Point Processing and Filtering

CS194: Intro to Comp. Vision, and Comp. Photo

Angjoo Kanazawa and Alexei Efros, UC Berkeley, Fall 2022

About me

Angjoo Kanazawa

- First name pronounced like “Andrew”
- Grew up in Kobe, Japan

Saito et al. PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human ICCV 2019

Recitation & Project Party

Python Intro: **Thursday 11am-12pm**
BWW 1st floor 1216

Project 1 Party: **Friday 3-4:30pm**

All in the class calendar.

Piazza

What is an image?

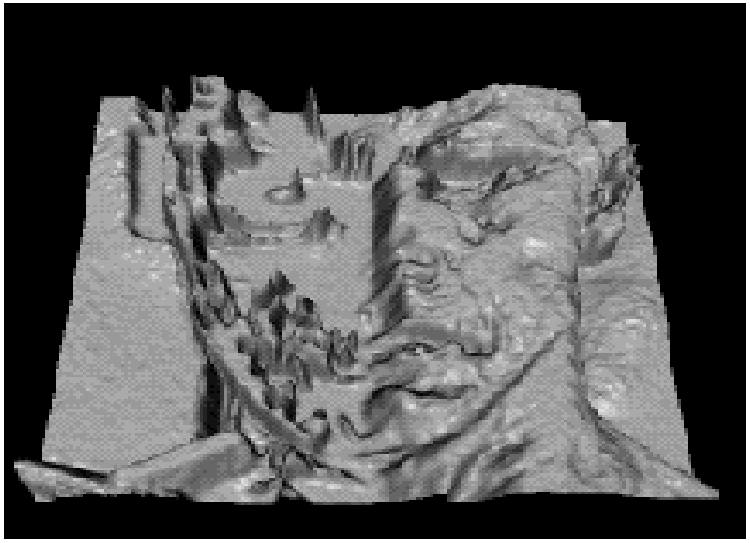
We can think of an **image** as a function, *f*, from R^2 to R:

- $f(x, y)$ gives the **intensity** at position (x, y)
- Realistically, we expect the image only to be defined over a rectangle, with a finite range:
  - $f: [a,b]\mathrm{x}[c,d] \rightarrow [0,1]$

A color image is just three functions pasted together. We can write this as a "vector-valued" function:

$$f(x,y) = \begin{bmatrix} r(x,y) \\ g(x,y) \\ b(x,y) \end{bmatrix}$$

Images as functions



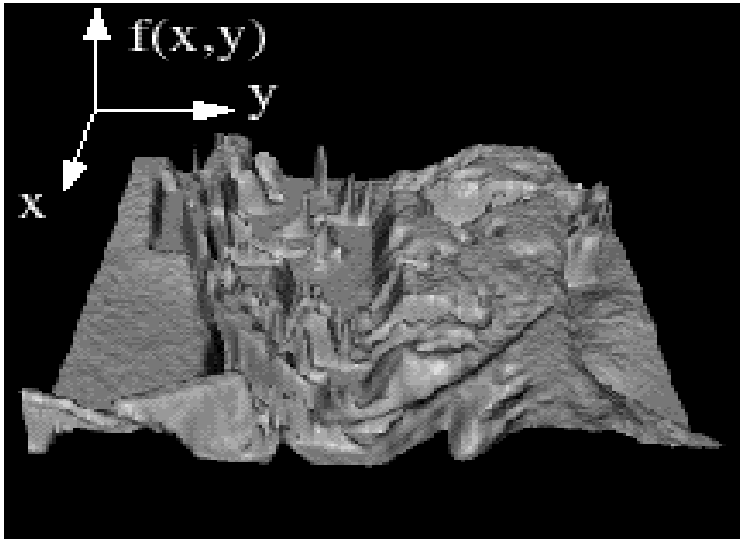
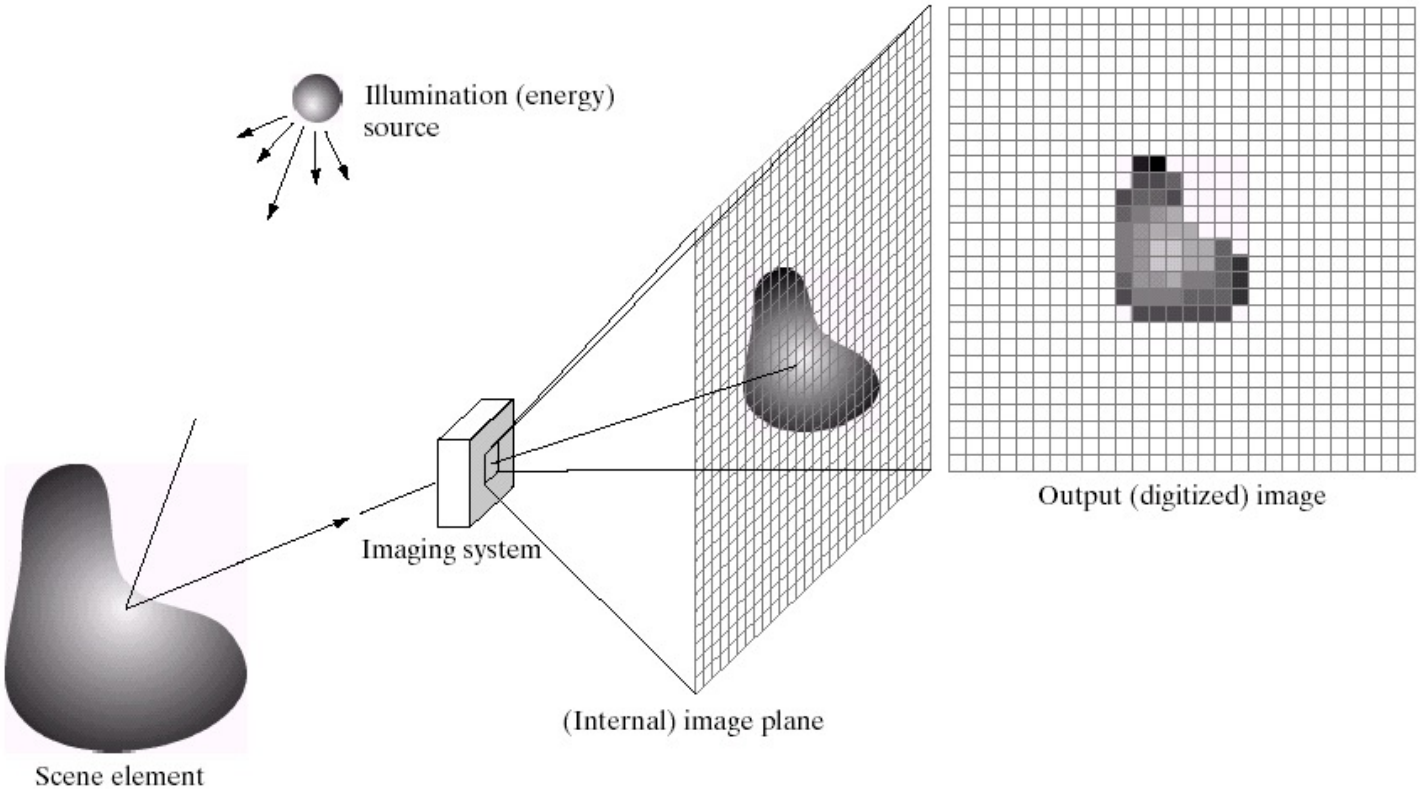


Image Formation



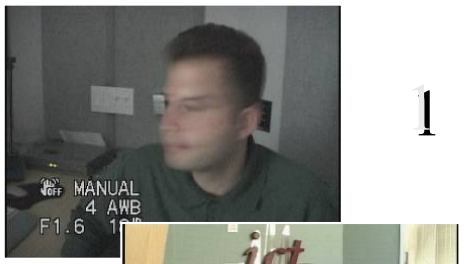
a
b c d e

FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

*f(x,y) = reflectance(x,y) * illumination(x,y)*

Reflectance in [0,1], illumination in [0,inf]

