Image Processing 1

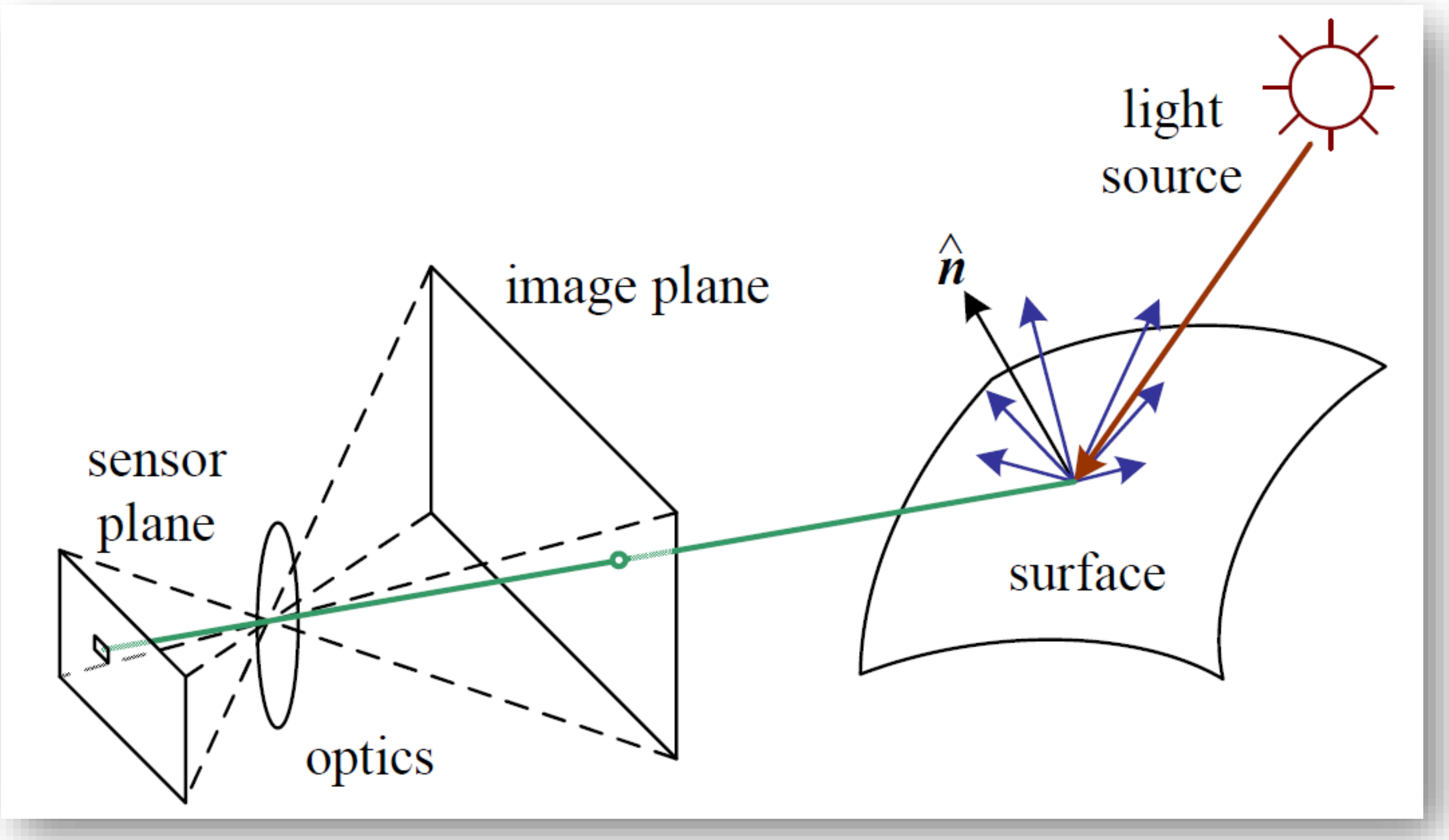
Hakan Bilen
University of Edinburgh

Computer Graphics
Fall 2017

This week

- What is an image?
- What is a digital image?
- Image processing
  - Point operations
  - Linear operations (anti-aliasing)
  - Fourier transformation
  - Global optimization (data-driven) based methods

What is an image?



An image as a function, I from R^2 to R

- $I(x, y): R^2 \rightarrow R$ outputs grayscale value at position (x, y)
- In practice $I(x, y): S \rightarrow V$

  $S = [a, b] \times [c, d]$ and $V = [0,1]$ (0->black, 1->white)

A color image $I(x, y)$ is a vector-valued function (e.g. RGB)

$$I(x, y) = \begin{pmatrix} r(x, y) \\ g(x, y) \\ b(x, y) \end{pmatrix}$$

What is a digital image?

Why am I not photogenic ☺

In computer graphics and vision, we typically use discrete images

- Sample the spatial space

  M rows, N columns

  $Resolution = M \times \mathrm{N}$

- Quantize each sample

  $L_{min} < f(x, y) < L_{max}$

  $[L_{min}, L_{max}] \rightarrow [0, L-1]$

- C-channel image storage size

  $L = 2^b$

  $S = M \times N \times C \times b$

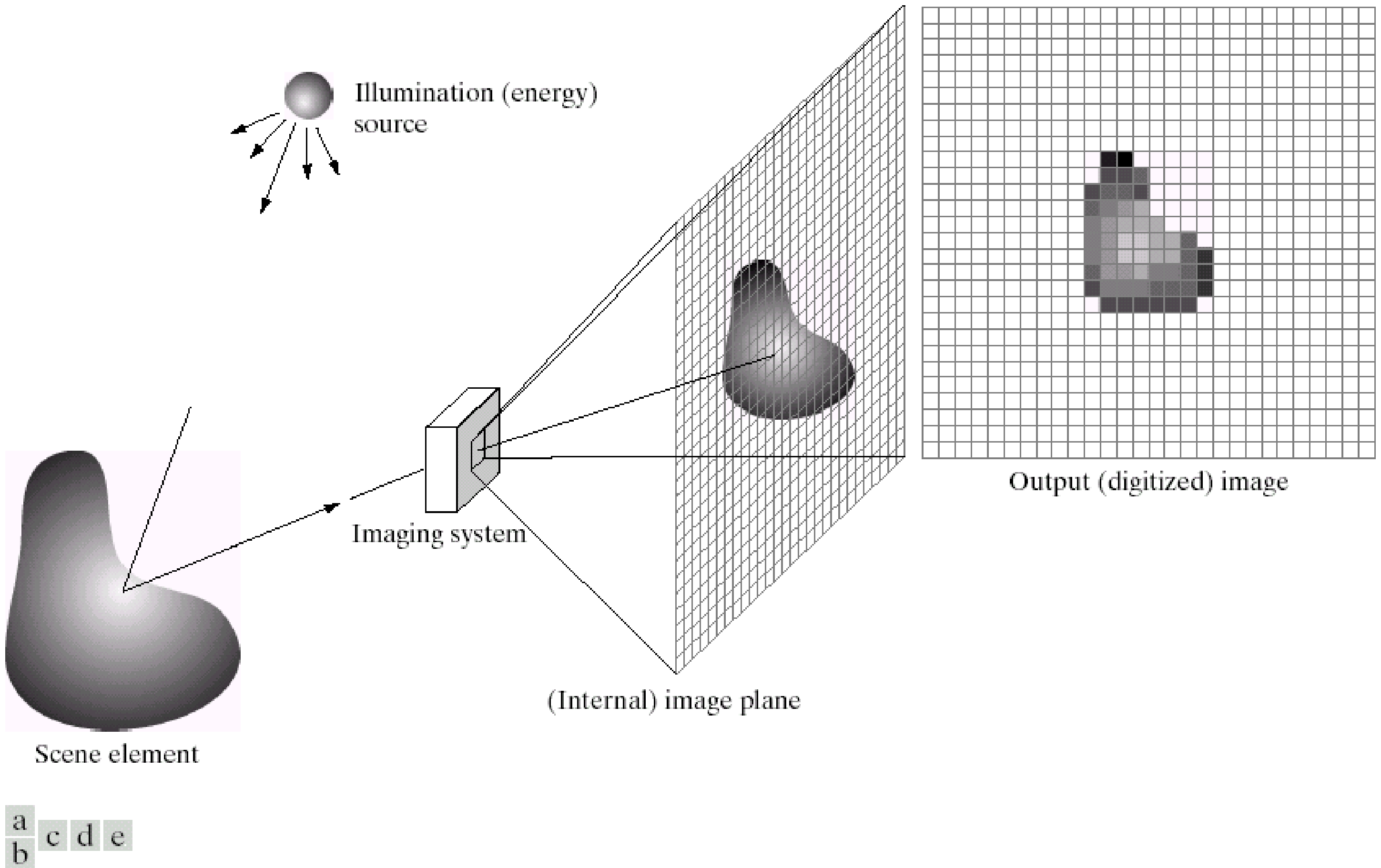
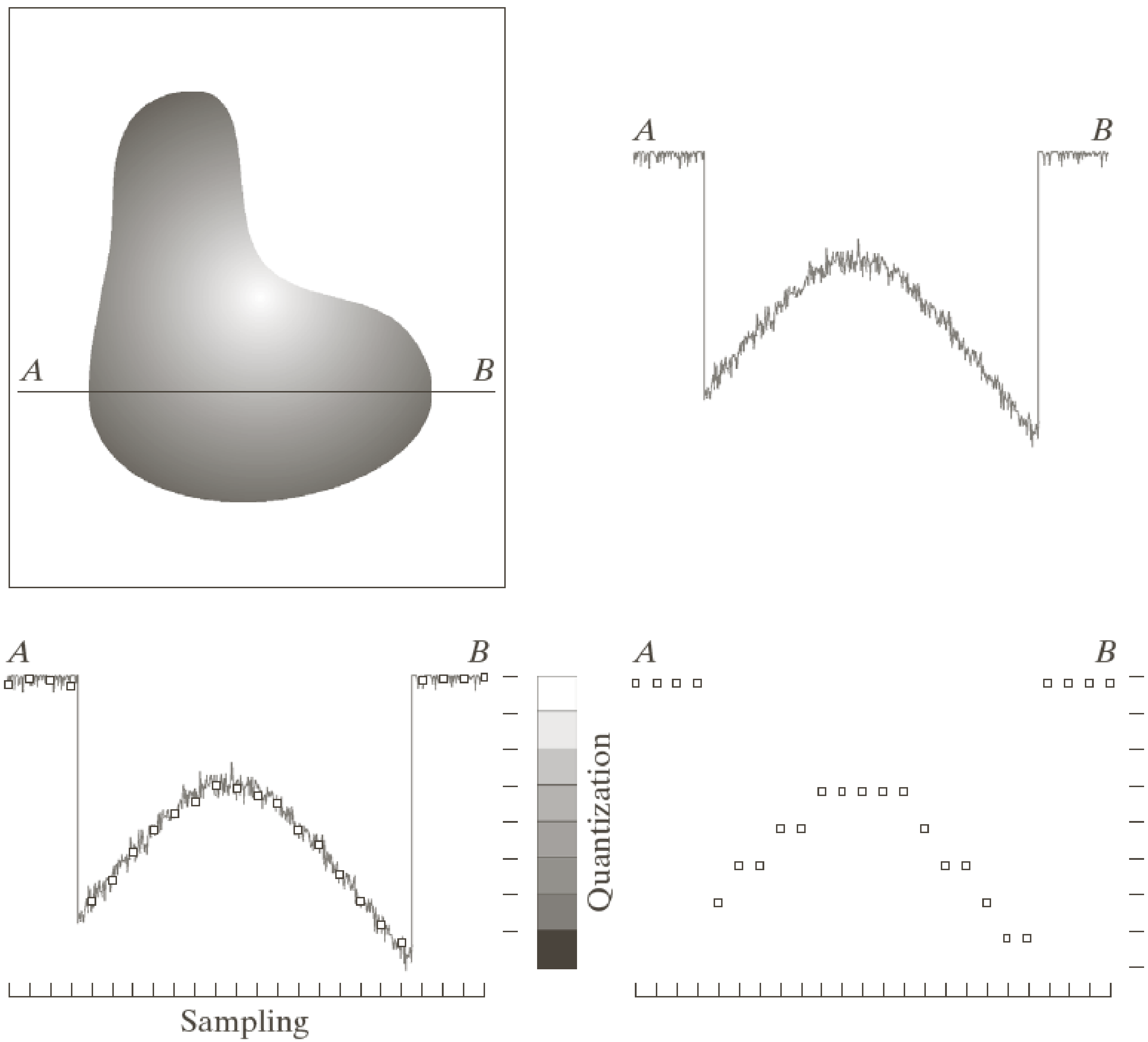


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

R. C. Gonzales & R. E. Woods

Image Sampling & Quantization



a b
c d

FIGURE 2.16 Generating a digital image. (a) Continuous image. (b) A scan line from *A* to *B* in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

R. C. Gonzales & R. E. Woods

Problem: Real world is high dynamic range

$I(x, y) \sim \text{illumination}(x, y) \times \text{reflectance}\ (x, y)$

$0 < \text{illumination}(x, y) < \inf$

$0 < \text{reflectance}(x, y) < 1$

1

1,500

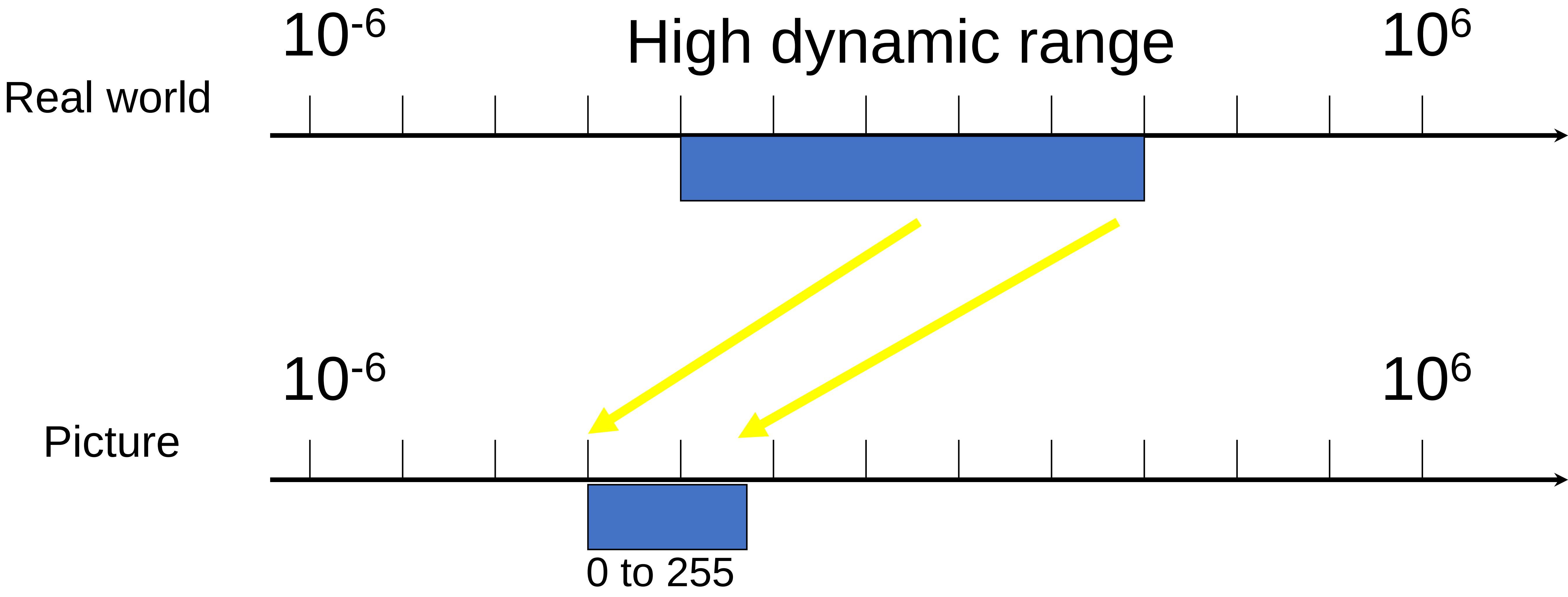
25,000

400,000

2,000,000

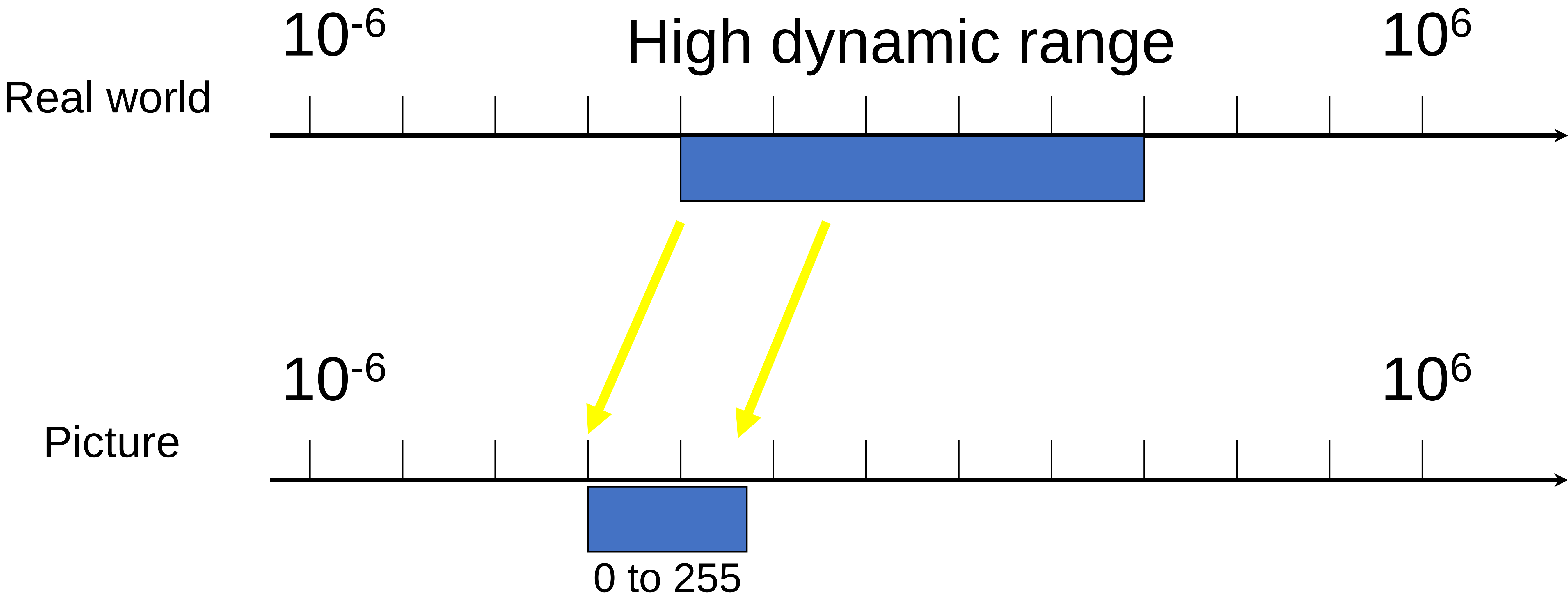
Image credit: Paul Debevec

Short Exposure



Slide credits: Alyosha Efros

Long Exposure



Slide credits: Alyosha Efros

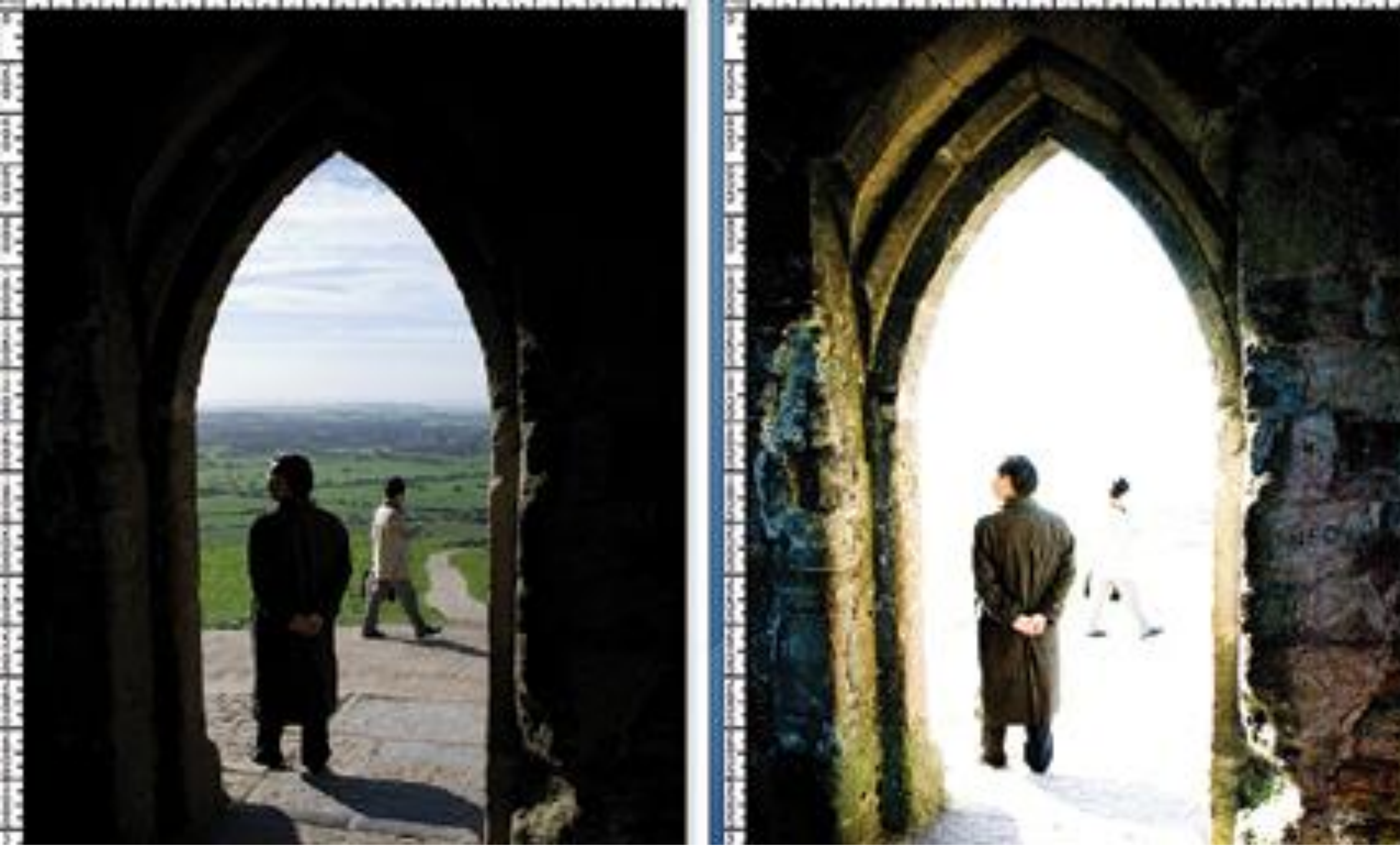


Image credits: shutterbug.com

Quantization I

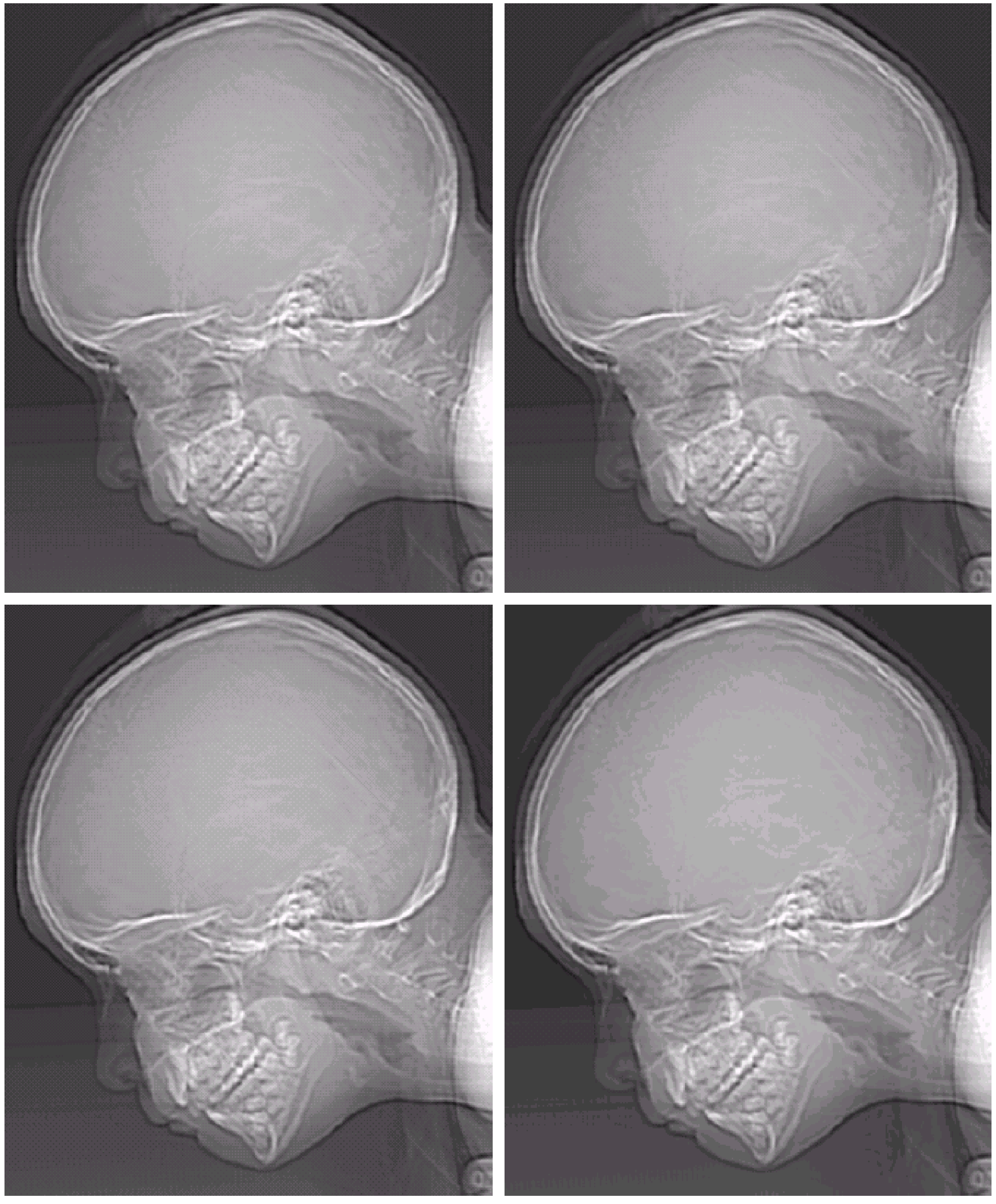


FIGURE 2.21 (a) 452 × 374, 256-level image. (b)–(d) Image displayed in 128, 64, and 32 gray levels, while keeping the spatial resolution constant.

