
Example

4 bits × R,G,B =
12 bits per pixel

© Marc Levoy

Example

3 bits × R,G,B =
9 bits per pixel



© Marc Levoy

Example

256 colors (8 bits) uniformly distributed across RGB cube, patterned dithering in Photoshop

© Marc Levoy

Example

256 colors (8 bits) adaptively distributed across RGB cube, patterned dithering in Photoshop

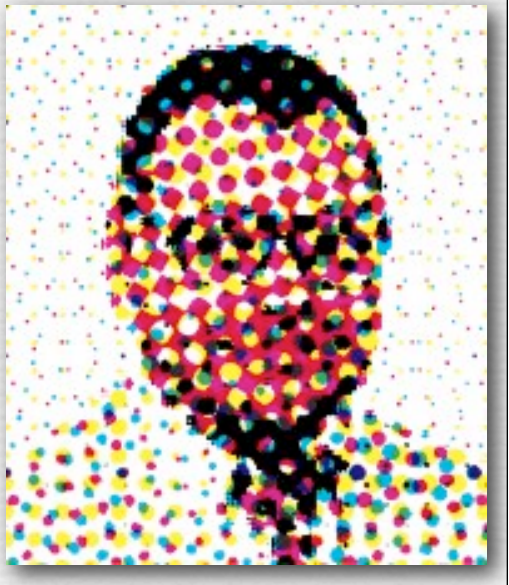
A student asked about the relationship between dithering and what's called halftoning for printing. They're different. Halftoning is a bit off-topic for the course, but not that far off-topic, so I'll cover it in class on 4/20. I've also added a slide on it to this PDF file (see next slide).

© 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 primary colors
  - 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



(wikimedia)

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

For an example of black and white halftoning, see http://en.wikipedia.org/

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

© 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*
  - even if it is low, 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*
  - even if it is low, you can reduce contouring by *dithering* (if displaying the image on a screen) or *halftoning* (if printing it on paper)

Questions?

©Marc Levoy

Slide credits

- Pat Hanrahan

- Cornsweet, T.N., *Visual Perception,* Kluwer Academic Press, 1970.
- Foley, J.D., van Dam, A., Feiner, S.K.,, and Hughes, J.F., *Computer Graphics: Principles and Practice*, second edition in C, Addison-Wesley.

© Marc Levoy