Problem: Dynamic Range

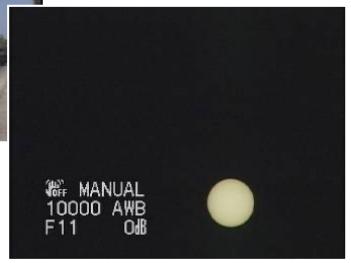


1

1500

25,000

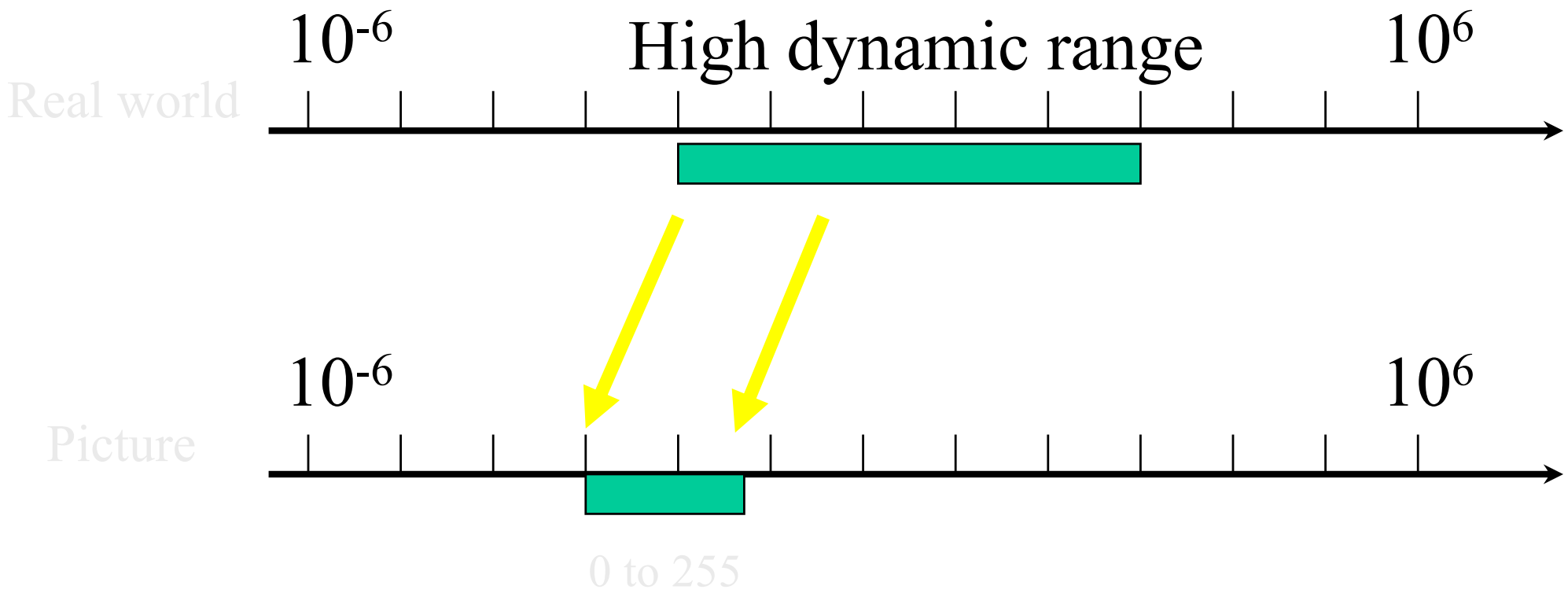
400,000



2,000,000,000

The real world is
High dynamic range

Long Exposure



Short Exposure

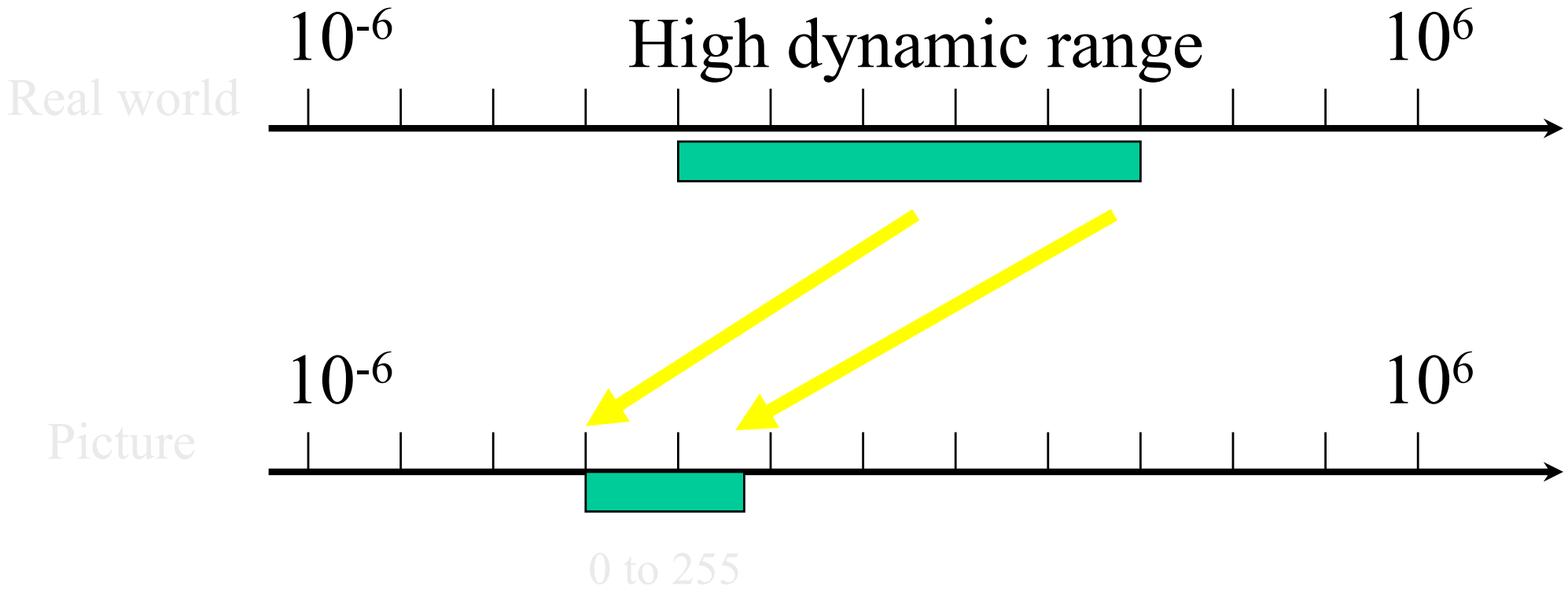
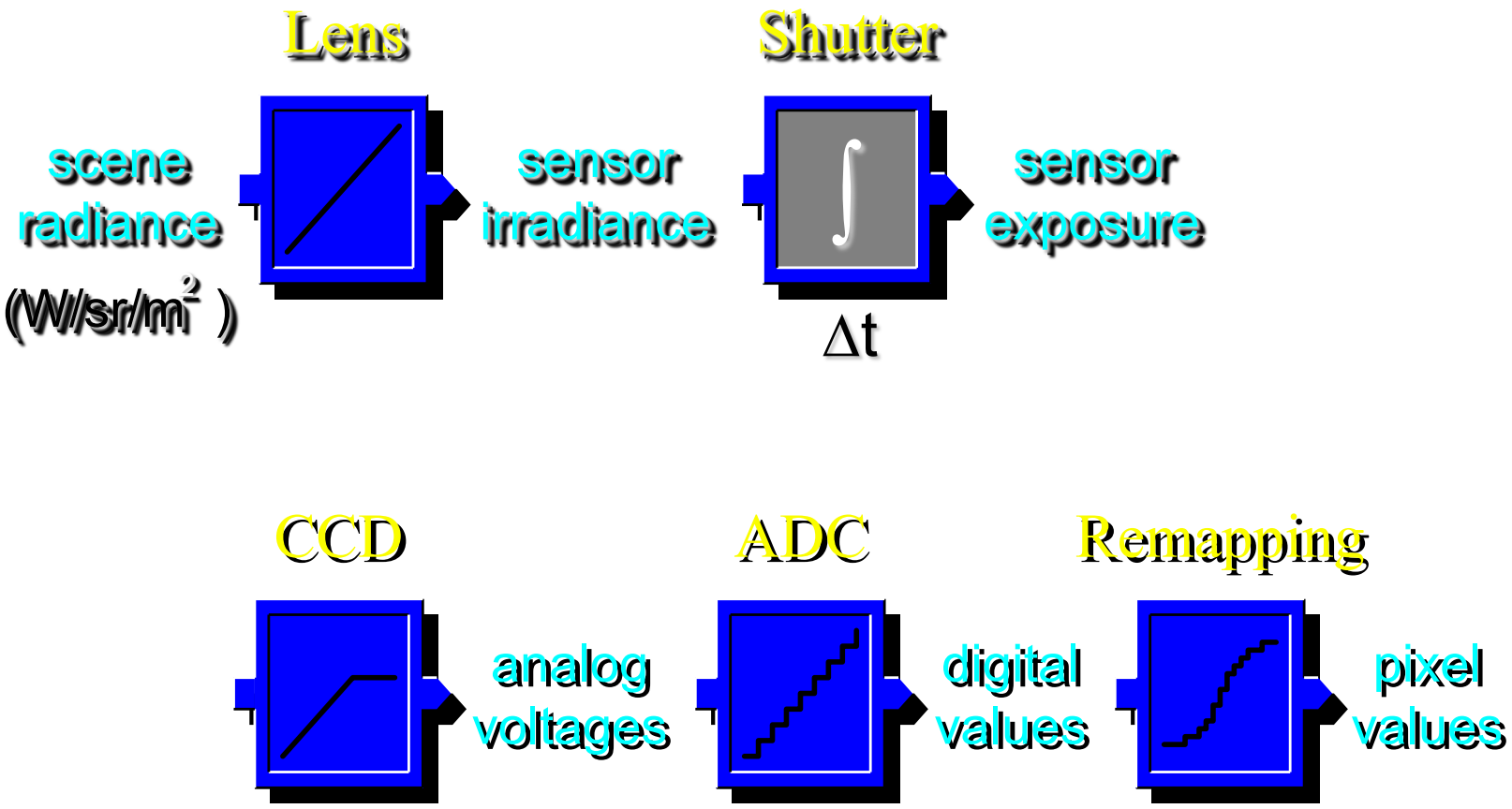


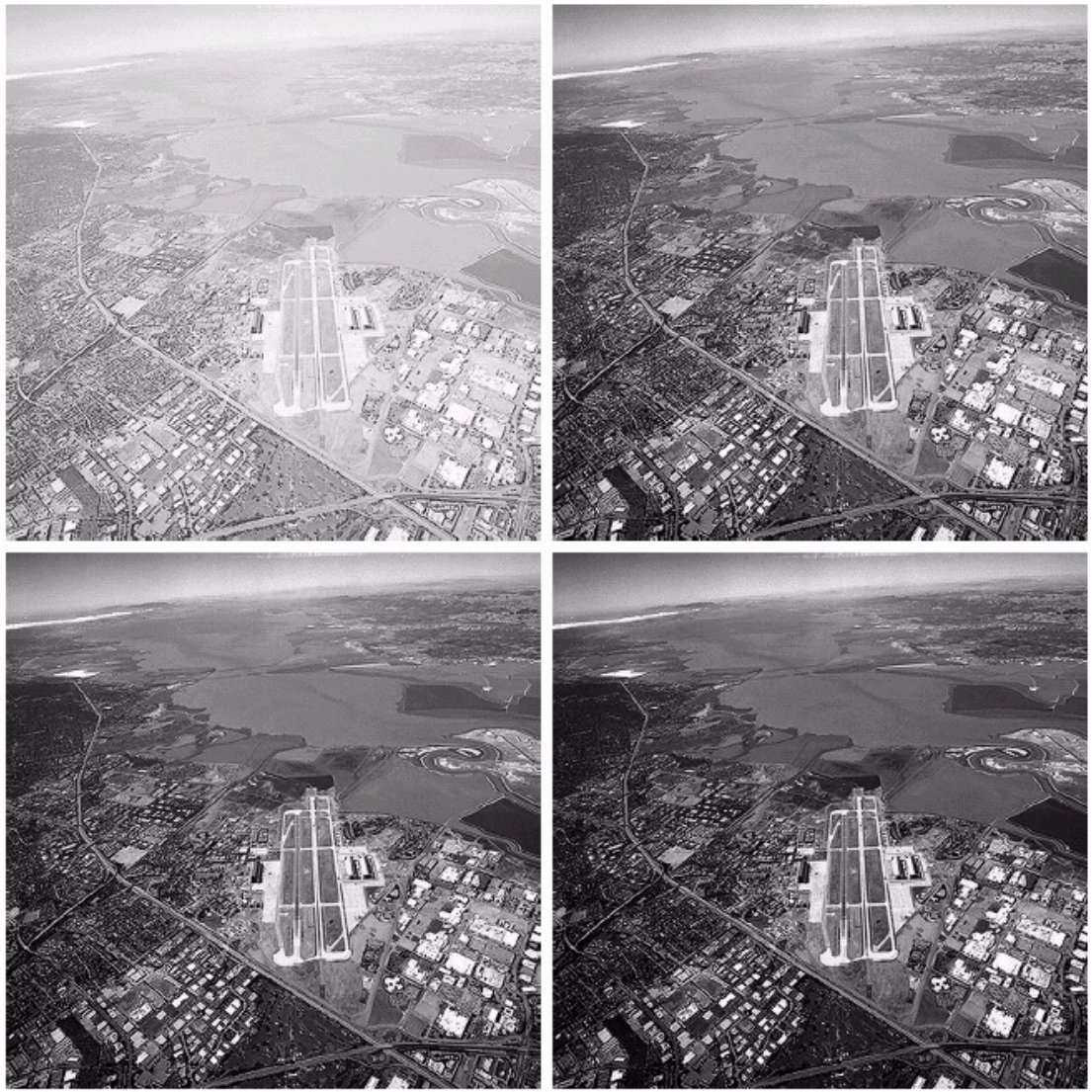
Image Acquisition Pipeline



Simple Point Processing: Enhancement

a b
c d

FIGURE 3.9
(a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0$, and 5.0, respectively. (Original image for this example courtesy of NASA.)



Power-law transformations

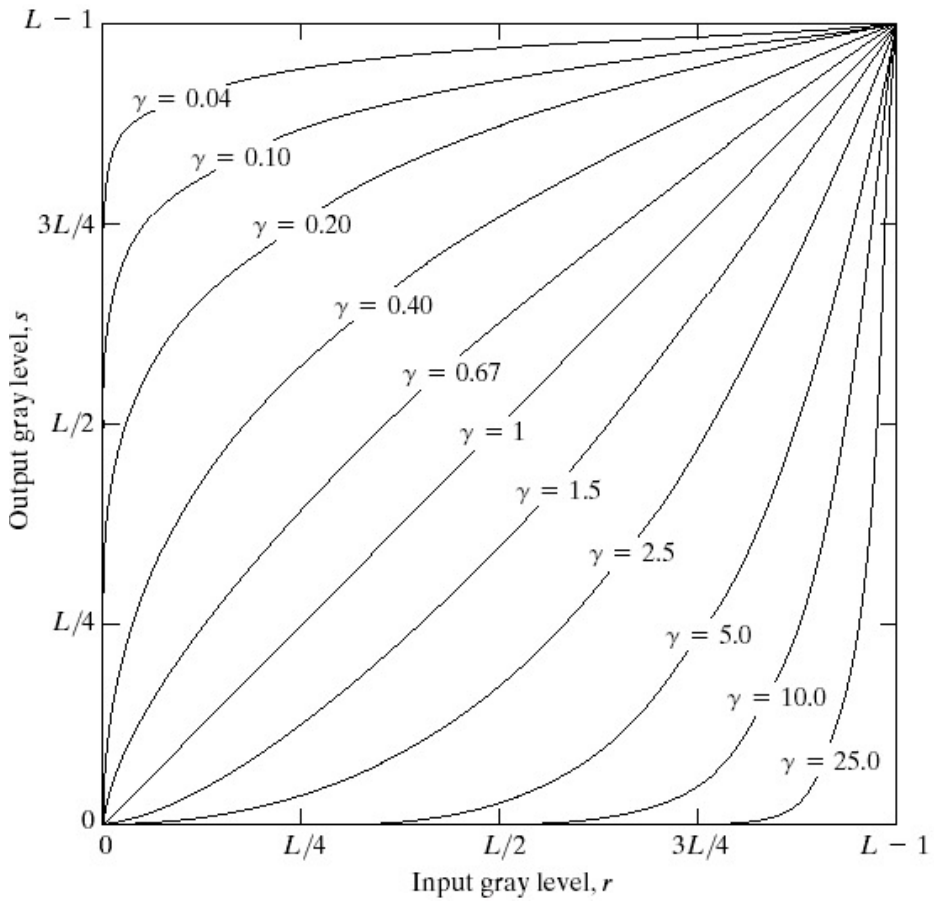
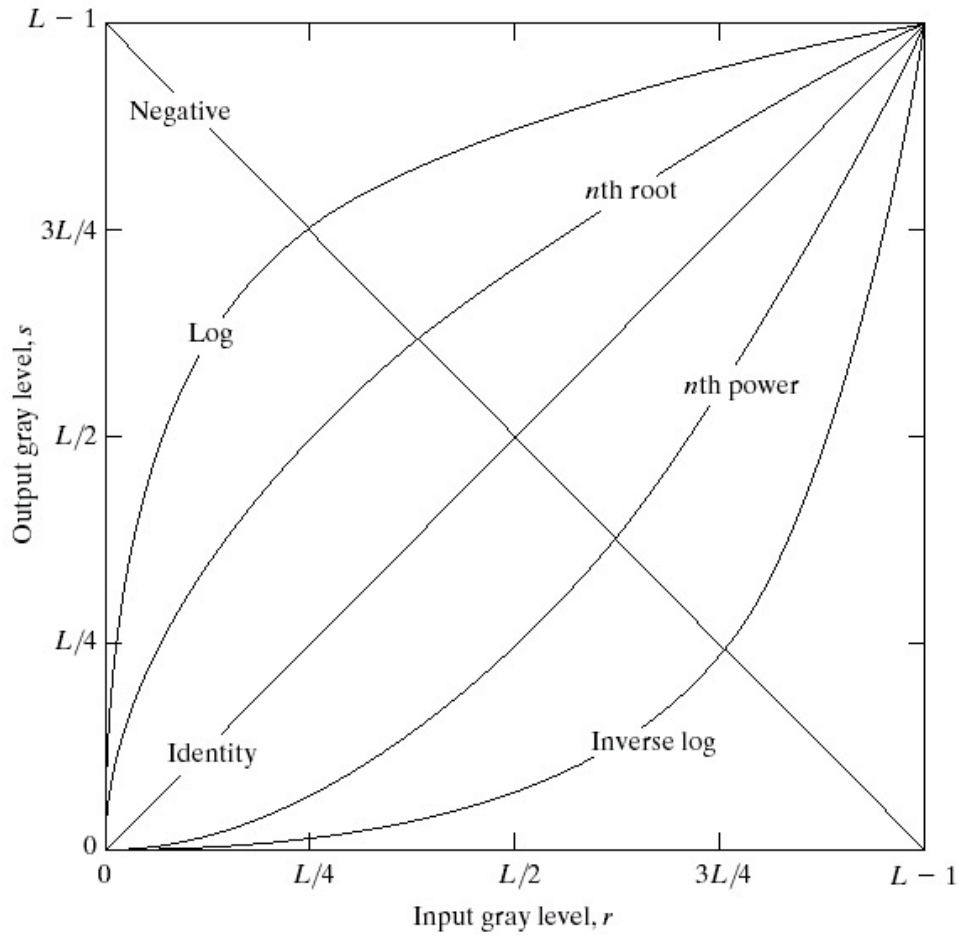


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

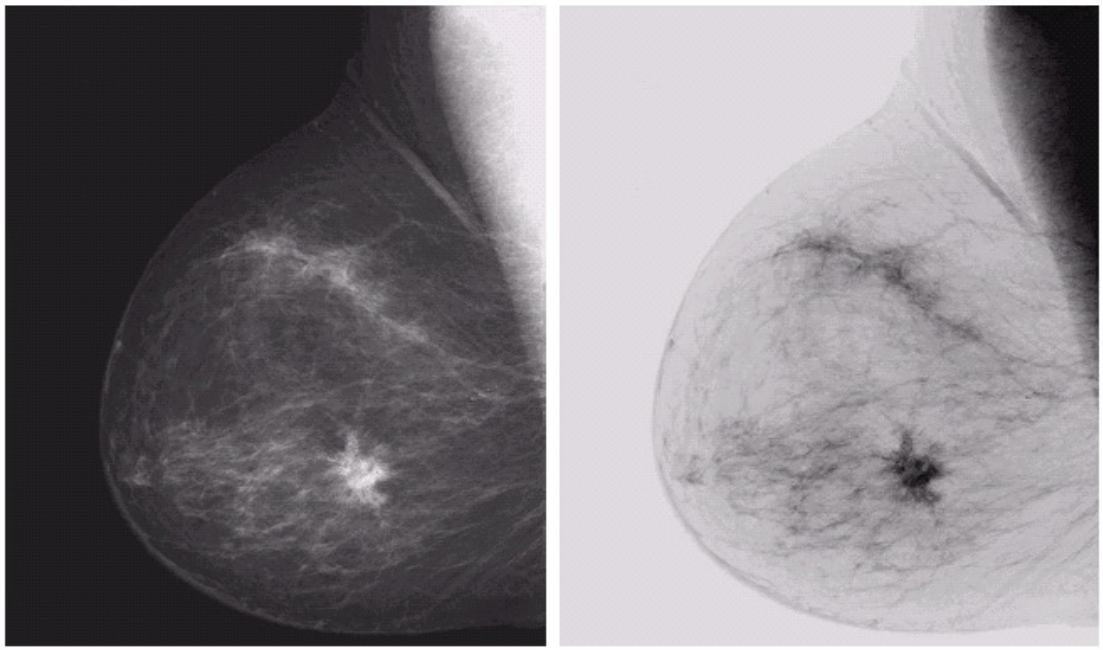
$$s = cr^{\gamma}$$

Basic Point Processing

FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.