Quantization II

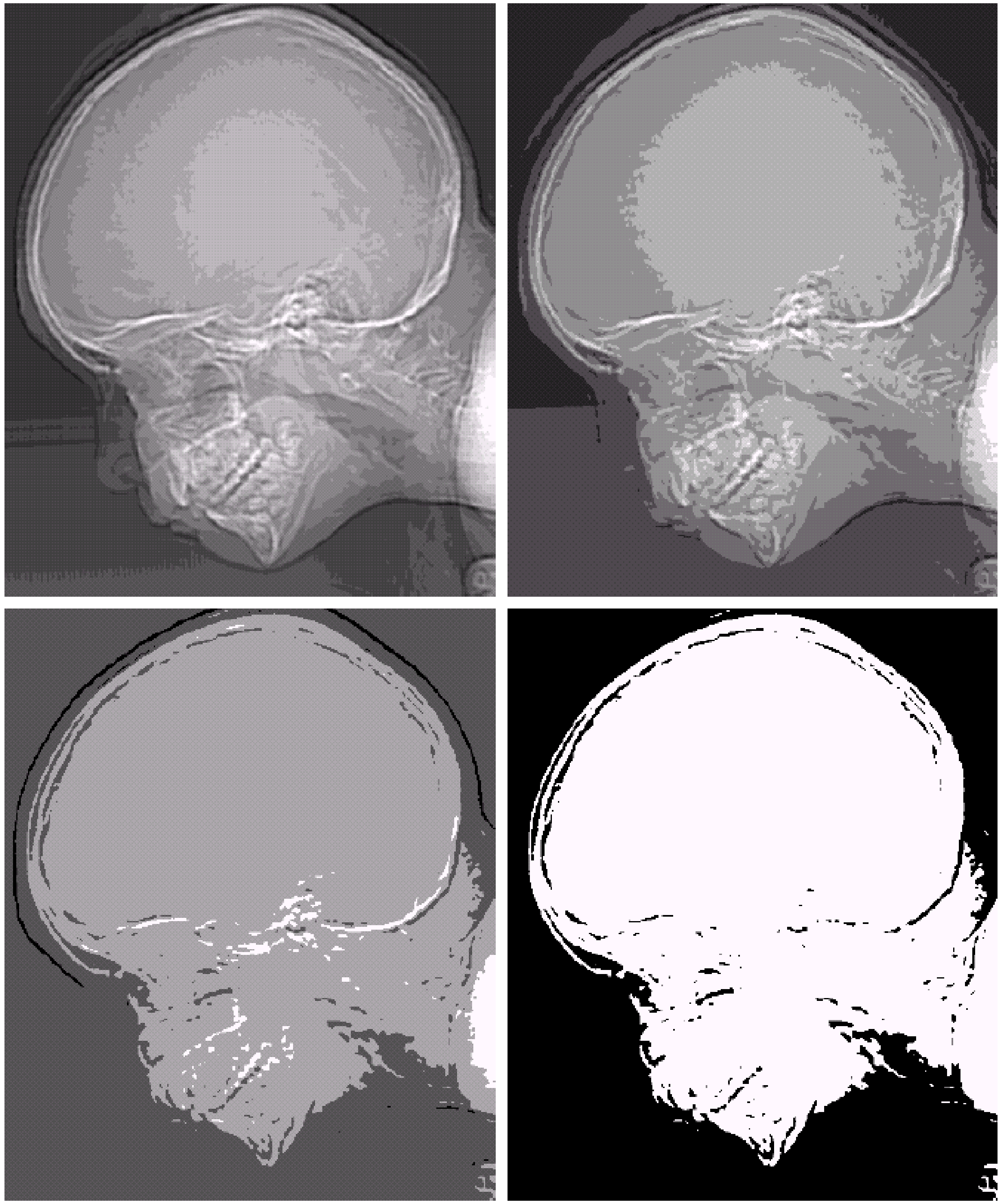


FIGURE 2.21 *(Continued)* (e)–(h) Image displayed in 16, 8, 4, and 2 gray levels. (Original courtesy of Dr. David R. Pickens, Department of Radiology & Radiological Sciences, Vanderbilt University Medical Center.)

What is image processing?

How can I look better in photos?

Definition: to preprocess the image and convert it into a form suitable for further analysis (Szelisky)

read an image, process it and write the result

```
>>convert chinese_chess.jpg -contrast chinese_contrast.png
```

Image credit: imagemagick.org

Image Processing Operations

Point processing

- Brightness
- Contrast
- Gamma
- Histogram eq.
- Black & white
- Saturation
- White balance

- Global optimization strategies
  - Model based
  - Deep learning

Filtering

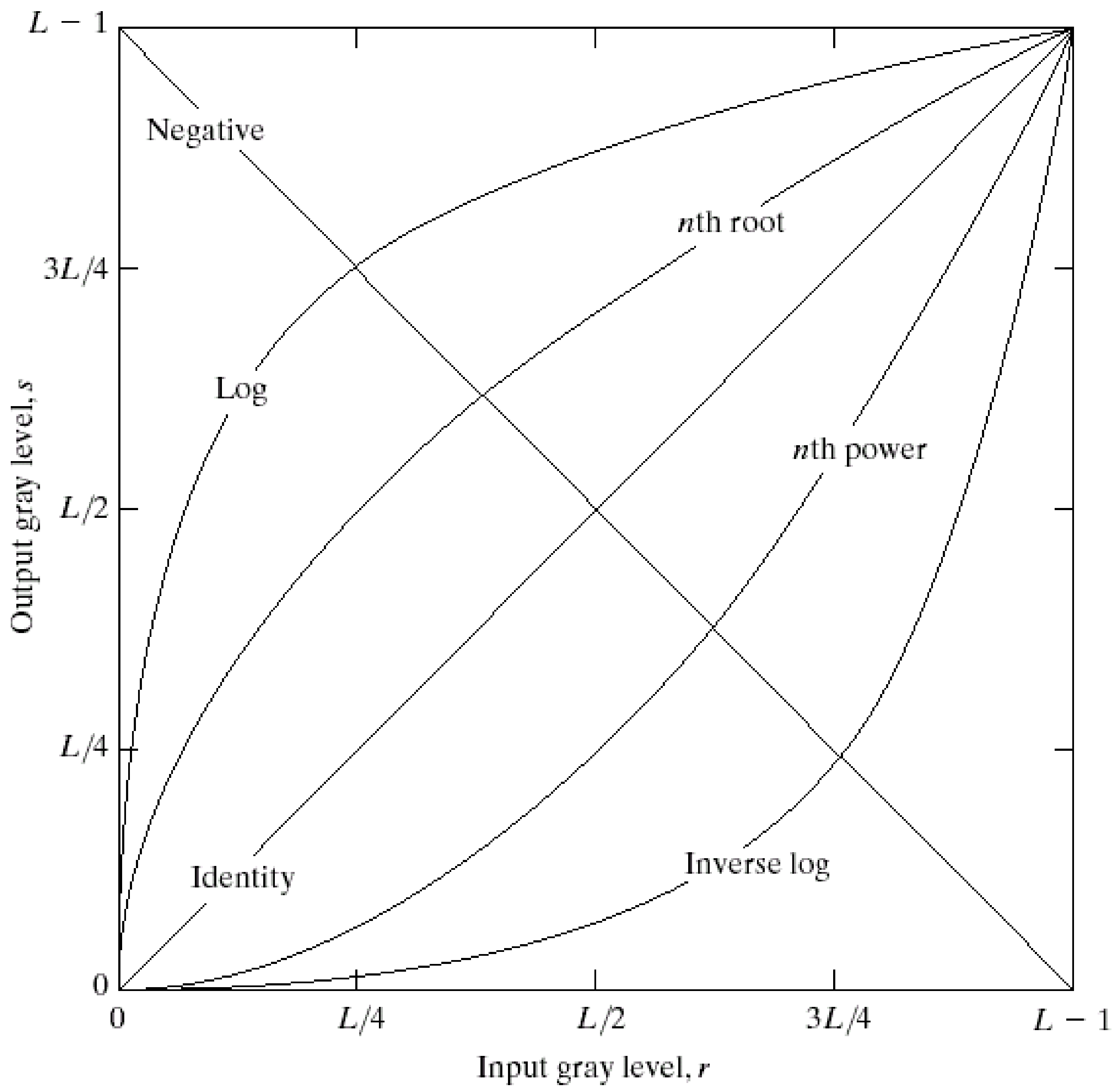
- Linear
  - Blurring
  - Sharpening
  - Edge detection
- Non-linear Filtering
  - Median Filtering
  - Bilateral Filtering
- Fourier transform

Point Processing

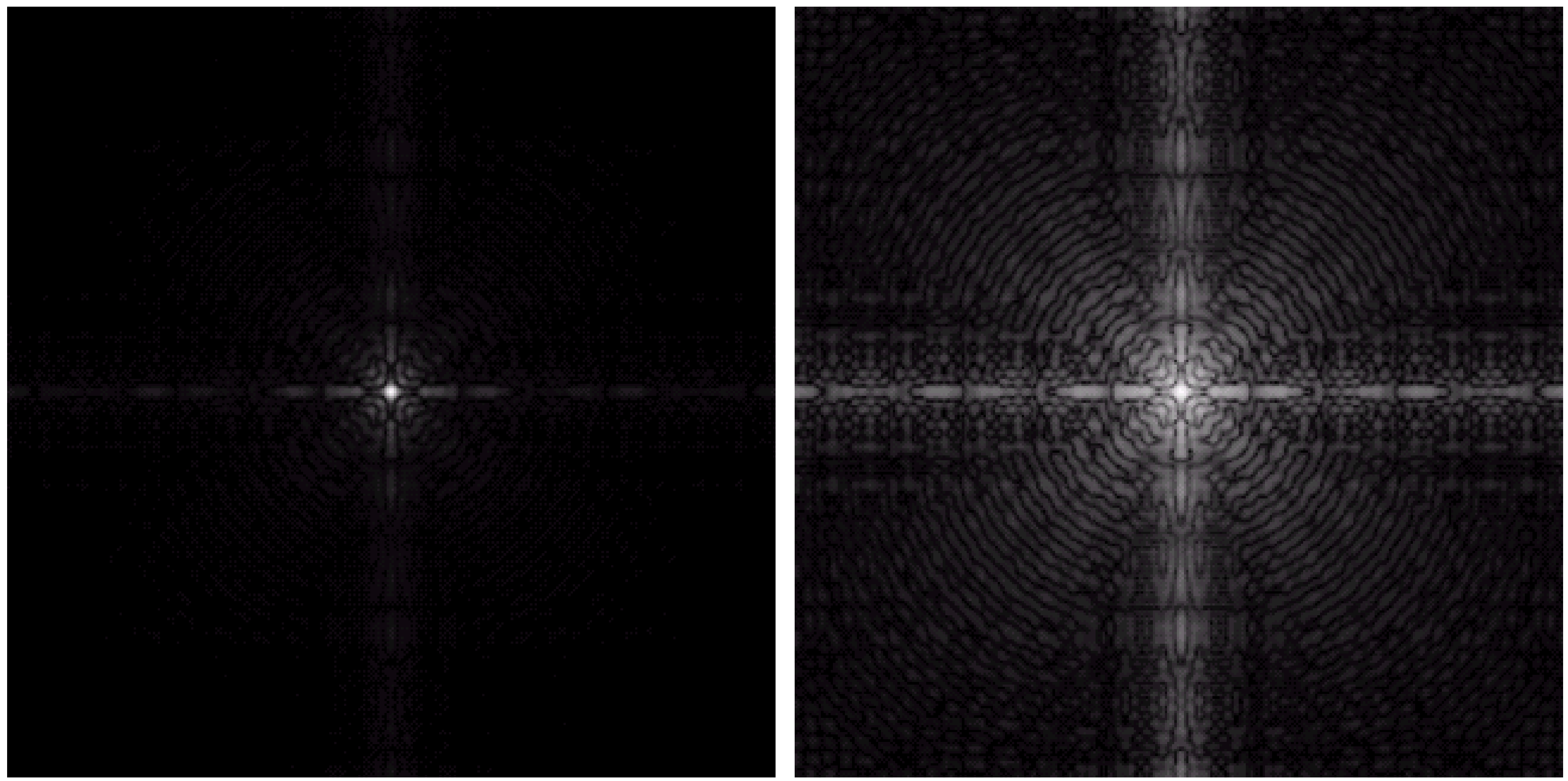
Log

$$s = c \cdot log(1 + r)$$

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



Log



a b

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

Point Processing

Power-law transformations

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

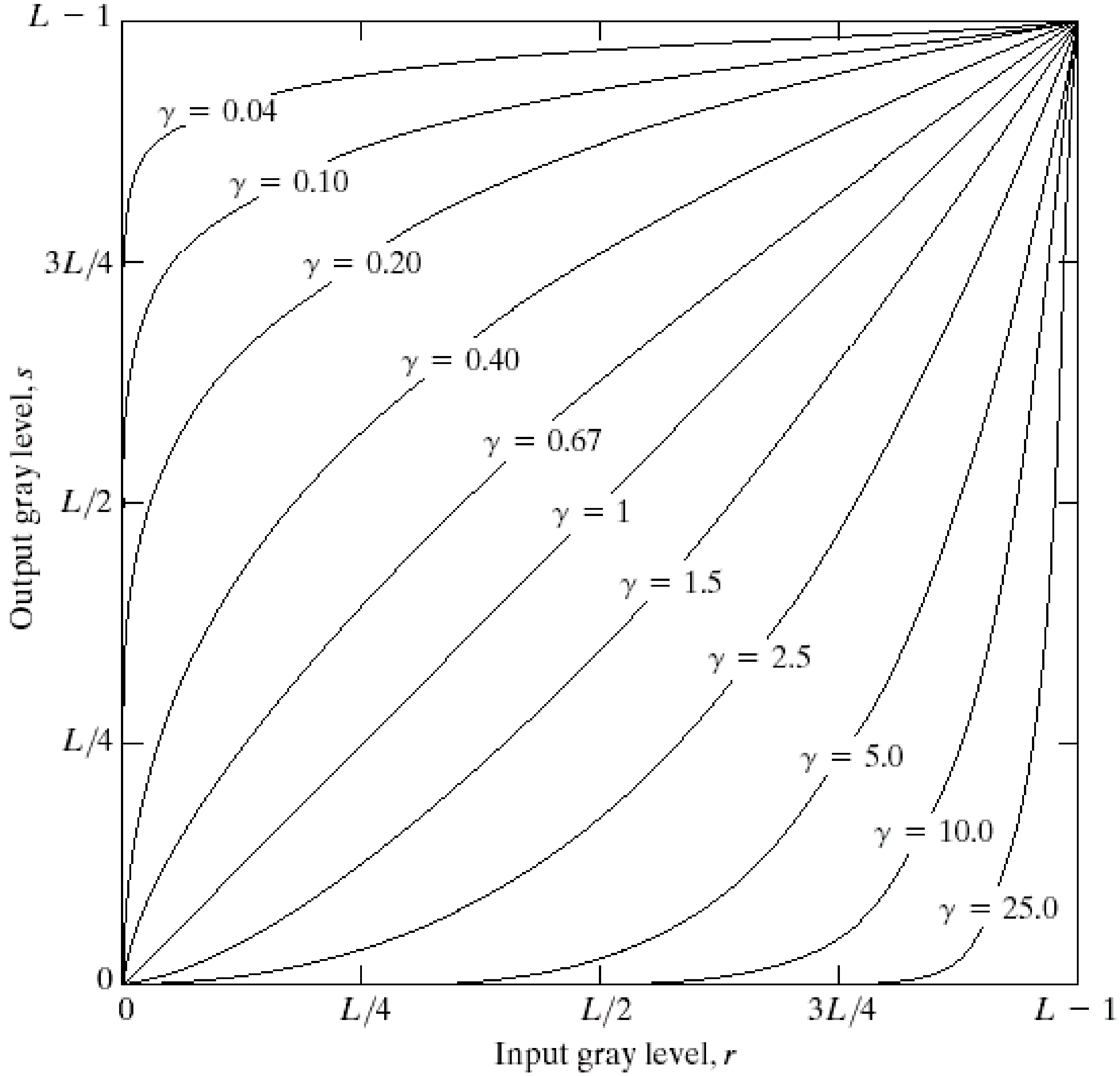
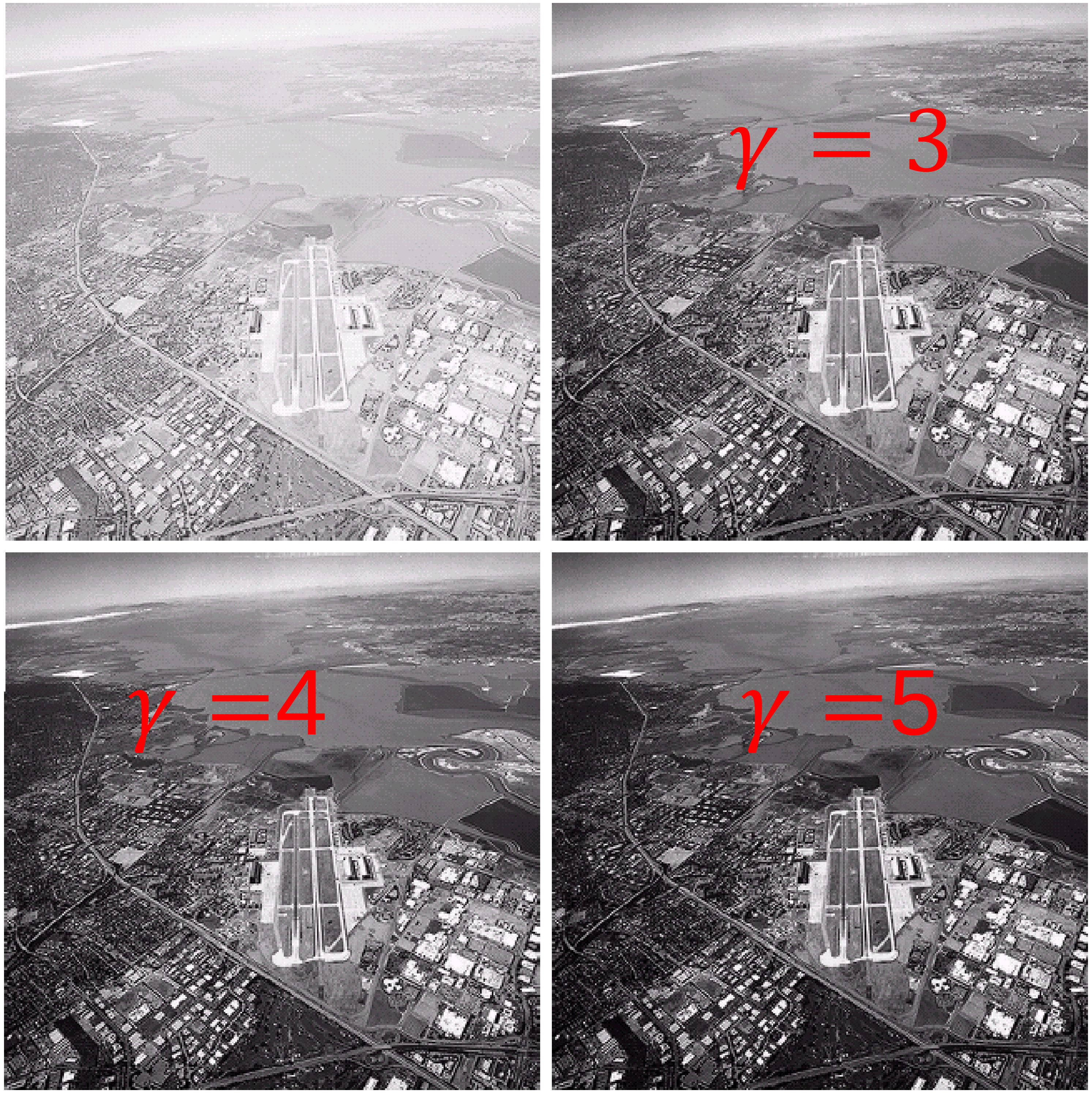


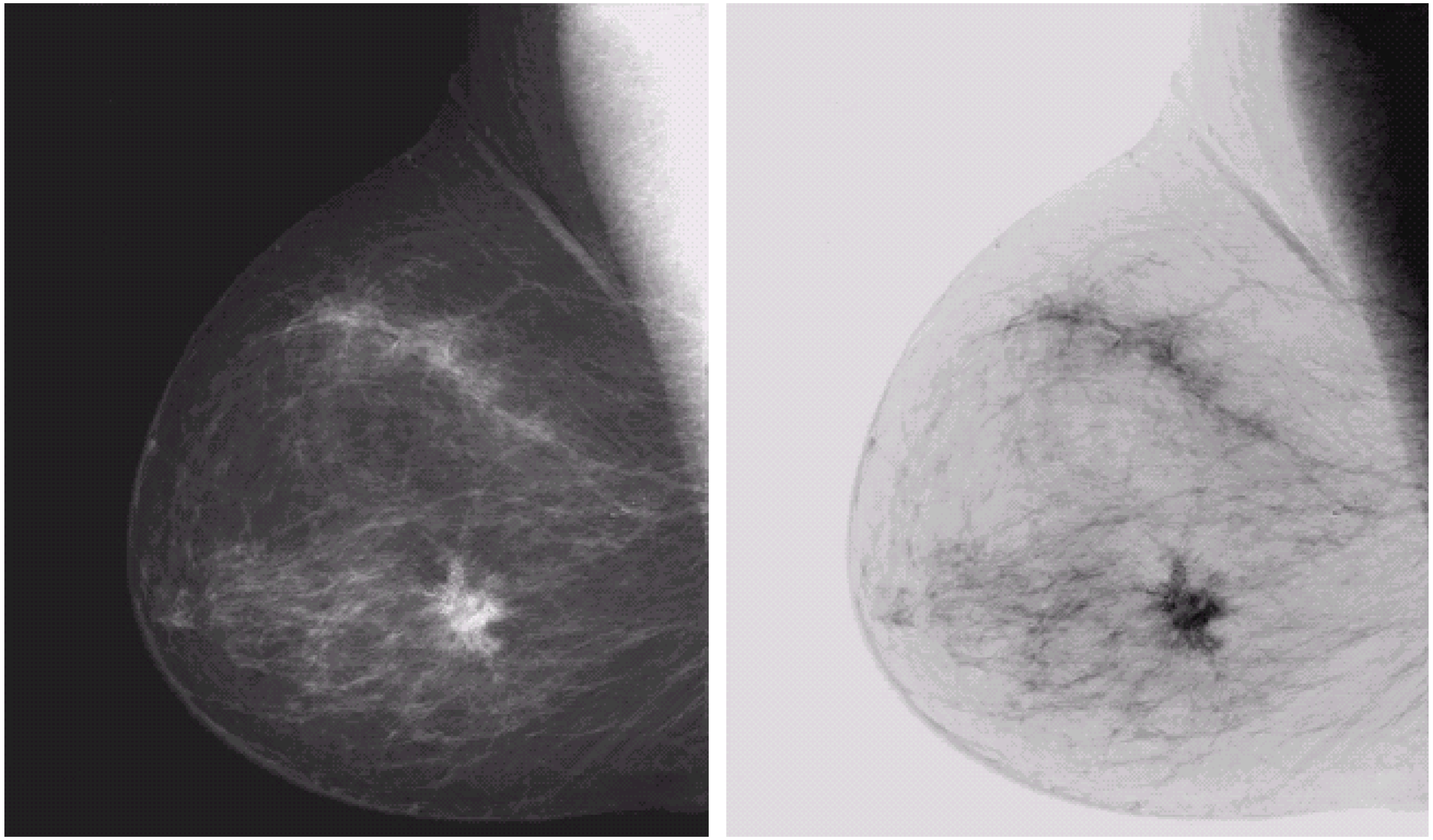
FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

Q: What does high and low γ do?

Point Processing



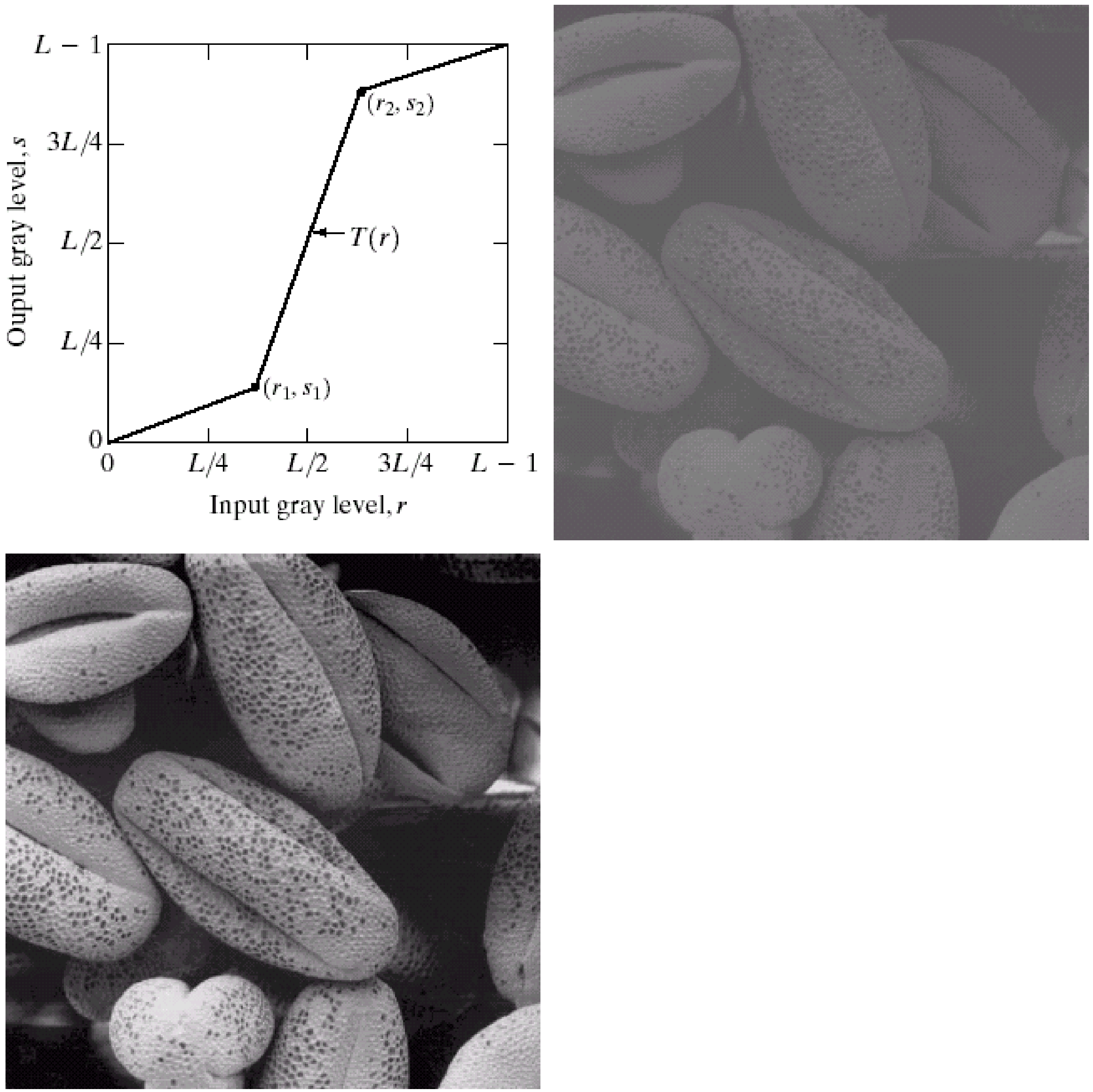
Negative



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.)

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

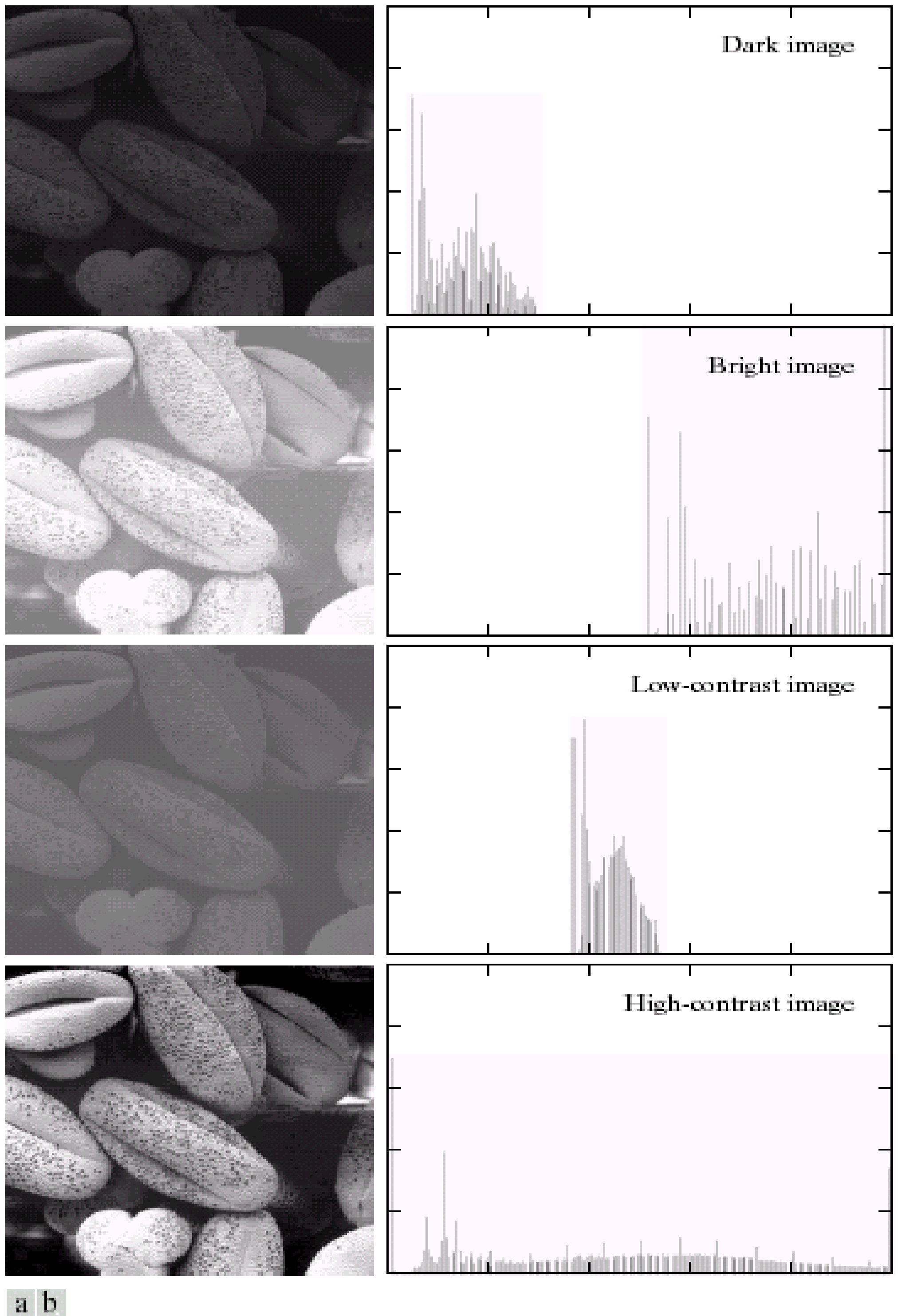
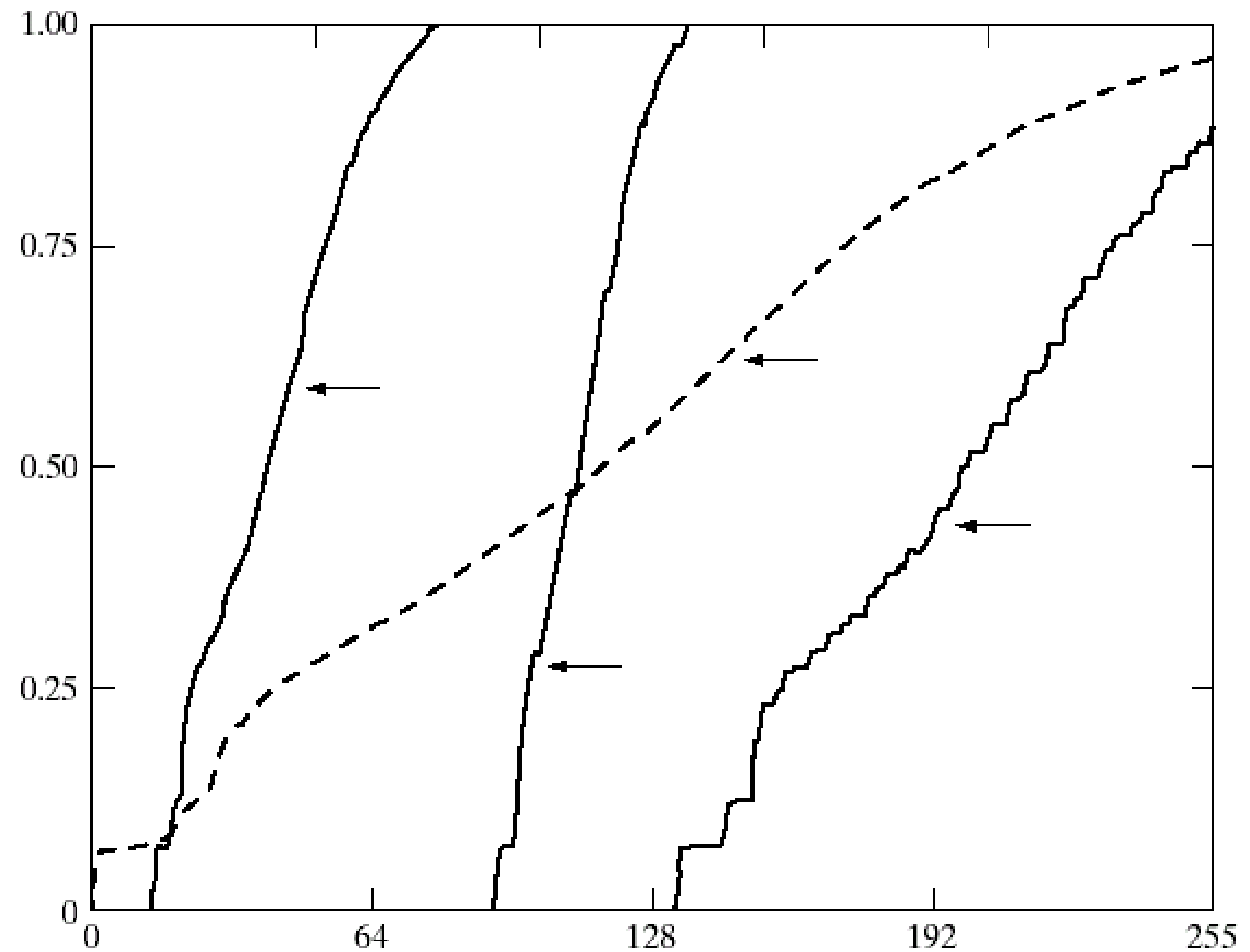


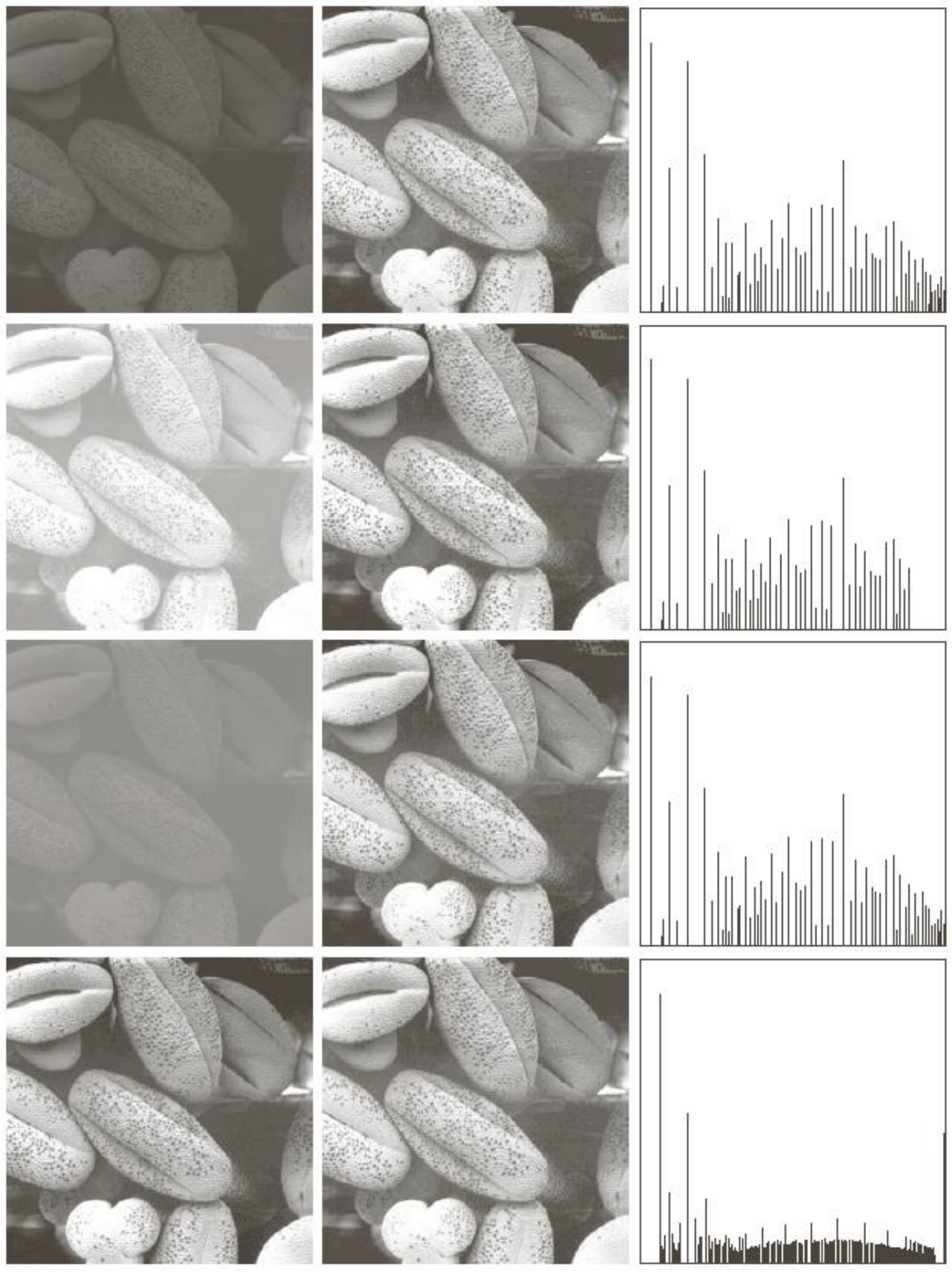
FIGURE 3.15 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)



Cumulative Histograms

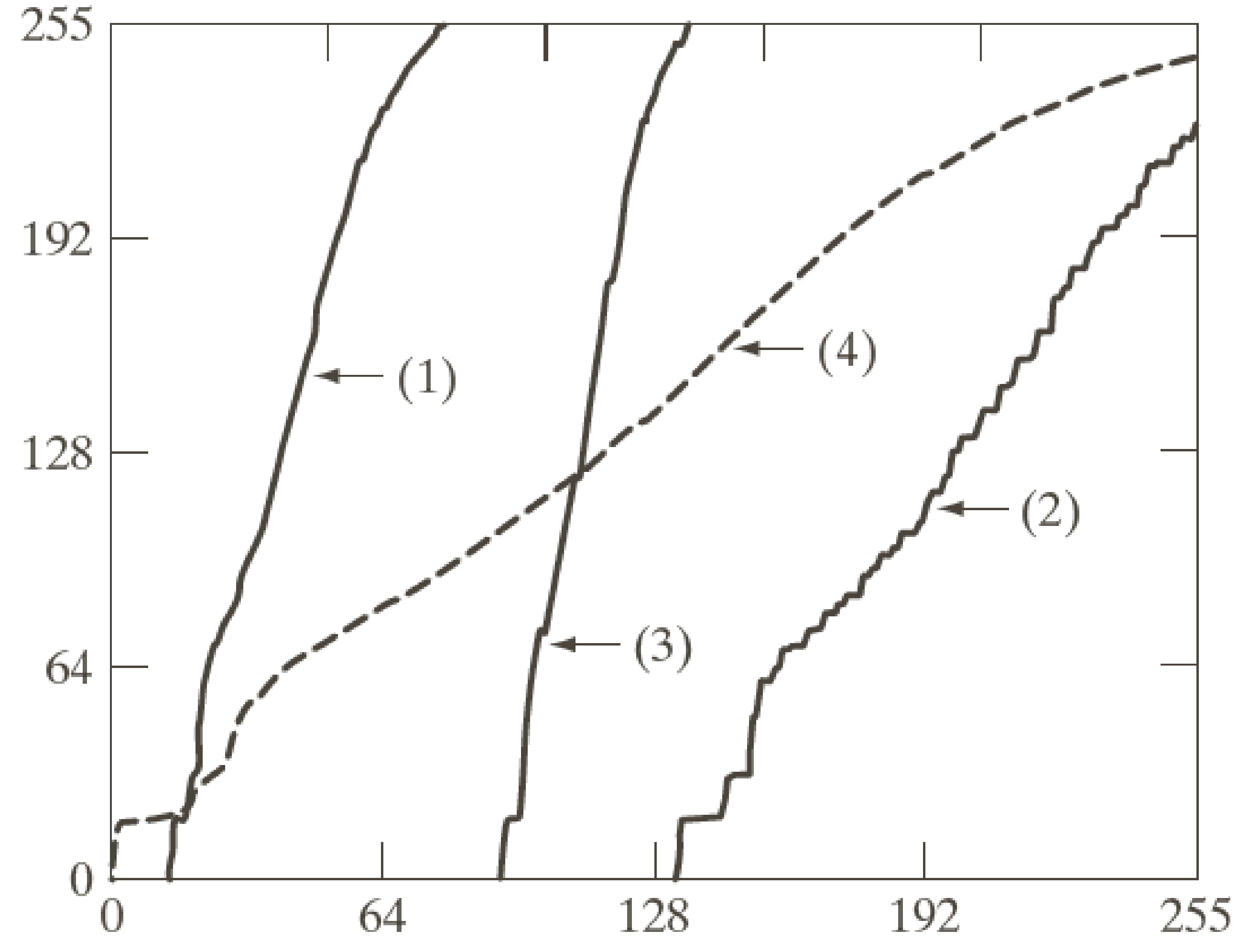
$$s = T(r) = \int_0^r p_r(w)dw$$

Histogram Equalization

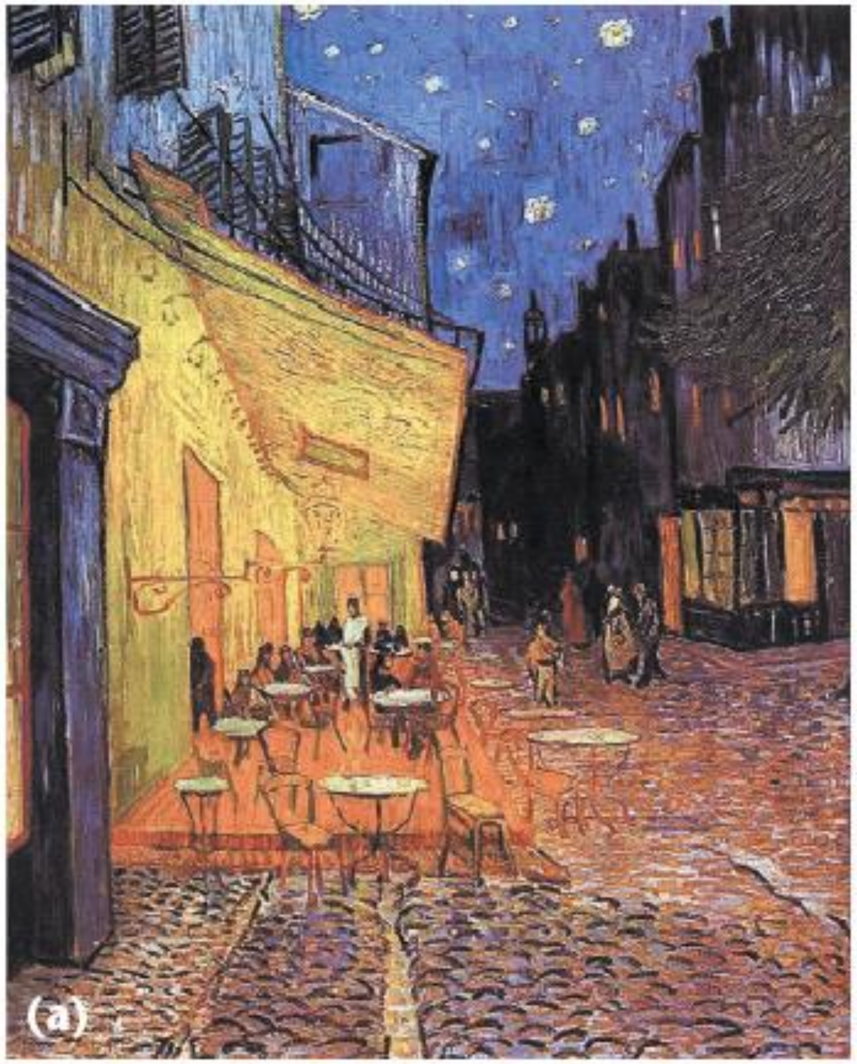


a b c

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



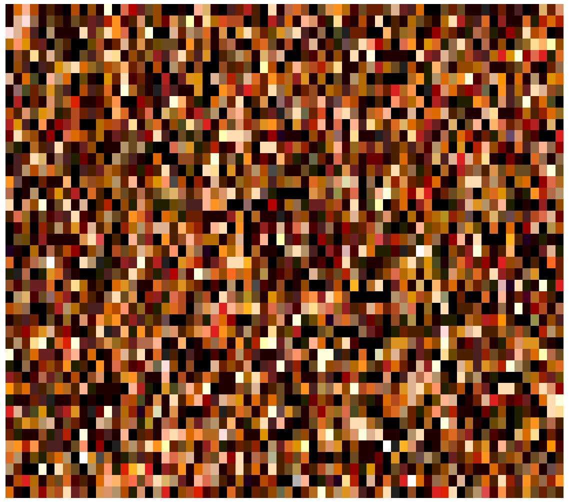
Color Transfer [Reinhard, et al, 2001]



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…

They don't know about their neighbors, edges, textures, etc.