Negative



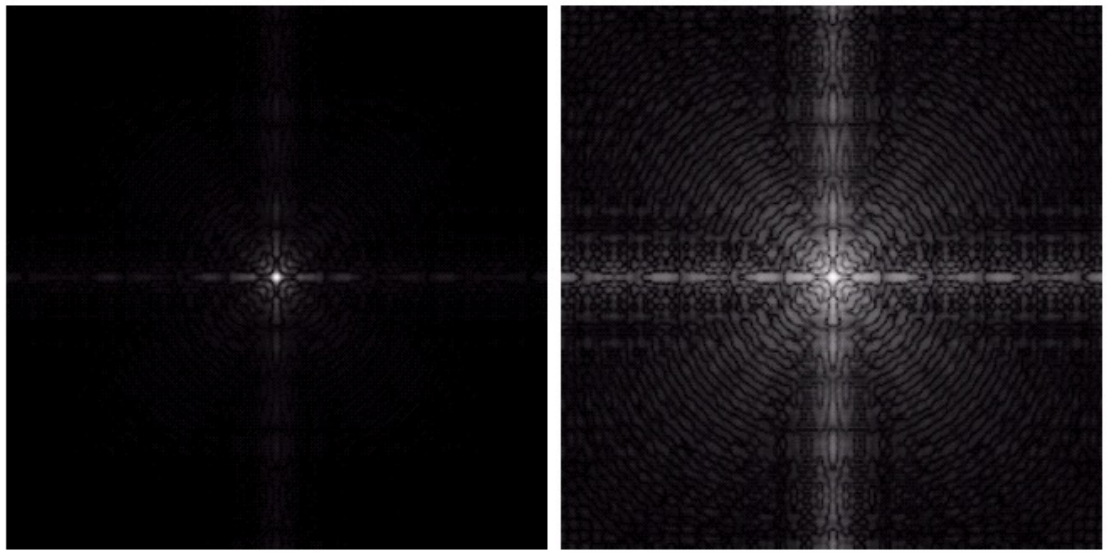
a b

FIGURE 3.4 (a) Original digital mammogram. (b) Negative image obtained using the negative transformation in Eq. (3.2-1). (Courtesy of G.E. Medical Systems.)

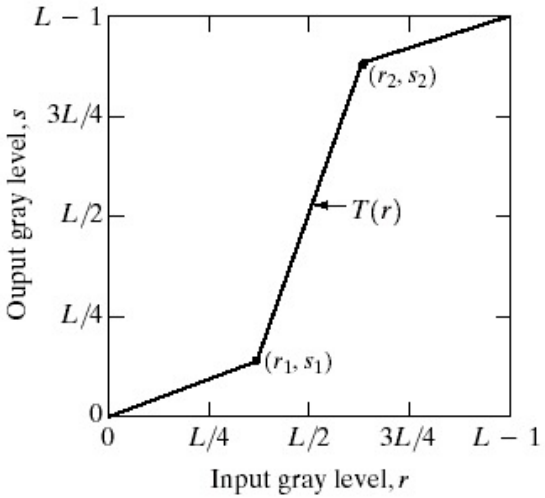
Log

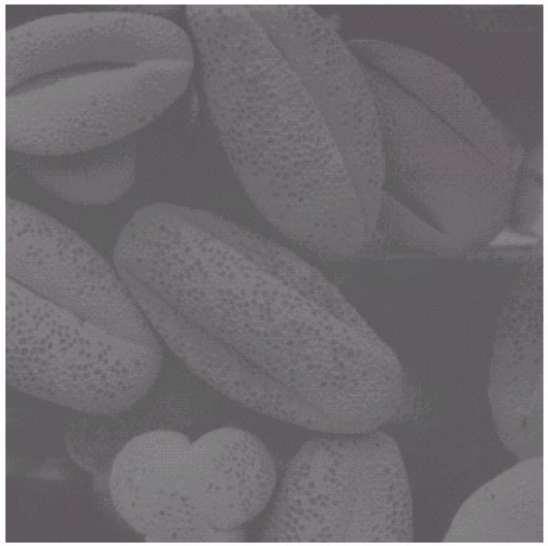
a b

FIGURE 3.5
(a) Fourier spectrum.
(b) Result of applying the log transformation given in Eq. (3.2-2) with $c = 1$.



Contrast Stretching

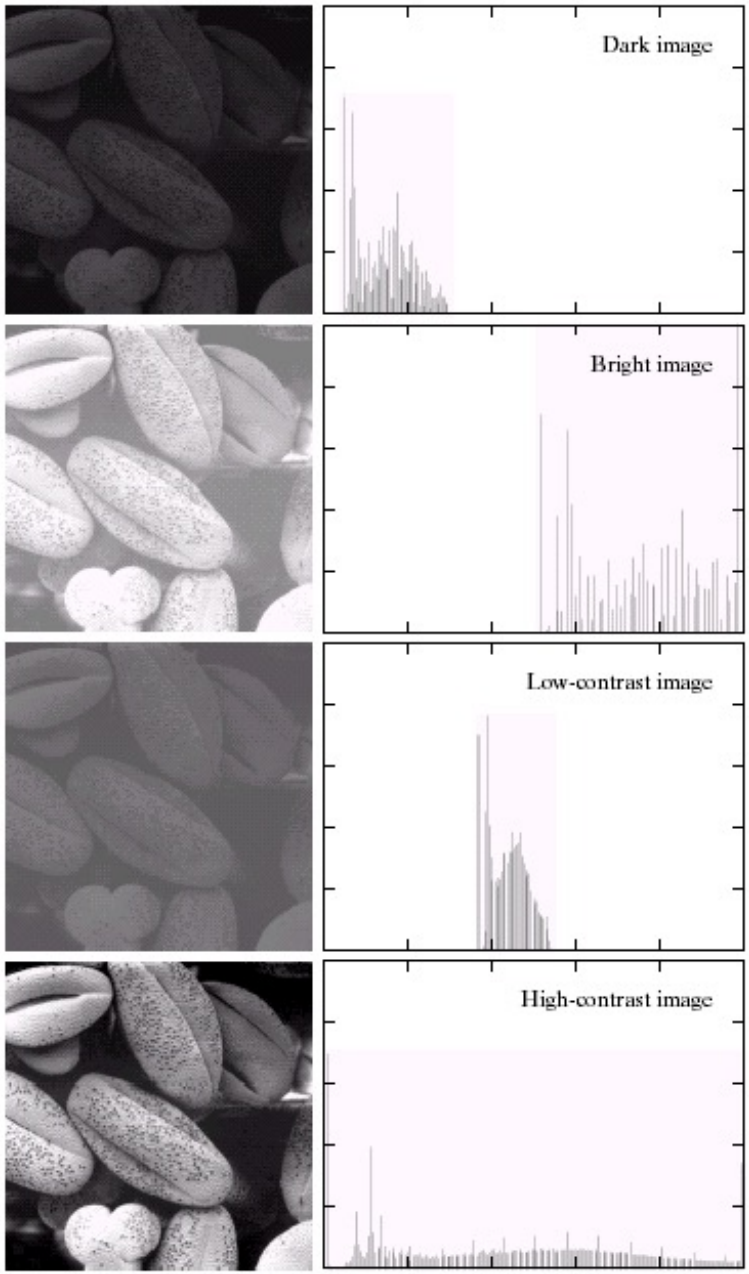




a b
c d

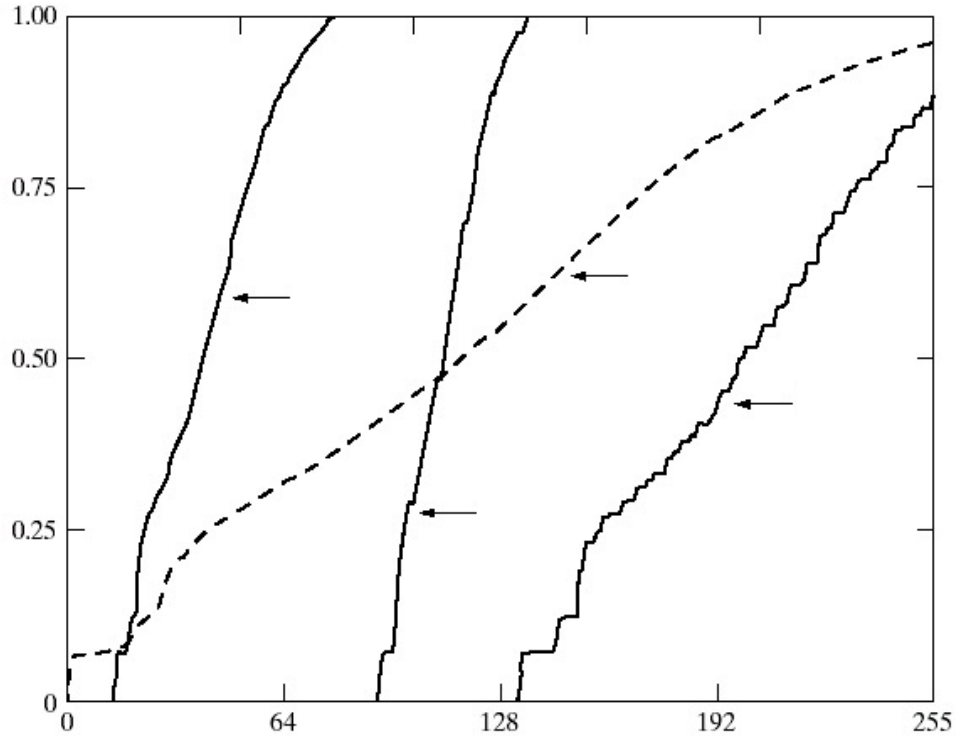
FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Image Histograms