Slide credits: Alyosha Efros

What Point Operations Can't Do

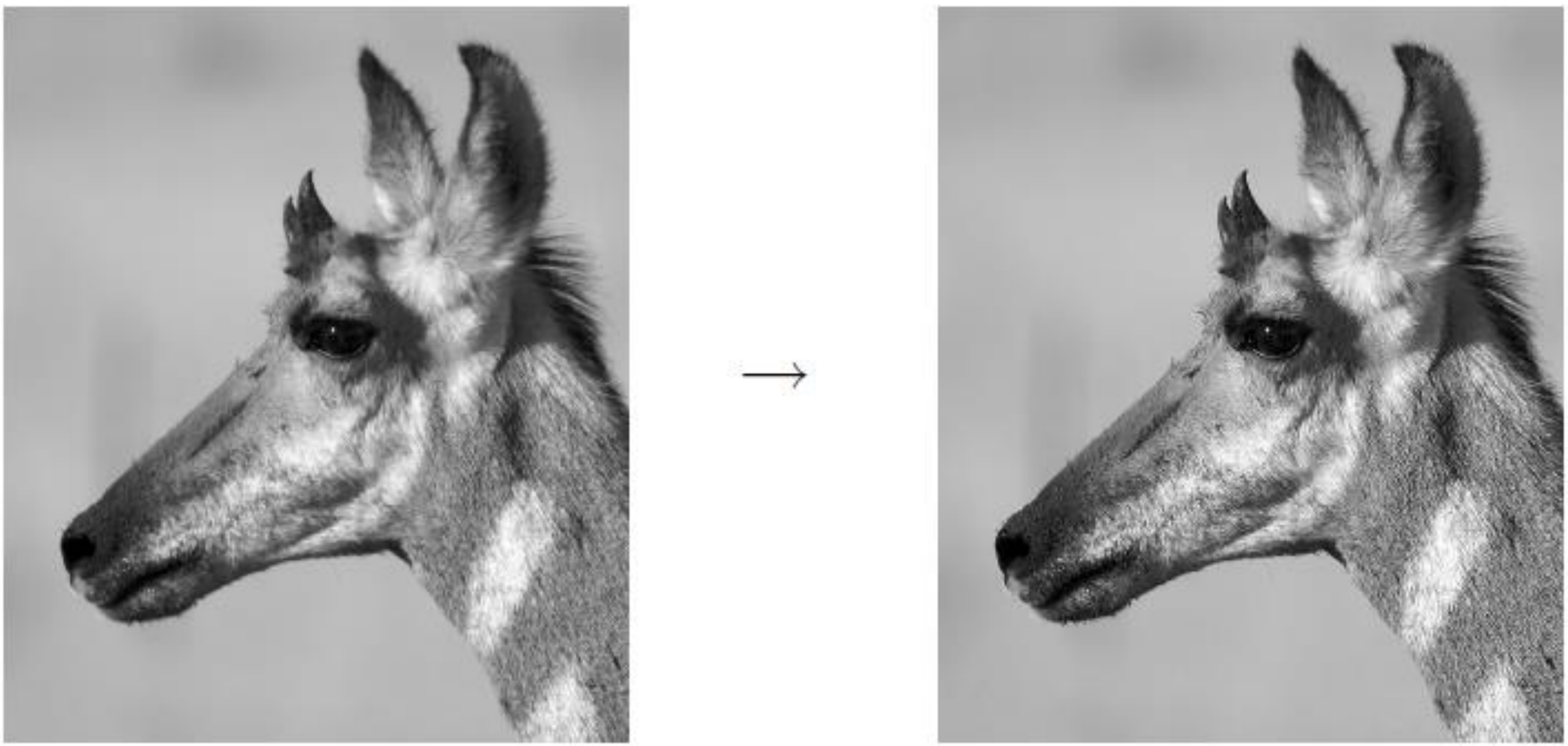
Blurring / Smoothing

→

Slide credits: Tom Fletcher

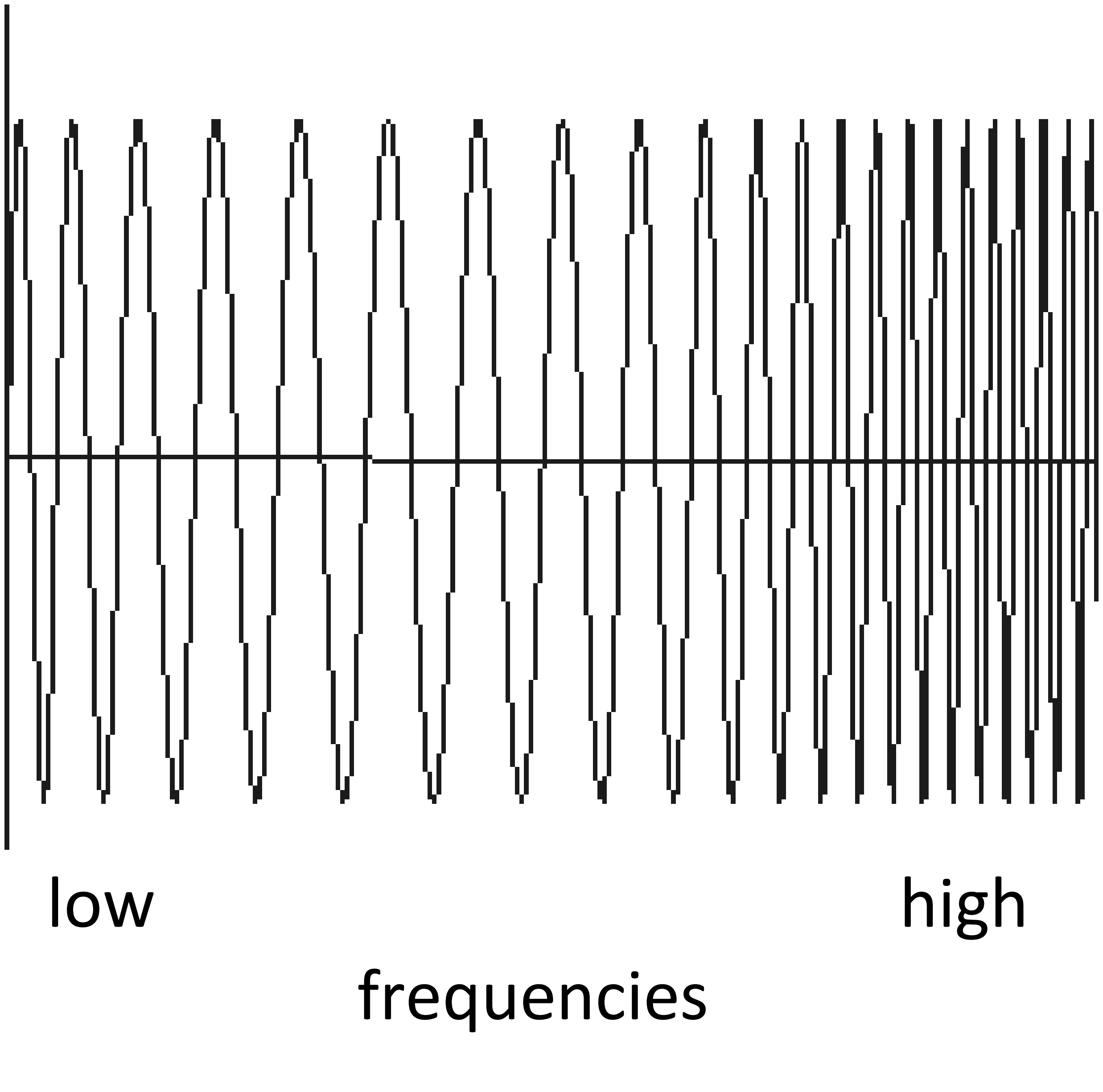
What Point Operations Can't Do

Sharpening



Slide credits: Tom Fletcher

1D Example: Audio

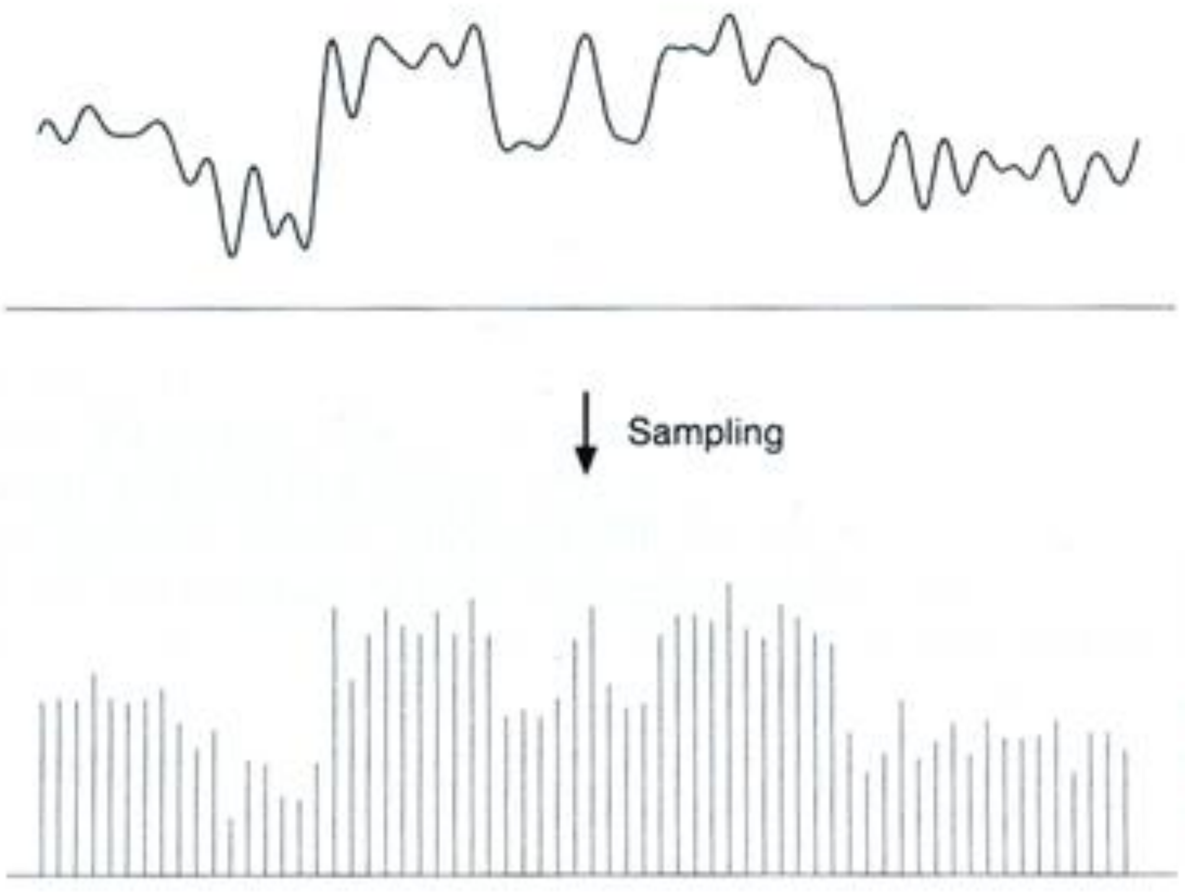


Sampled representations

How to store and compute with continuous functions?

Common scheme for representation: samples

- write down the function's values at many points



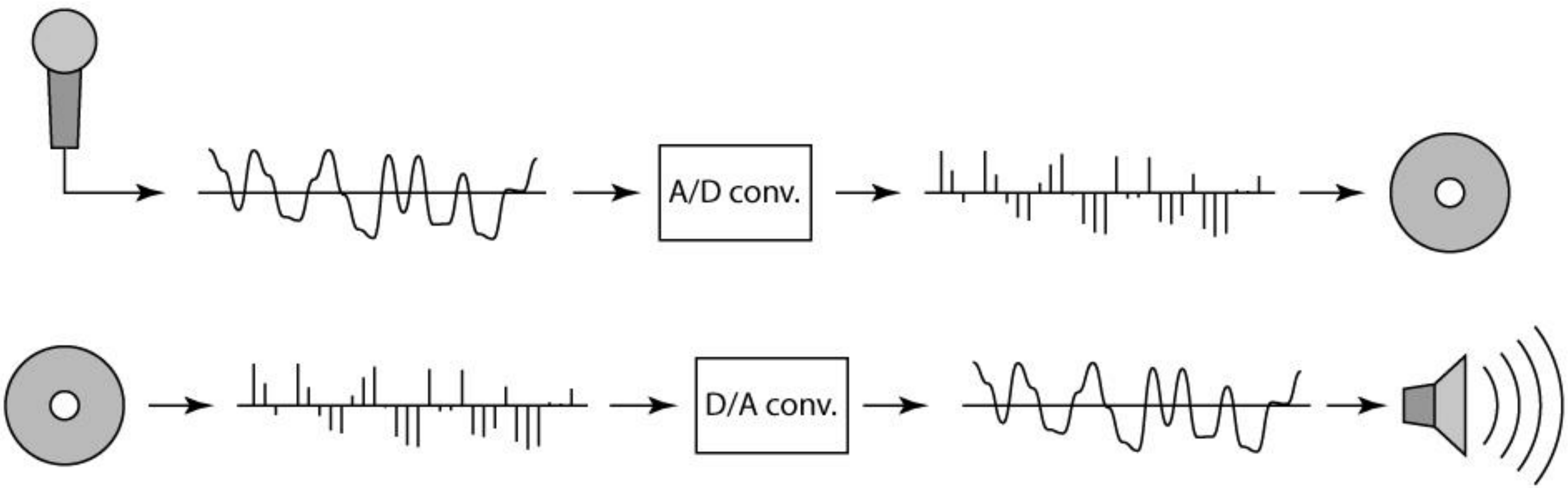
[FvDFH fig.14.14b / Wolberg]

Sampling in digital audio

Recording: sound to analog to samples to disc

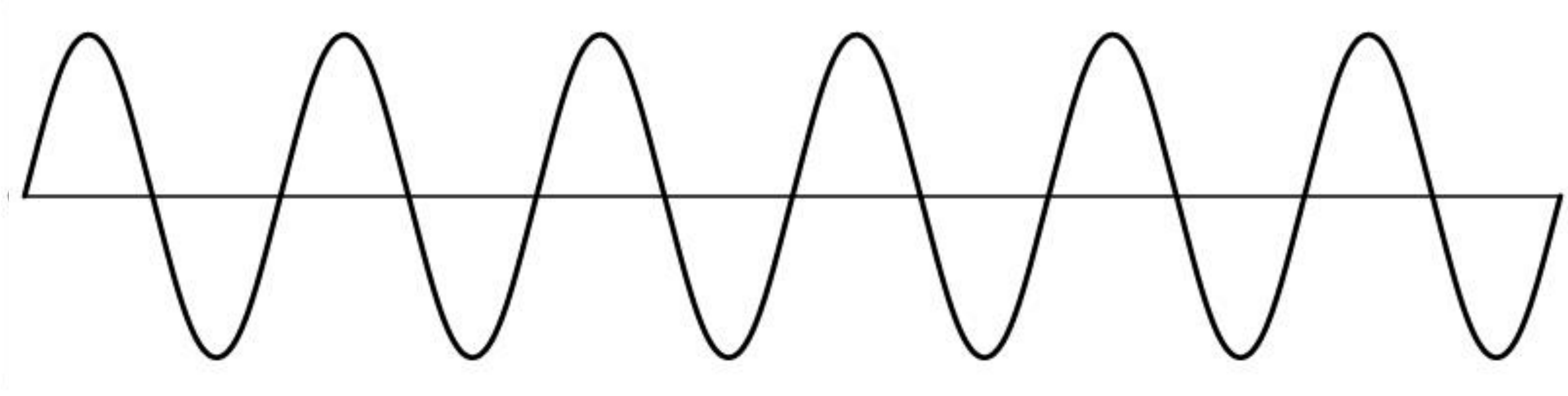
Playback: disc to samples to analog to sound again

- how can we be sure we are filling in the gaps correctly?
- Problem 1: undersampling artifact
- Problem 2: reconstruction artifact



Sampling and Reconstruction

Simple example: a sign wave

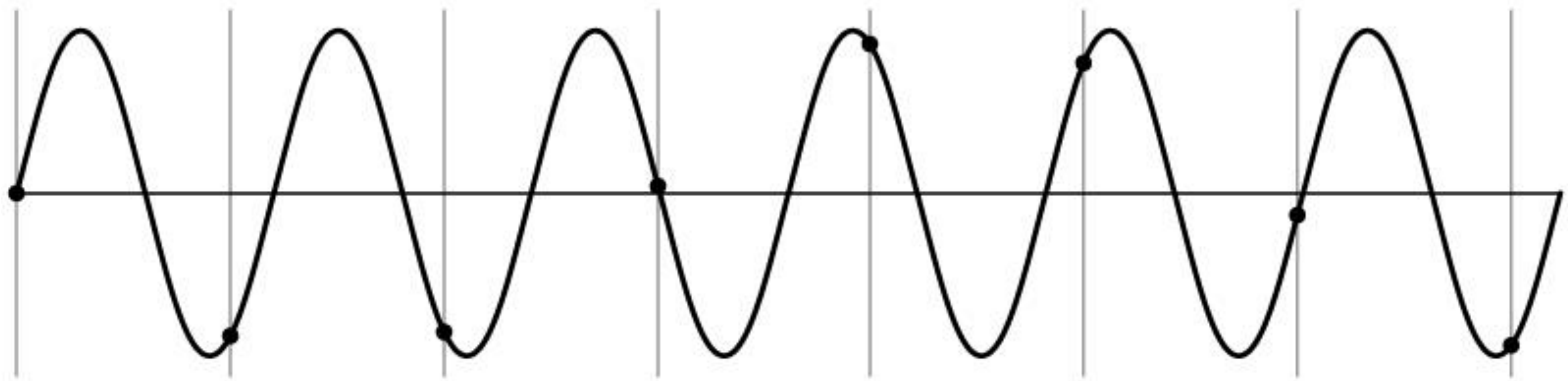


Undersampling

What if we “missed” things between the samples?

Simple example: undersampling a sine wave

- unsurprising result: information is lost

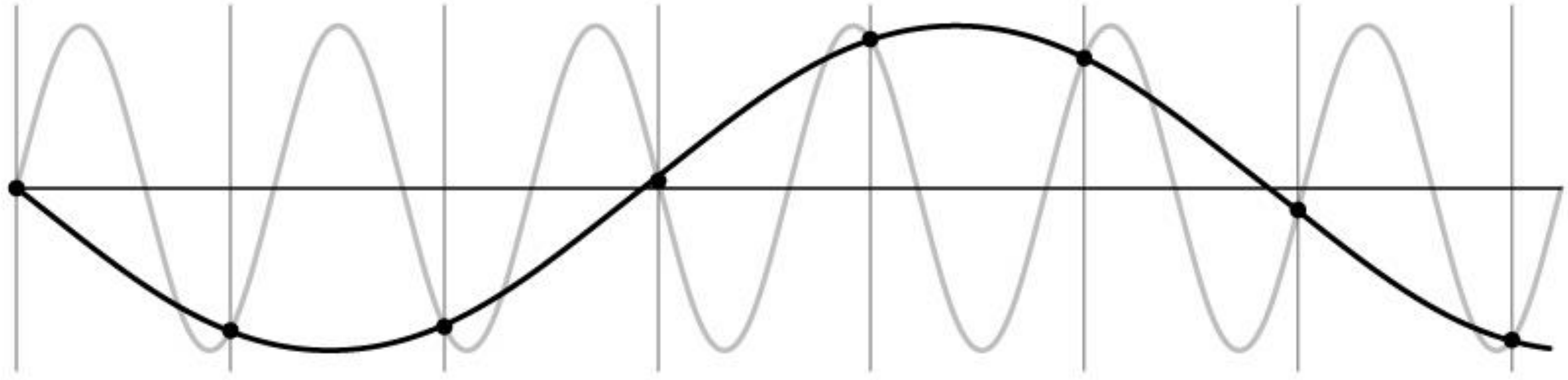


Undersampling

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

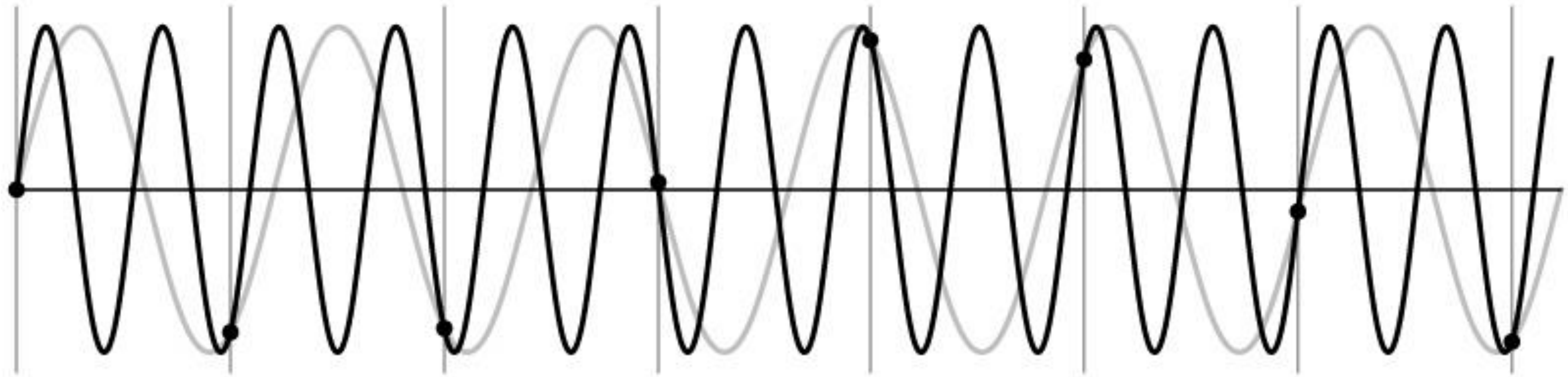


Undersampling

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



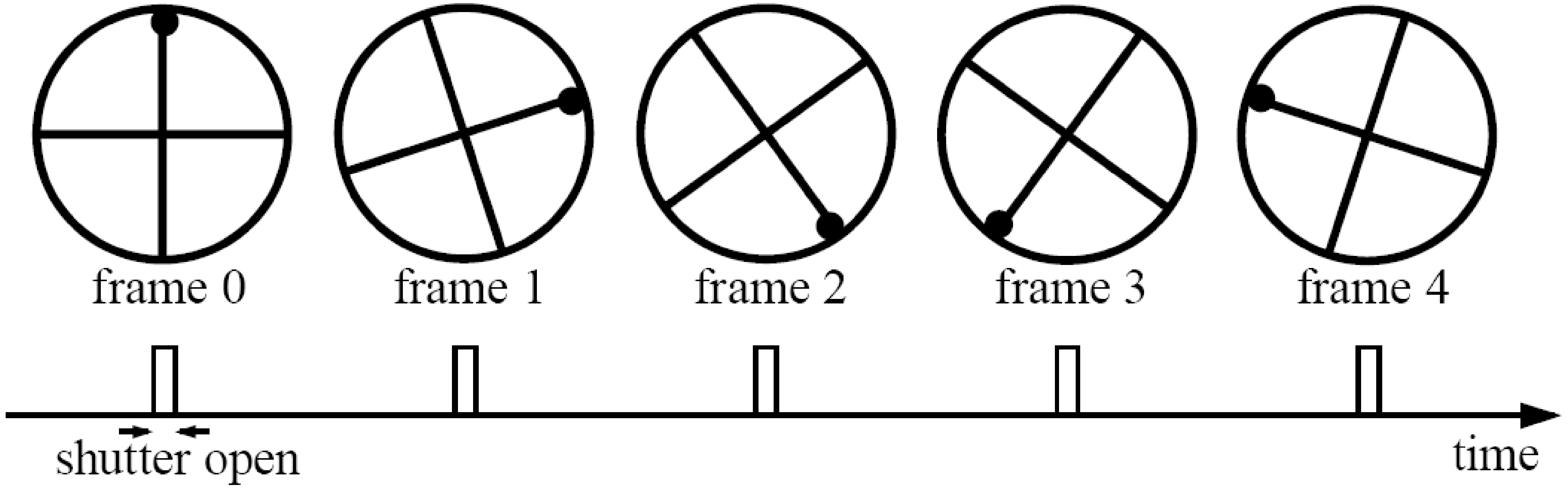
Aliasing in video

https://www.youtube.com/watch?v=ByTsISFXUoY

Aliasing in video

Imagine a spoked wheel moving to the right (rotating clockwise). Mark wheel with dot so we can see what's happening.

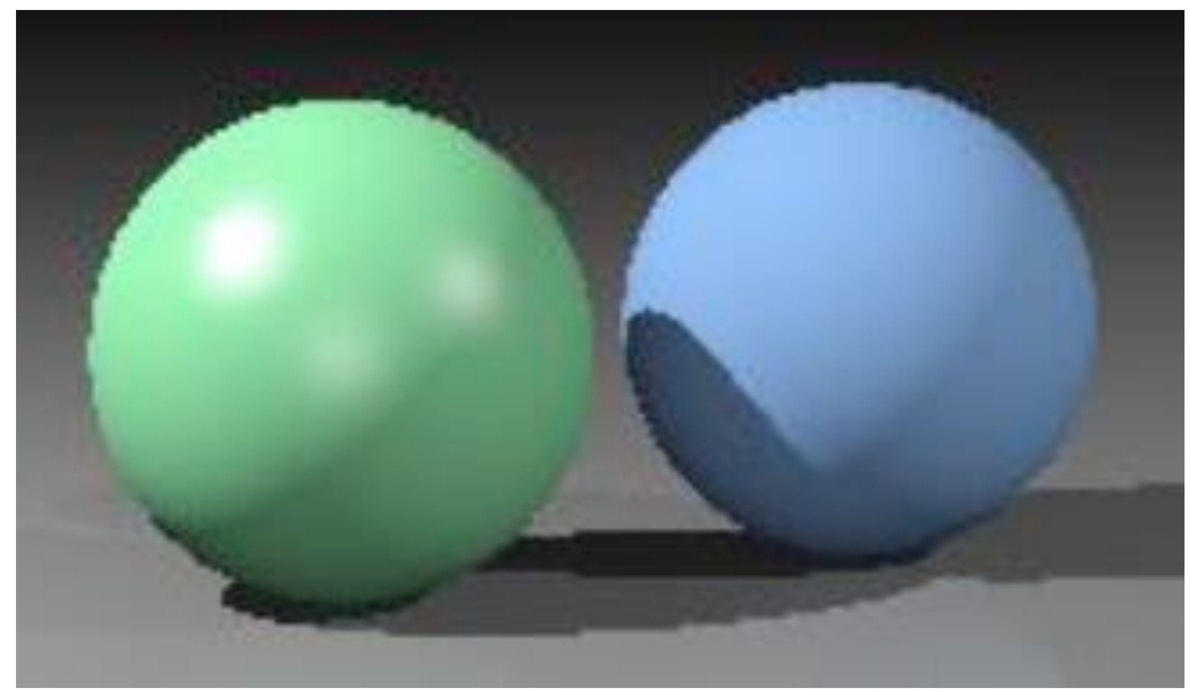
If camera shutter is only open for a fraction of a frame time (frame time = 1/30 sec. for video, 1/24 sec. for film):



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

Slide by Steve Seitz

More aliasing examples



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

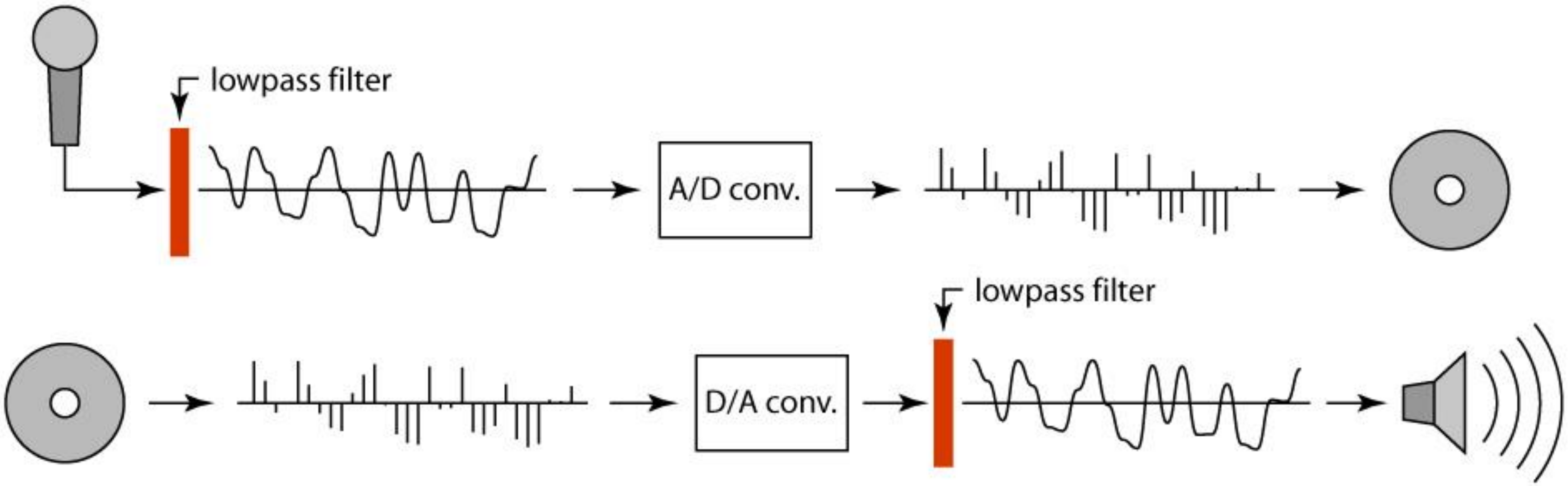
Make the signal less "wiggly"

- 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

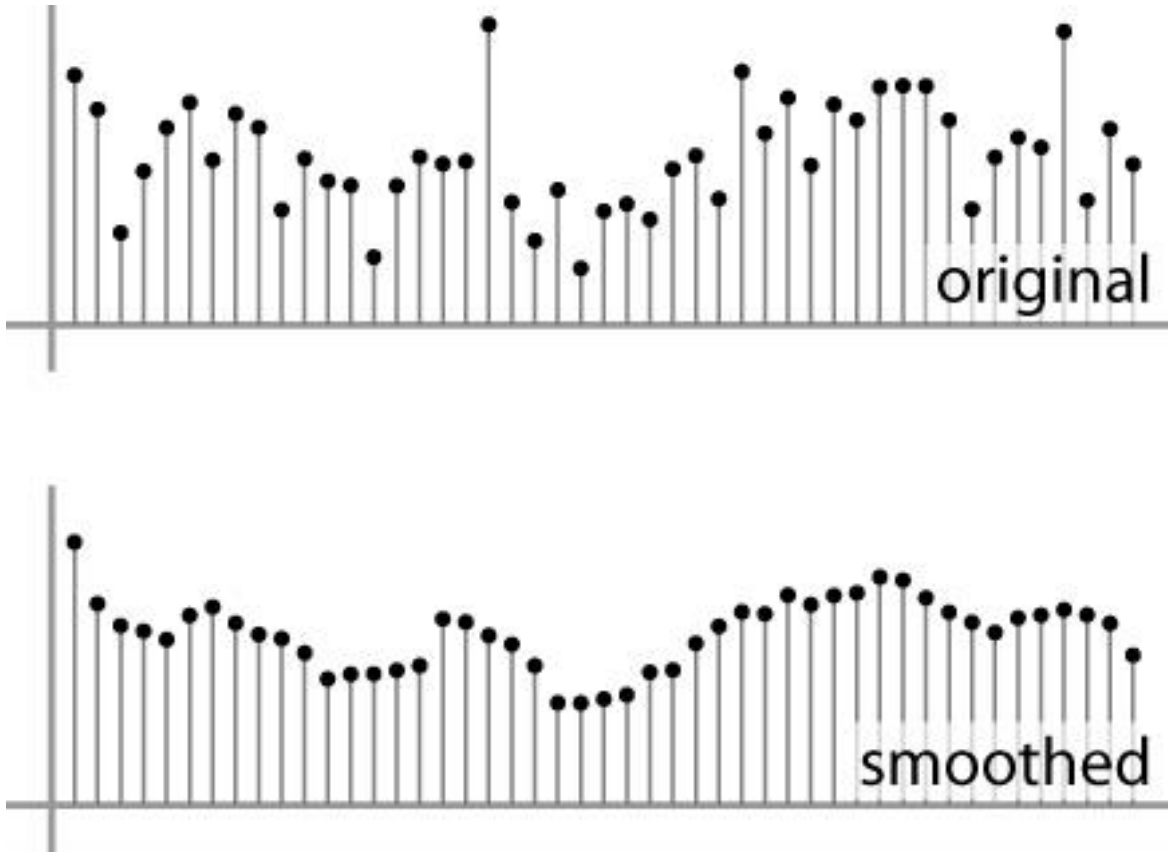
- linearity: $filter(\alpha f + g) = \alpha filter(f) + filter(g)$
- shift invariance: behavior invariant to shifting the input
  - delaying an audio signal
  - sliding an image around

Can be modeled mathematically by *convolution*

Moving Average

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

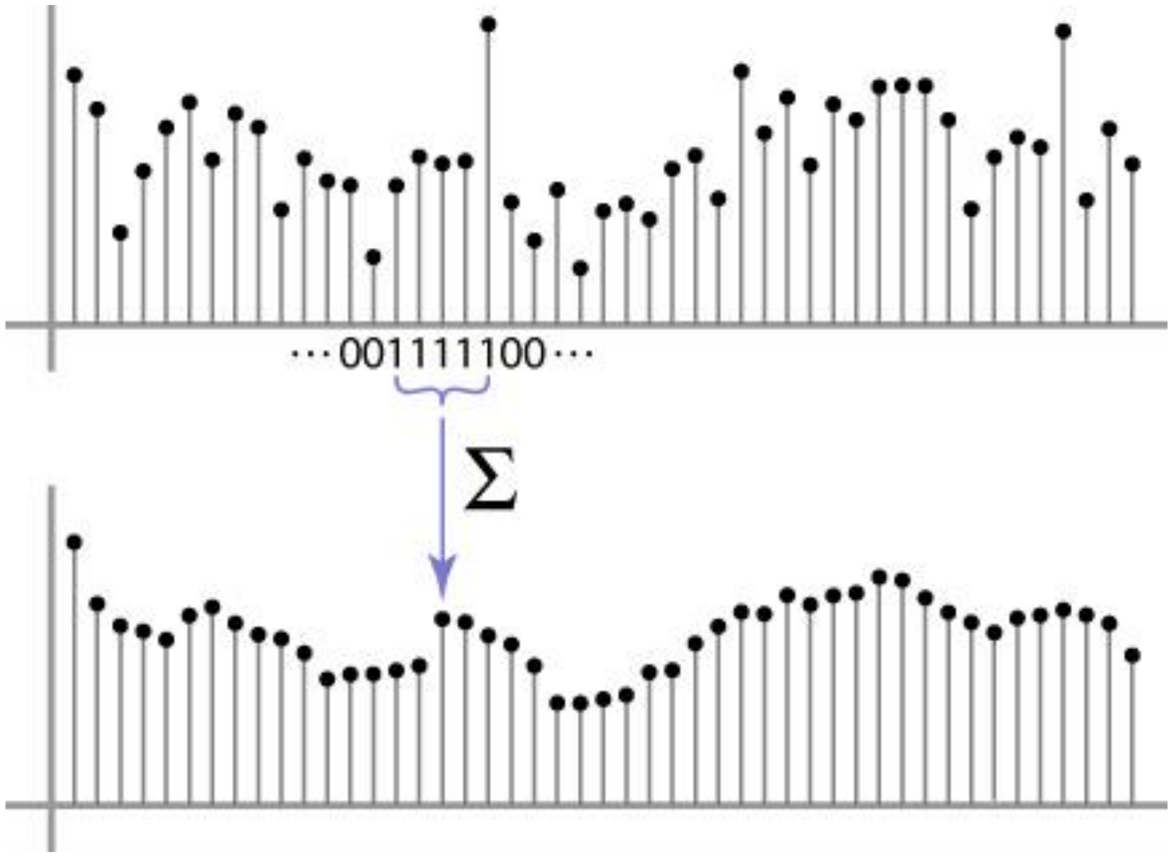
a simple example to start off: smoothing



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 $(2k+1) \times (2k+1)$, and G be the output image

$$G(x, y) = \sum_{u=-k}^{u=k} \sum_{v=-k}^{v=k} H(u, v) F(x+u, y+v)$$

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

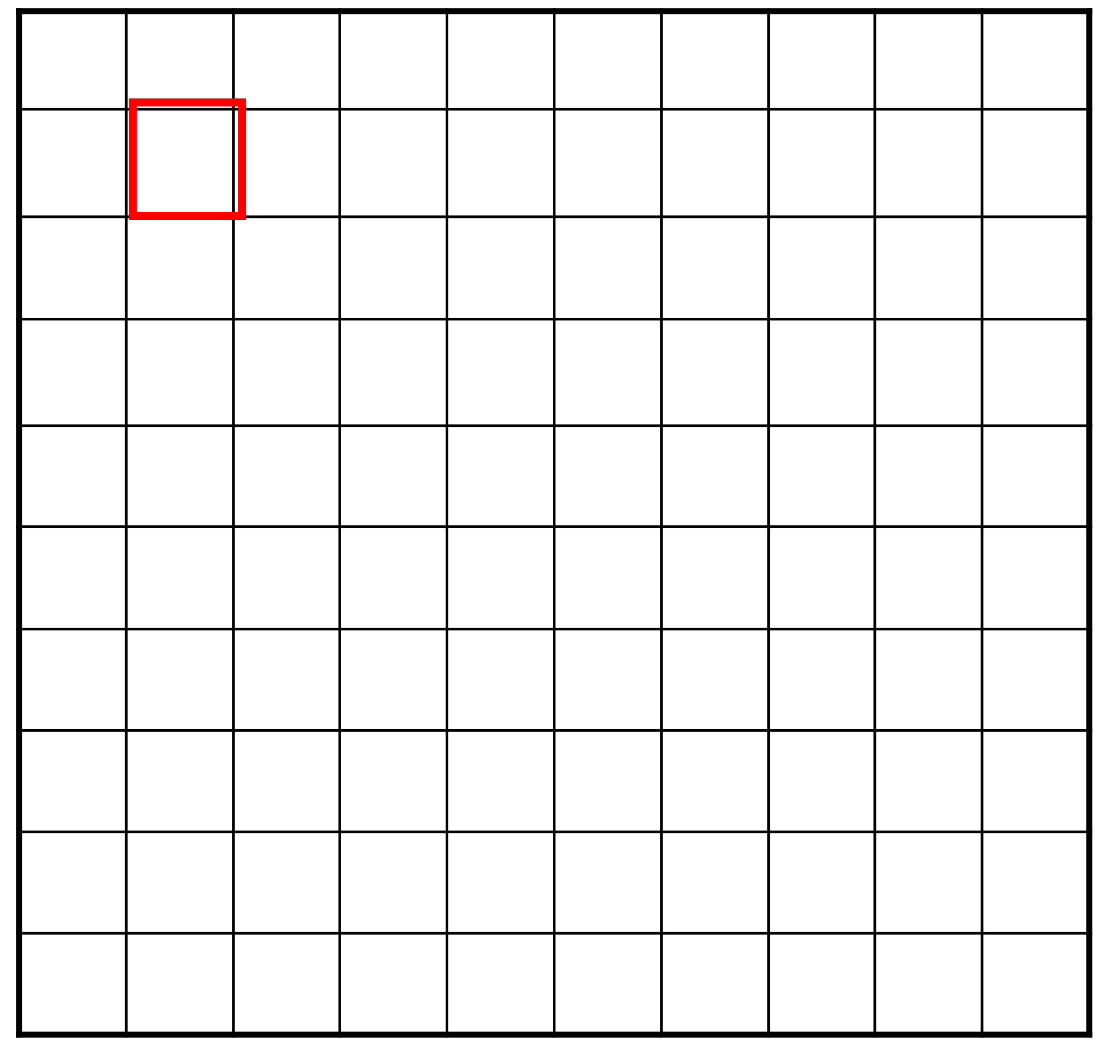
$h(\cdot,\cdot)$ $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f(\cdot,\cdot)$

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

$g(\cdot,\cdot)$



$$g(u,v) = \sum_{x,y} h(x,y)\ f(u+x, v+y)$$

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							

$$g(u,v)=\sum_{x,y} h(x,y)\; f(u+x,v+y)$$

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						

$$g(u,v) = \sum_{x,y} h(x,y)\; f(u+x, v+y)$$

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					

$$g(u,v)=\sum_{x,y} h(x,y)\ f(u+x,v+y)$$

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				

$$g(u,v) = \sum_{x,y} h(x,y)\ f(u+x, v+y)$$

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				
				?					

$$g(u,v) = \sum_{x,y} h(x,y)\, f(u+x, v+y)$$

Credit: S. Seitz

Image filtering

$$h(\cdot,\cdot) \quad \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$f(\cdot,\cdot)$

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					

$$g(u,v) = \sum_{x,y} h(x,y)\; f(u+x, v+y)$$

Credit: S. Seitz

Image filtering

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

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

	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	

Credit: S. Seitz

Box Filter

What does it do?

- Replaces each pixel with an average of its neighborhood
- Achieve smoothing effect (remove sharp features)

$$h(\cdot\,,\cdot\,)$$

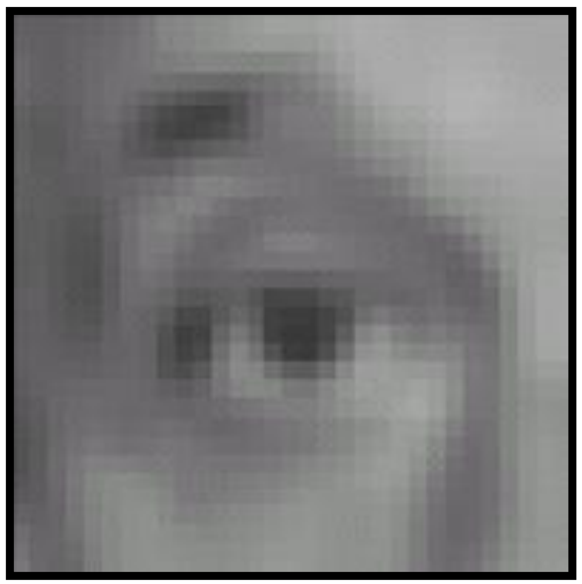
$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

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
1	0	0
0	0	0

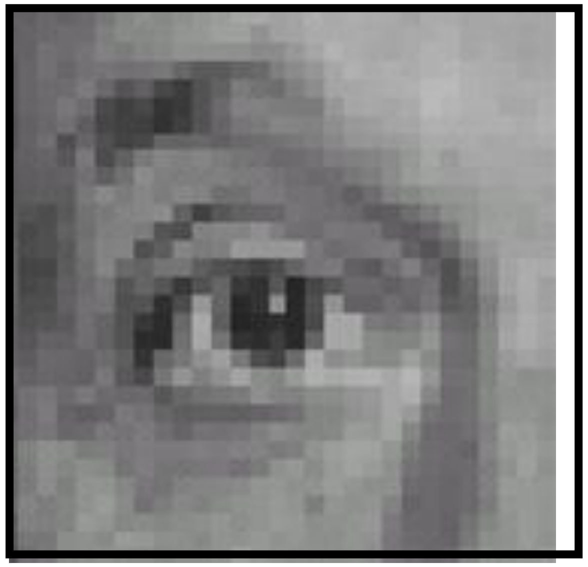
?

Source: D. Lowe

Practice with linear filters

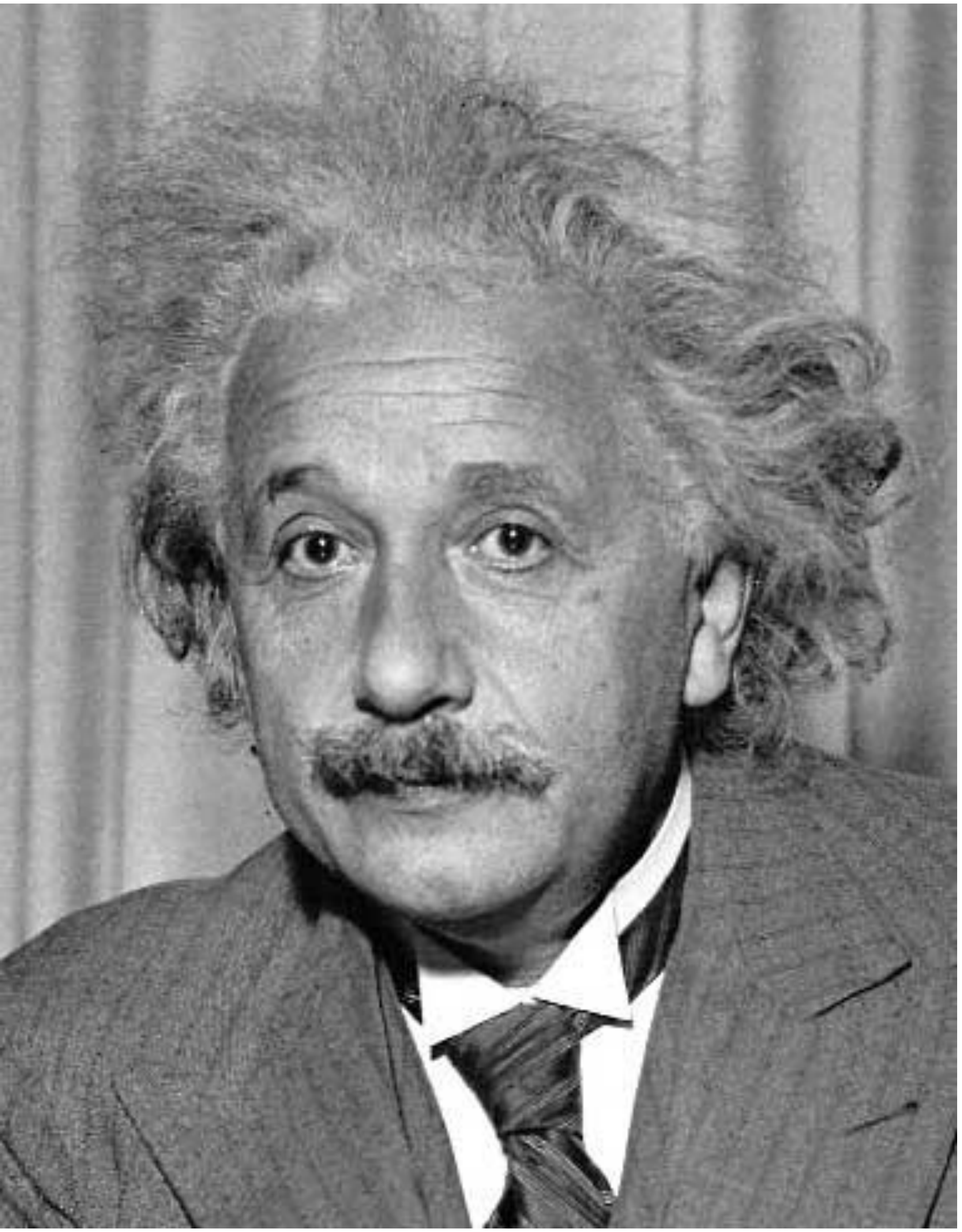
Original

0	0	0
1	0	0
0	0	0



Shifted left
By 1 pixel

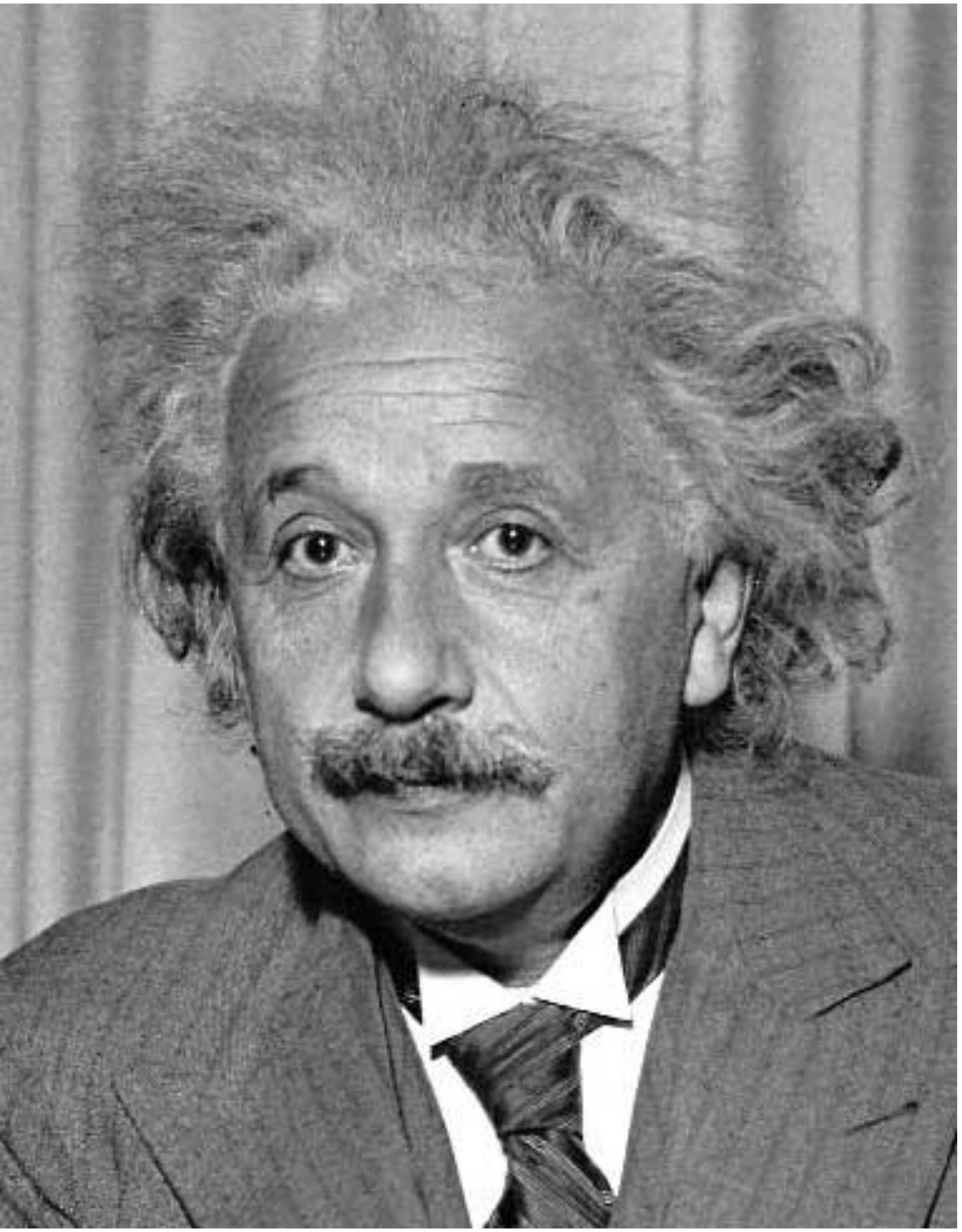
Source: D. Lowe



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

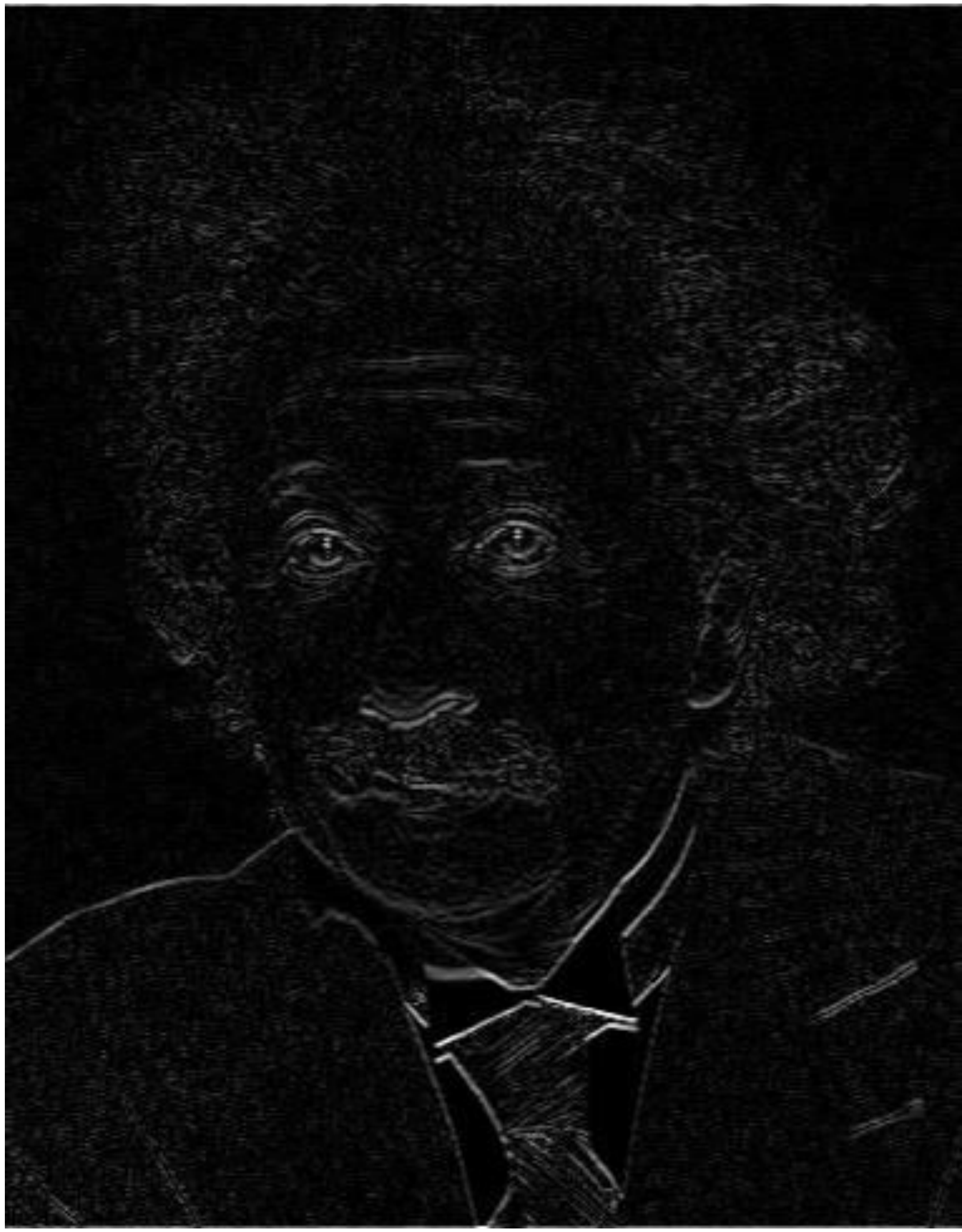
Sobel

Vertical Edge
(absolute value)