x-axis: bins

y-axis: count



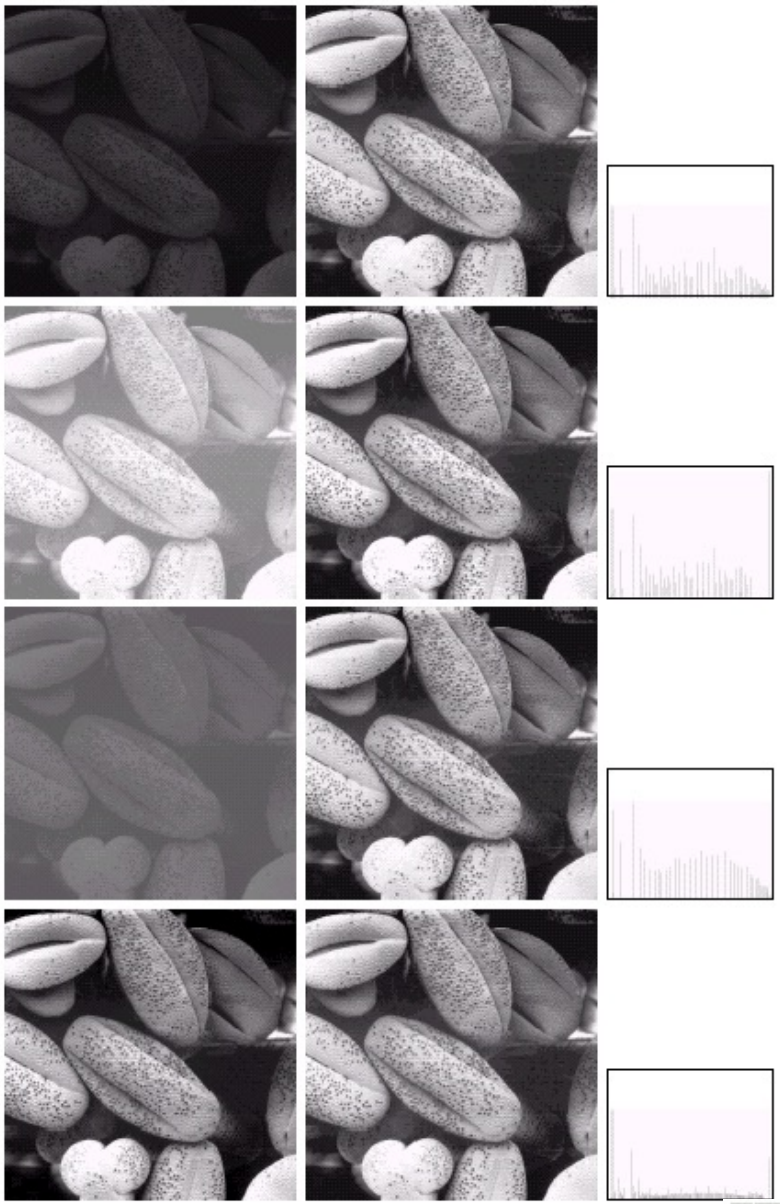
Cumulative Histograms

$s = T(r)$

a b

FIGURE 3.15 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

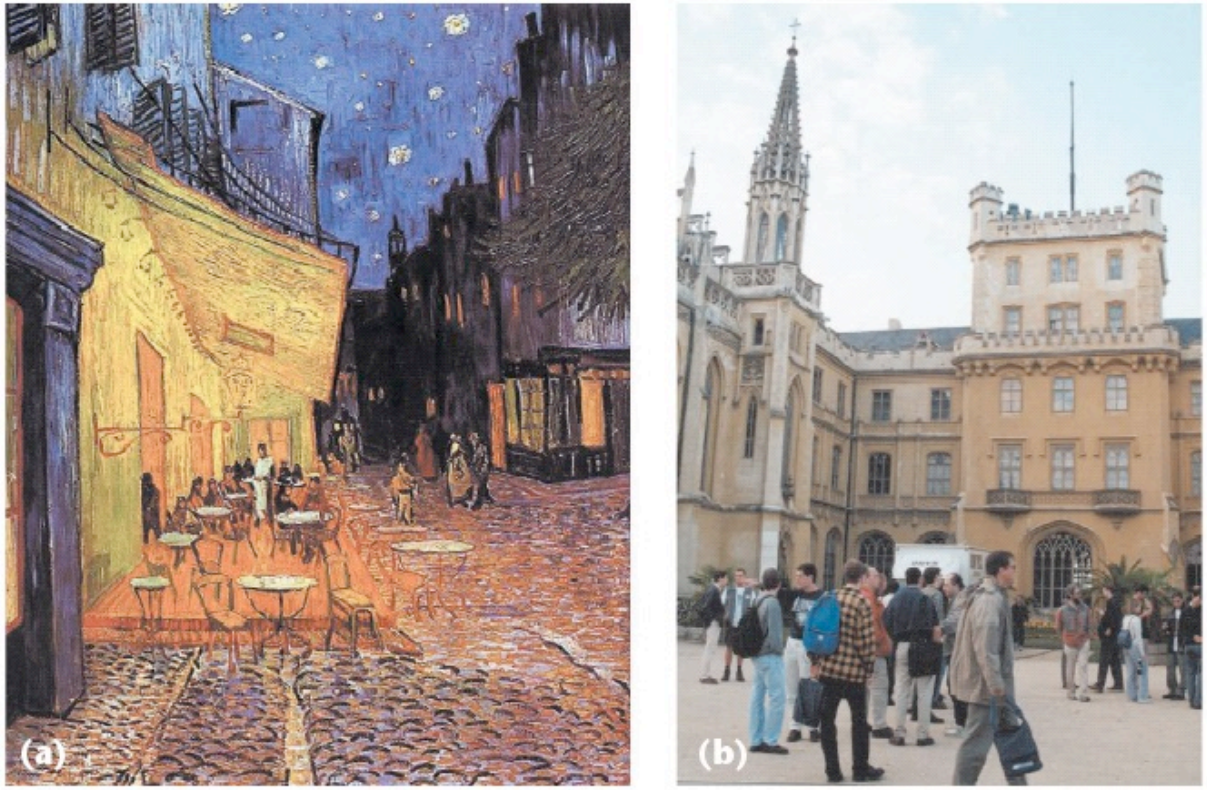
Histogram Equalization



a b c

FIGURE 3.17 (a) Images from Fig. 3.15. (b) Results of histogram equalization. (c) Corresponding histograms.

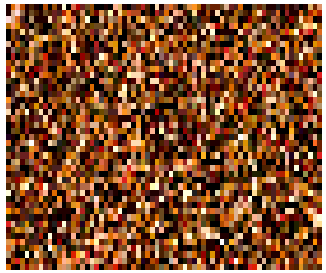
Color Transfer [Reinhard, et al, 2001]



Erik Reinhard, Michael Ashikhmin, Bruce Gooch, Peter Shirley, **Color Transfer** between Images. *IEEE Computer Graphics and Applications*, 21(5), pp. 34–41. September 2001.

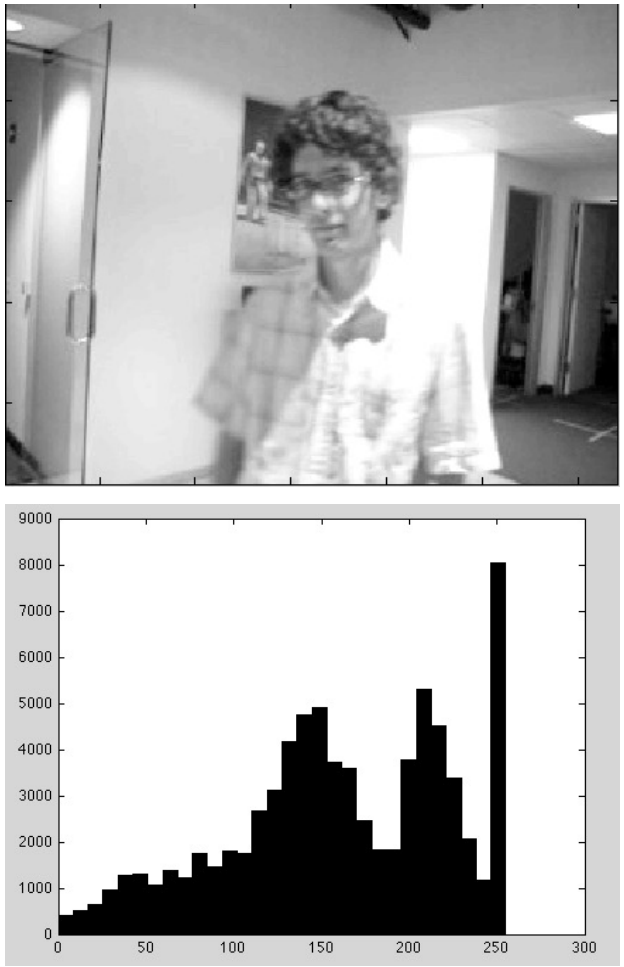
Limitations of Point Processing

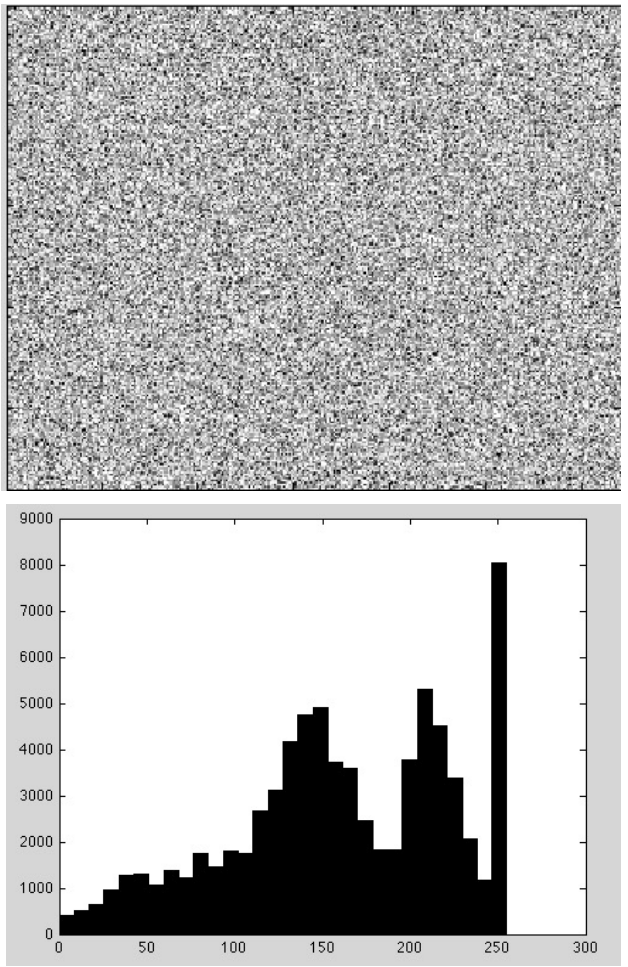
Q: What happens if I reshuffle all pixels within the image?



A: It's histogram won't change. No point processing will be affected…

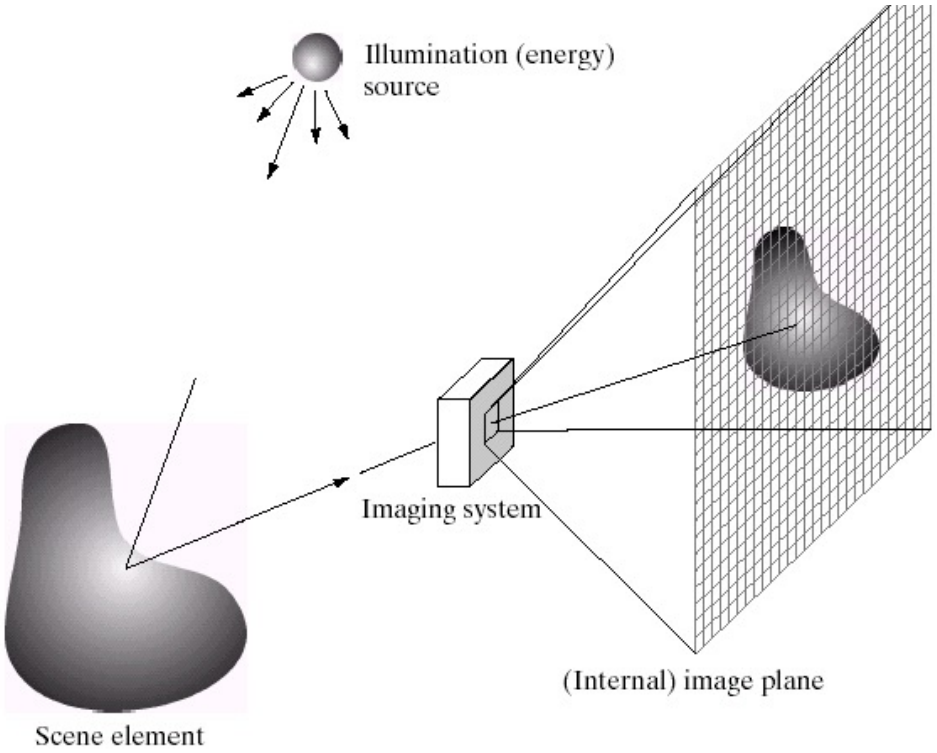
Limitations of Point Processing…

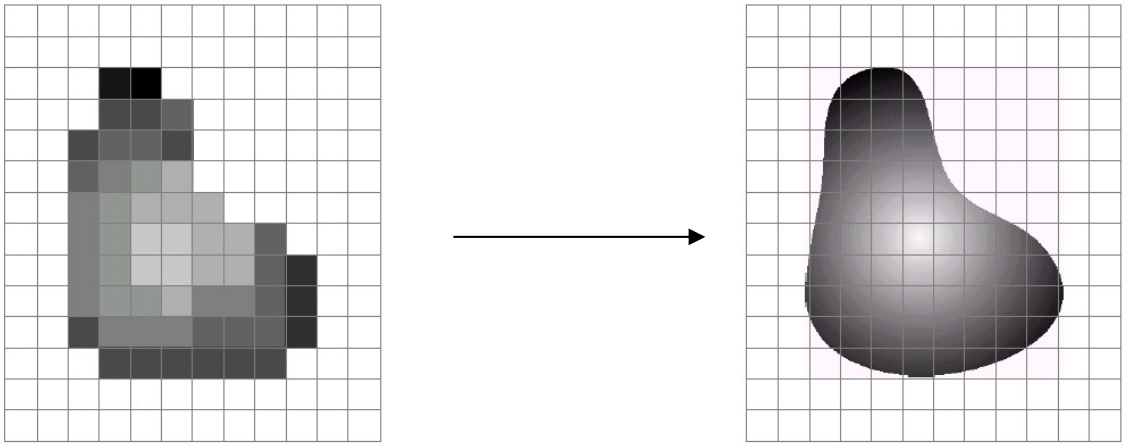




Slide by Erik Learned-Miller

Sampling and Reconstruction



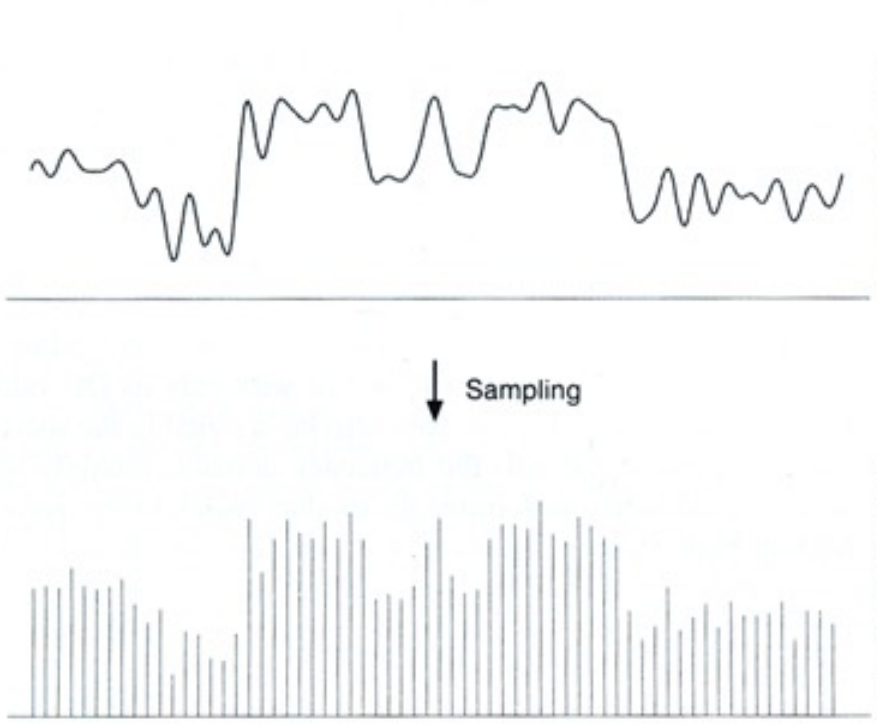


Sampled representations

How to store and compute with continuous functions?