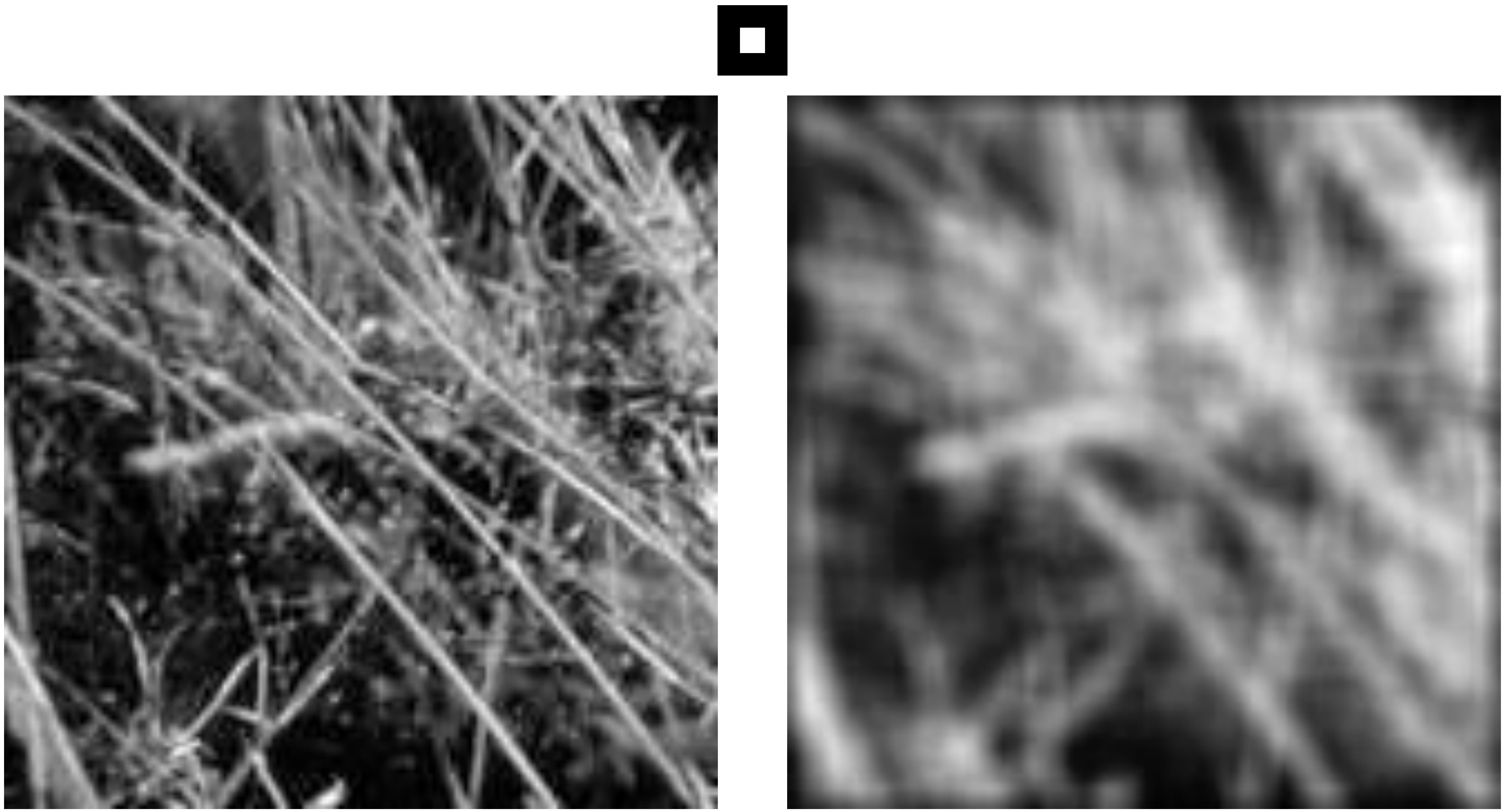
1	2	1
0	0	0
-1	-2	-1

Sobel



Horizontal Edge
(absolute value)

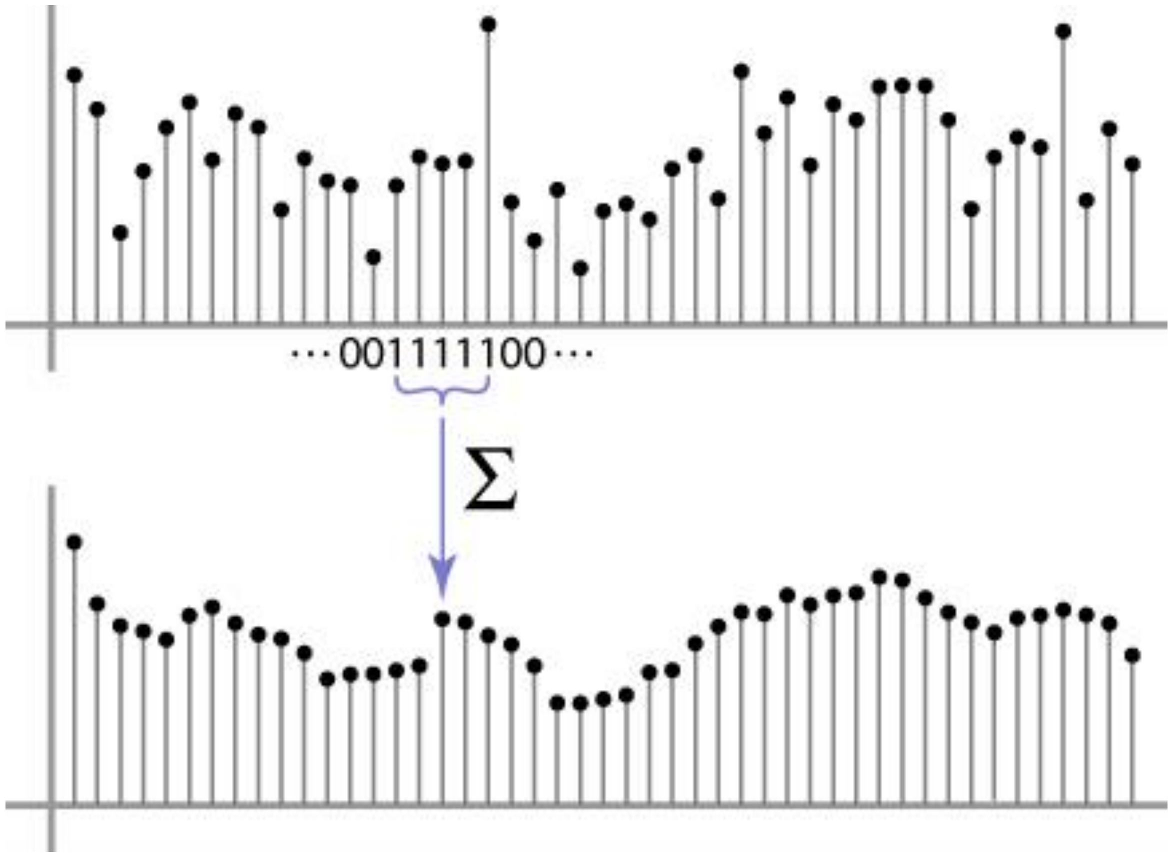
Back to the box filter



Moving Average

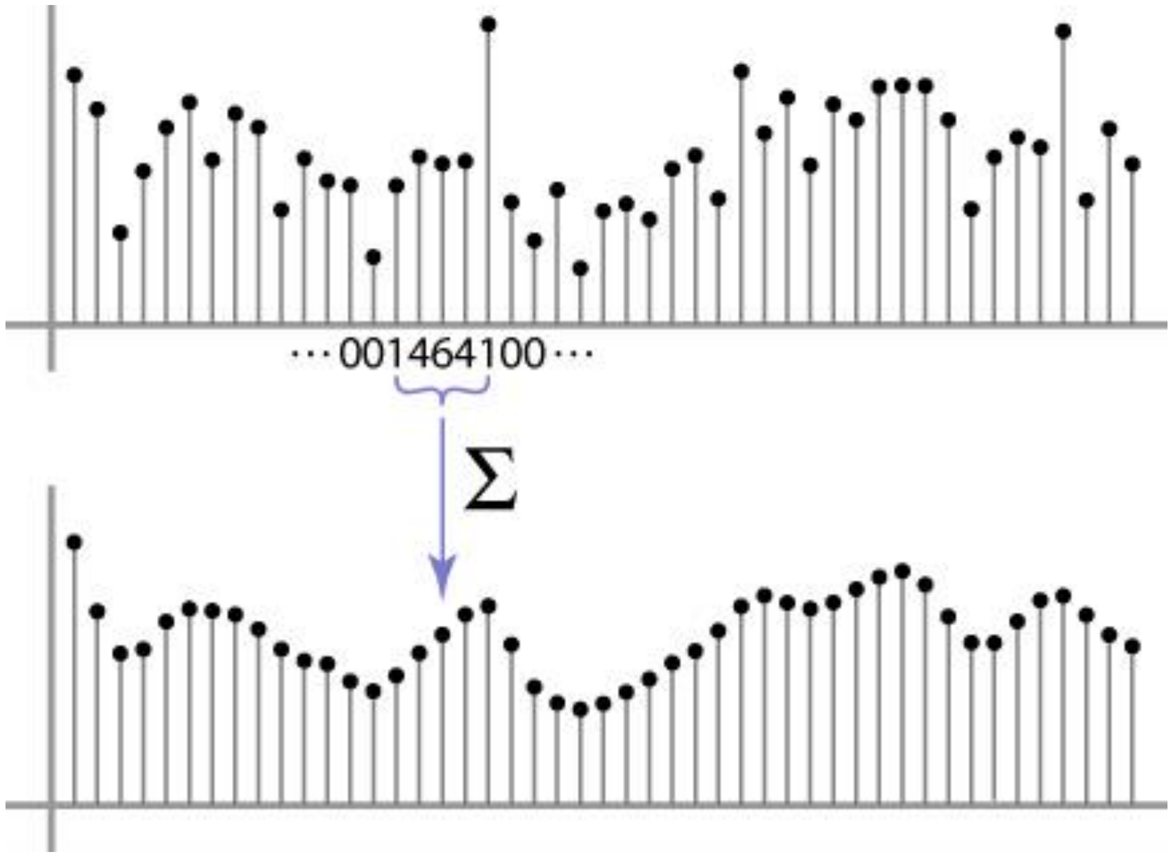
Can add weights to our moving average

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



Weighted Moving Average

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



Moving Average In 2D

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$F(u, v)$

$H(x, y)$

Moving Average In 2D

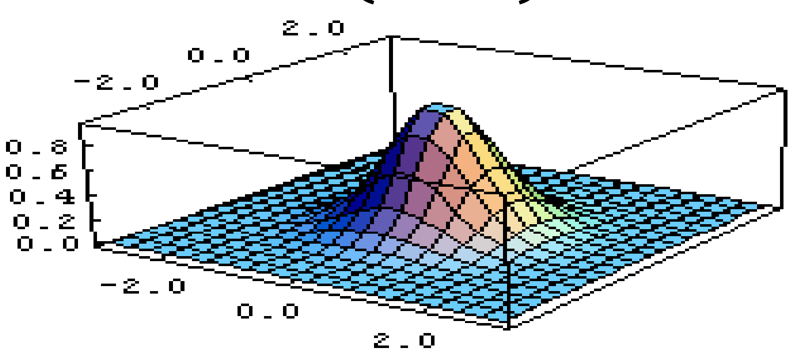
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$\frac{1}{16}$

1	2	1
2	4	2
1	2	1

$H(x, y)$

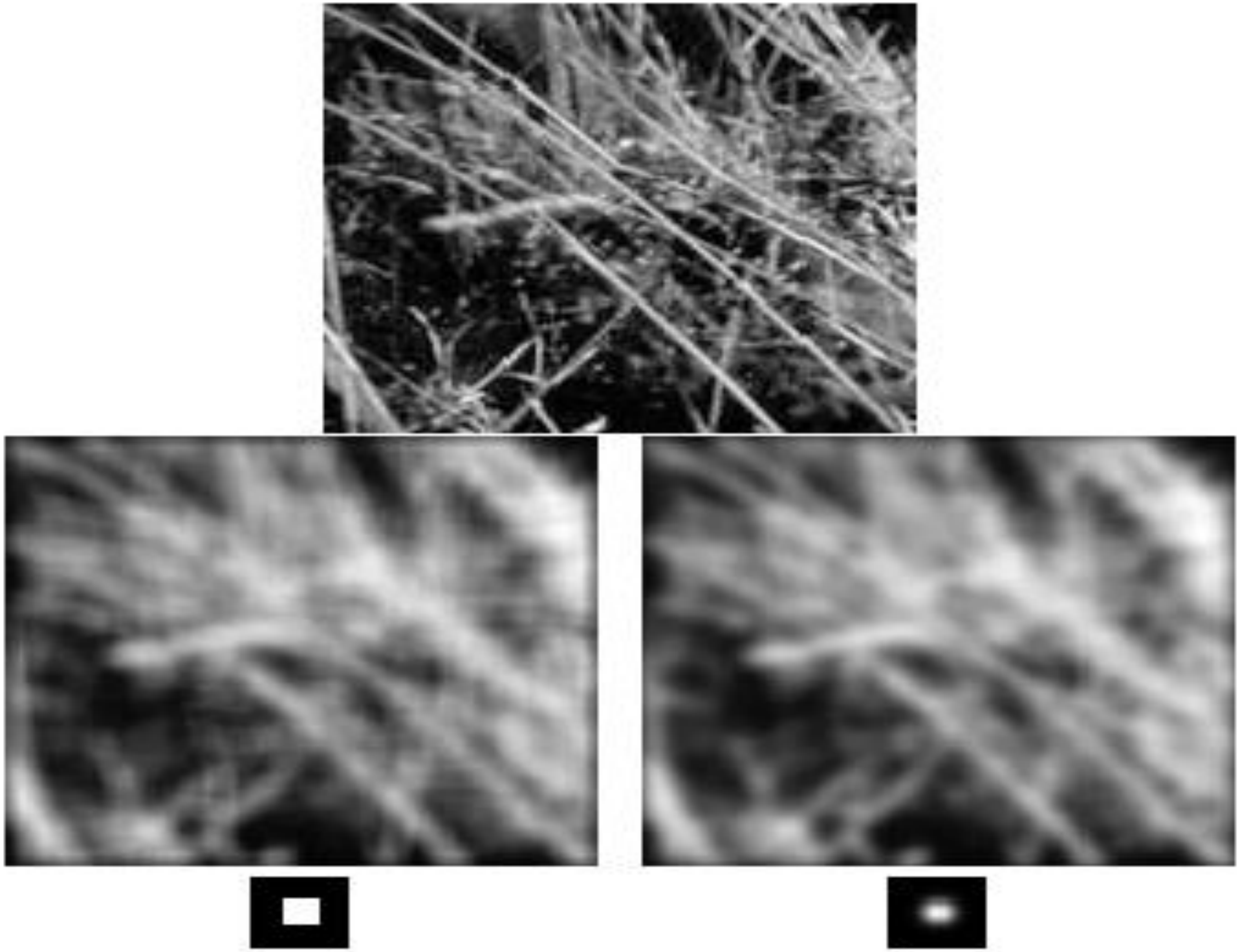
$F(u, v)$



$$h(i, j) = \frac{1}{2\pi\sigma^2}\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right)$$

Slide Credit: S. Seitz ad Marschner

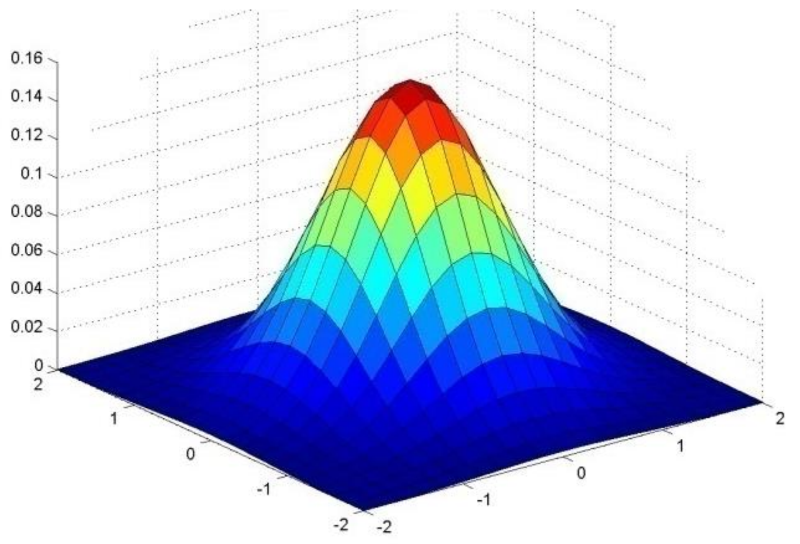
Mean vs. Gaussian filtering

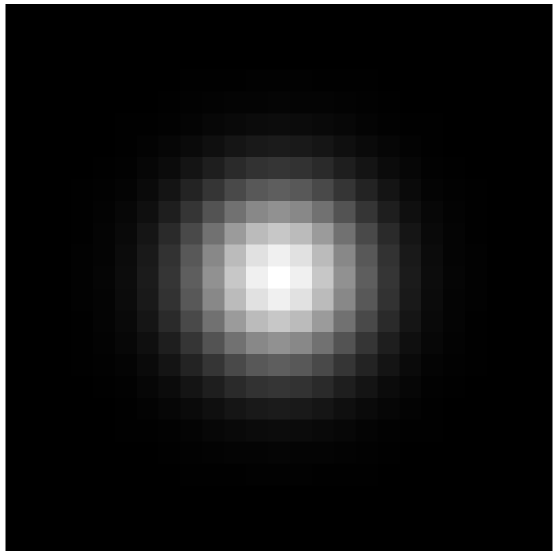


Slide by Steve Seitz

Important filter: Gaussian

Weight contributions of neighboring pixels by nearness





0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

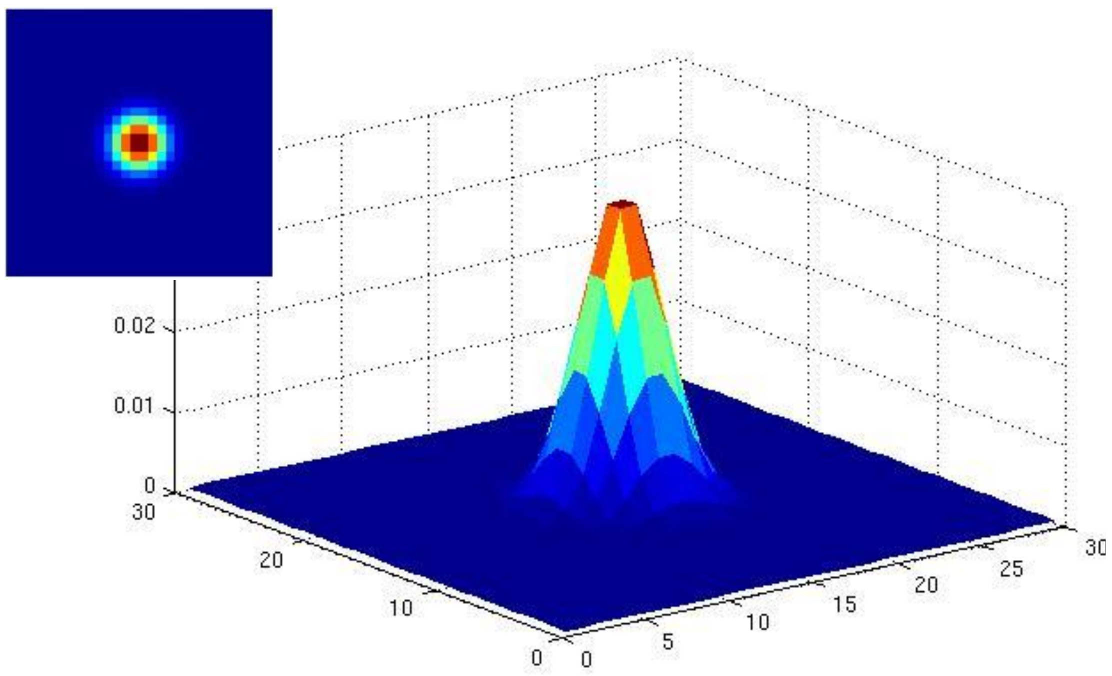
5 x 5, σ = 1

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

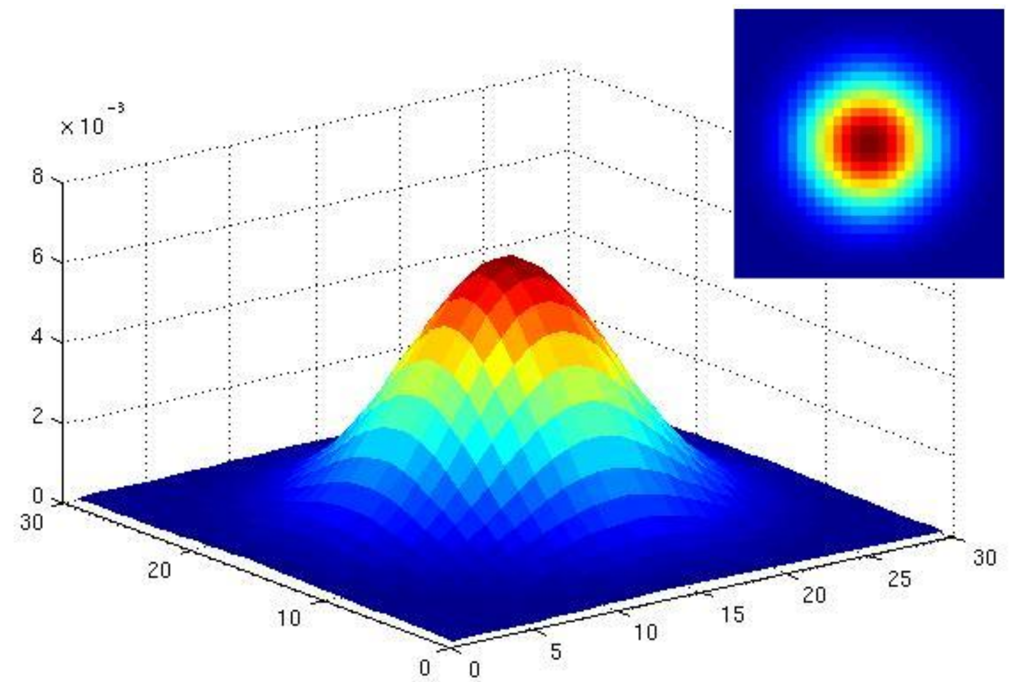
Slide credit: Christopher Rasmussen

Gaussian Kernel

- Standard deviation σ: determines extent of smoothing



σ = 2 with 30 x 30 kernel

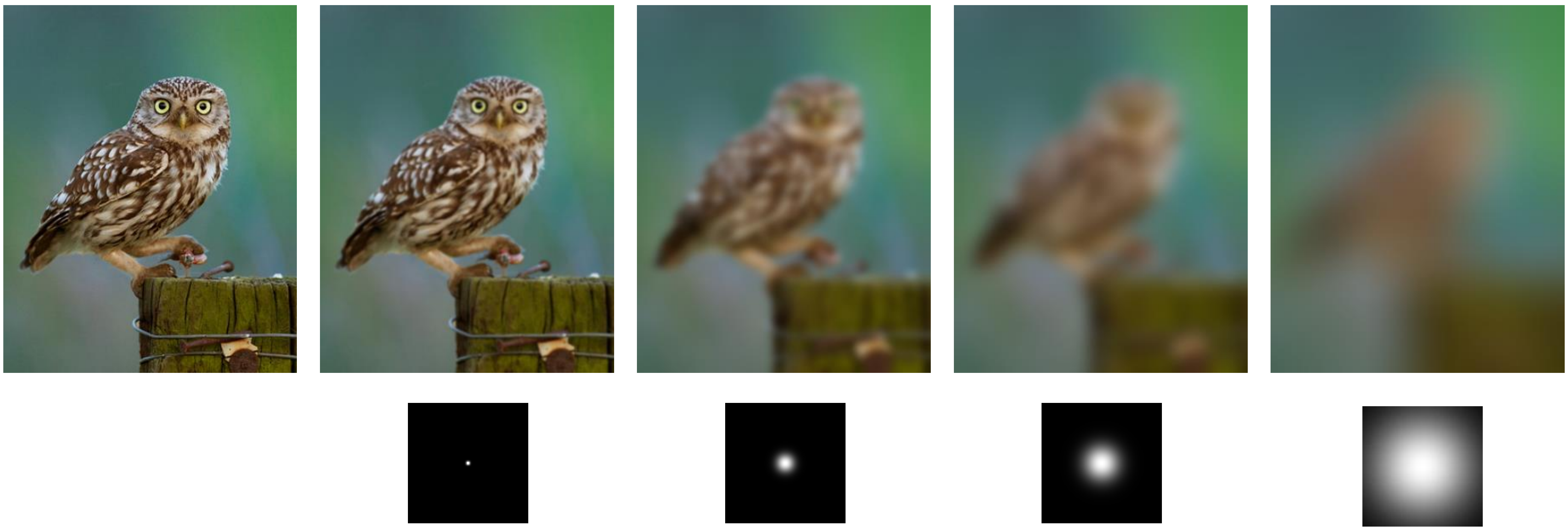


σ = 5 with 30 x 30 kernel

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Source: K. Grauman

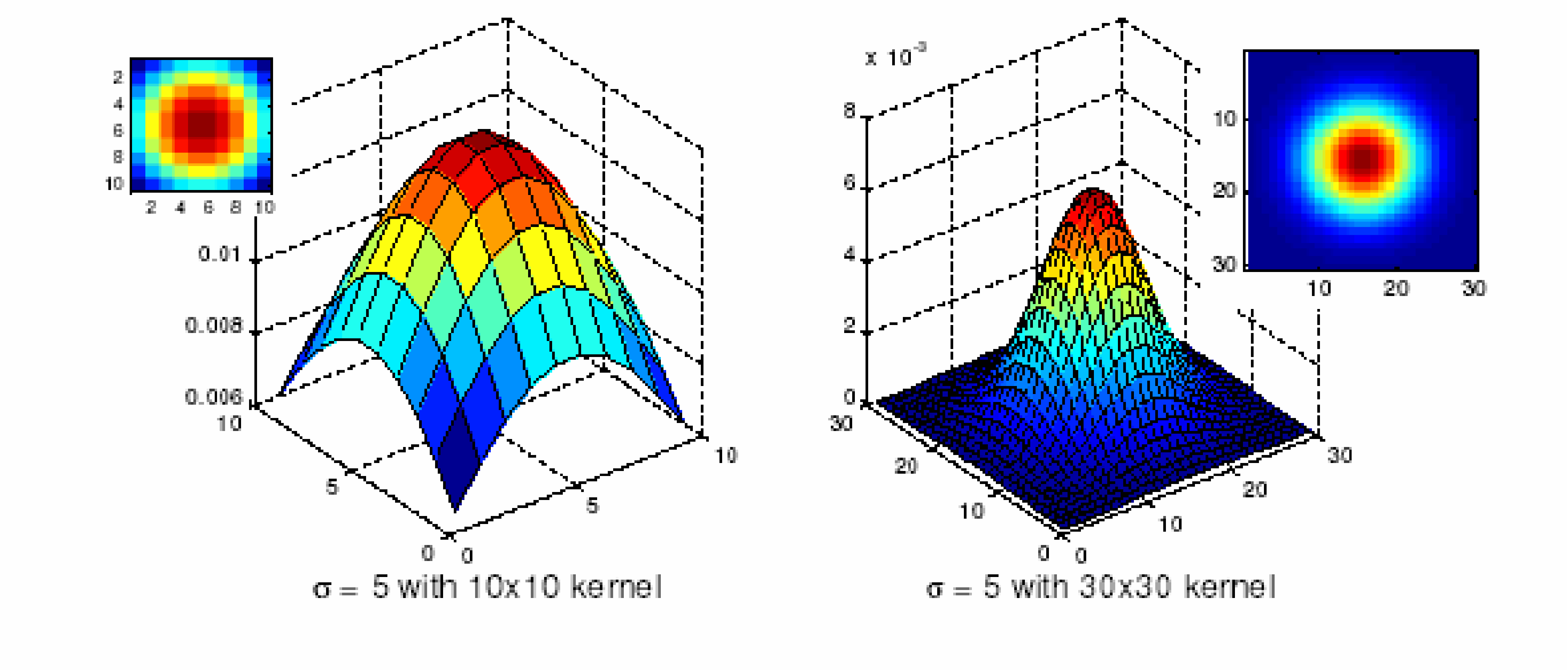
Gaussian filters



σ = 1 pixel σ = 5 pixels σ = 10 pixels σ = 30 pixels

Choosing kernel width

•The Gaussian function has infinite support, but discrete filters use finite kernels



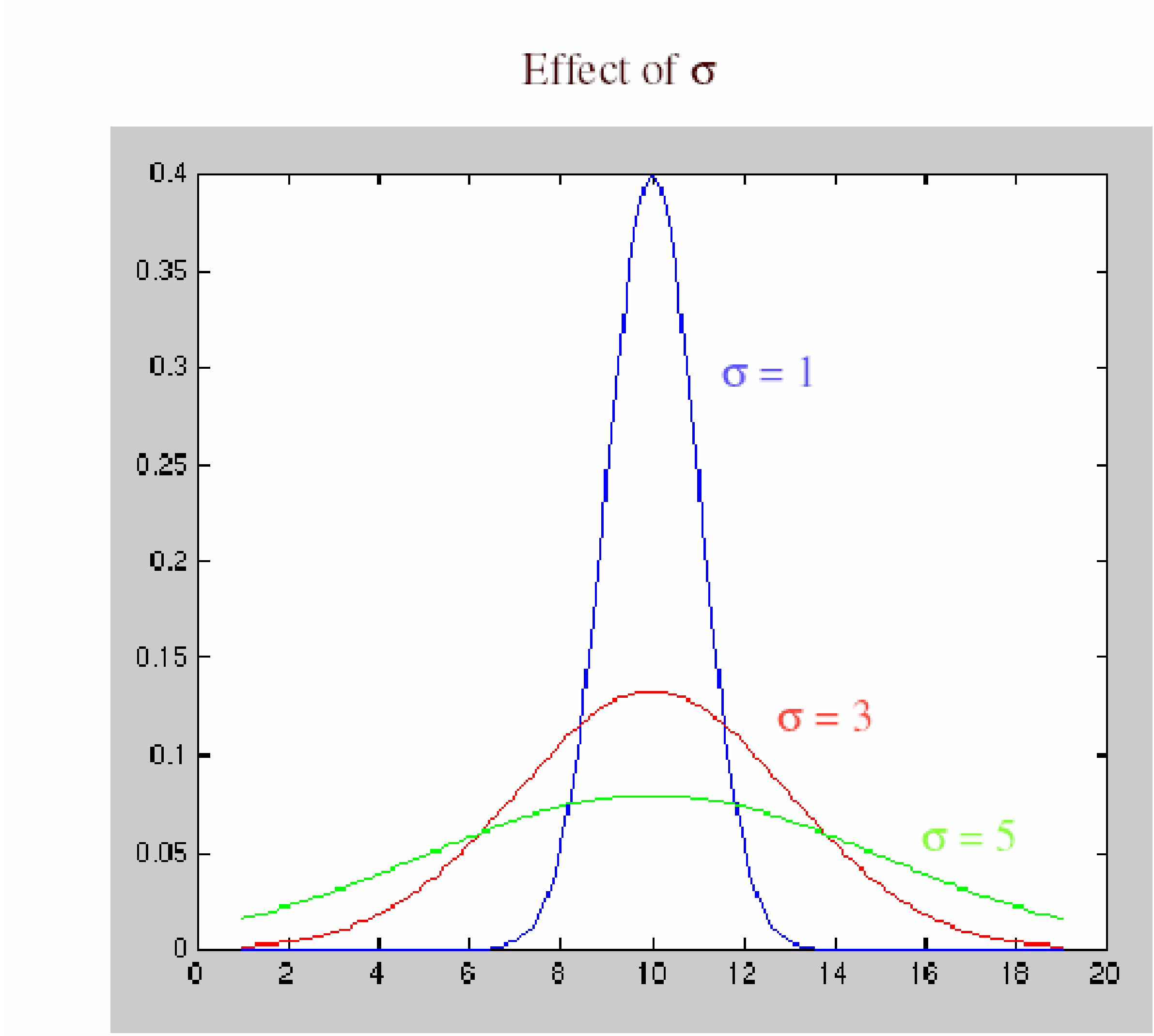
Source: K. Grauman

Practical matters

How big should the filter be?

Values at edges should be near zero

Rule of thumb for Gaussian: set filter half-width to about 3 *σ*



Side by Derek Hoiem

Image half-sizing

This image is too big to fit on the screen. How can we reduce it?

How to generate a half-sized version?

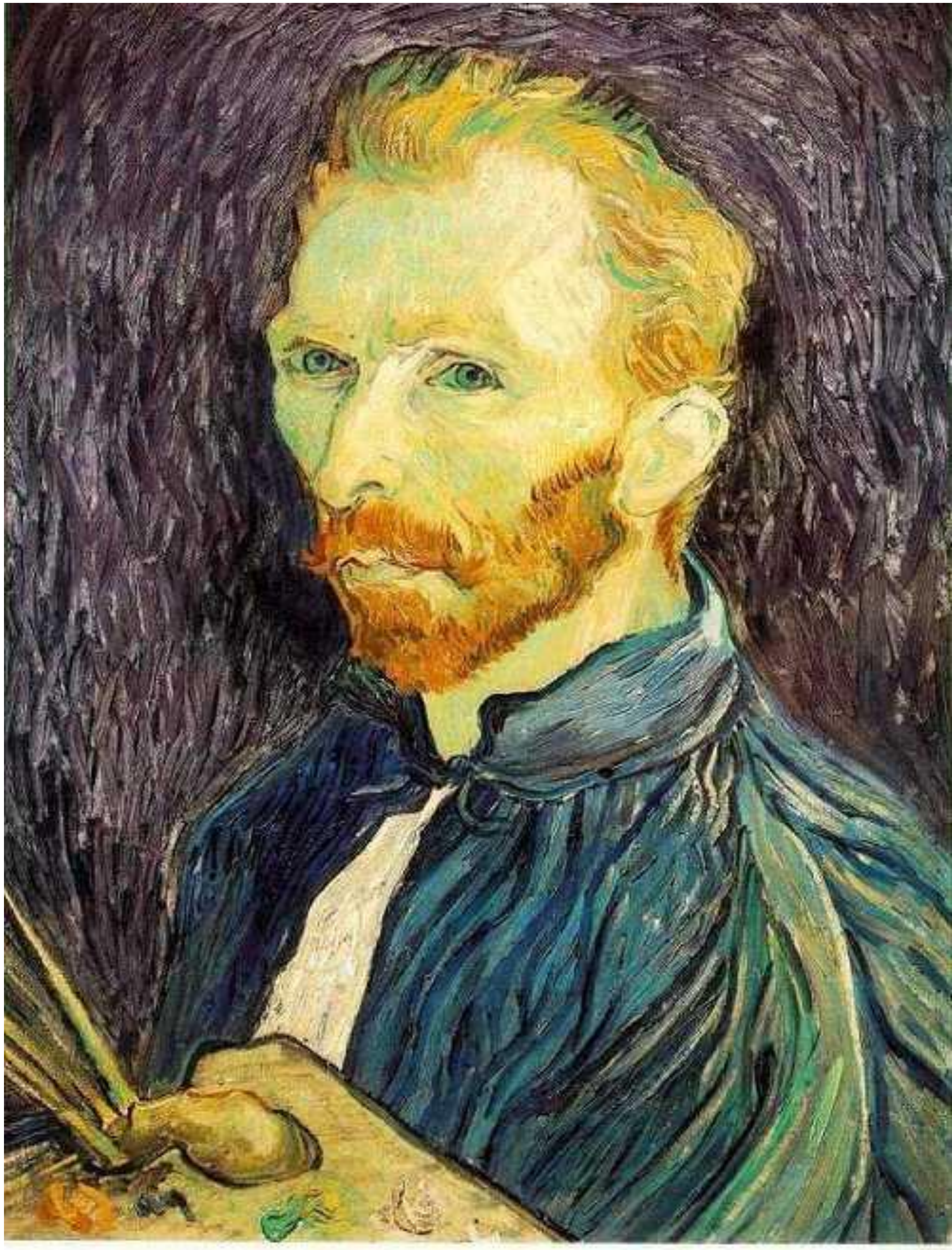
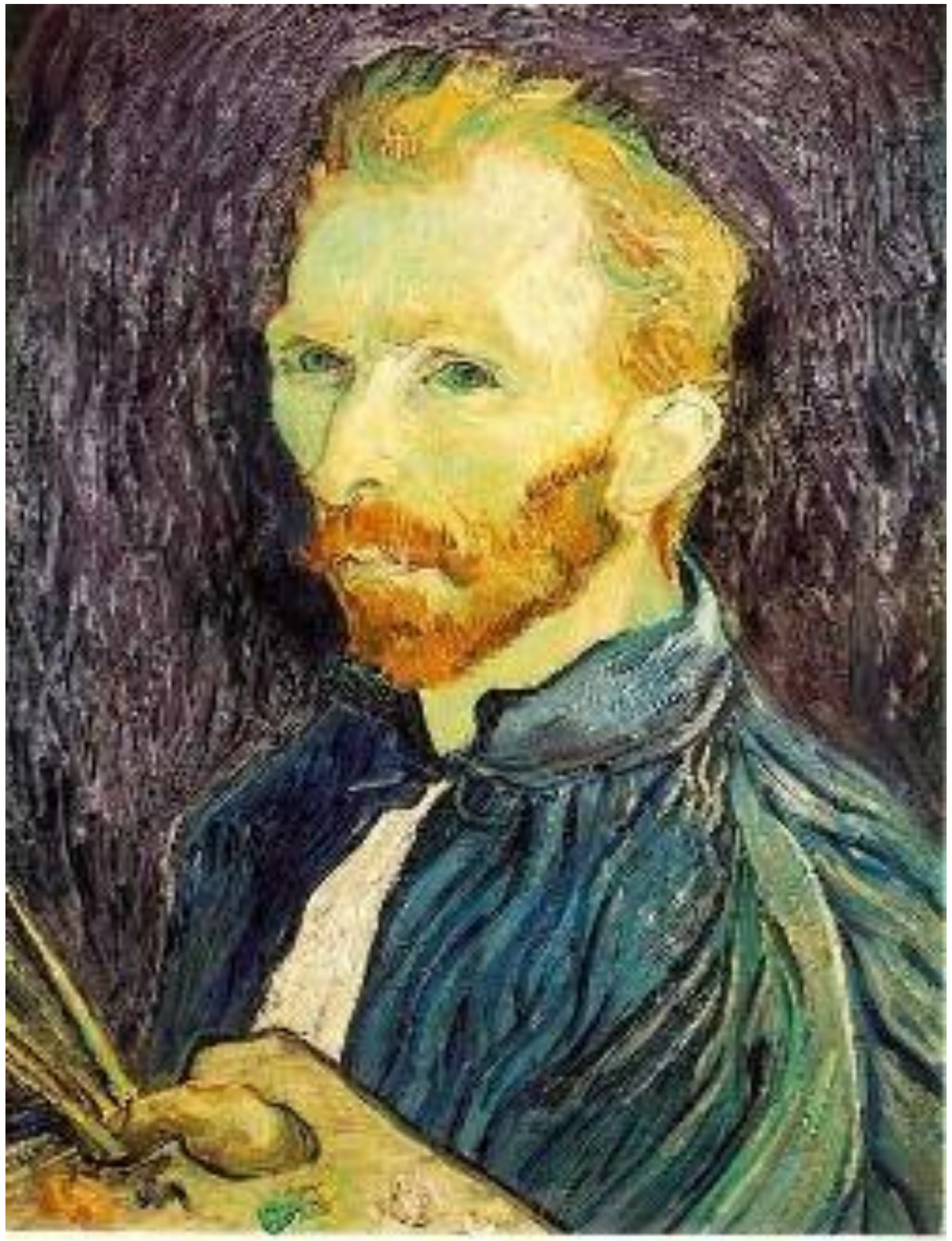


Image sub-sampling



1/2

1/4

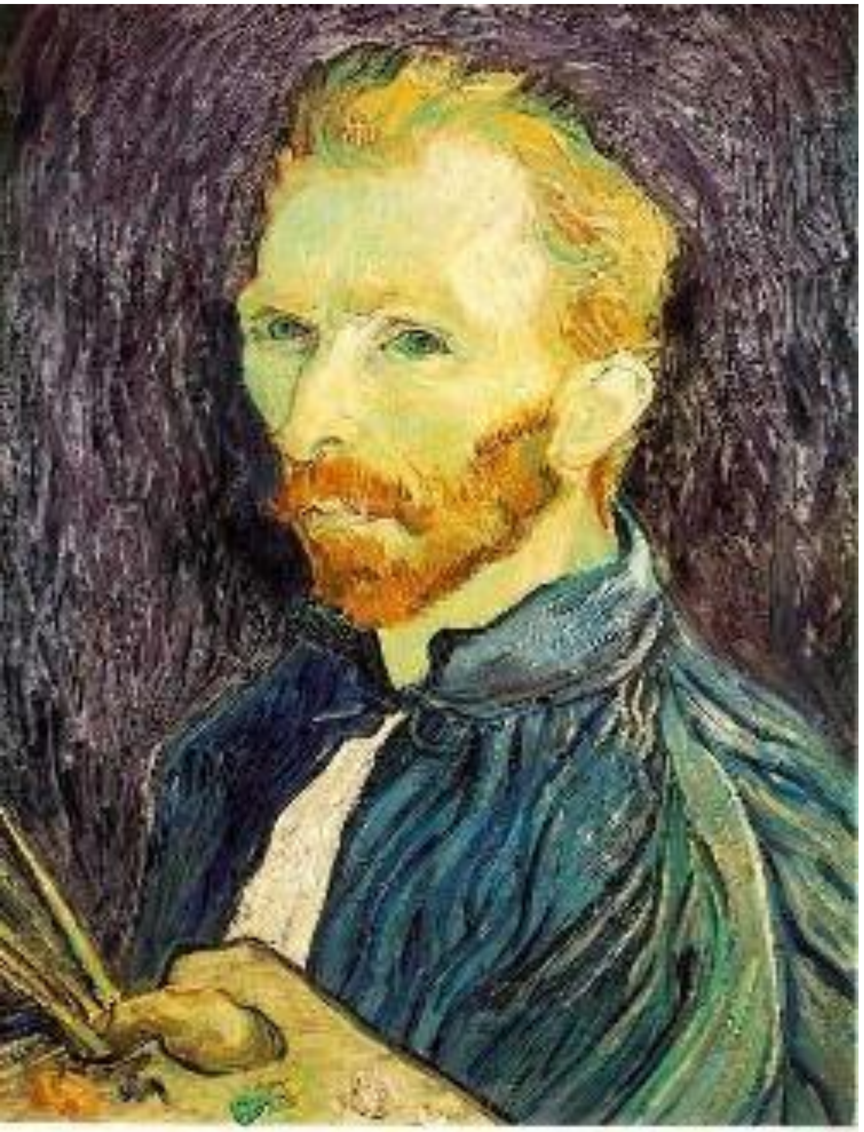
1/8

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

- called *image sub-sampling*

Slide by Steve Seitz

Image sub-sampling



1/2

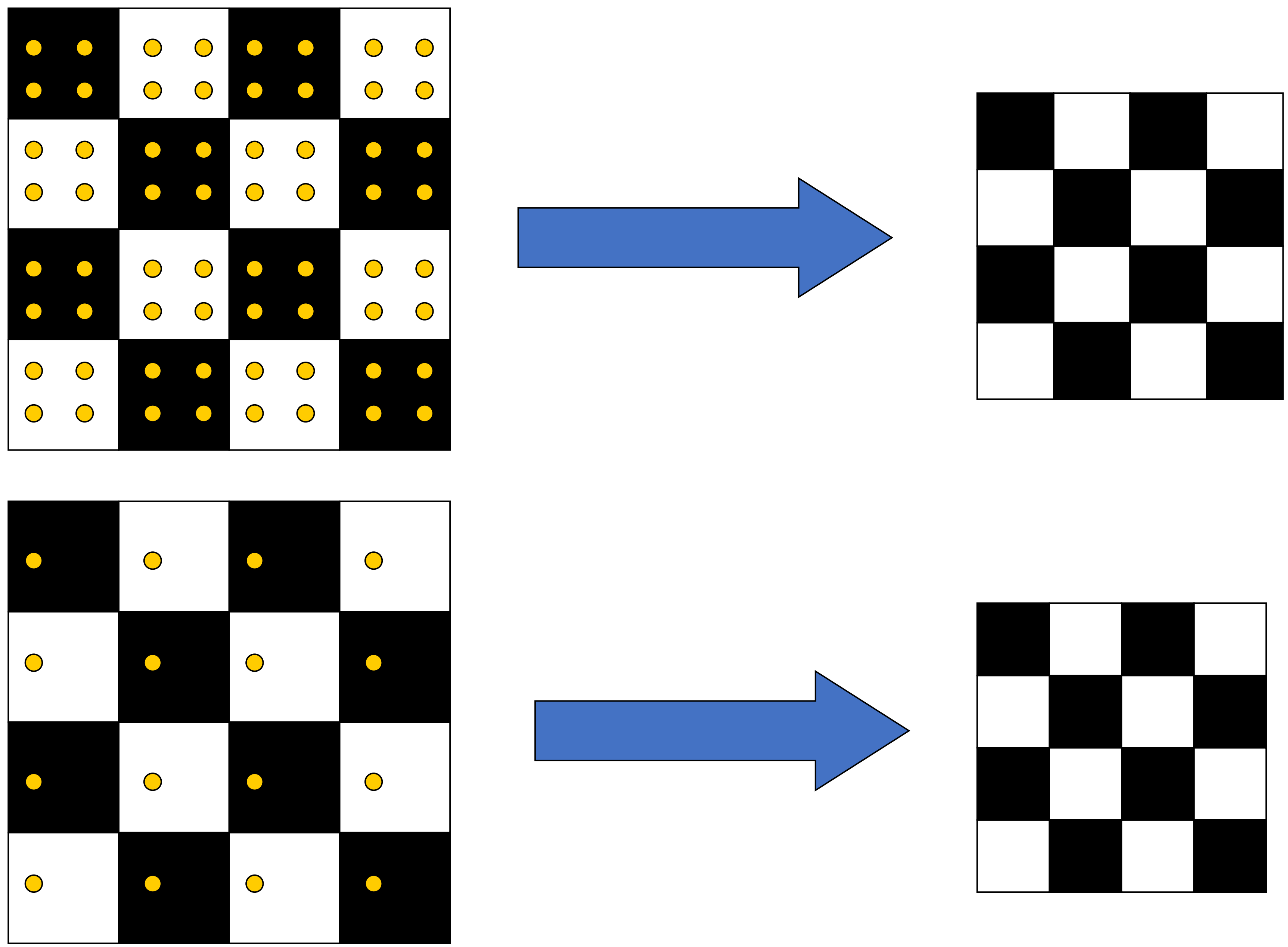
1/4 (2x zoom)

1/8 (4x zoom)

Aliasing! What do we do?