Common scheme for representation: samples

- write down the function's values at many points

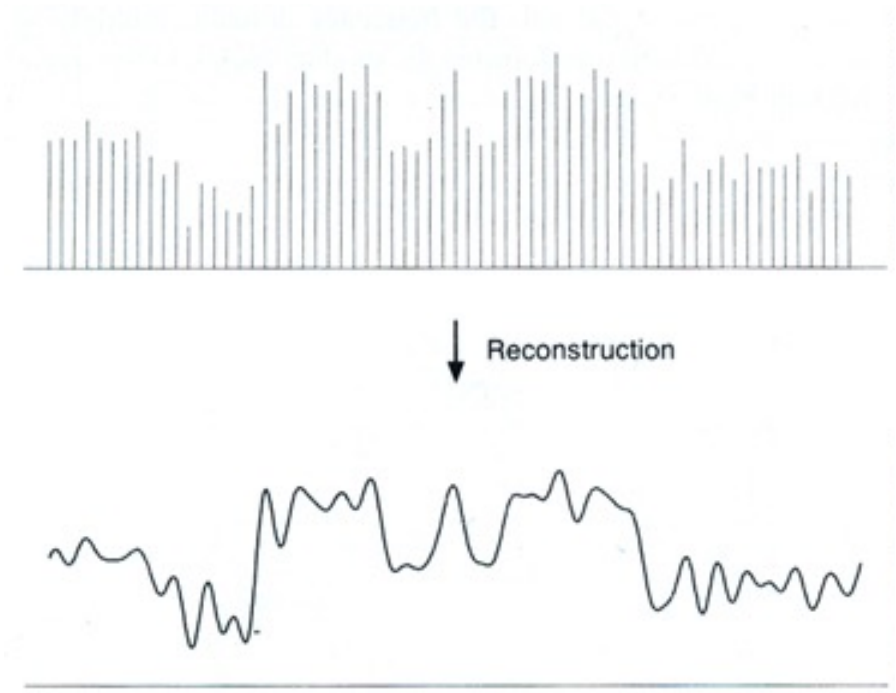


[FvDFH fig.14.14b / Wolberg]

Reconstruction

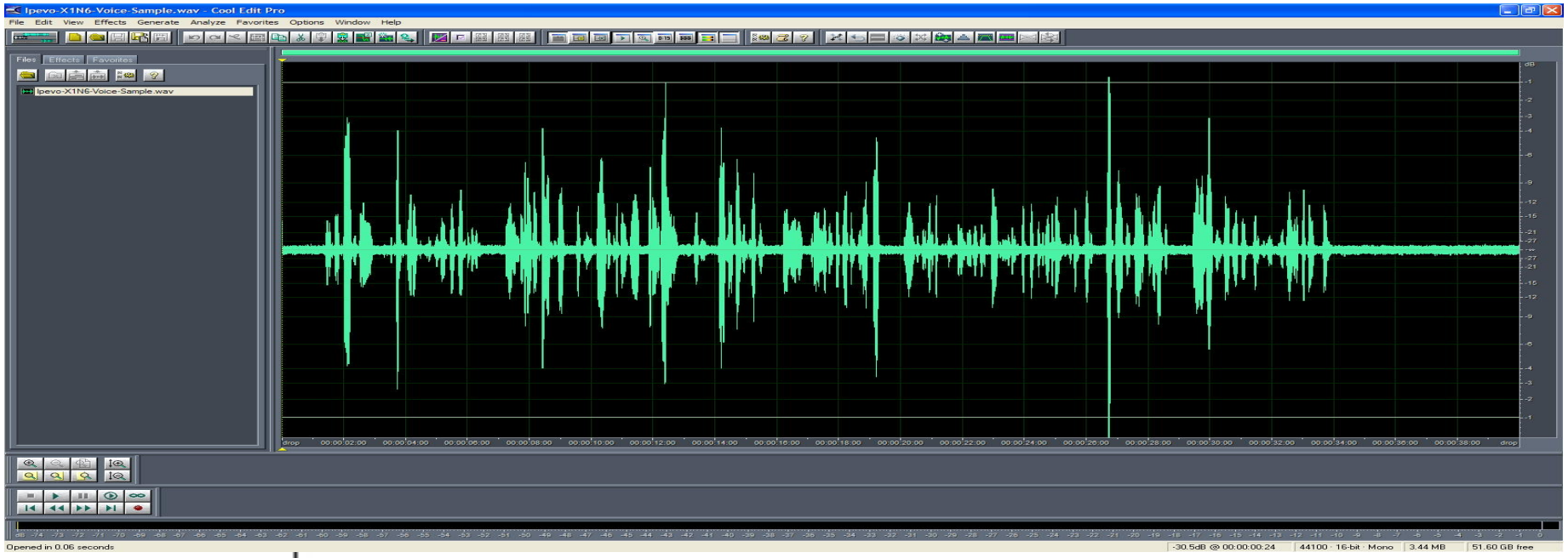
Making samples back into a continuous function

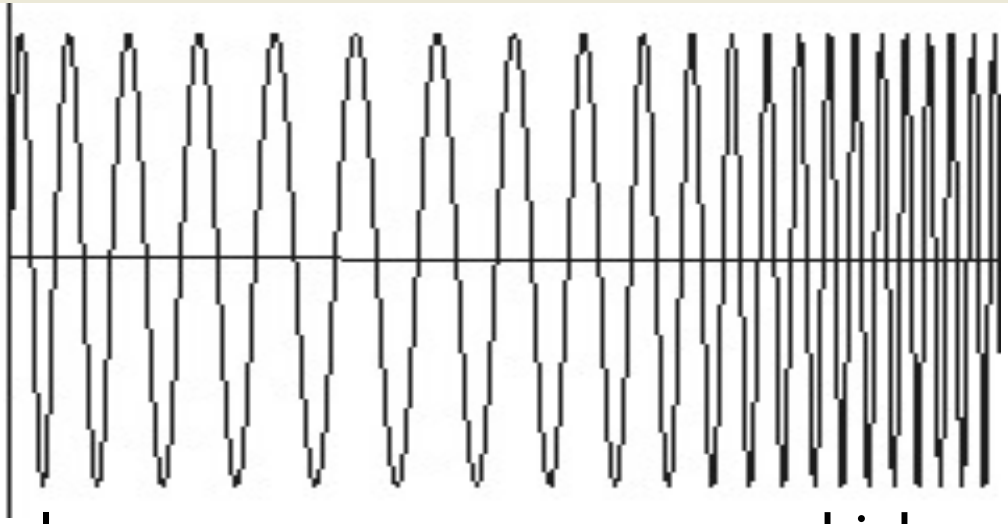
- for output (need realizable method)
- for analysis or processing (need mathematical method)
- amounts to “guessing” what the function did in between



[FvDFH fig.14.14b / Wolberg]

1D Example: Audio





low high

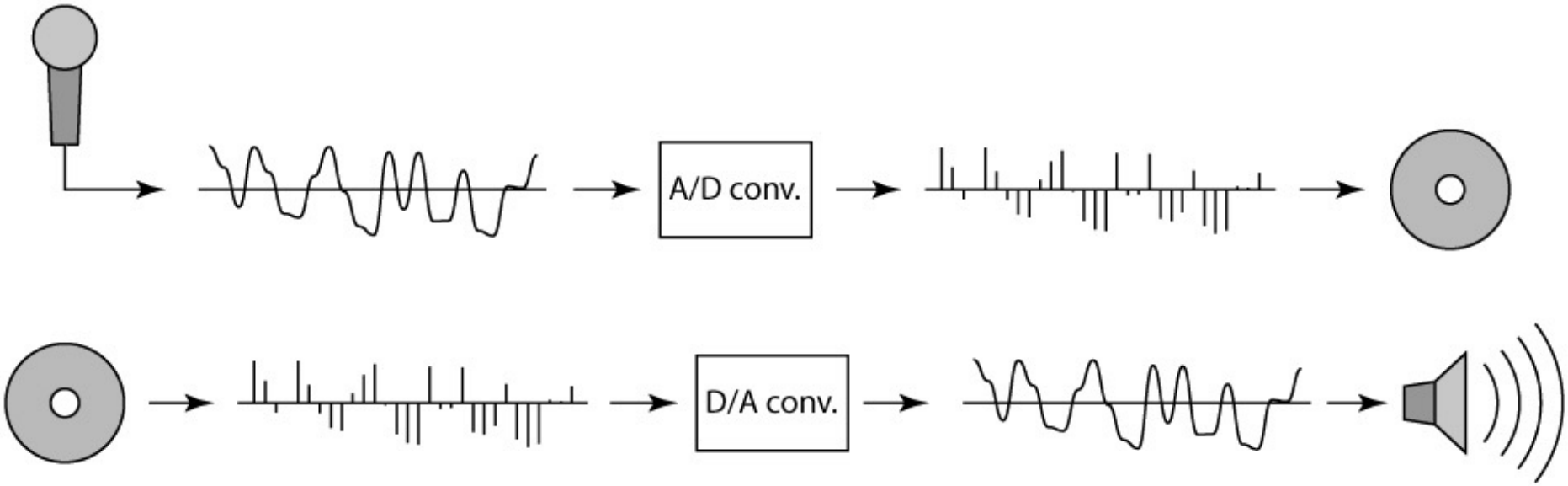
frequencies

Sampling in digital audio

Recording: sound to analog to samples to disc

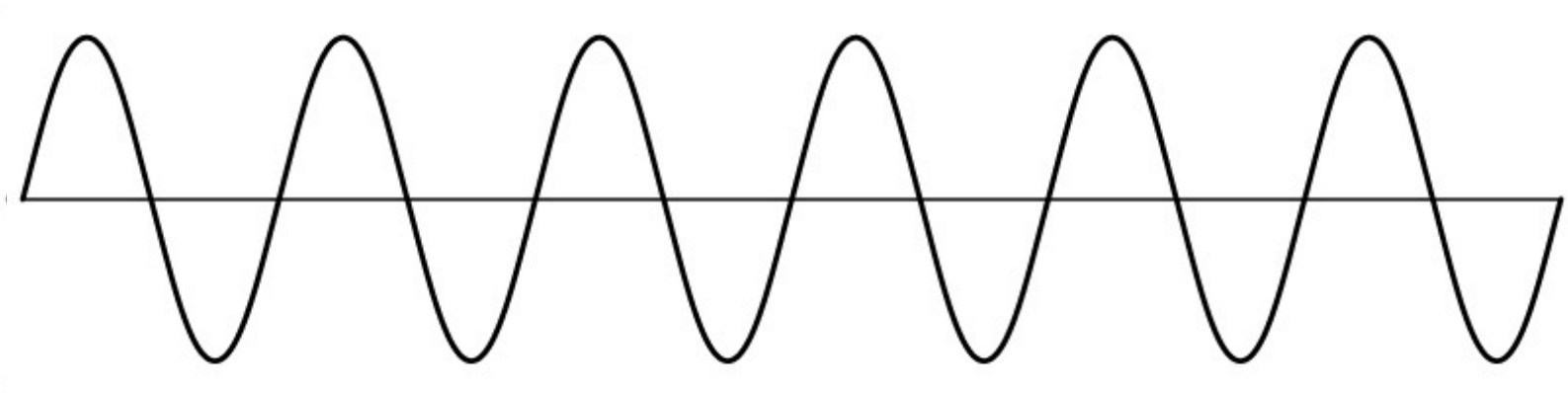
Playback: disc to samples to analog to sound again

- how can we be sure we are filling in the gaps correctly?



Sampling and Reconstruction

Simple example: a sign wave

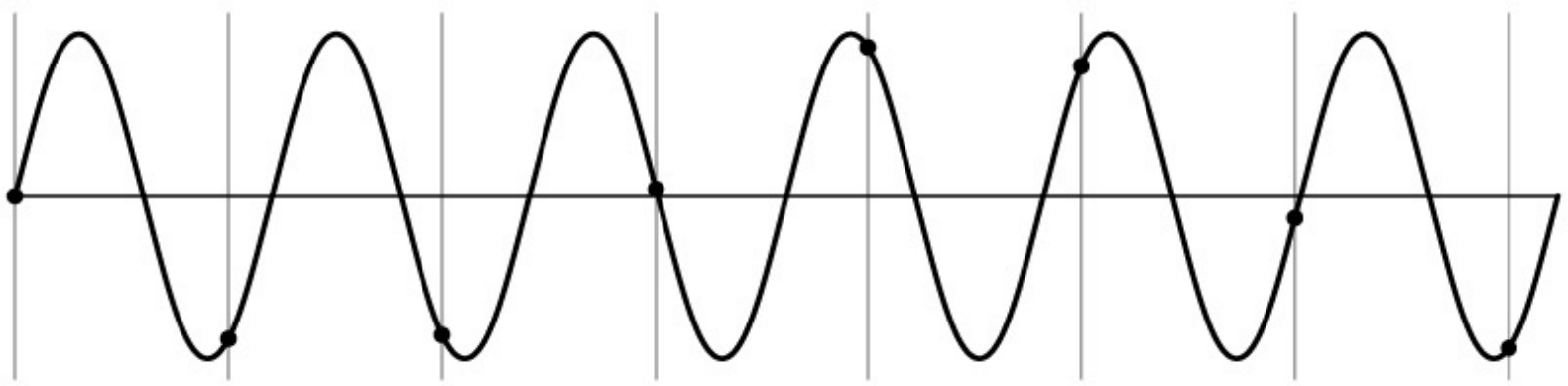


Undersampling

What if we “missed” things between the samples?

Simple example: undersampling a sine wave

- unsurprising result: information is lost

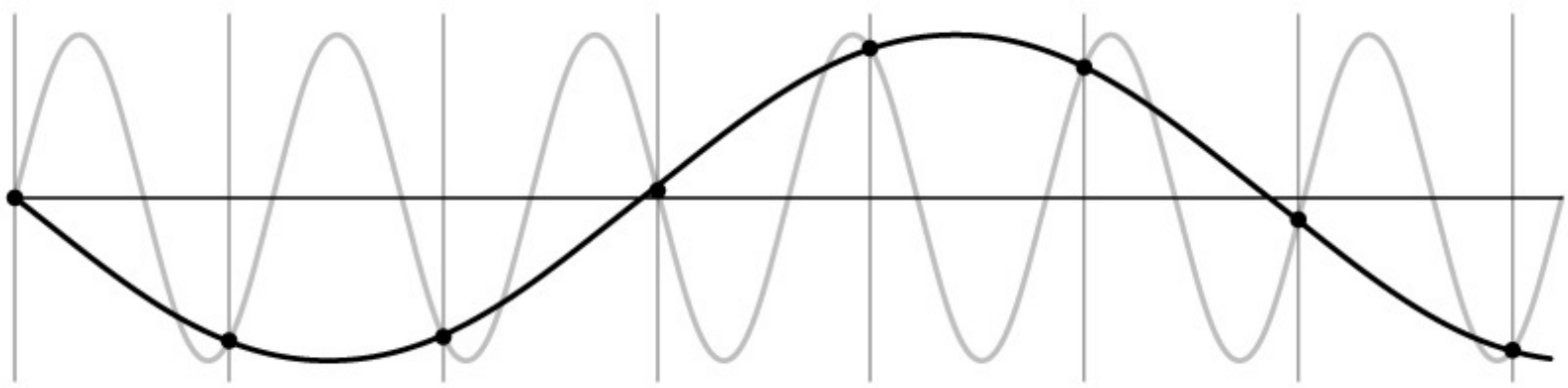


Undersampling

What if we “missed” things between the samples?

Simple example: undersampling a sine wave

- unsurprising result: information is lost
- surprising result: indistinguishable from lower frequency

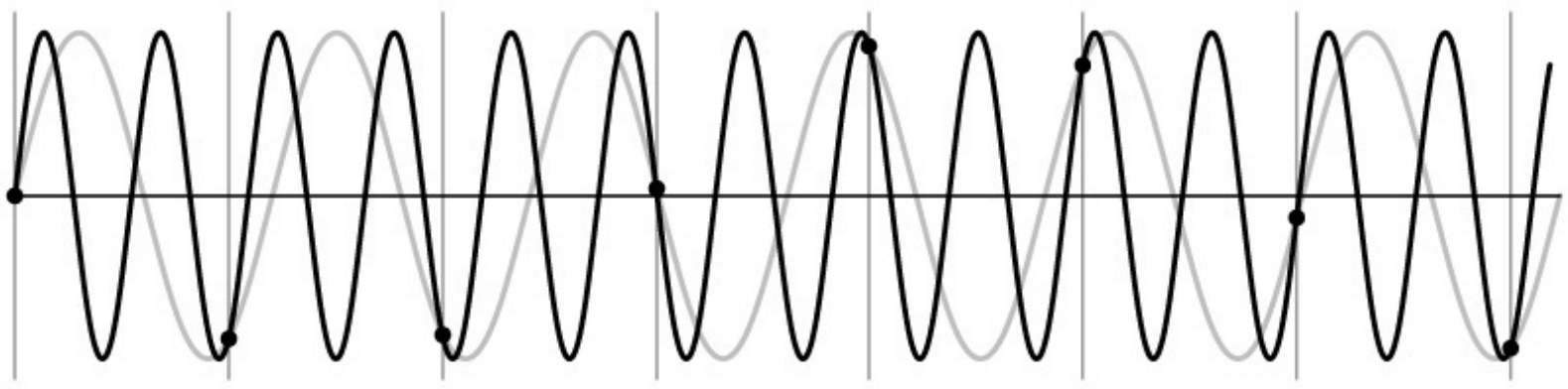


Undersampling

What if we “missed” things between the samples?

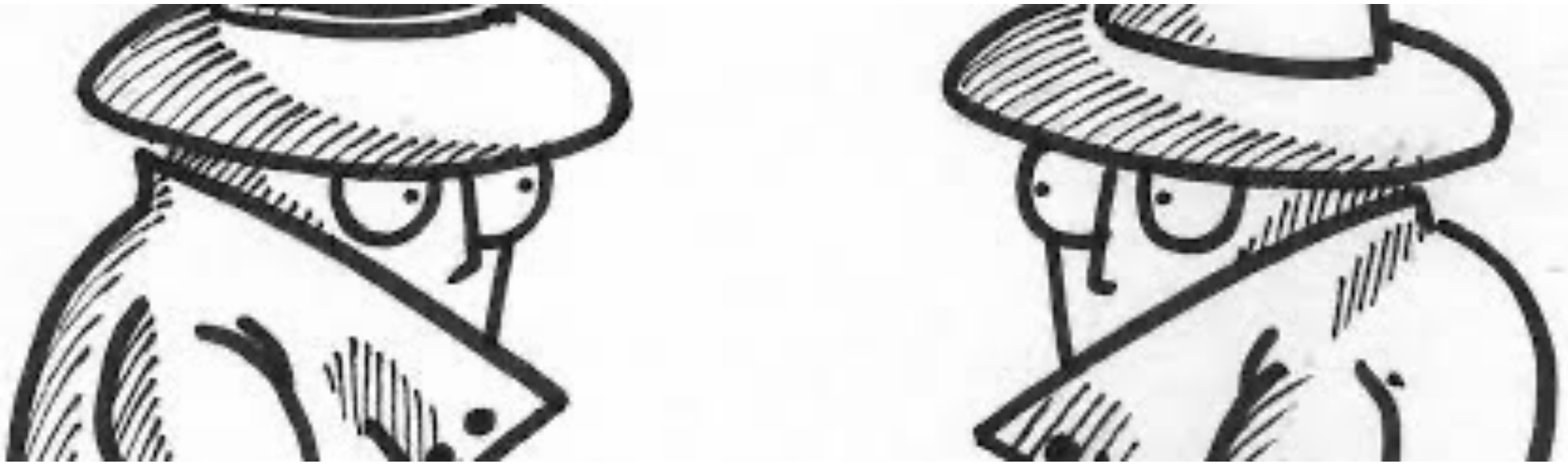
Simple example: undersampling a sine wave

- unsurprising result: information is lost
- surprising result: indistinguishable from lower frequency
- also, was always indistinguishable from higher frequencies
- *aliasing*: signals “traveling in disguise” as other frequencies



Aliasing!

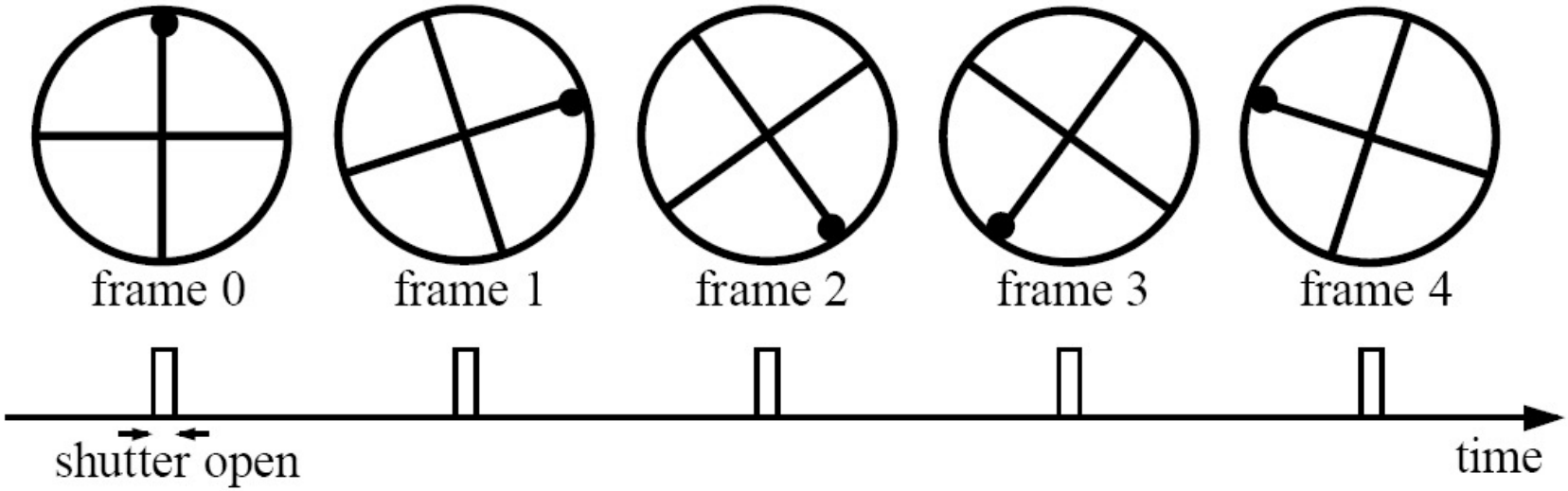
Alias = Pseudonym, False Identity



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):

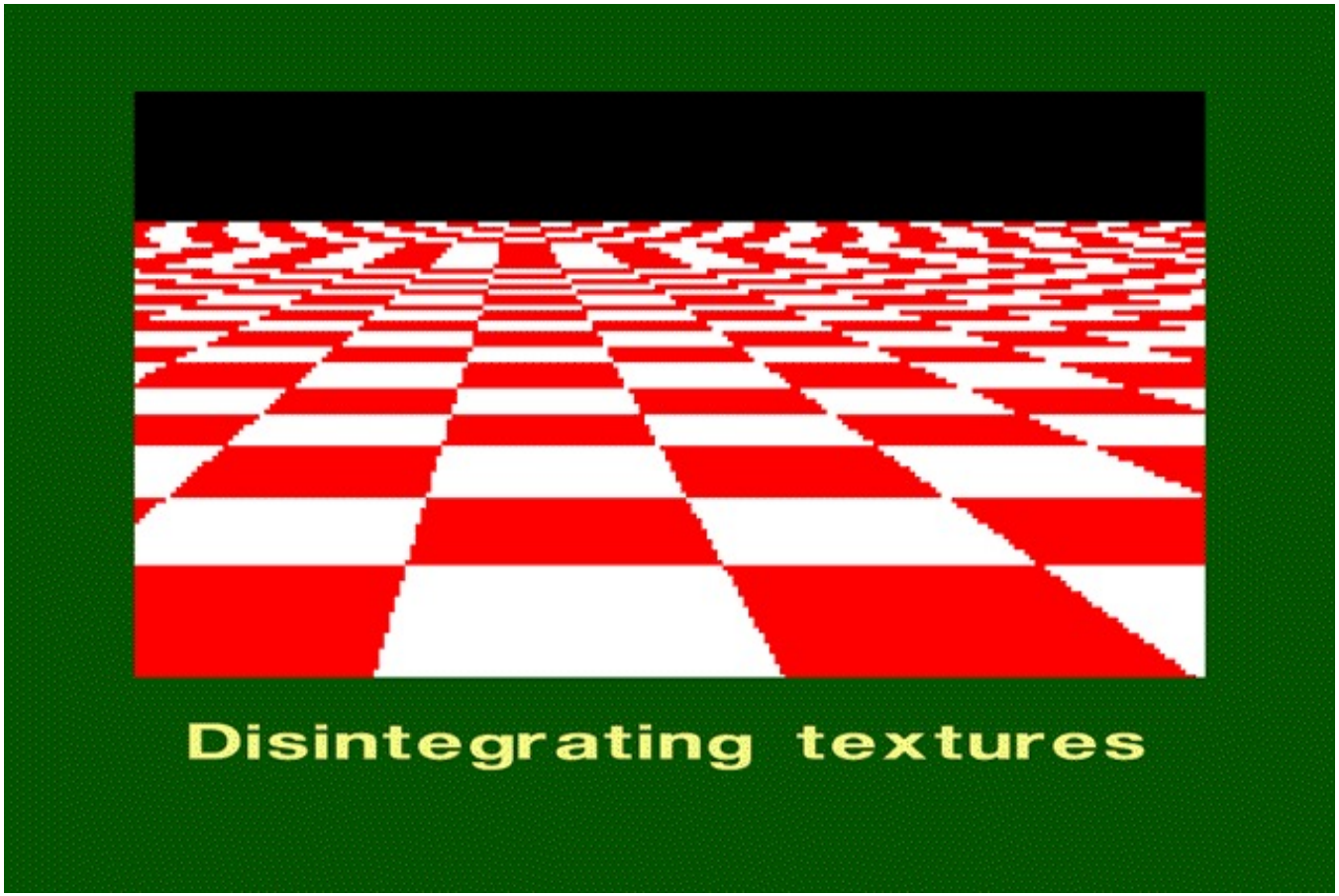


Without dot, wheel appears to be rotating slowly backwards! (counterclockwise)

Slide by Steve Seitz

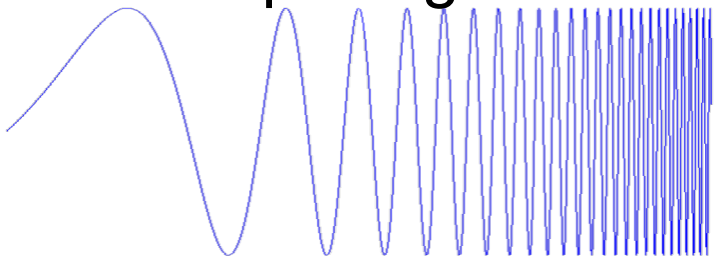

TEACH ME

Aliasing in images

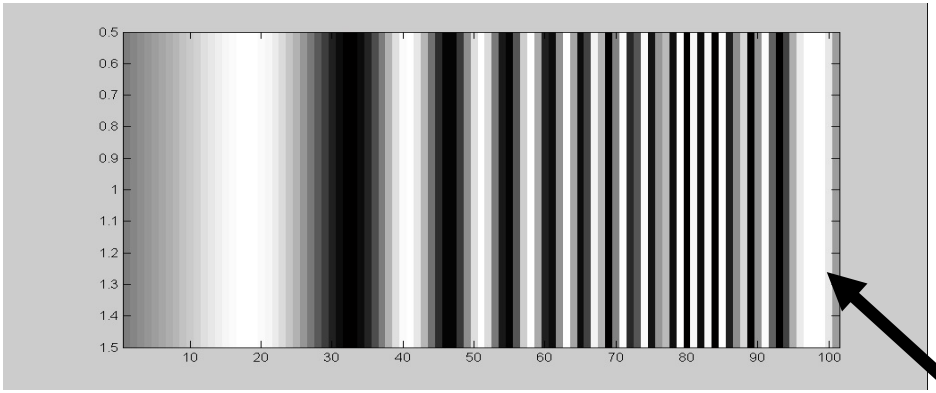


What's happening?

Input signal:

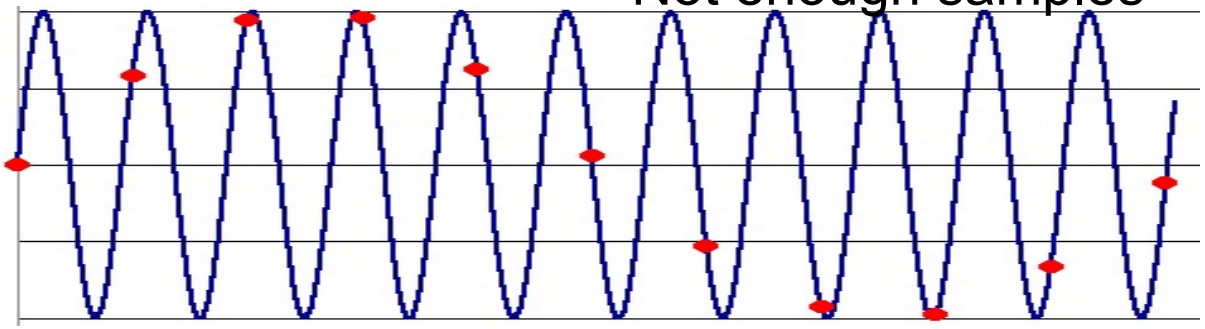


Plot as image:



x = 0:.05:5; imagesc(sin((2.^x).*x))

Alias!
Not enough samples



Antialiasing

What can we do about aliasing?

Sample more often

- Join the Mega-Pixel craze of the photo industry
- But this can’t go on forever

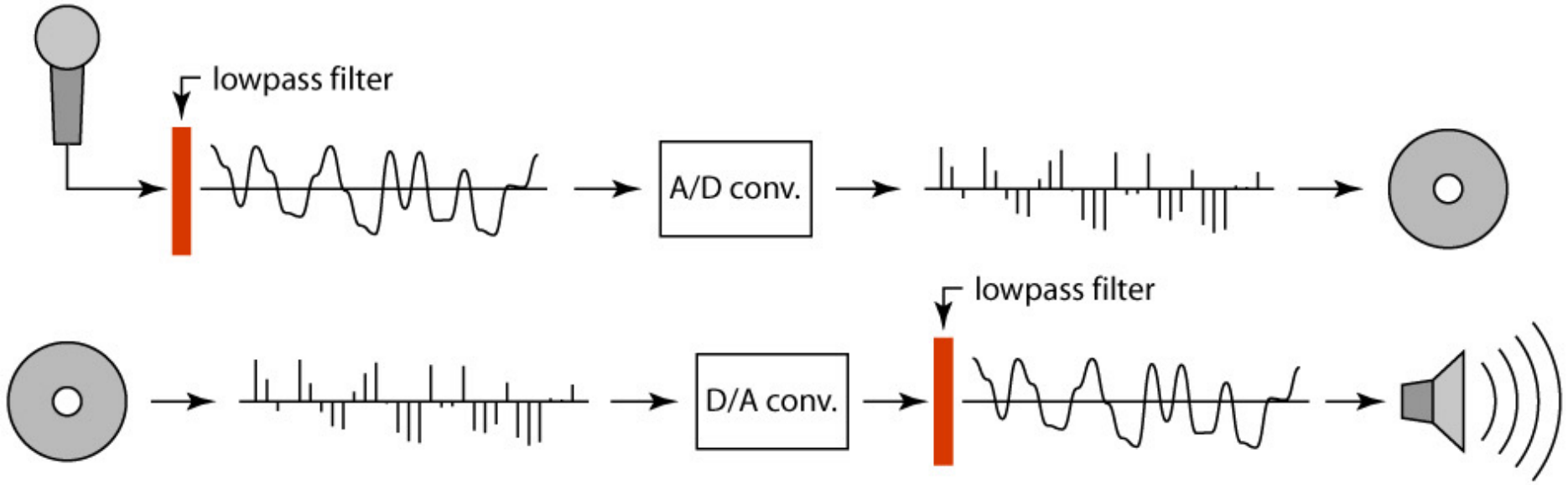
Make the signal less “wiggly”

- Get rid of some high frequencies
- Will loose information
- But it’s better than aliasing

Preventing aliasing

Introduce lowpass filters:

- remove high frequencies leaving only safe, low frequencies
- choose lowest frequency in reconstruction (disambiguate)



Linear filtering: a key idea

Transformations on signals; e.g.:

- bass/treble controls on stereo
- blurring/sharpening operations in image editing
- smoothing/noise reduction in tracking

Key properties

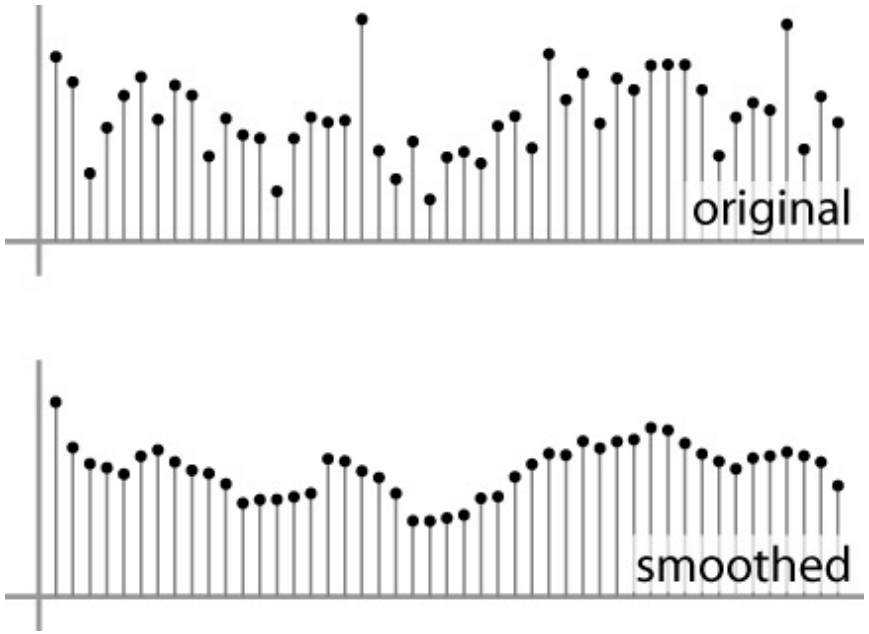
- linearity: filter(f + g) = filter(f) + filter(g)
- shift invariance: behavior invariant to shifting the input
  - delaying an audio signal
  - sliding an image around

Can be modeled mathematically by *convolution*

Moving Average

basic idea: define a new function by averaging over a sliding window

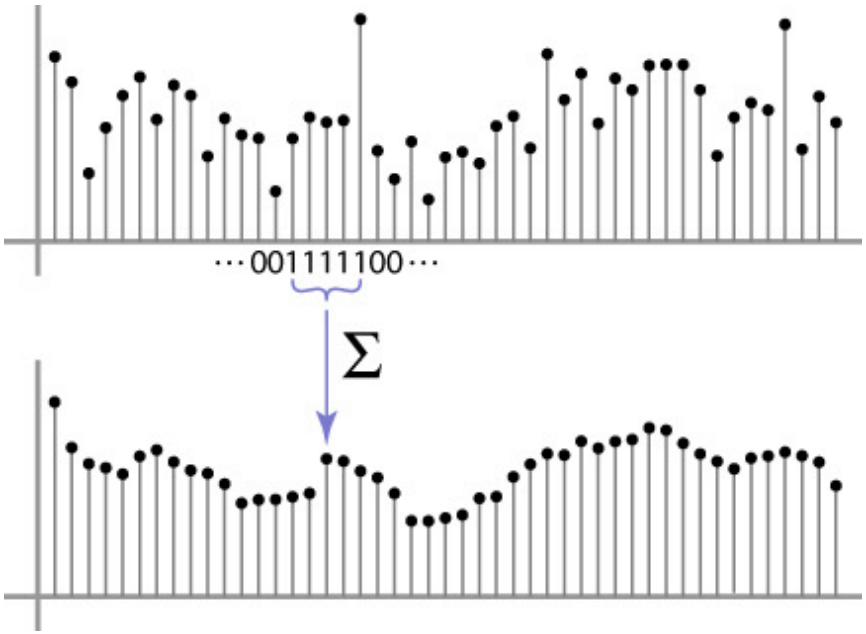
a simple example to start off: smoothing



Moving Average

Can add weights to our moving average

Weights […, 0, 1, 1, 1, 1, 1, 0, …] / 5



Cross-correlation

Let F be the image, H be the kernel (of size 2*k*+1 x 2*k*+1), and G be the output image

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i+u, j+v]$$

This is called a **cross-correlation** operation:

$$G = H \otimes F$$

- Can think of as a "dot product" between local neighborhood and kernel for each pixel

In 2D: box filter

$h[\cdot,\cdot]$

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

Slide credit: David Lowe (UBC)

Image filtering

$h[\cdot,\cdot]$

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[.,.]$

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

$g[.,.]$

$$g[m,n] = \sum_{k,l} h[k,l]\, f[m+k, n+l]$$

Credit: S. Seitz

Image filtering

$h[.,.]$

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[.,.]$

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

$g[.,.]$

	0	10							

$$g[m,n] = \sum_{k,l} h[k,l]\, f[m+k, n+l]$$

Credit: S. Seitz

Image filtering

$h[\cdot,\cdot]$ $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[\cdot,\cdot]$

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

$g[\cdot,\cdot]$

	0	10	20						

Credit: S. Seitz

Image filtering

$h[\cdot,\cdot]$ $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[\cdot,\cdot]$

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

$g[\cdot,\cdot]$

	0	10	20	30					

Credit: S. Seitz

Image filtering

$h[\cdot,\cdot]$ $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[\cdot,\cdot]$

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

$g[\cdot,\cdot]$

	0	10	20	30	30				

Credit: S. Seitz

Image filtering

$h[\cdot,\cdot]$ $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[\cdot,\cdot]$

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

$g[\cdot,\cdot]$

	0	10	20	30	30				
				?					

Credit: S. Seitz

Image filtering

$h[\cdot,\cdot]$ $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[\cdot,\cdot]$

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

$g[\cdot,\cdot]$

	0	10	20	30	30				
						?			
				50					

Credit: S. Seitz

Image filtering

$h[\cdot,\cdot]$ $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[.,.]$

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

$g[.,.]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

$$g[m,n] = \sum_{k,l} h[k,l]\, f[m+k, n+l]$$

Credit: S. Seitz

Box Filter

What does it do?

- Replaces each pixel with an average of its neighborhood

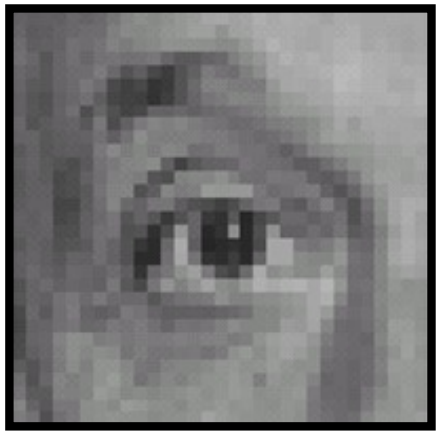
- Achieve smoothing effect (remove sharp features)

$$h[\cdot,\cdot]$$

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

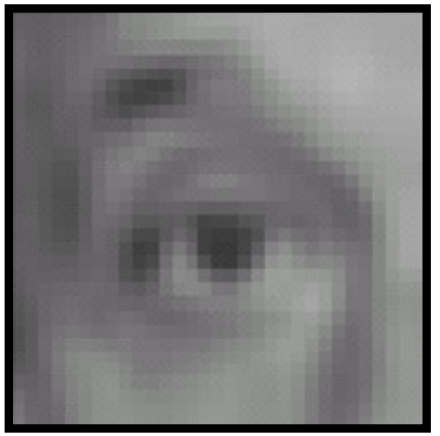
Slide credit: David Lowe (UBC)

Linear filters: examples



Original

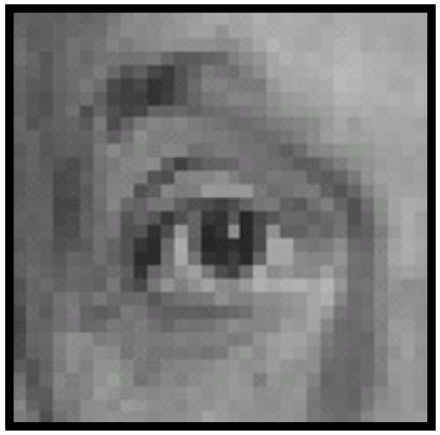
$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} =$$



Blur (with a mean filter)

Source: D. Lowe

Practice with linear filters



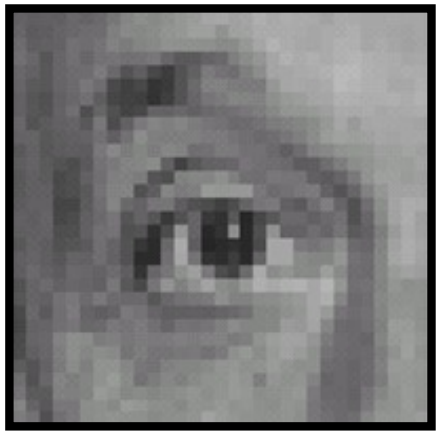
Original

0	0	0
0	1	0
0	0	0

?

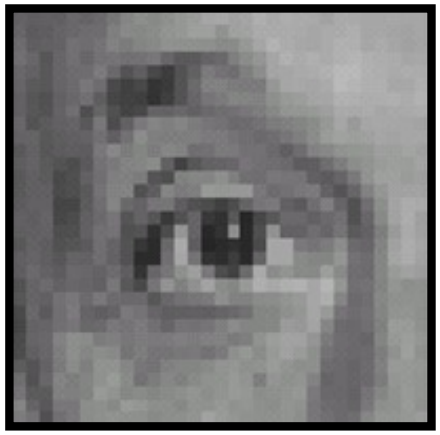
Source: D. Lowe

Practice with linear filters



Original

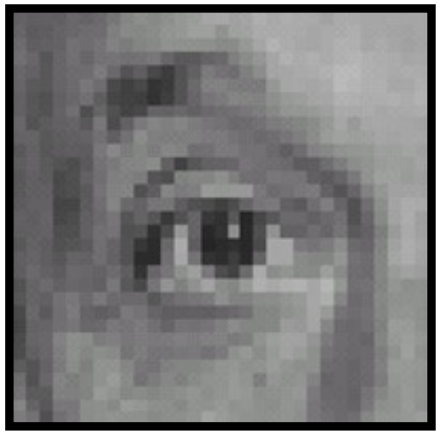
0	0	0
0	1	0
0	0	0



Filtered
(no change)

Source: D. Lowe

Practice with linear filters



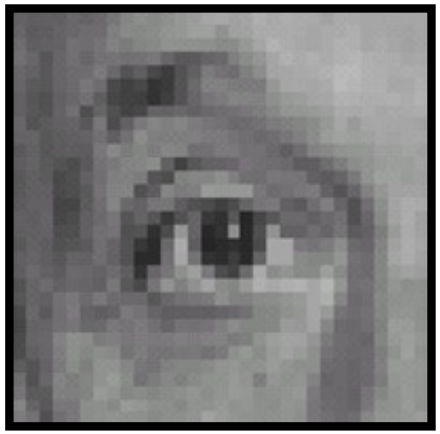
Original

0	0	0
0	0	1
0	0	0

?

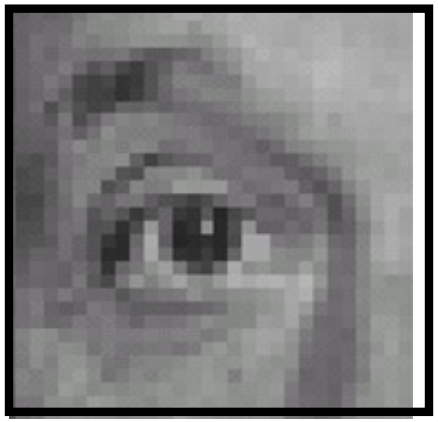
Source: D. Lowe

Practice with linear filters



Original

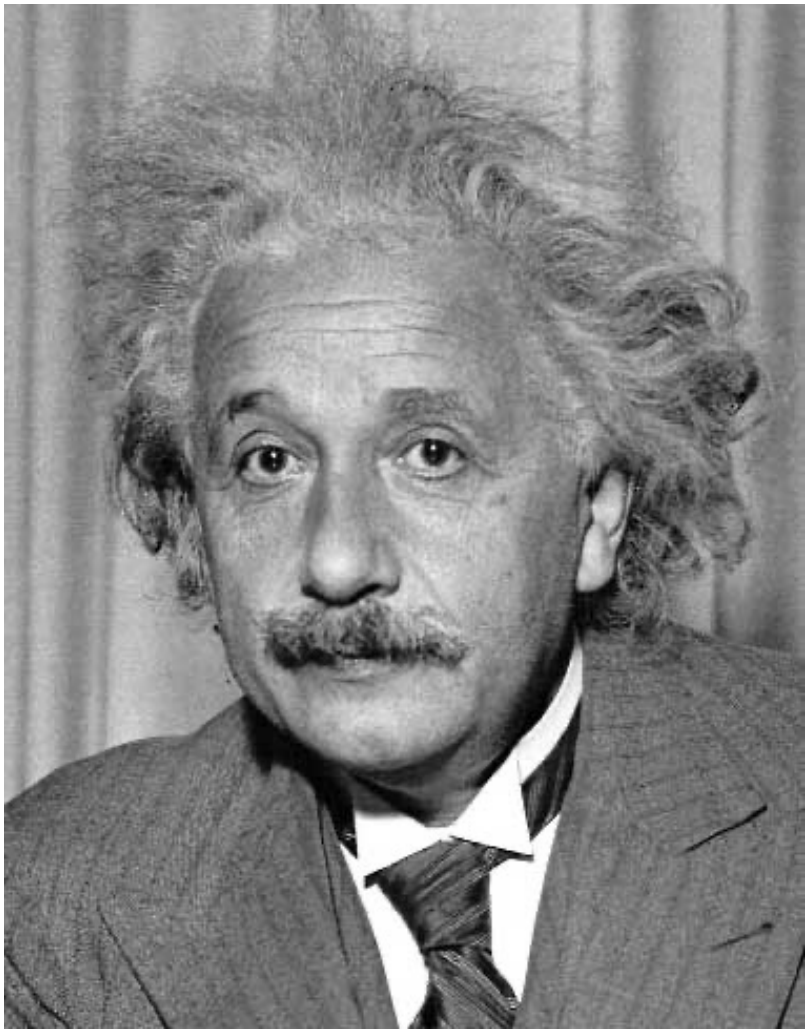
0	0	0
0	0	1
0	0	0



Shifted left
By 1 pixel

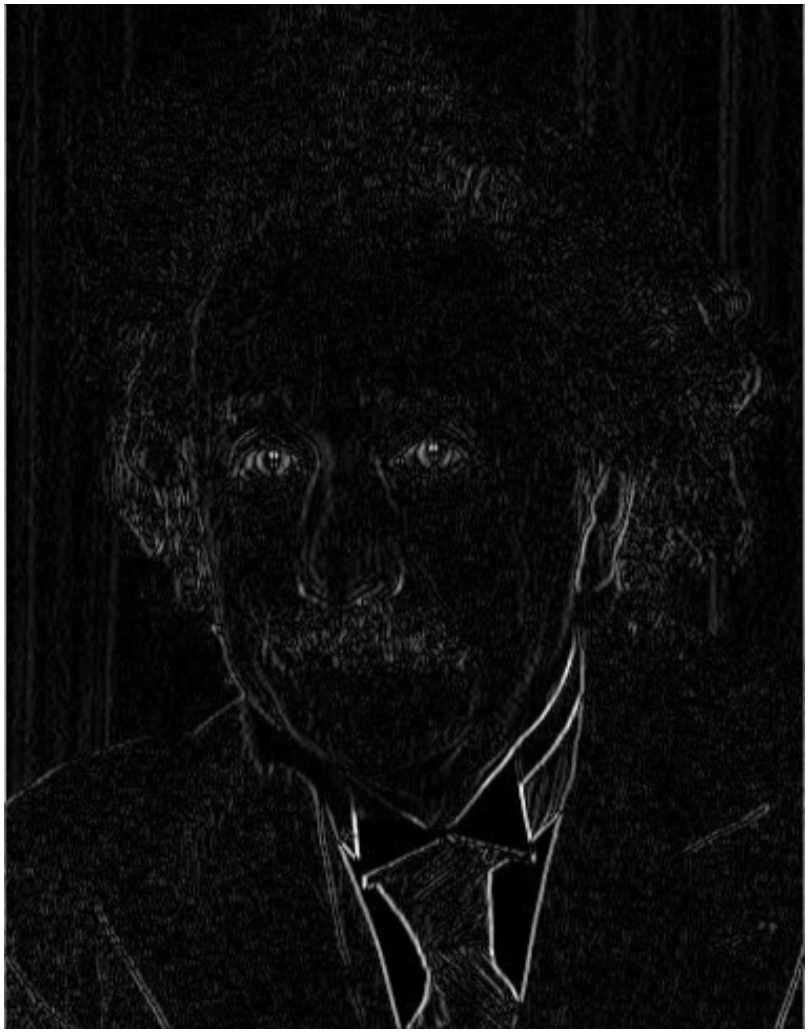
Source: D. Lowe

Other filters



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

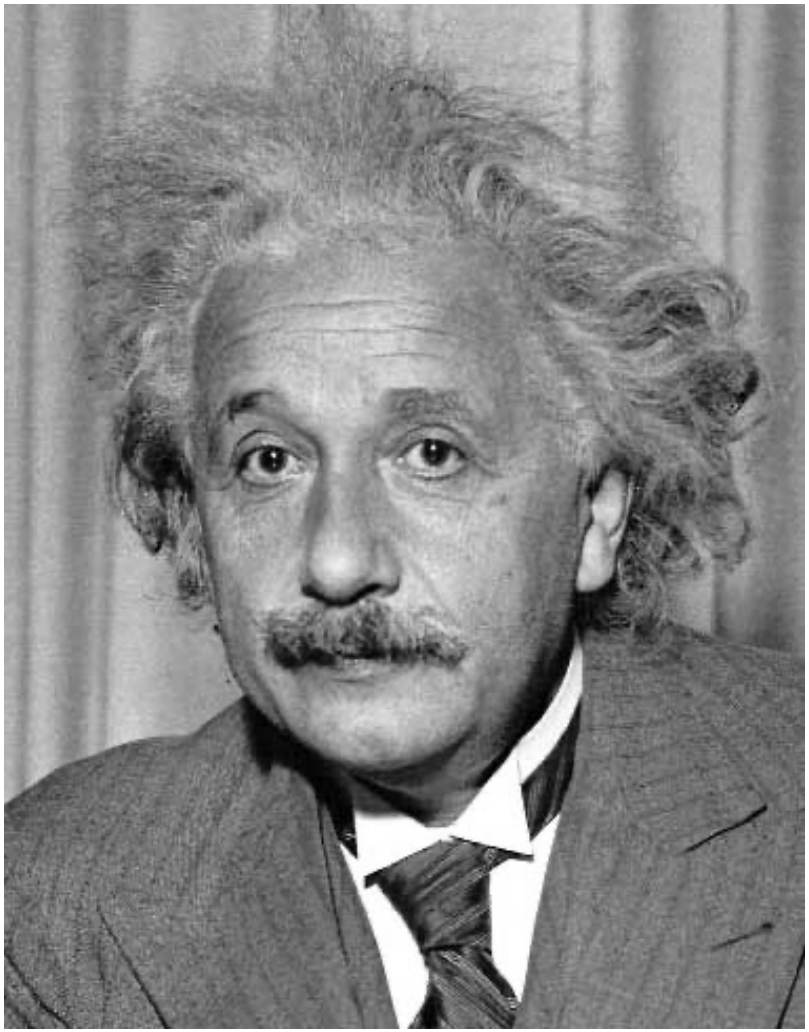
Sobel



Vertical Edge
(absolute value)

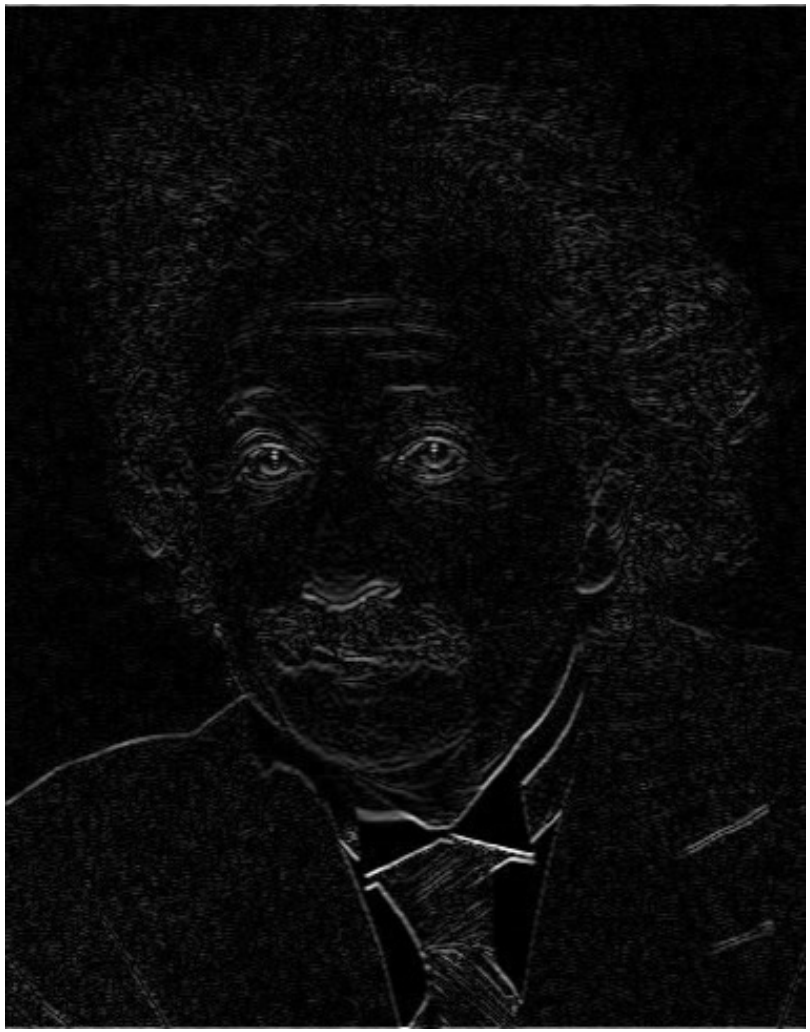
Q?

Other filters



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

Sobel



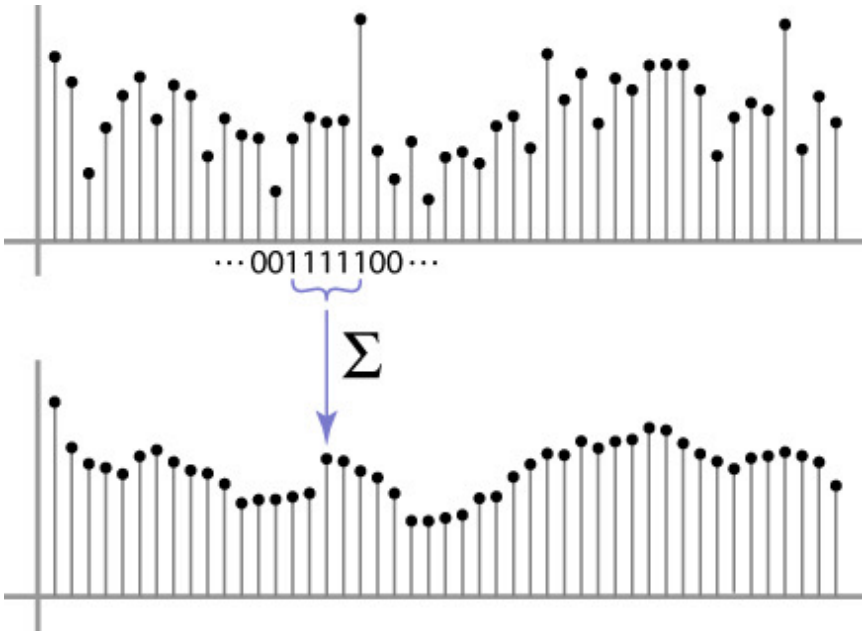
Horizontal Edge
(absolute value)

Back to the box filter

Moving Average

Can add weights to our moving average

Weights […, 0, 1, 1, 1, 1, 1, 0, …] / 5



Weighted Moving Average

bell curve (gaussian-like) weights […, 1, 4, 6, 4, 1, …]