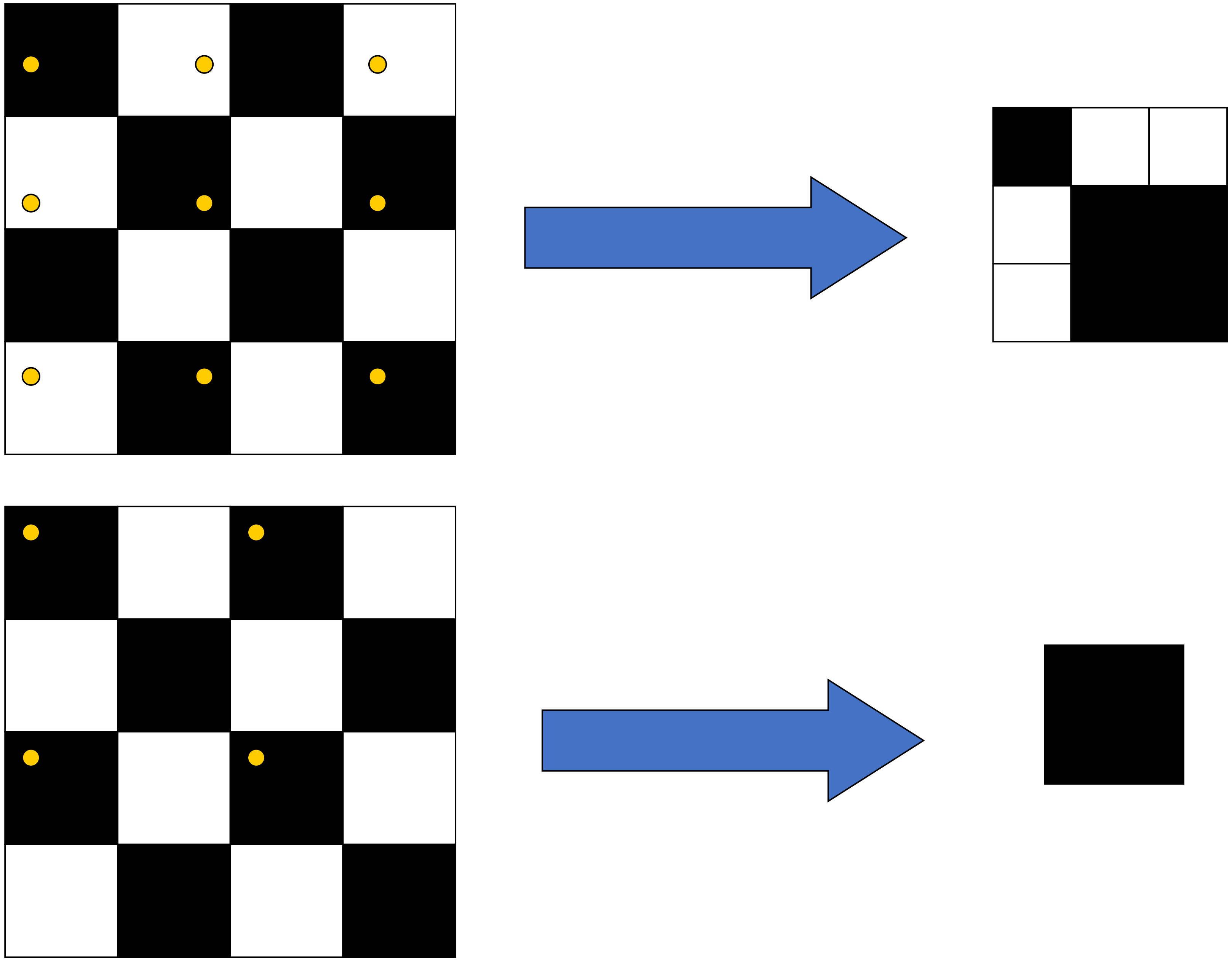
Slide by Steve Seitz

Sampling an image



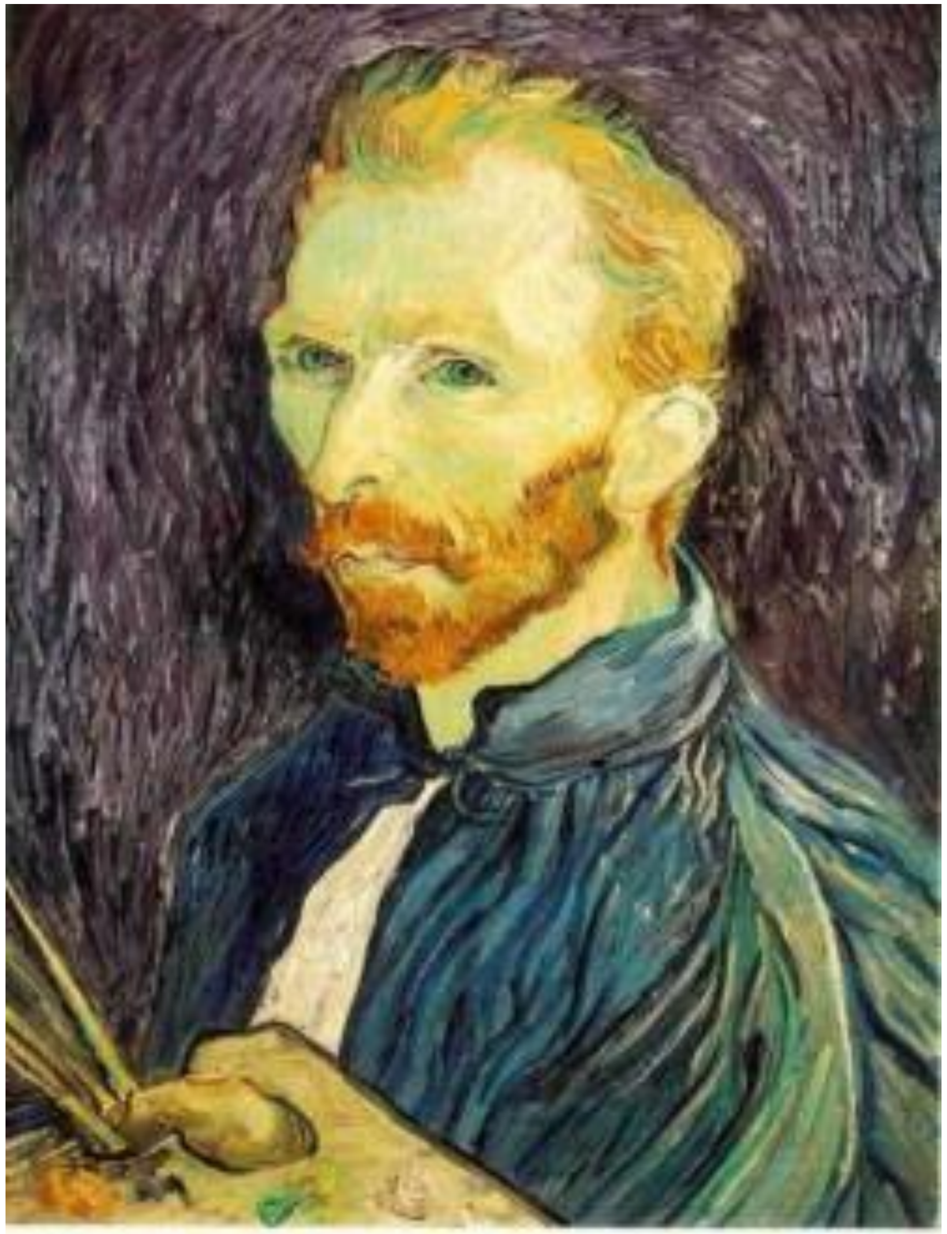
Examples of GOOD sampling

Undersampling



Examples of BAD sampling -> Aliasing

Gaussian (lowpass) pre-filtering



Gaussian 1/2

G 1/4

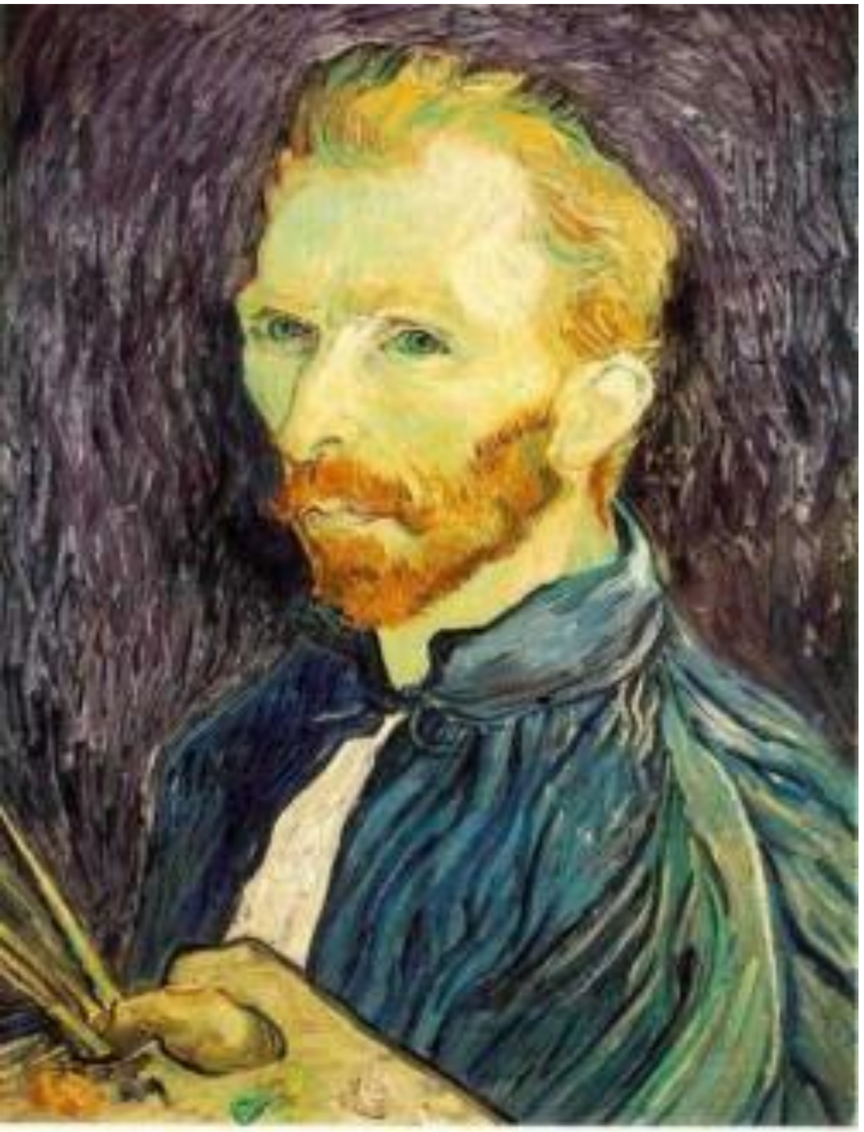
G 1/8

Solution: filter the image, *then* subsample

- Filter size should double for each ½ size reduction. Why?

Slide by Steve Seitz

Subsampling with Gaussian filtering



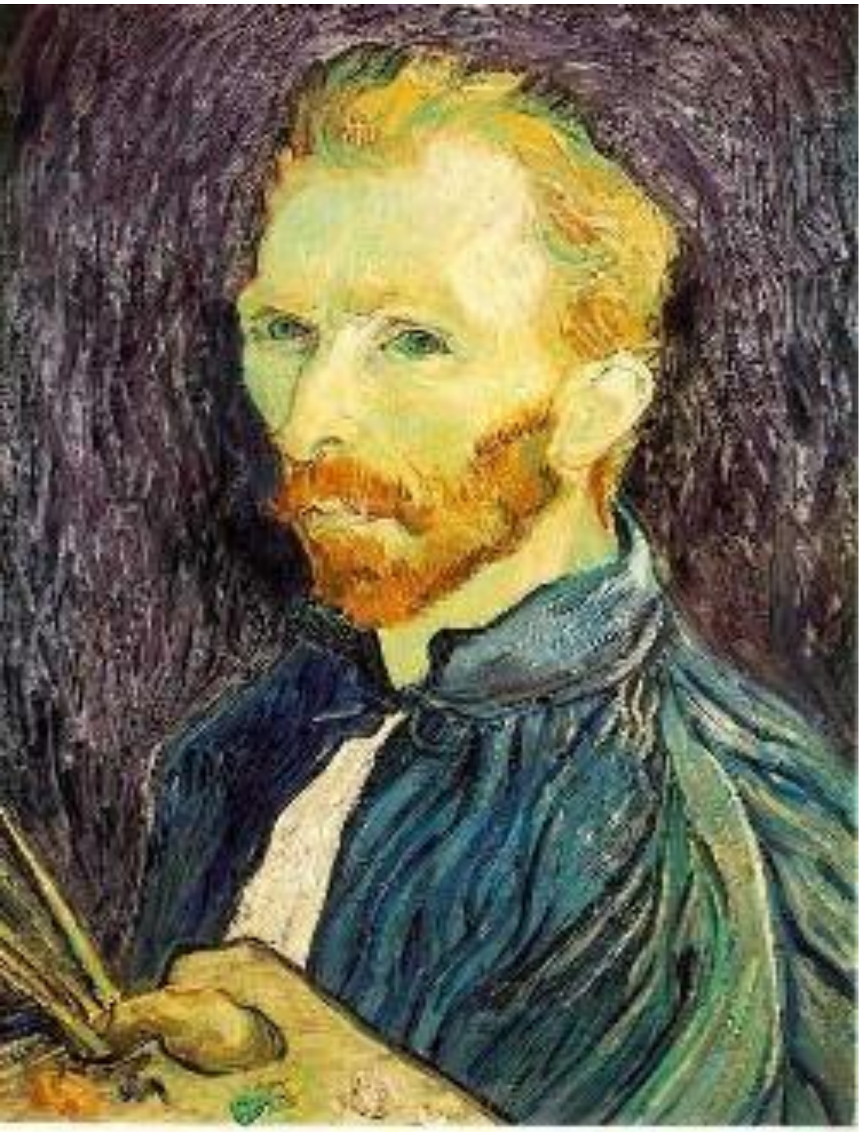
Gaussian 1/2

G 1/4

G 1/8

Slide by Steve Seitz

Compare with...



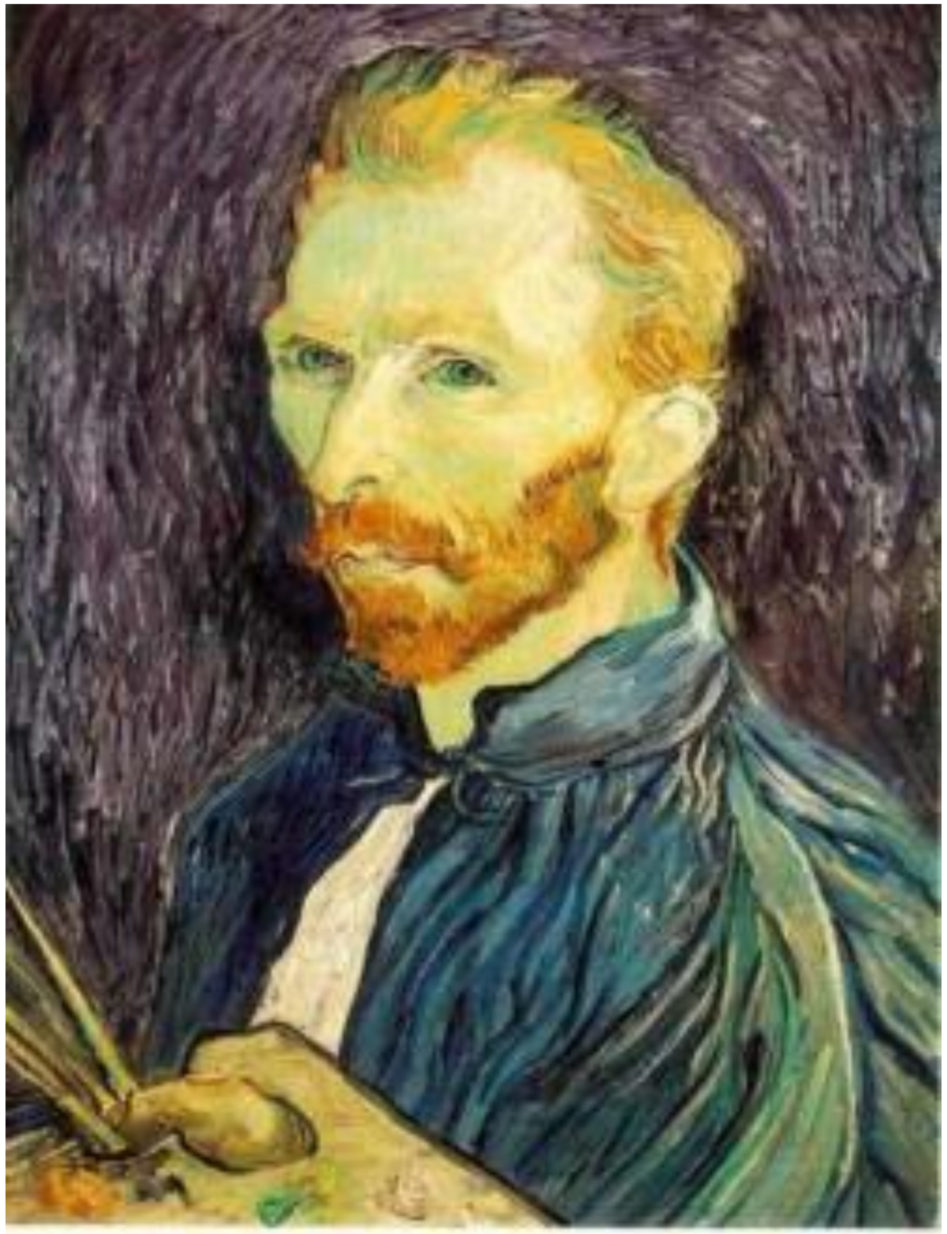
1/2

1/4 (2x zoom)

1/8 (4x zoom)

Slide by Steve Seitz

Gaussian (lowpass) pre-filtering



Gaussian 1/2

G 1/4

G 1/8

Solution: filter the image, *then* subsample

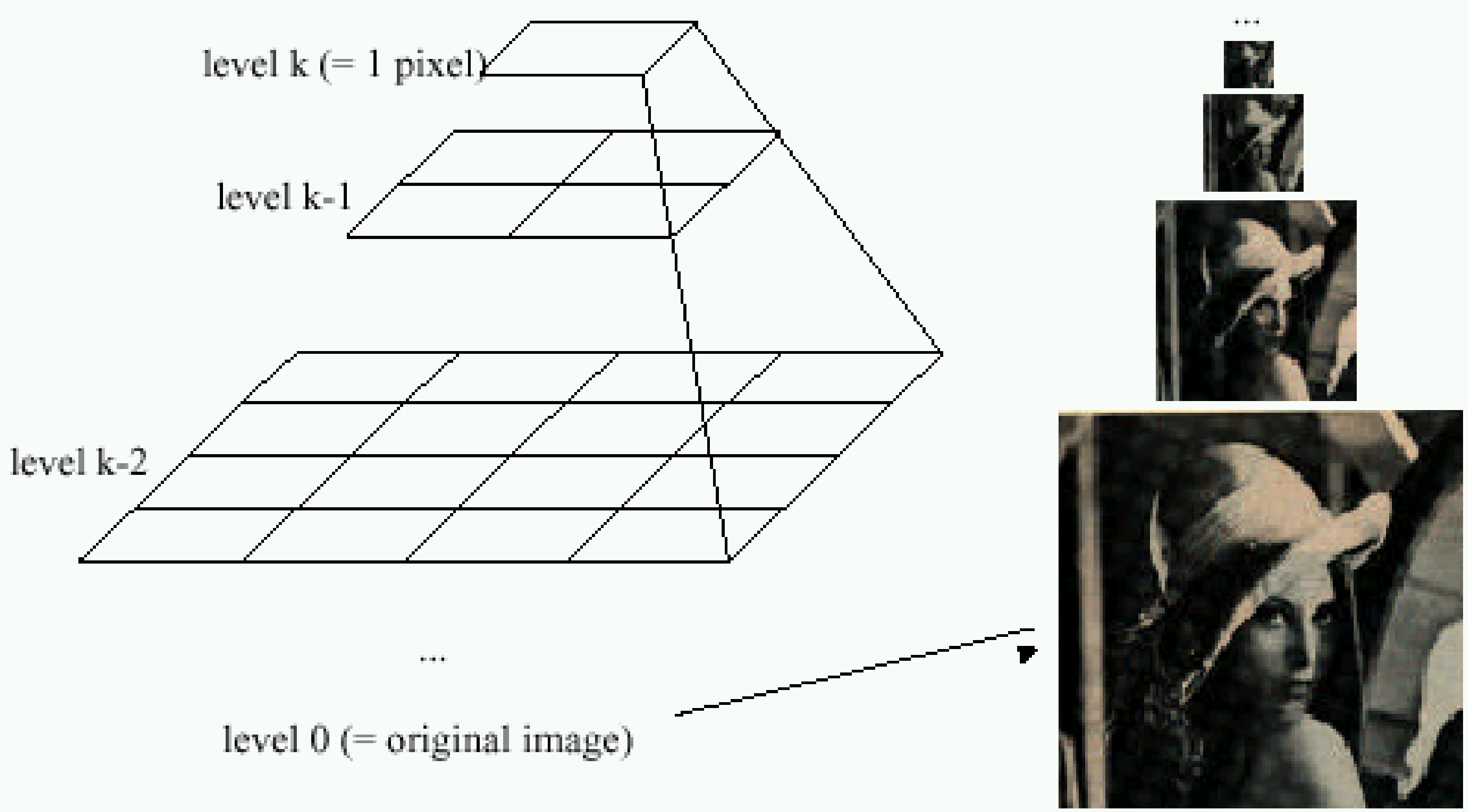
- Filter size should double for each ½ size reduction. Why?
- How can we speed this up?

Slide by Steve Seitz

Multi-resolution Representations

Image Pyramids

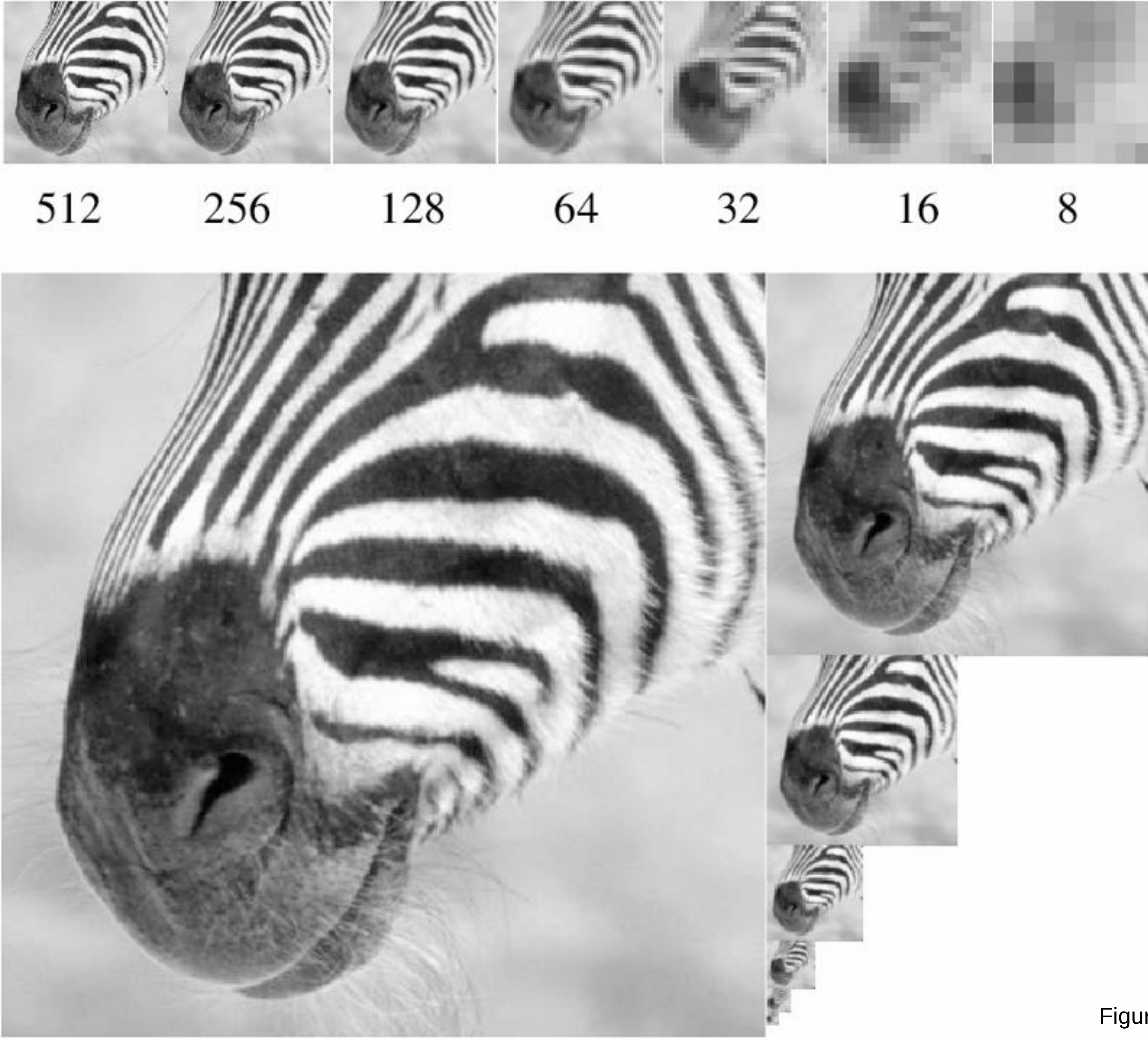
Idea: Represent NxN image as a "pyramid" of 1×1, 2×2, 4×4,…, $2^k\times2^k$ images (assuming $N=2^k$)



Known as a **Gaussian Pyramid** [Burt and Adelson, 1983]

- In computer graphics, a *mip map* [Williams, 1983]
- A precursor to *wavelet transform*

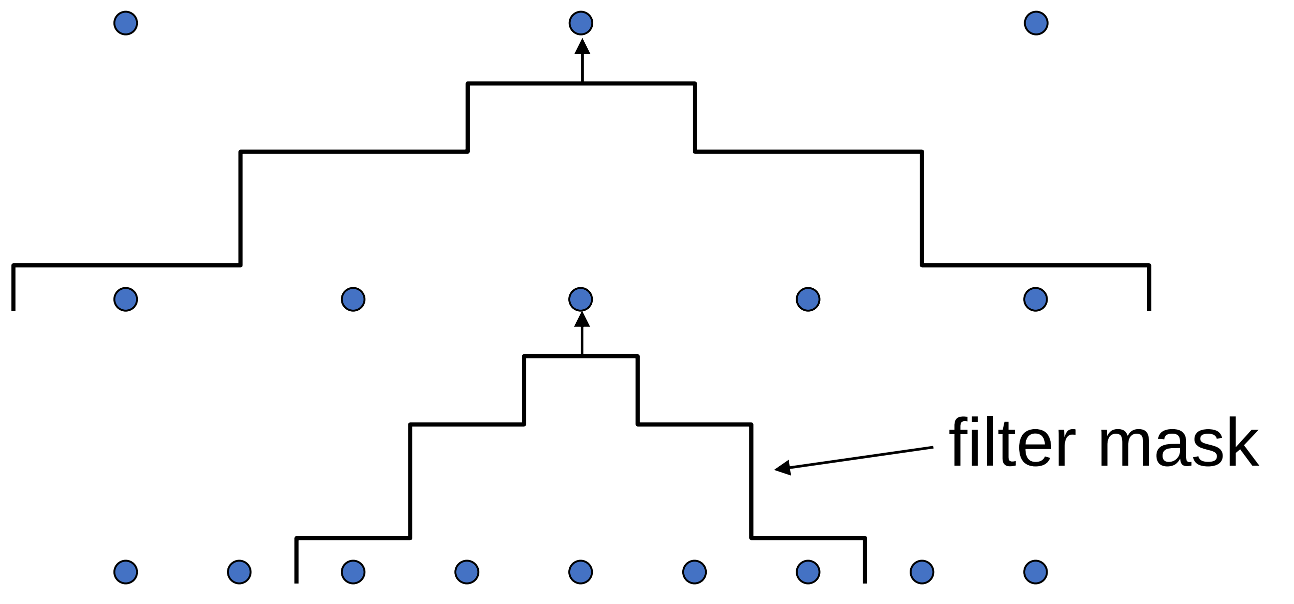
Slide by Steve Seitz



A bar in the big images is a hair on the zebra's nose; in smaller images, a stripe; in the smallest, the animal's nose

Figure from David Forsyth

Gaussian pyramid construction



Repeat

- Filter
- Subsample

Until minimum resolution reached

- can specify desired number of levels (e.g., 3-level pyramid)

The whole pyramid is only 4/3 the size of the original image!

Slide by Steve Seitz

What are they good for?

"Biological visual systems also operate on a hierarchy of scales" (Marr 1982)

Improve Search

- Search over translations
  - Classic coarse-to-fine strategy
- Search over scale
  - Template matching
  - E.g. find a face at different scales

Notes

- B1 (Marschner & Shirley) Chapter 3.2 and 9.1-9.4
- Additional Reading: Chapter 2.3-2.4 and 3, Digital Image Processing, Gonzalez & Woods
- Additional Reading: Point transformations: http://homepages.inf.ed.ac.uk/rbf/HIPR2/pntops.htm
- Additional Reading: Chapter 3.1-3.2 http://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf