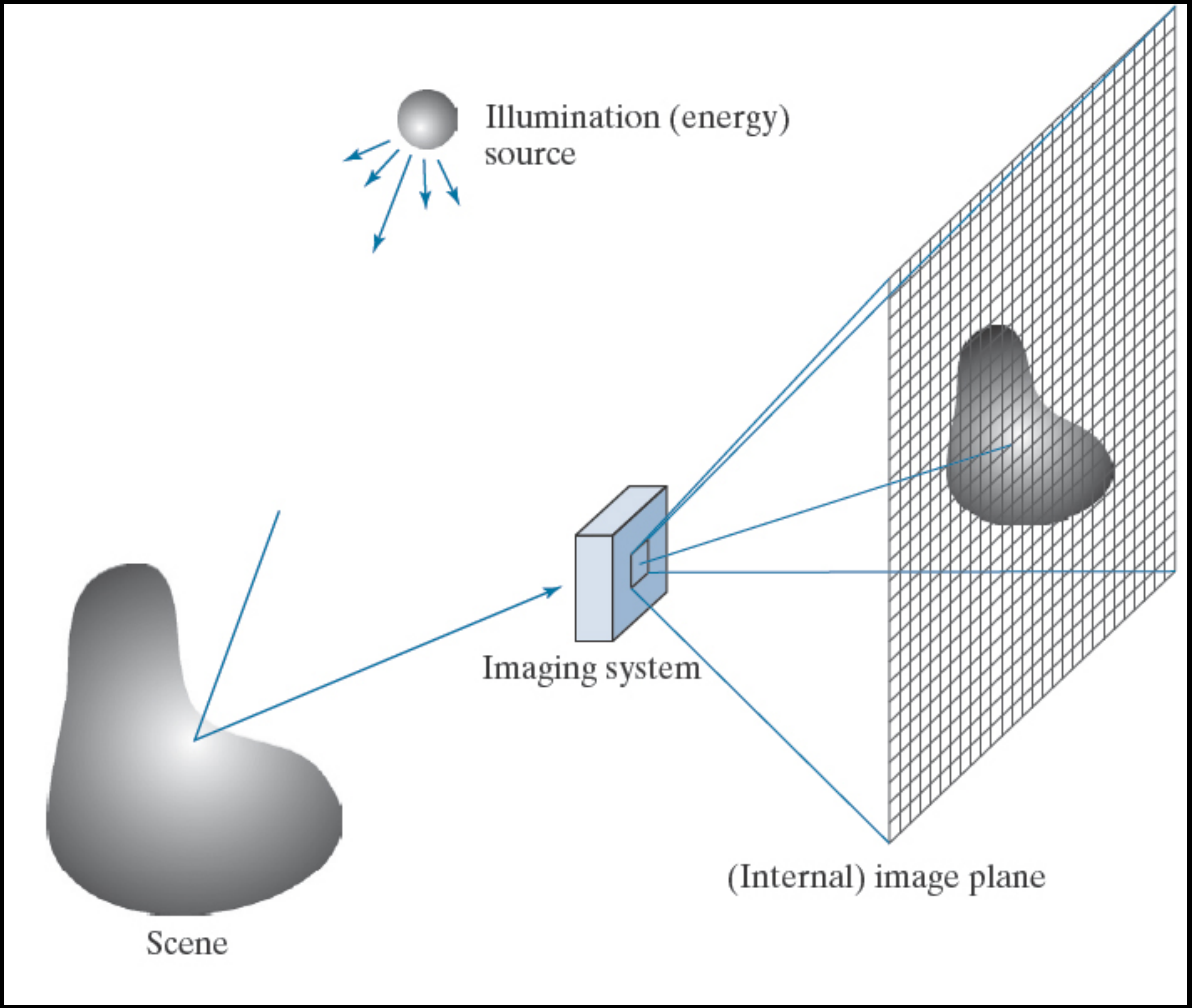
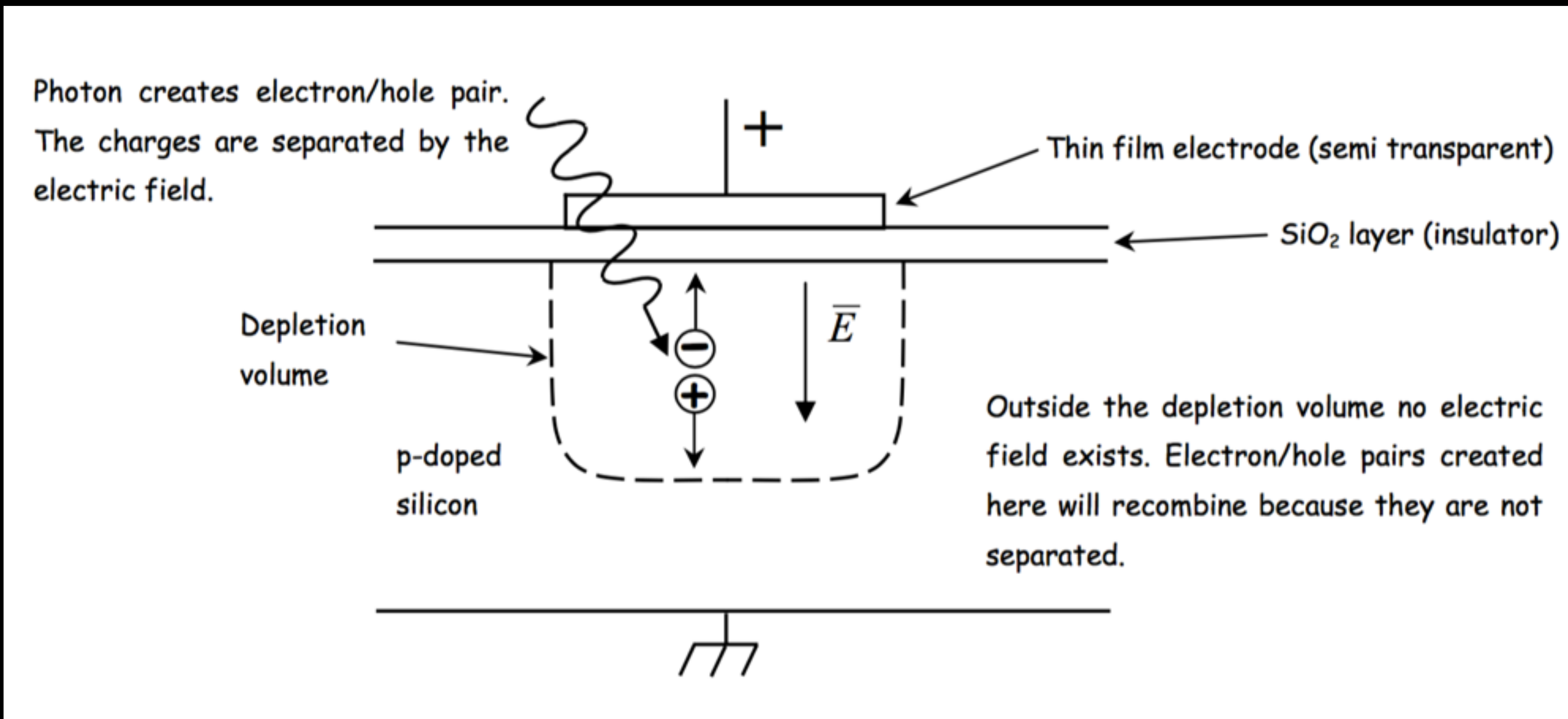
Digital imaging
and
image processing
Review

Final Exam Monday June 11th at 8am

Digital Data

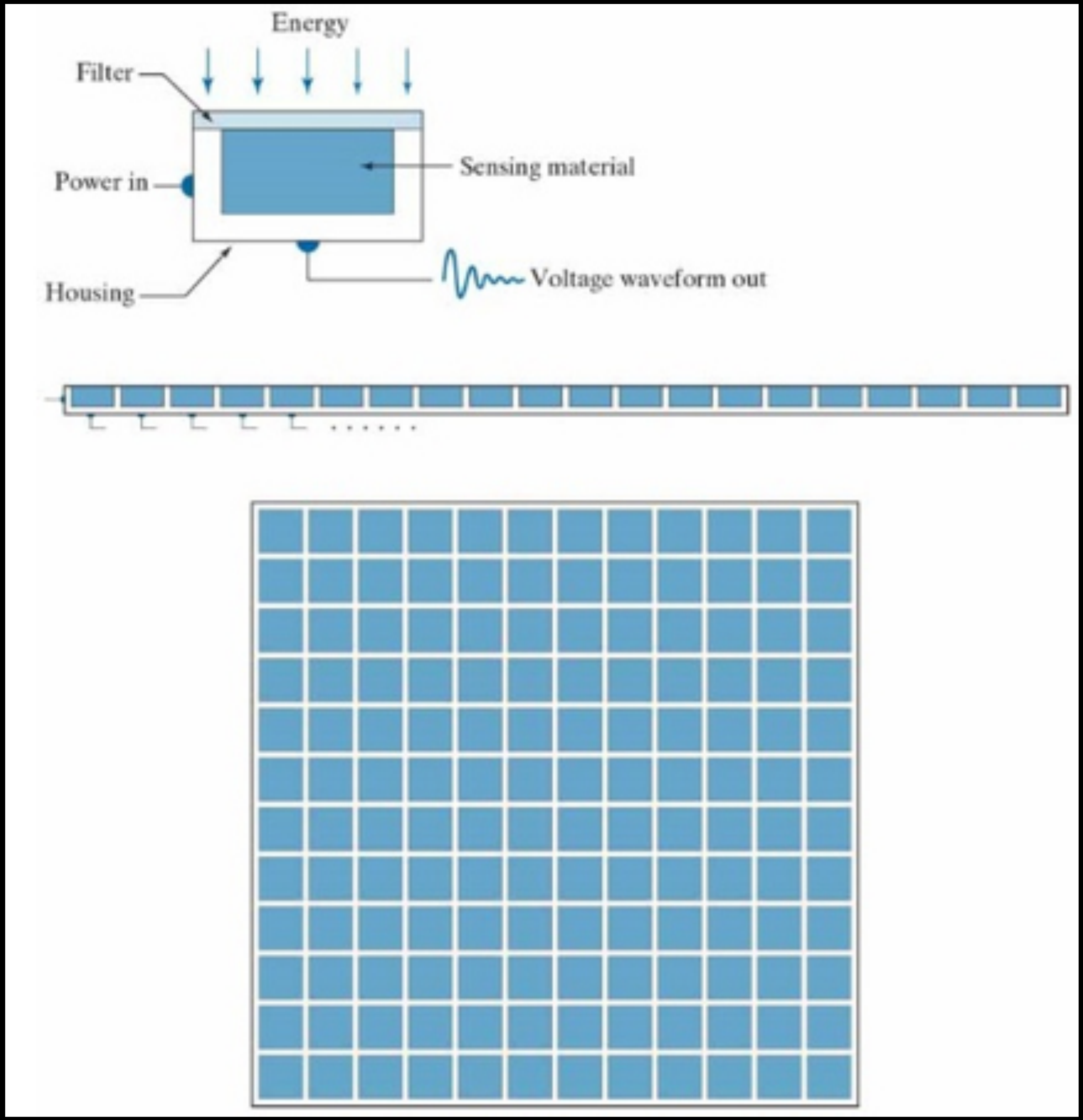


Pixels



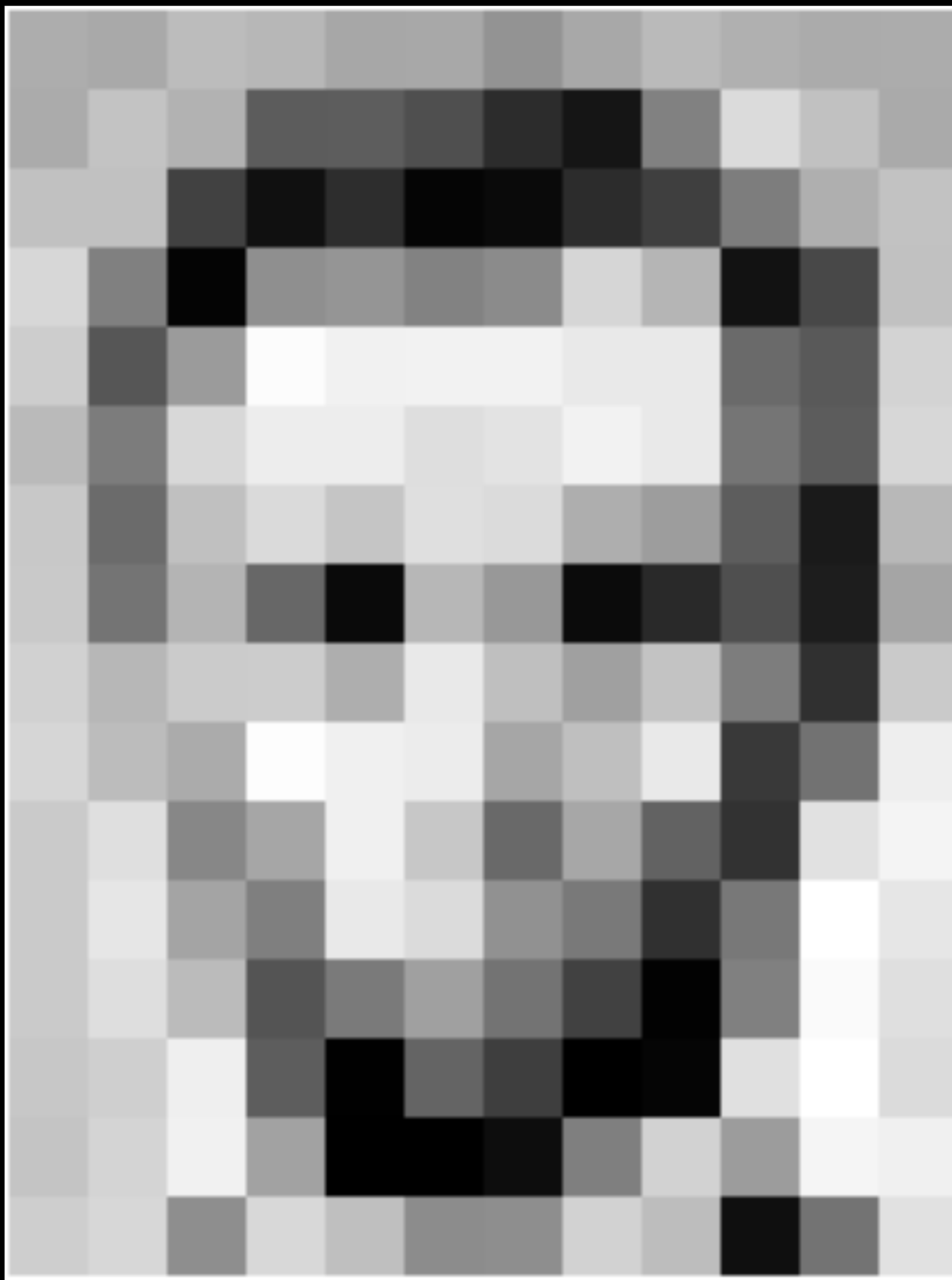
Photon creates electron/hole pair. The charges are separated by the electric field.
+
Thin film electrode (semi transparent)
SiO2 layer (insulator)
Depletion volume
$\overline{E}$
p-doped silicon
Outside the depletion volume no electric field exists. Electron/hole pairs created here will recombine because they are not separated.

Digital Sensors



(a) Single sensing element.
(b) Line sensor.
(c) Array sensor.

Digital Image



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	106	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Grayscale image

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Bit depth

- Binary: 0 and 1
- 8 bit: 0 up to (2^8 =) 256
- 16 bit: 0 up to (16^2 =) 65,536
- 32 bit: 0 up to (32^2 =) 4,294,967,296
- Color: RGB contains a red, green and blue matrix of the bit depth specified

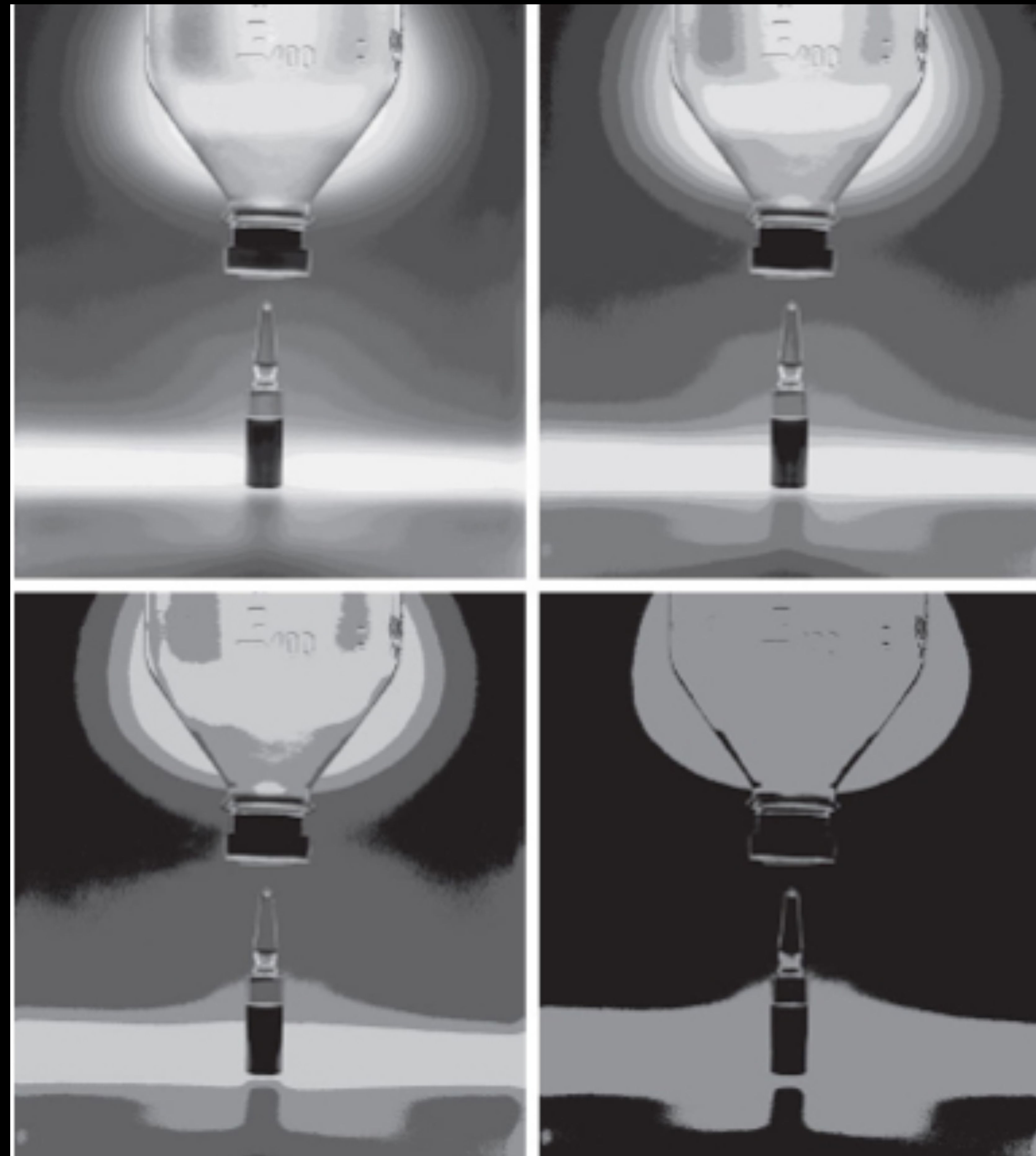
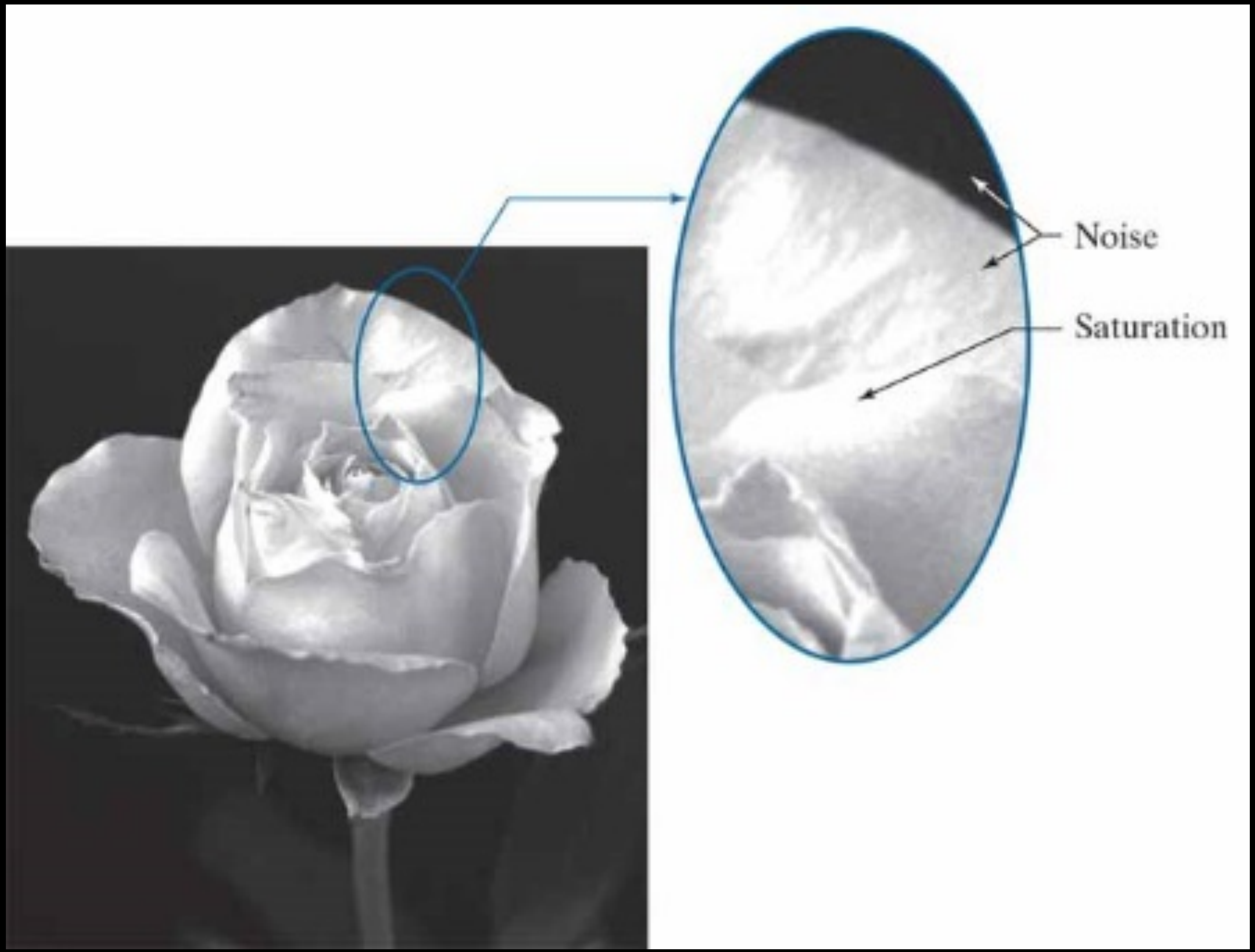
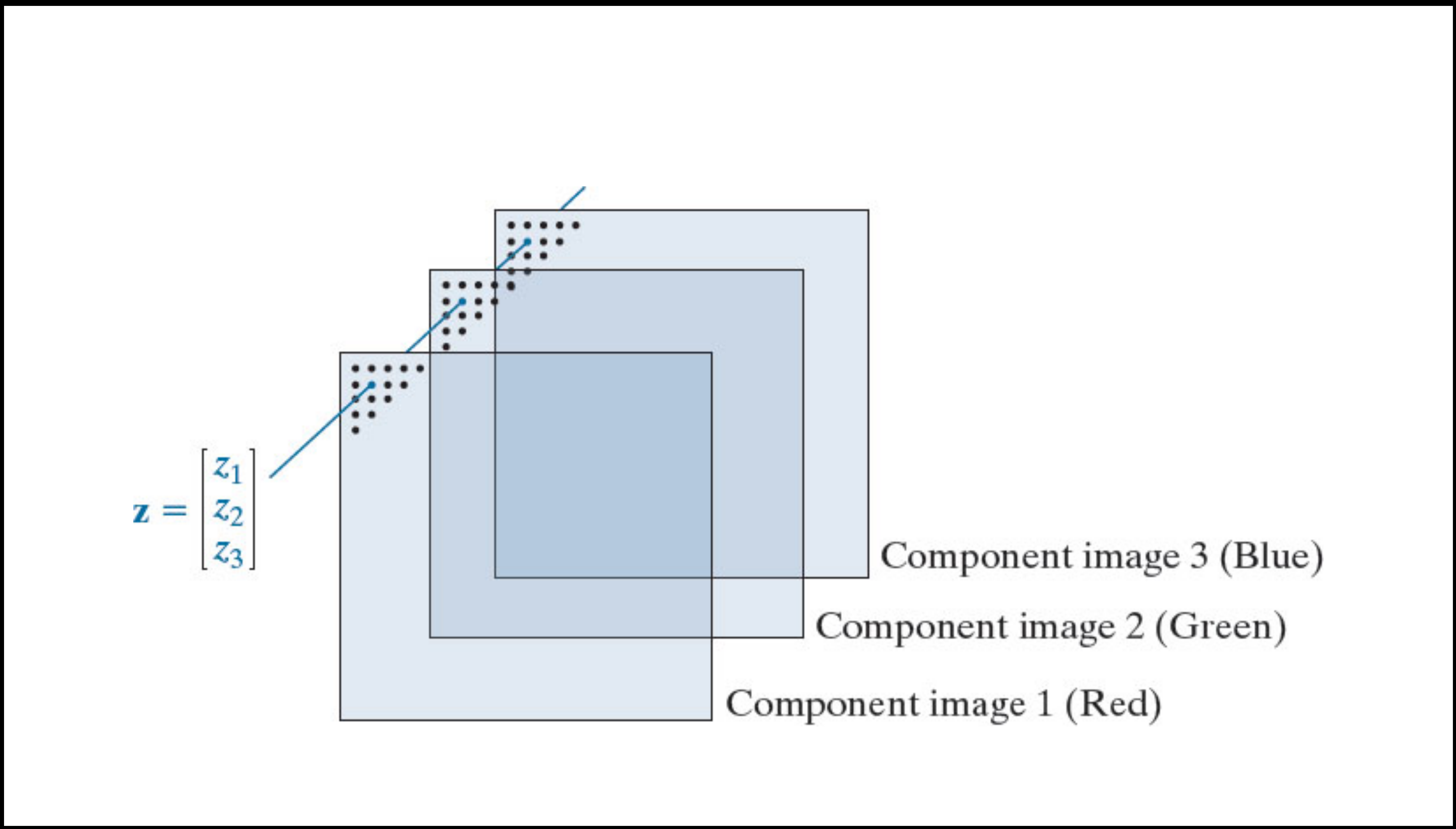


Image displayed in 32, 16, 8, 4, and 2 intensity levels.

Saturation

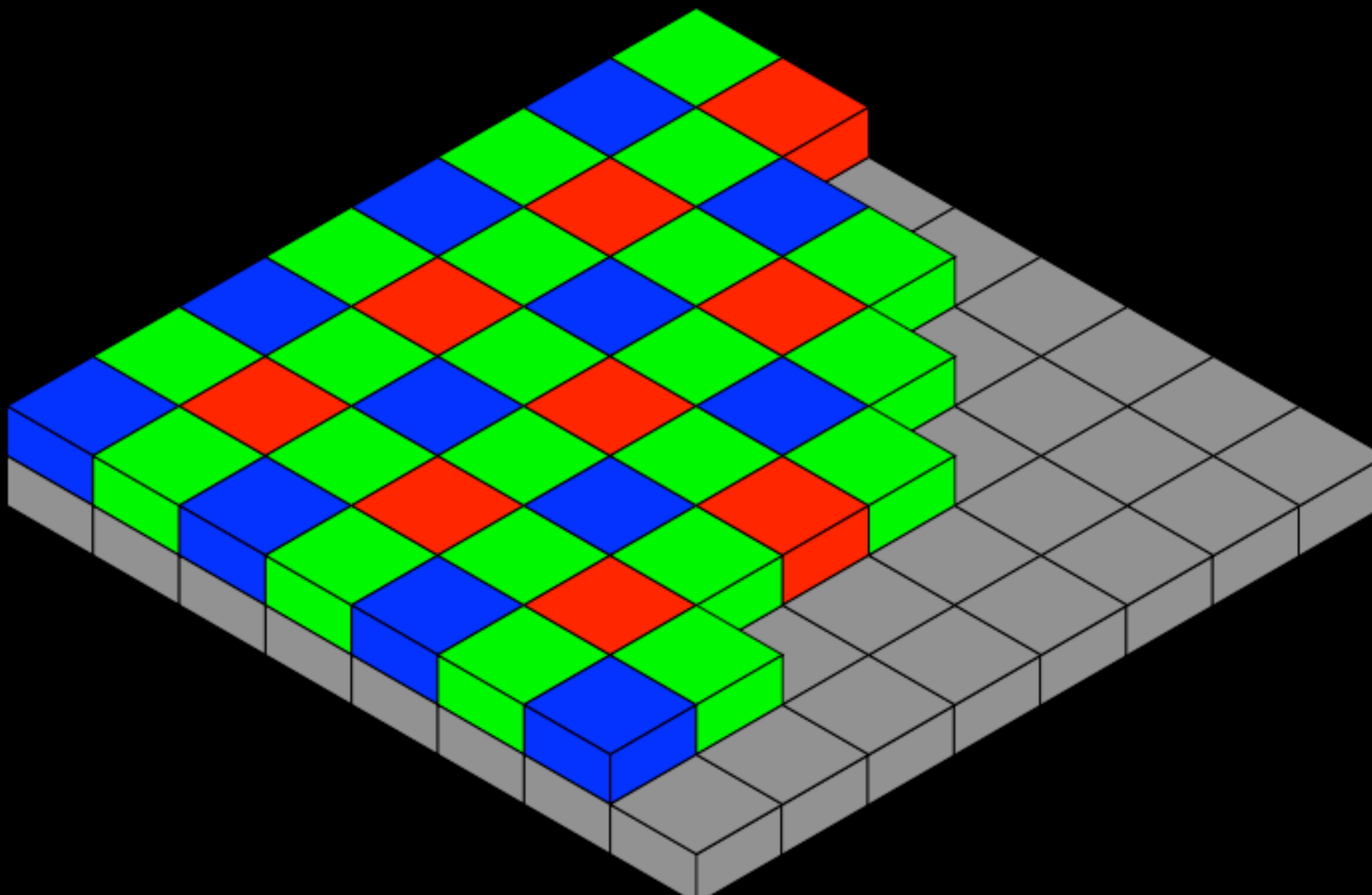


Forming a vector

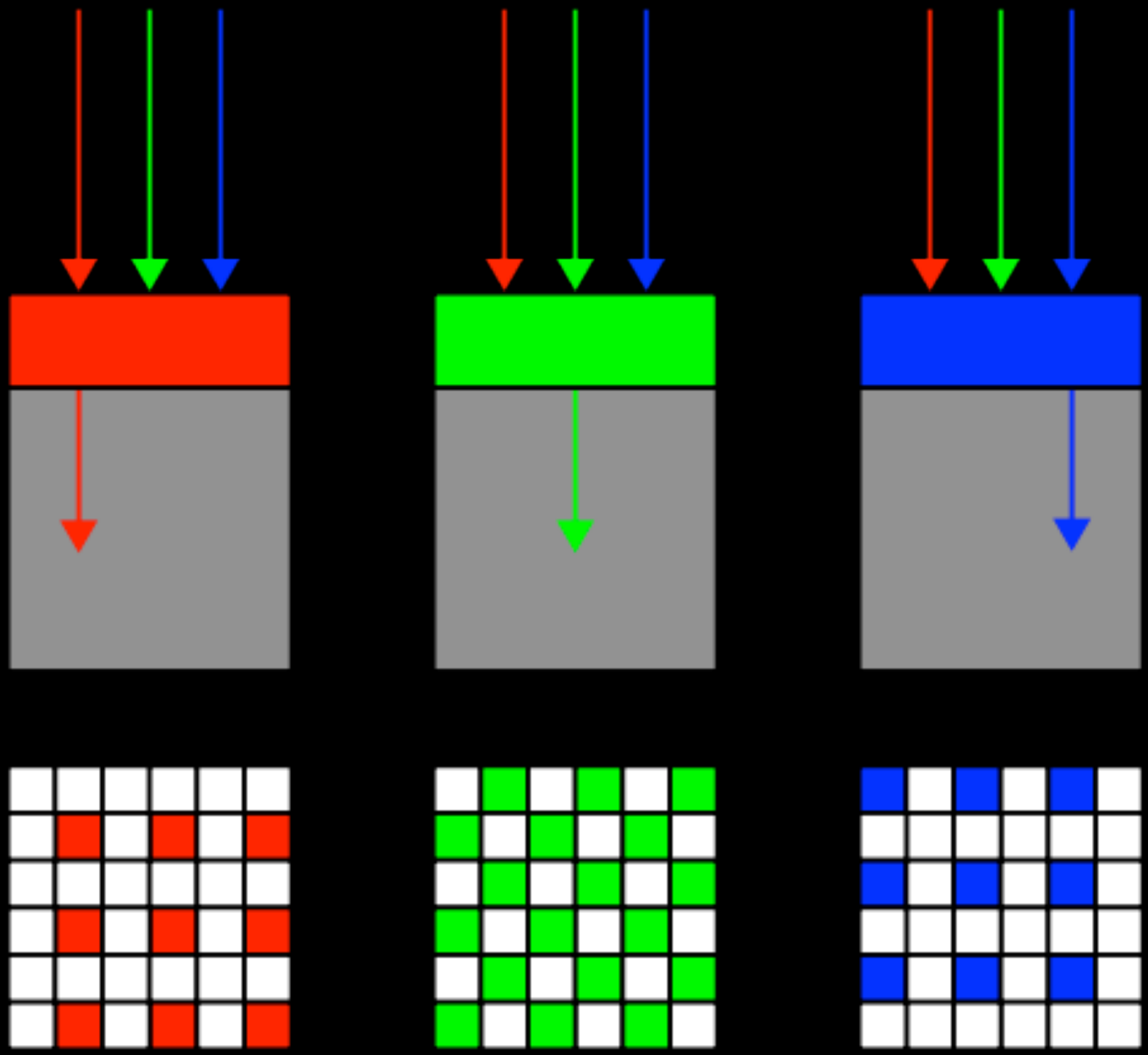


Color Images

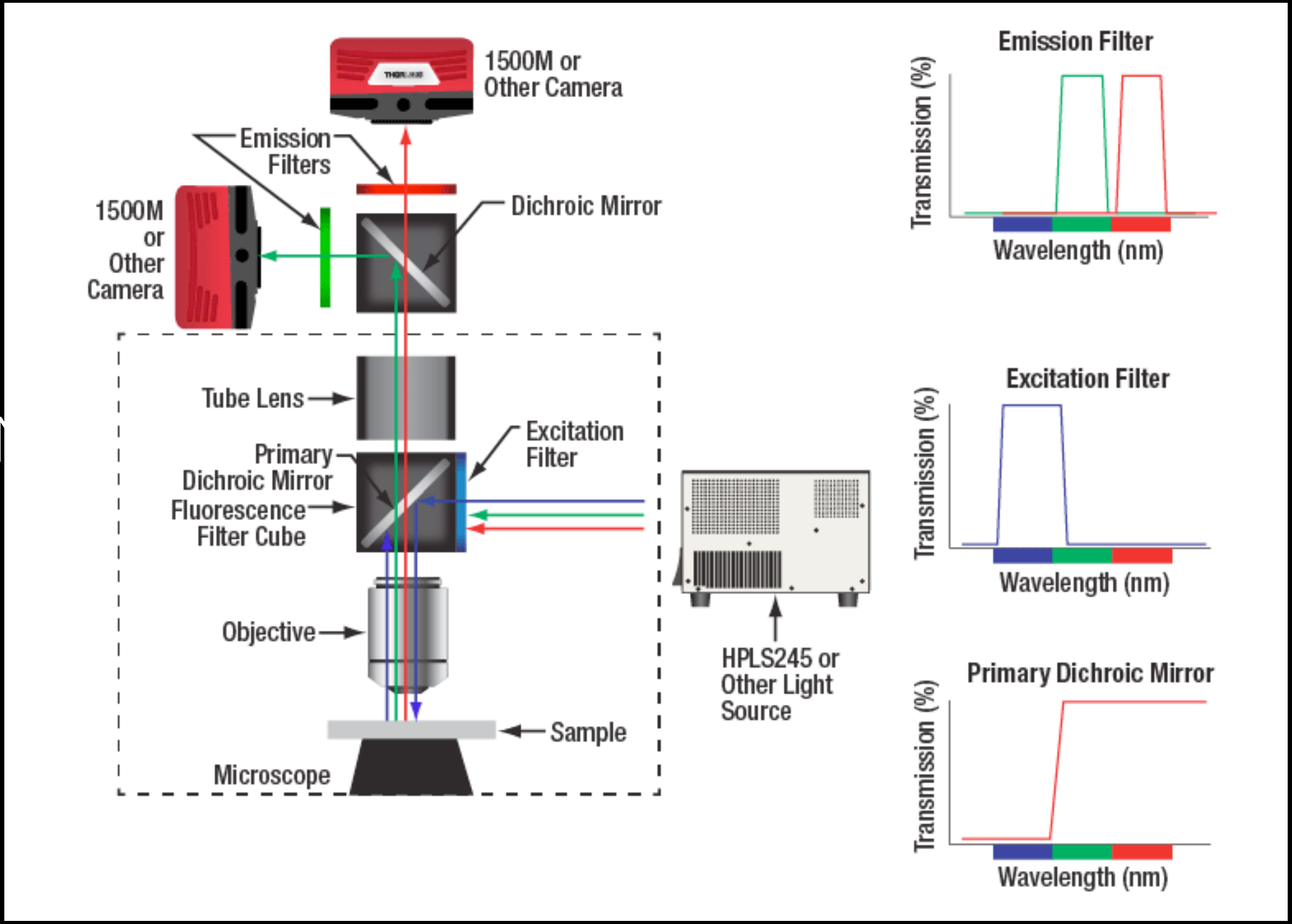
Color sensor: Bayer pattern



We lose light



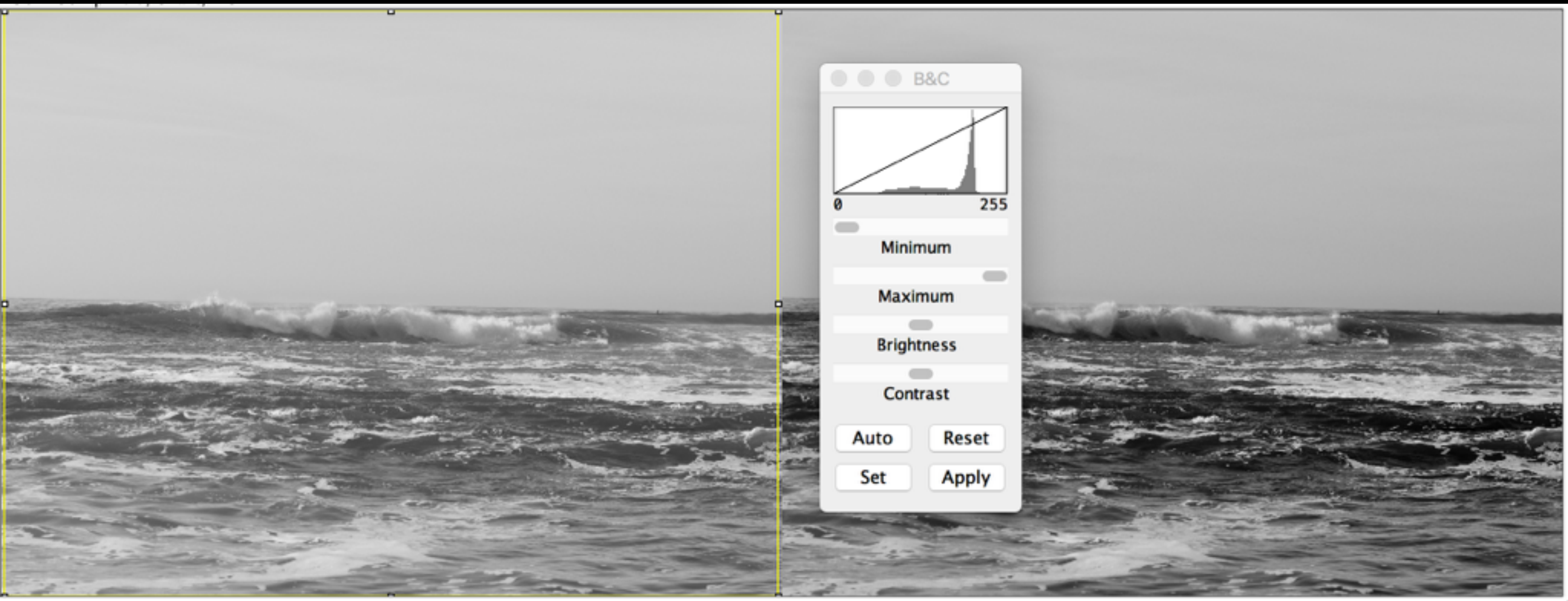
Dichroic Mirrors, Multiple Cameras



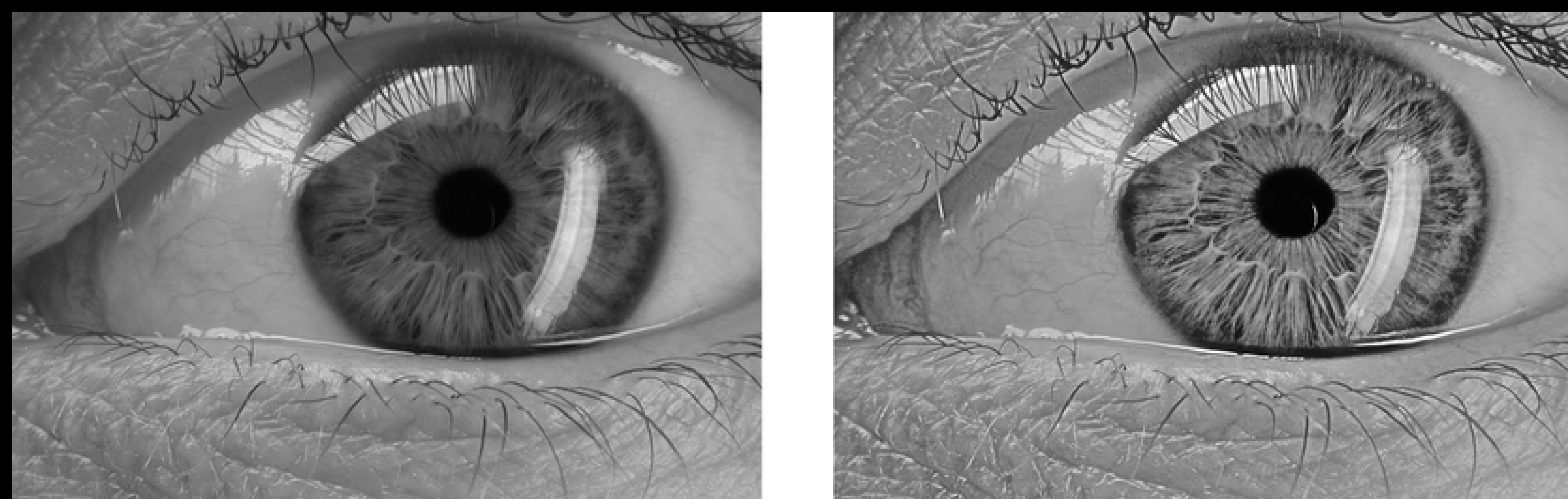
- N

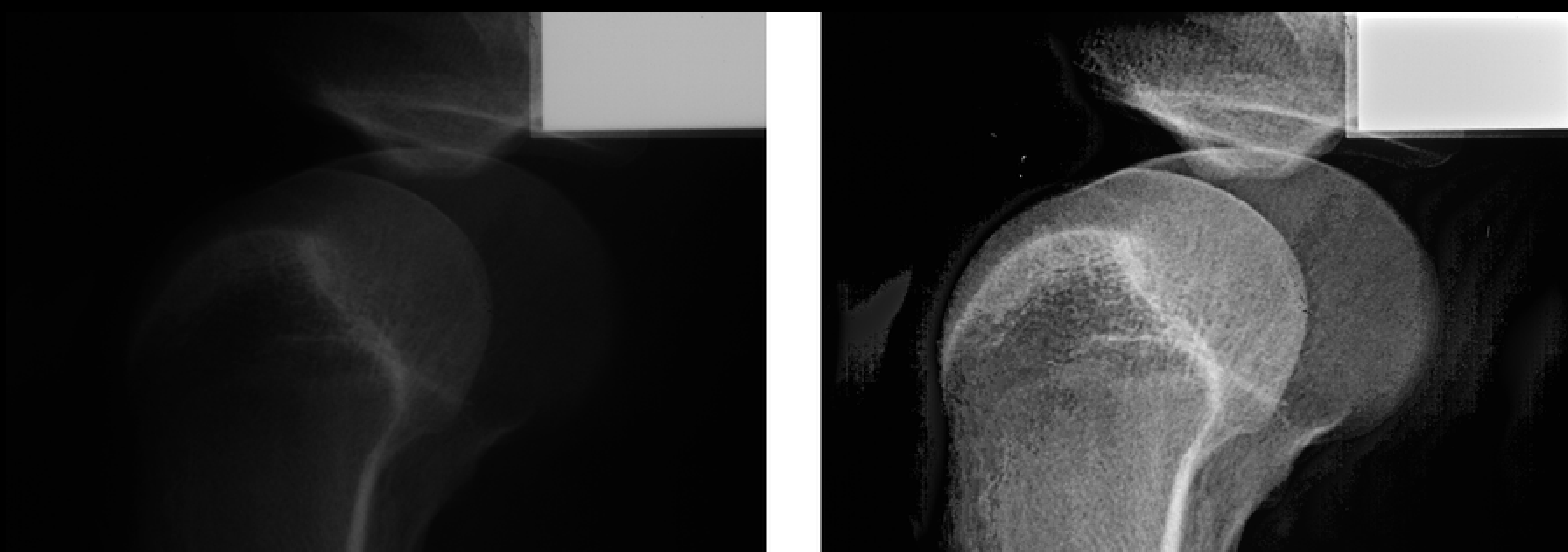
Image Display

Simplest contrast adjustment:
Set Min, Max of display



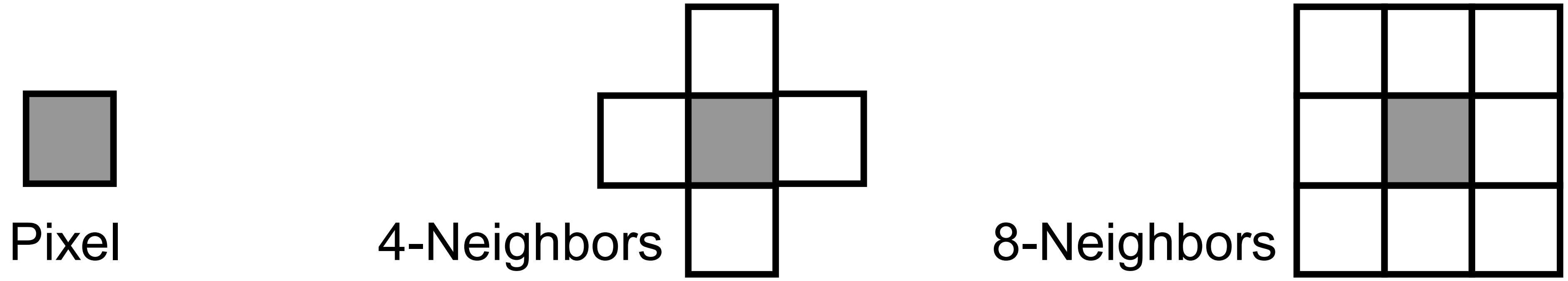
More powerful methods to improve contrast





In image space

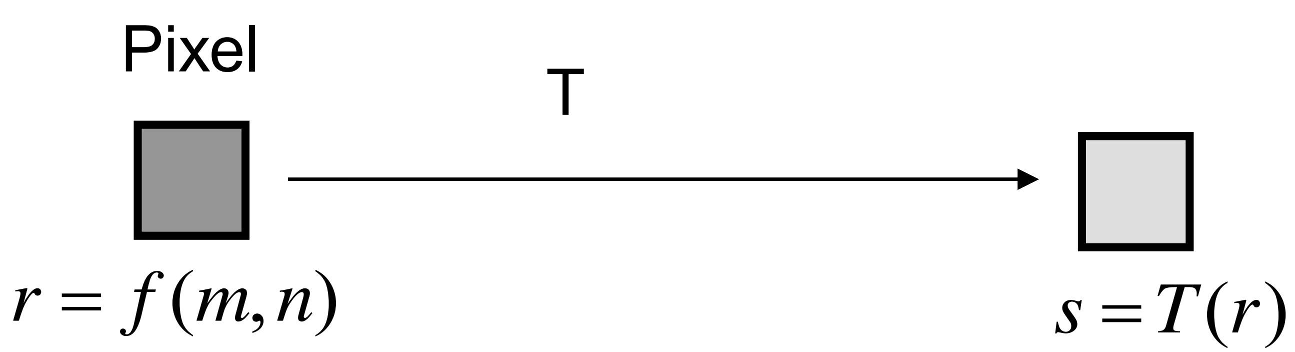
- We distinguish two domains:
  - Spatial or Pixel domain: $f(x,y)$ or $f(m,n)$
  - Frequency Domain: $F(w_x,w_y)$ or $F(u,v)$

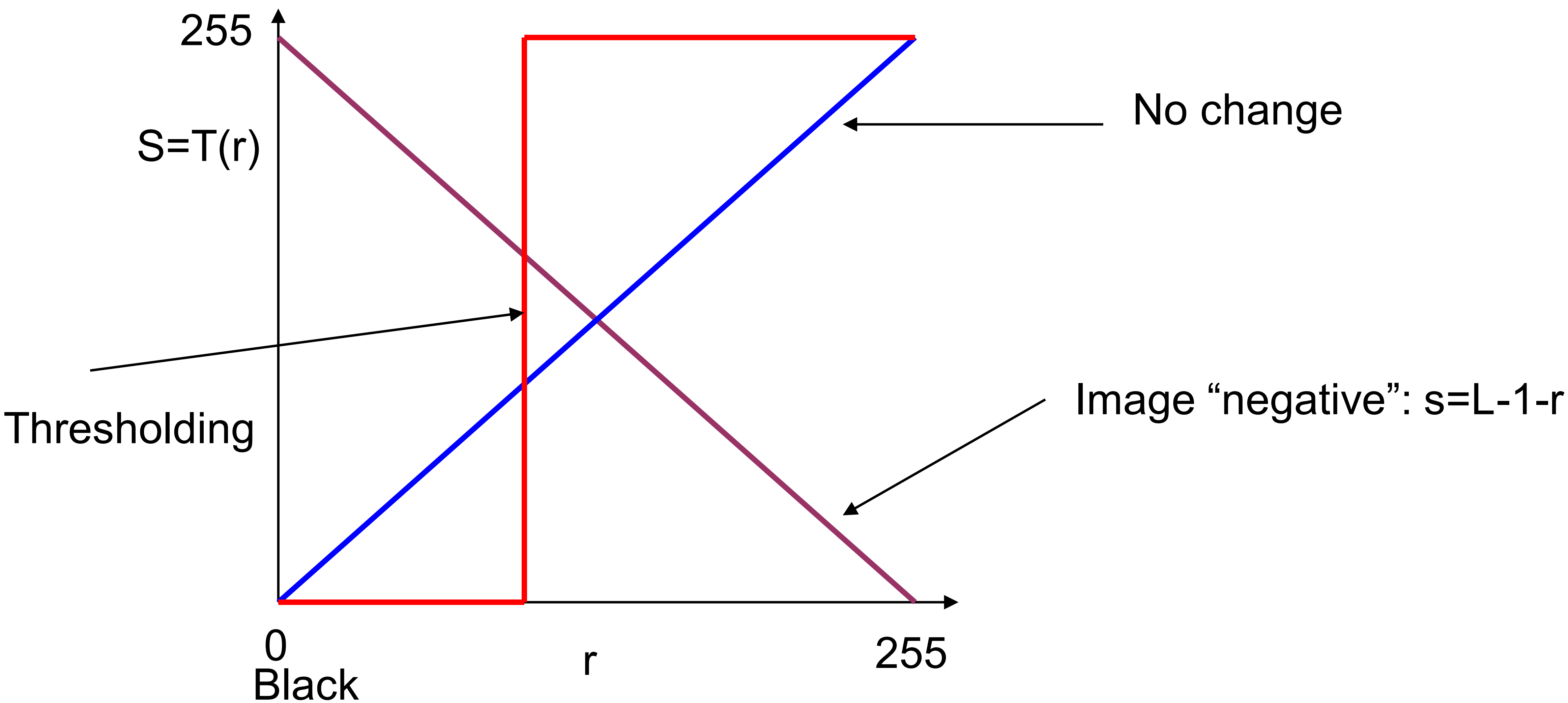


Operations

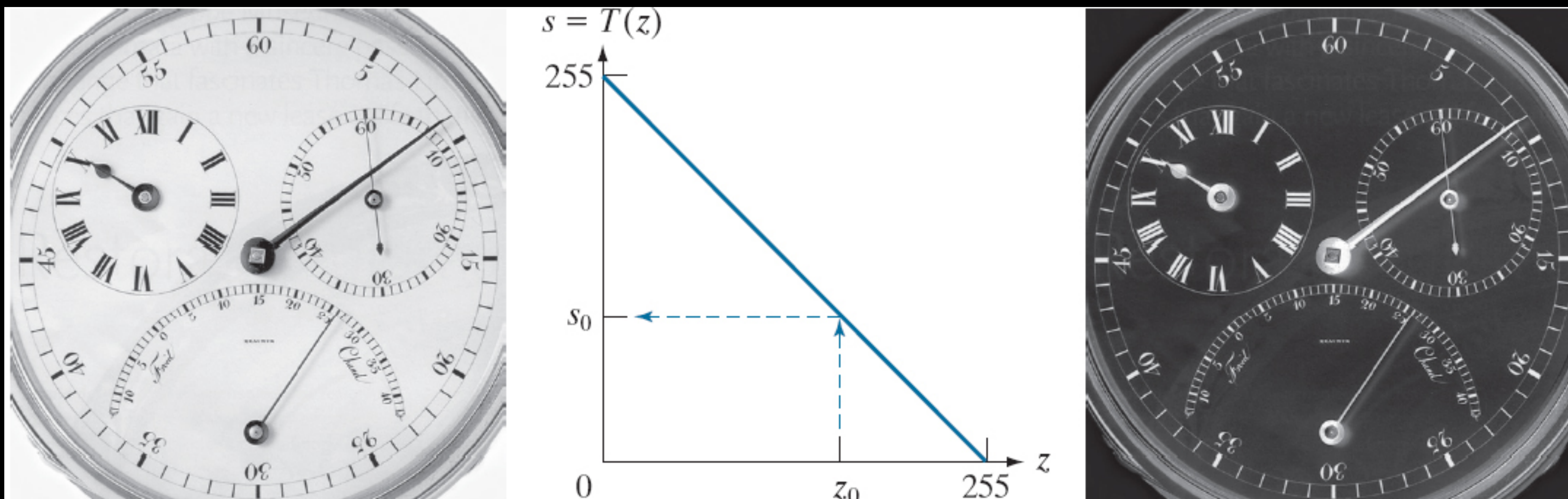
- Element-wise (pixel by pixel) *vs.* Matrix operations
- Single-pixel vs neighborhood:
  - Single-pixel: grab *e.g.* the value of the one nearest pixel)
  - Neighborhood (calculate and use e.g. the average / max / median / min or other calculated value of the nearest neighbors)

Simplest form of processing: Point Processing



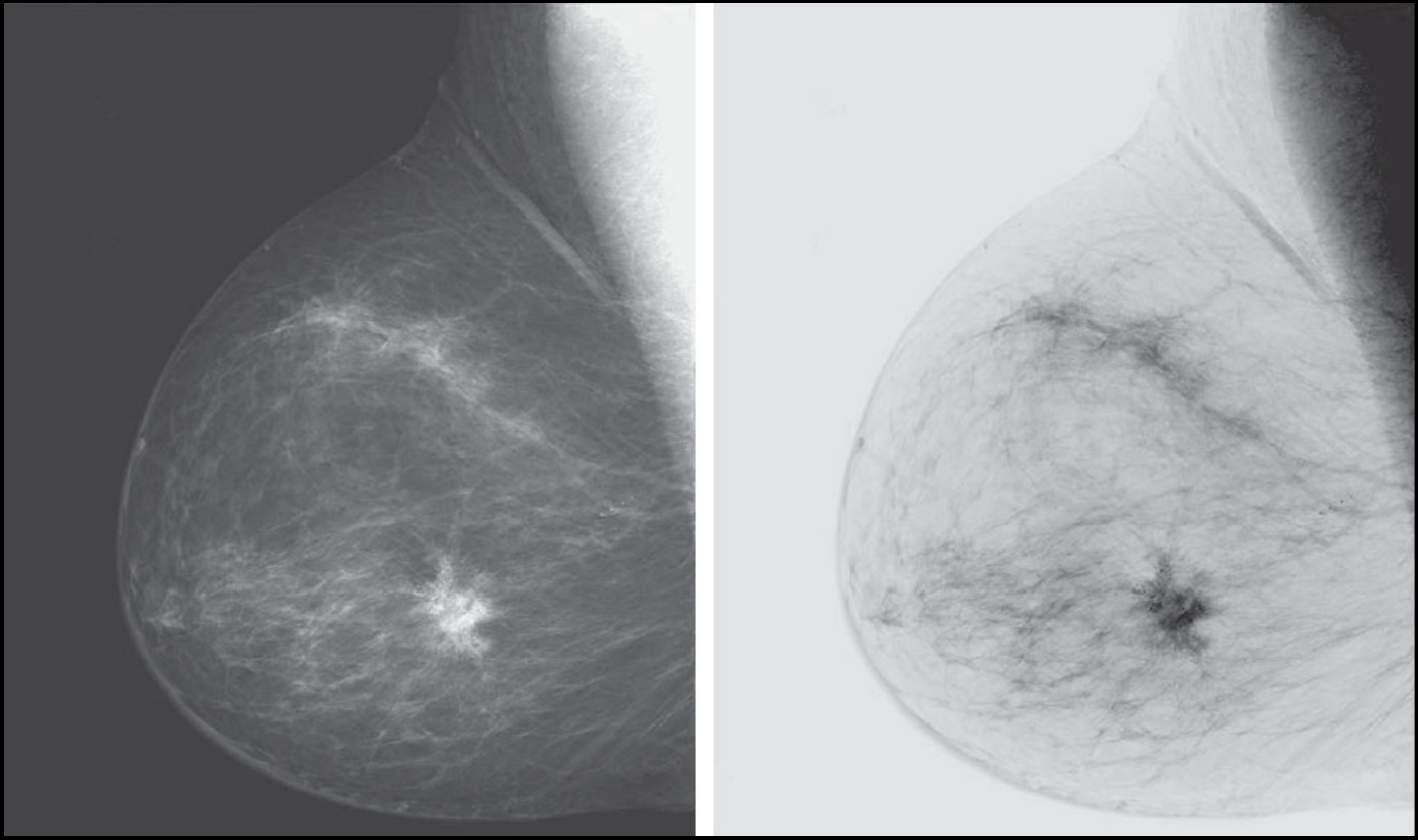


Inverse lookup table



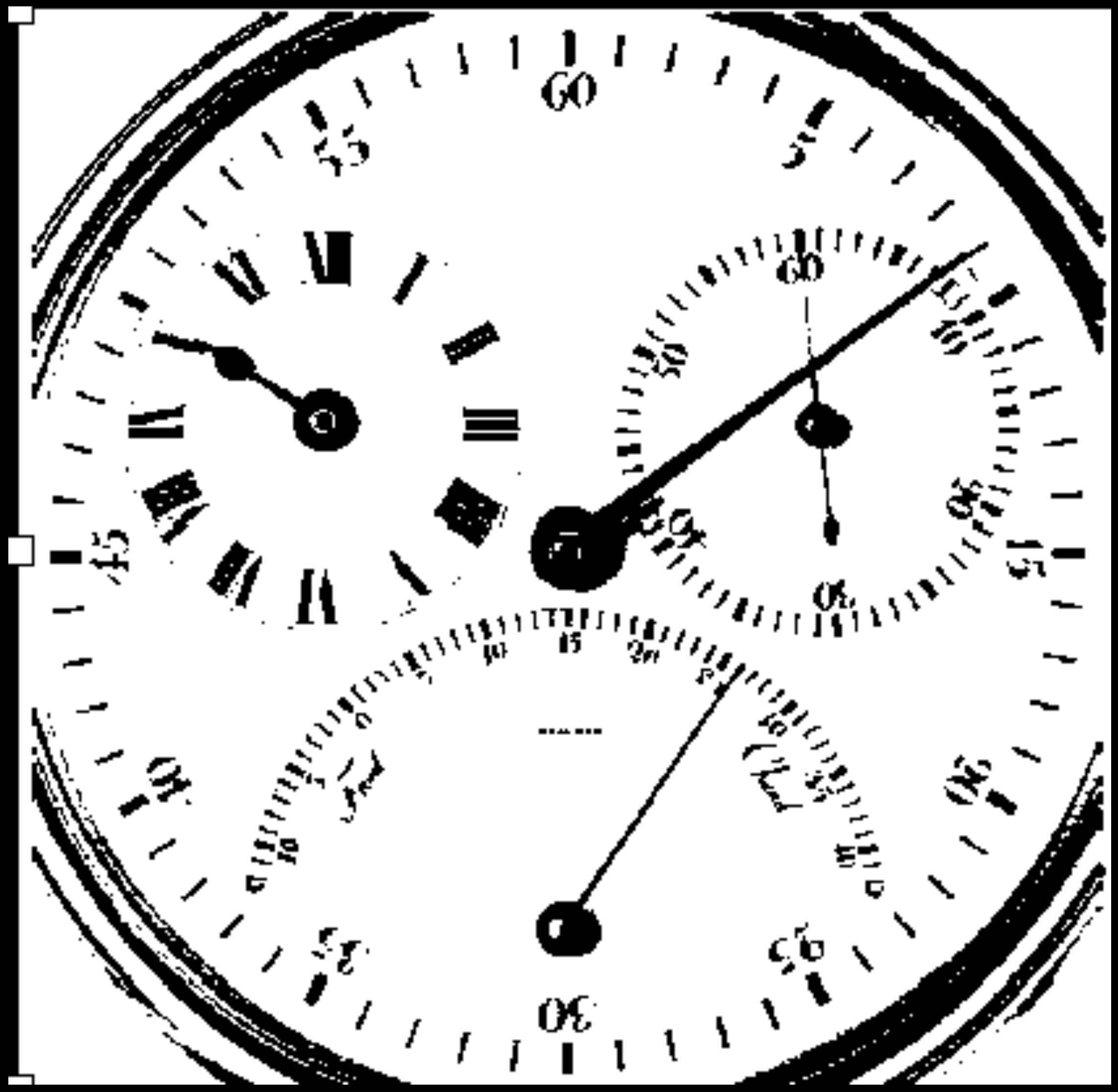
(a) 8-bit image. (b) Intensity transformation function used to obtain the digital equivalent of a “photographic” negative of an 8-bit image. The arrows show transformation of an arbitrary input intensity value z into its corresponding output value s0. (c) Negative of (a) obtained using (b)

Better visibility for display / diagnosis

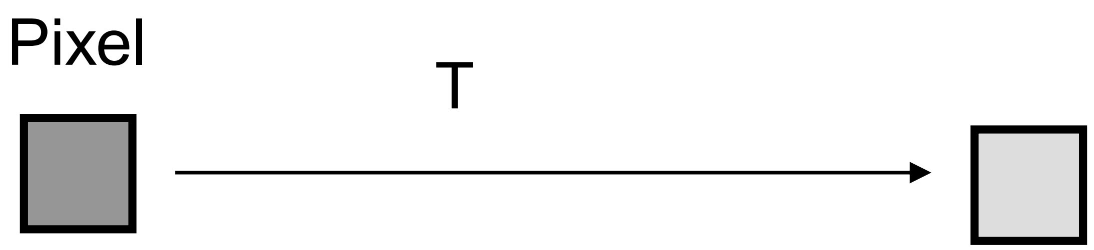


Binary

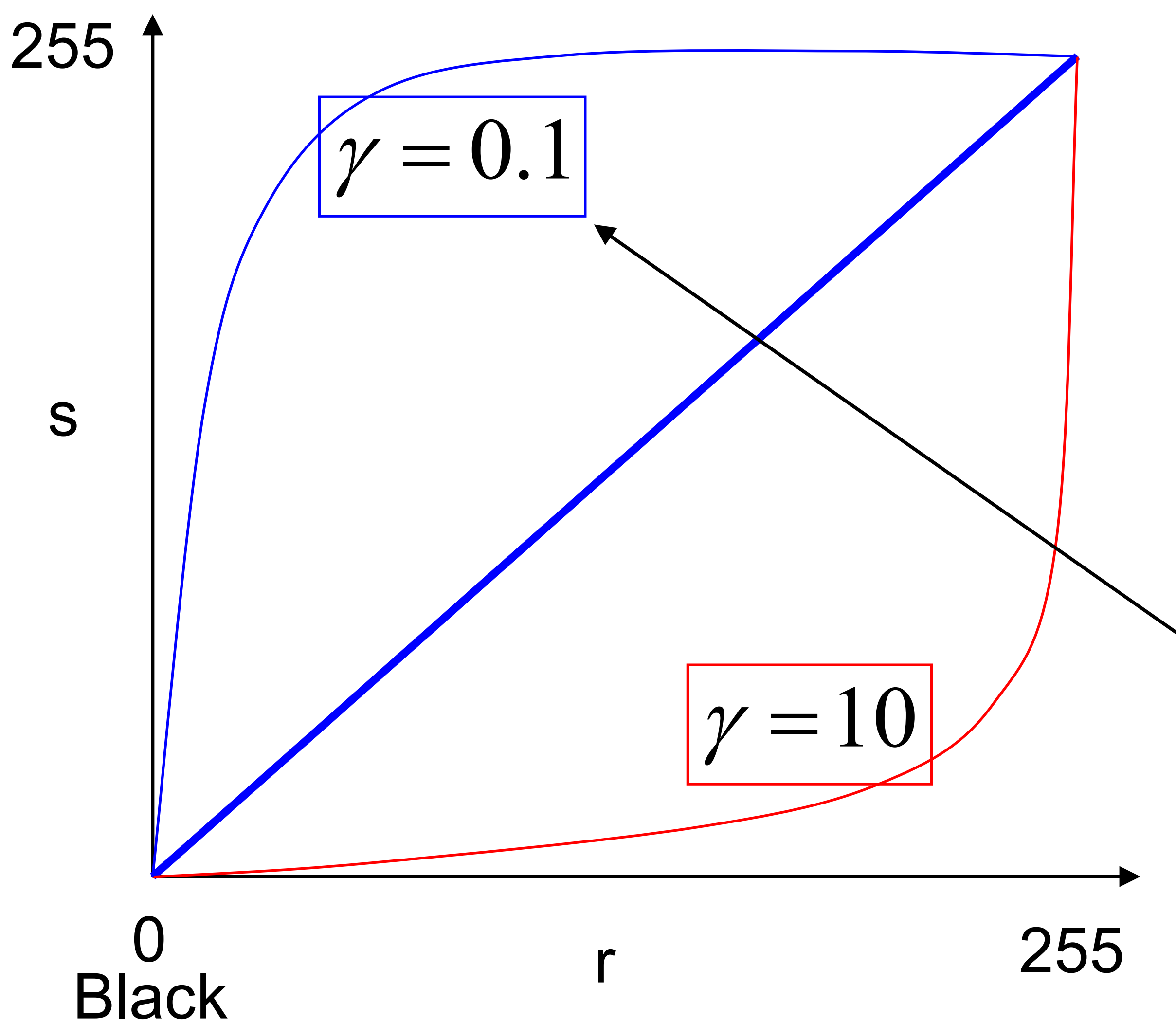
- Small storage
- Easier to apply some operations



Simplest form of processing: Point Processing



$r = f(m,n)$ $\qquad$ $s = T(r)$



Common Examples:

• Dynamic Range Compression

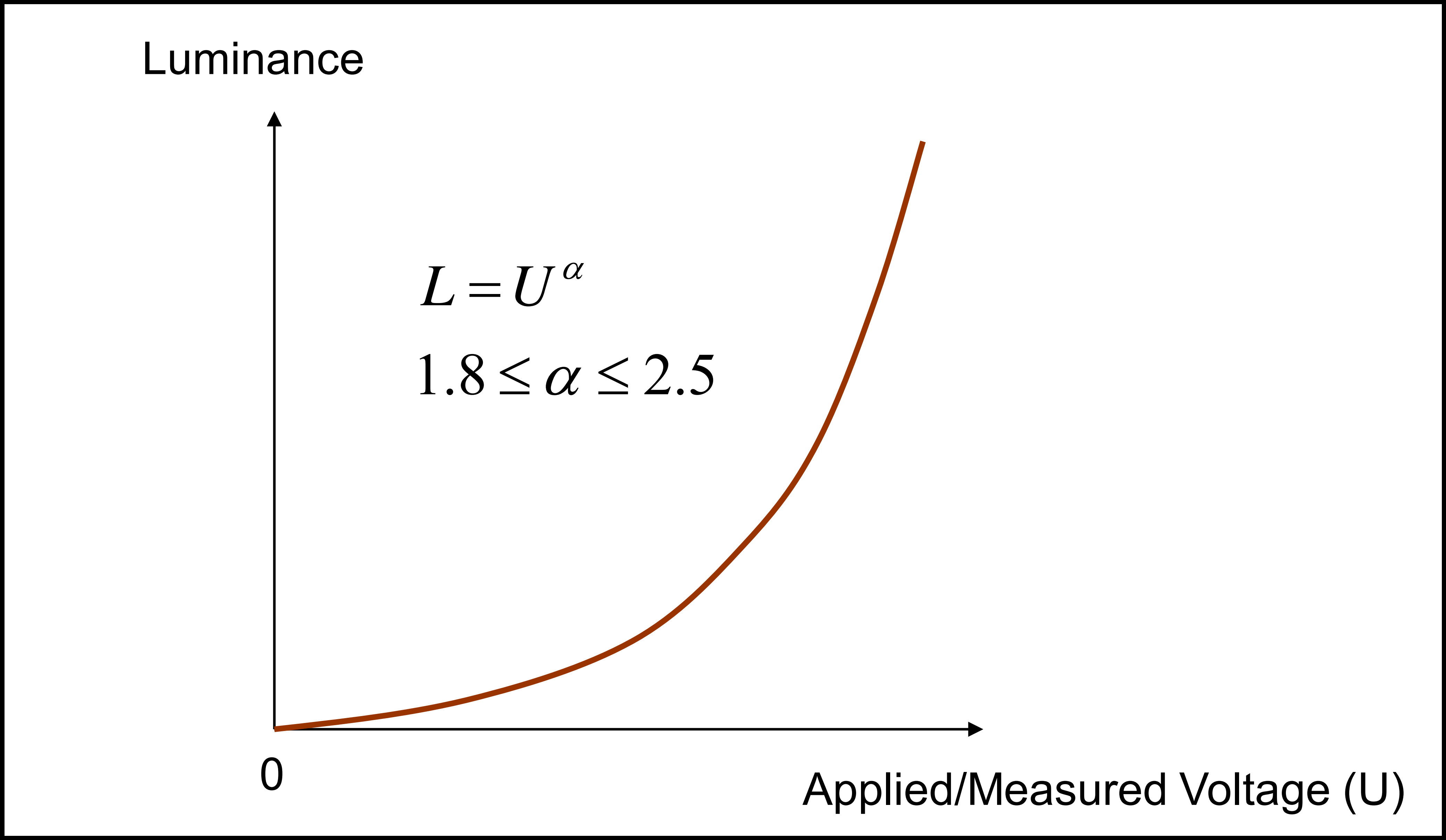
$$T(r) = c\log(1+r)$$

• Gamma Correction

$$T(r) = cr^{\gamma}$$

Narrow range of “dark” gets mapped to broad range of “gray”

Gamma Correction



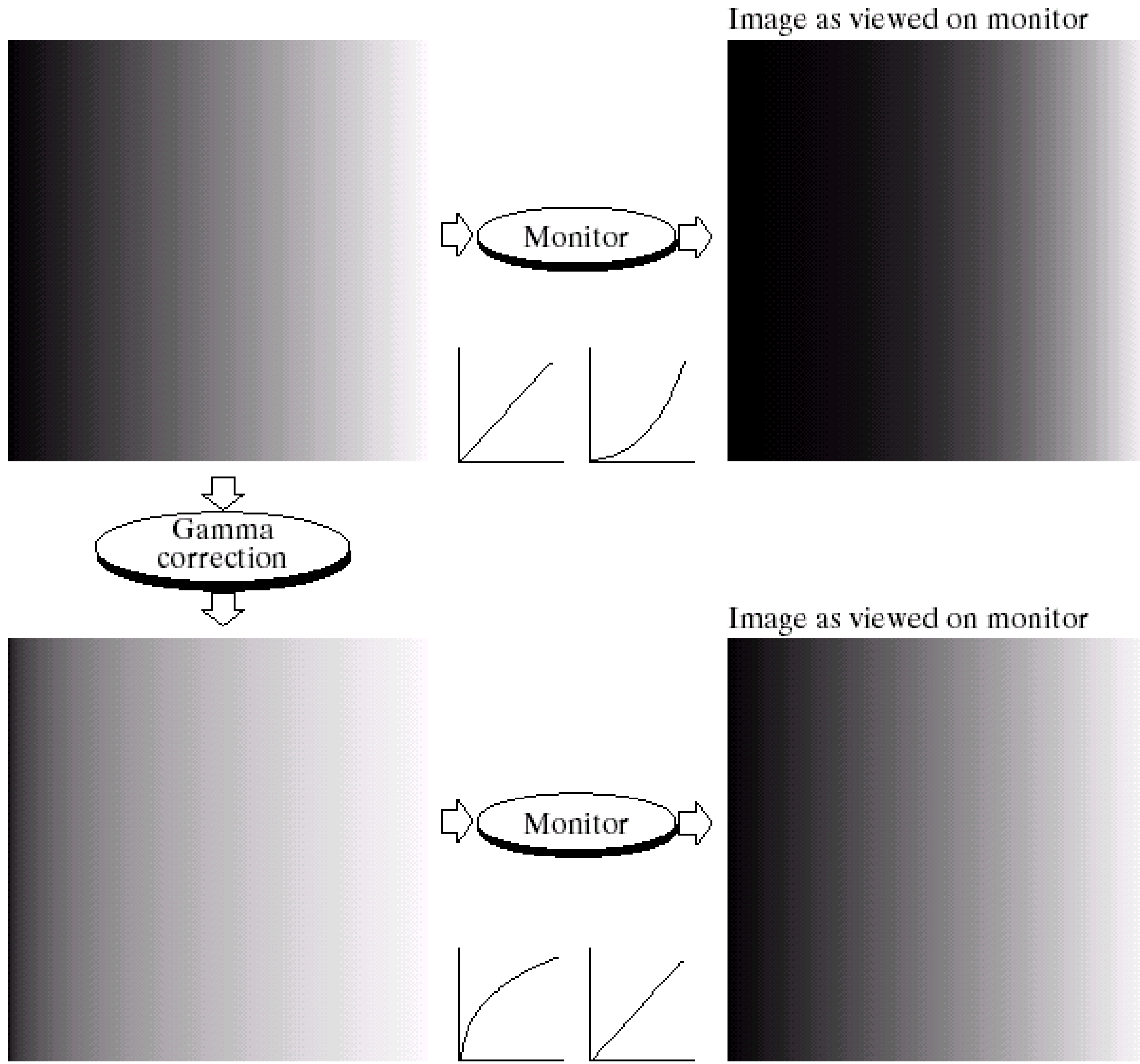
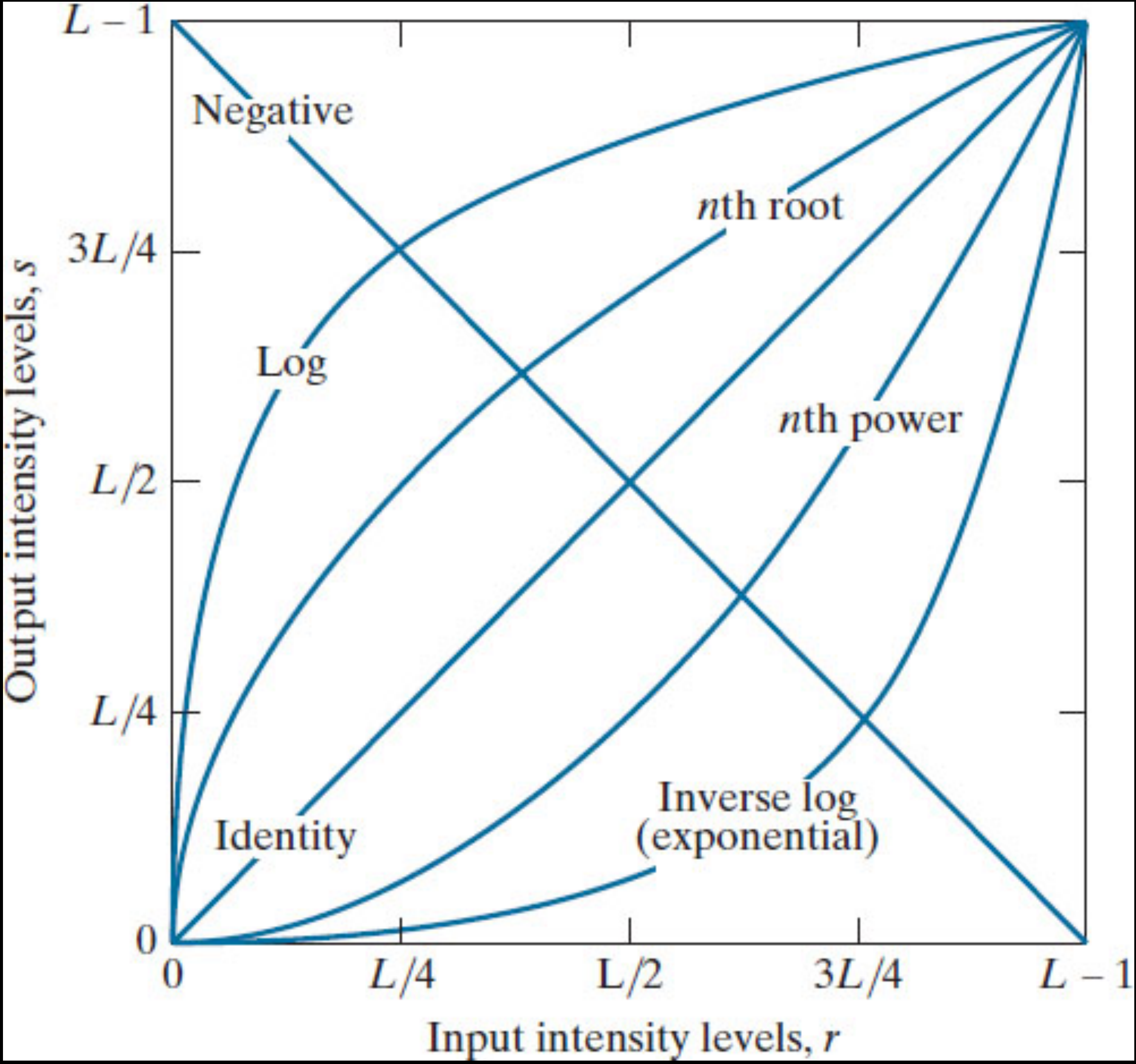


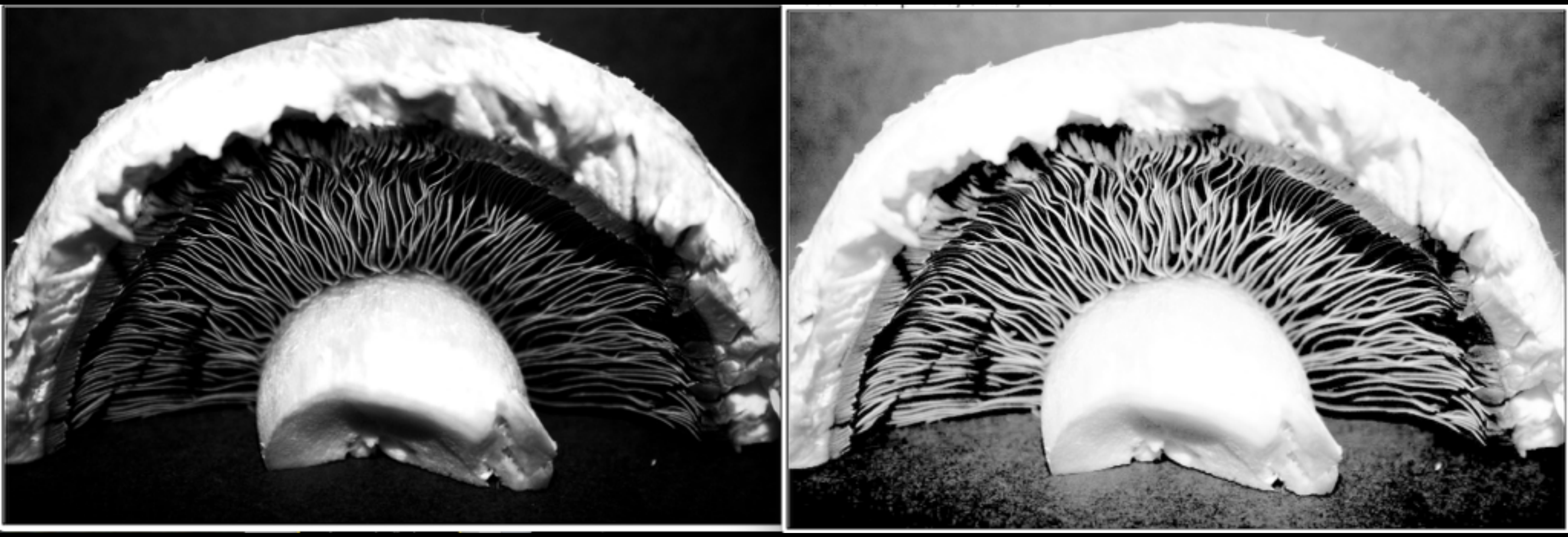
Image as viewed on monitor
Monitor
Gamma
correction
Image as viewed on monitor
Monitor

Scaling

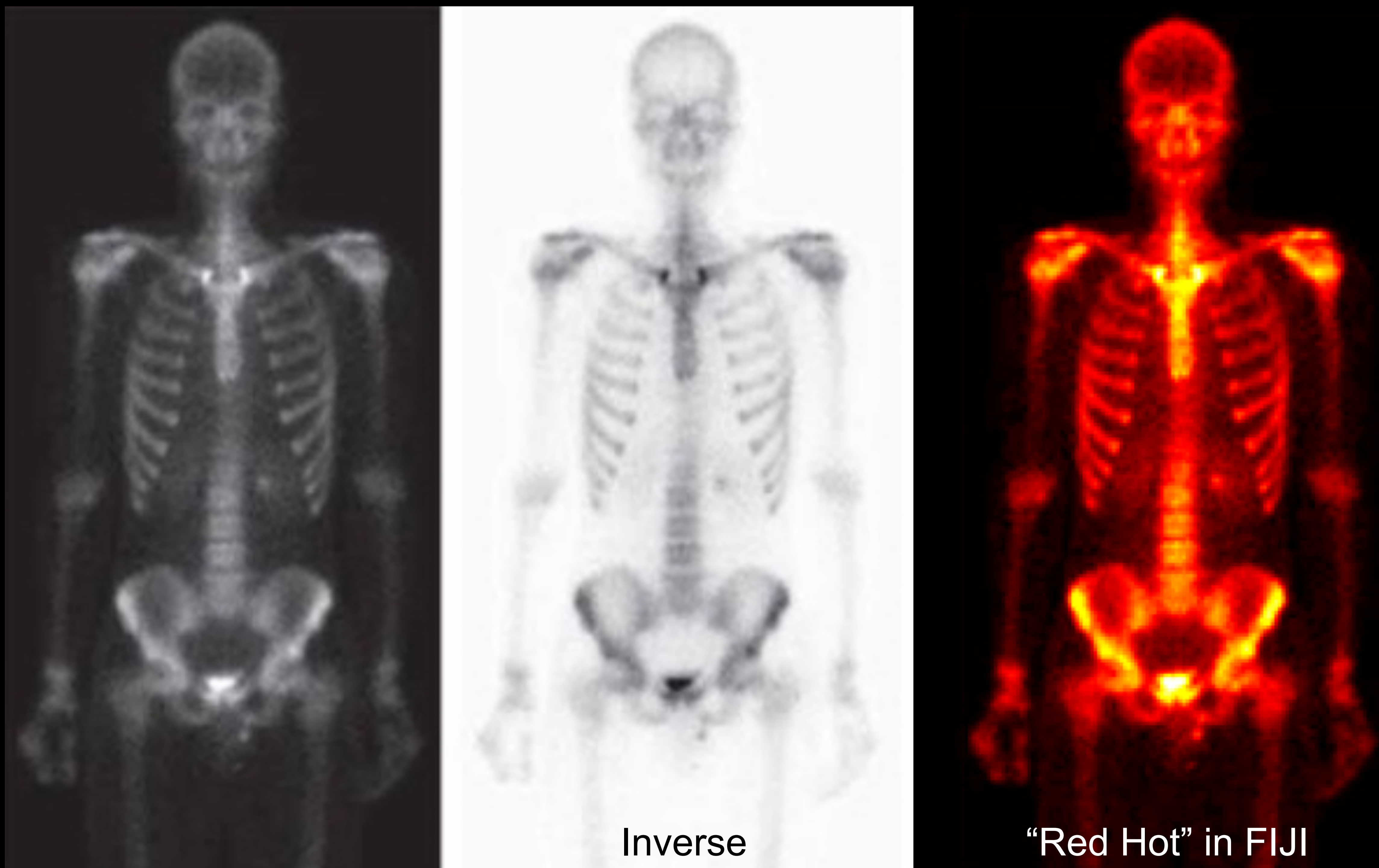


L − 1
3L/4
L/2
L/4
0
Output intensity levels, s
Negative
Log
nth root
Identity
nth power
Inverse log
(exponential)
0
L/4
L/2
3L/4
L − 1
Input intensity levels, r

Log scale display



Synthetic lookup tables

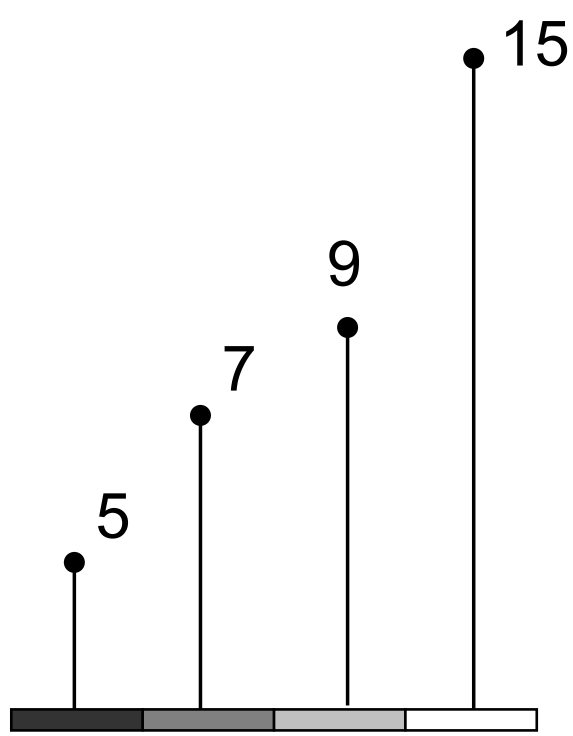


Chasing the right one can make it easier to see stuff — and to get published…

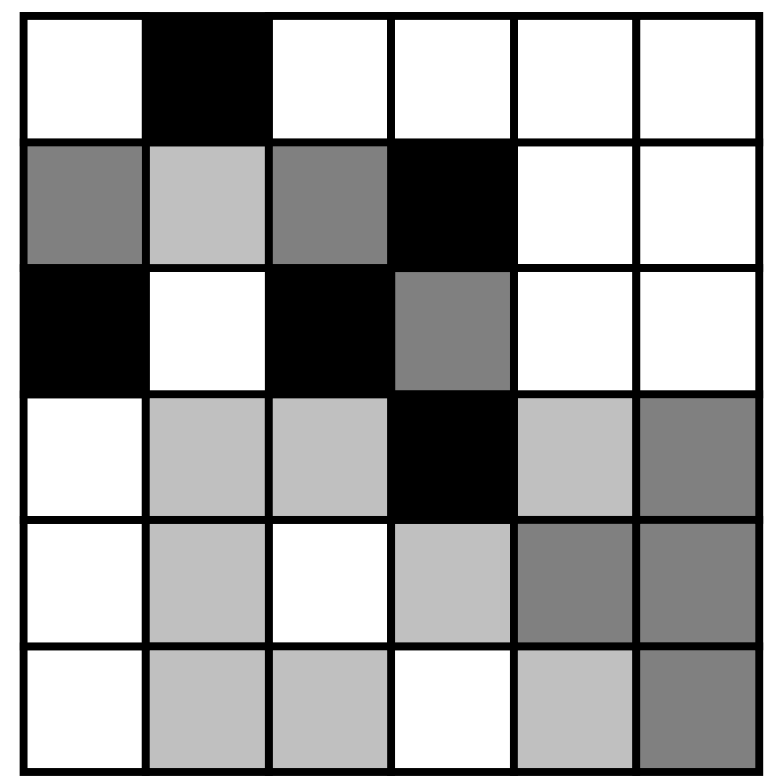
Histogram Processing:

- Distribution of gray-levels can be judged by measuring a Histogram

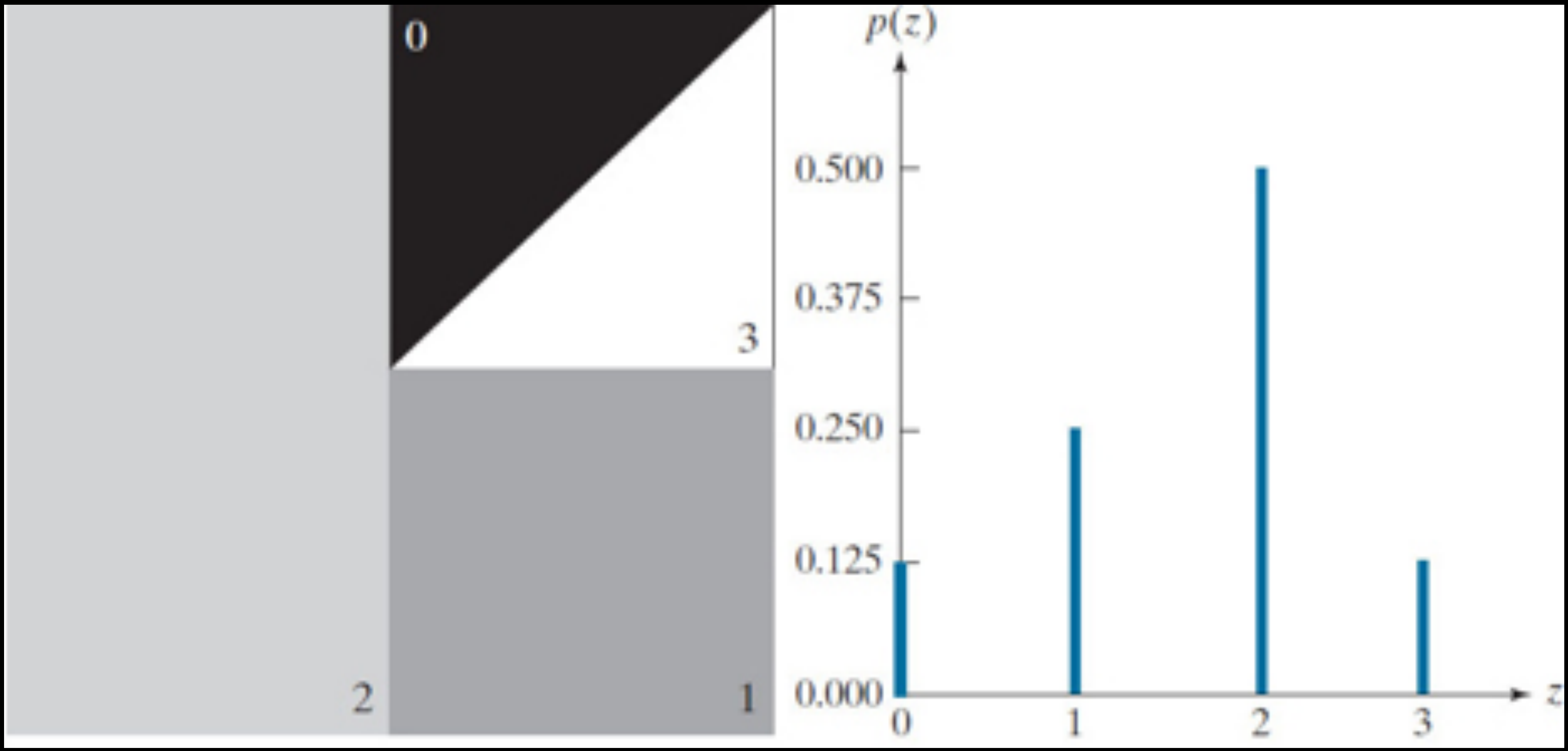
Histogram:



Graylevel



Histograms



Histogram manipulation

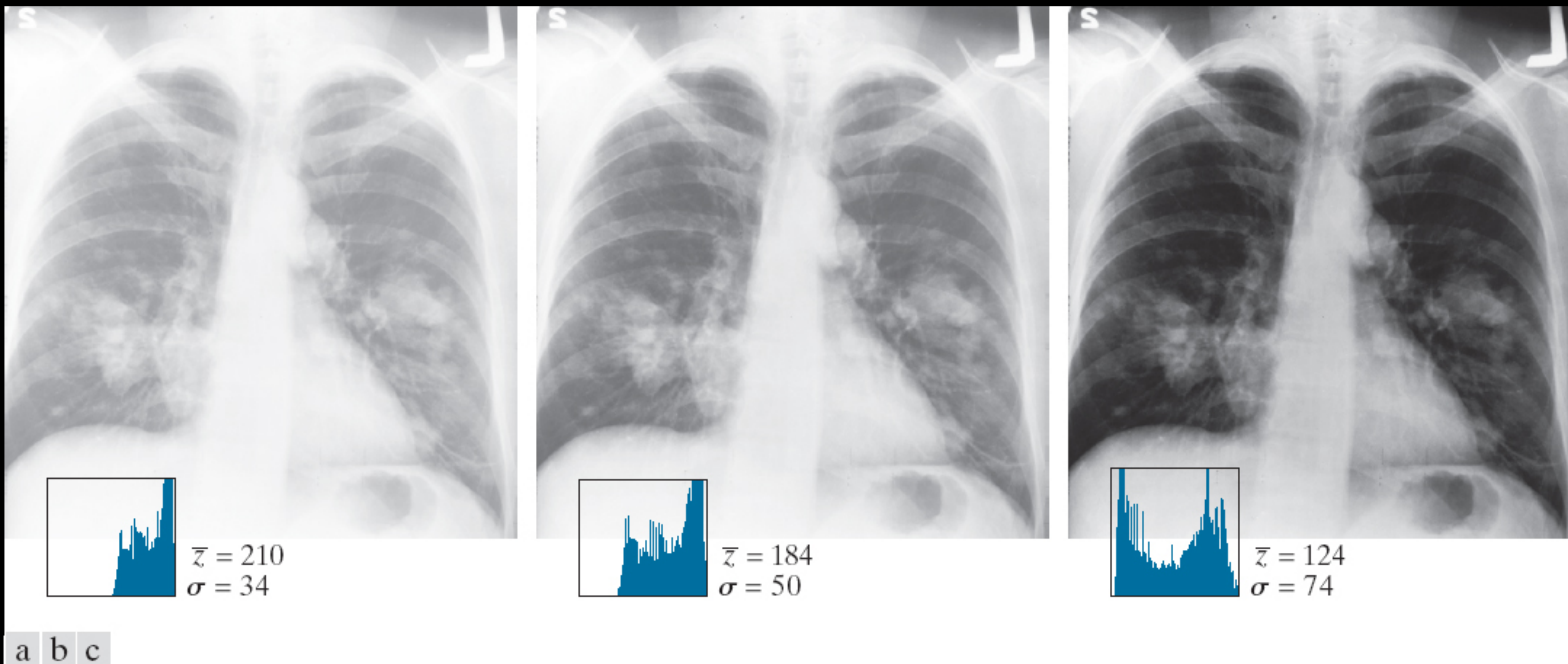
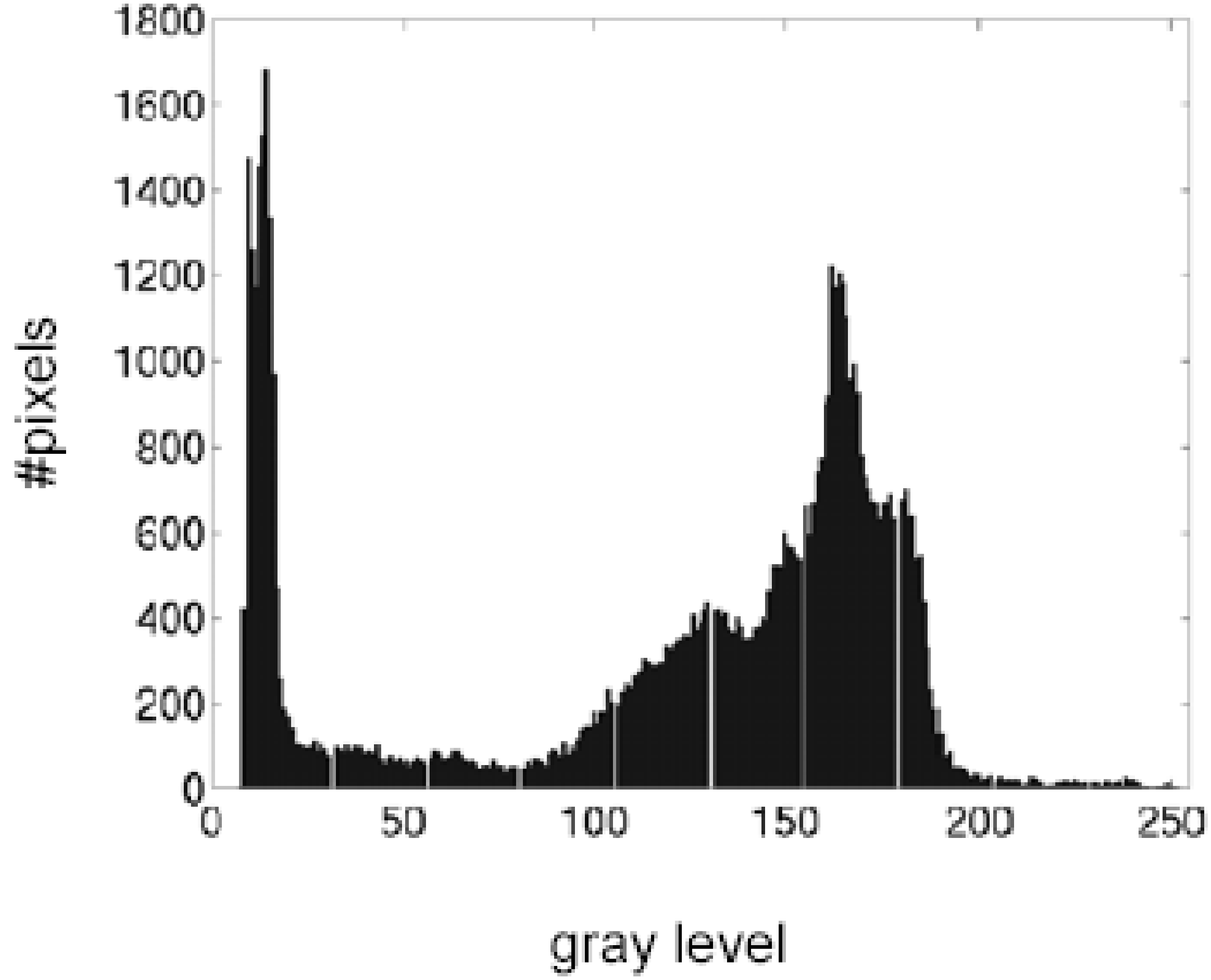


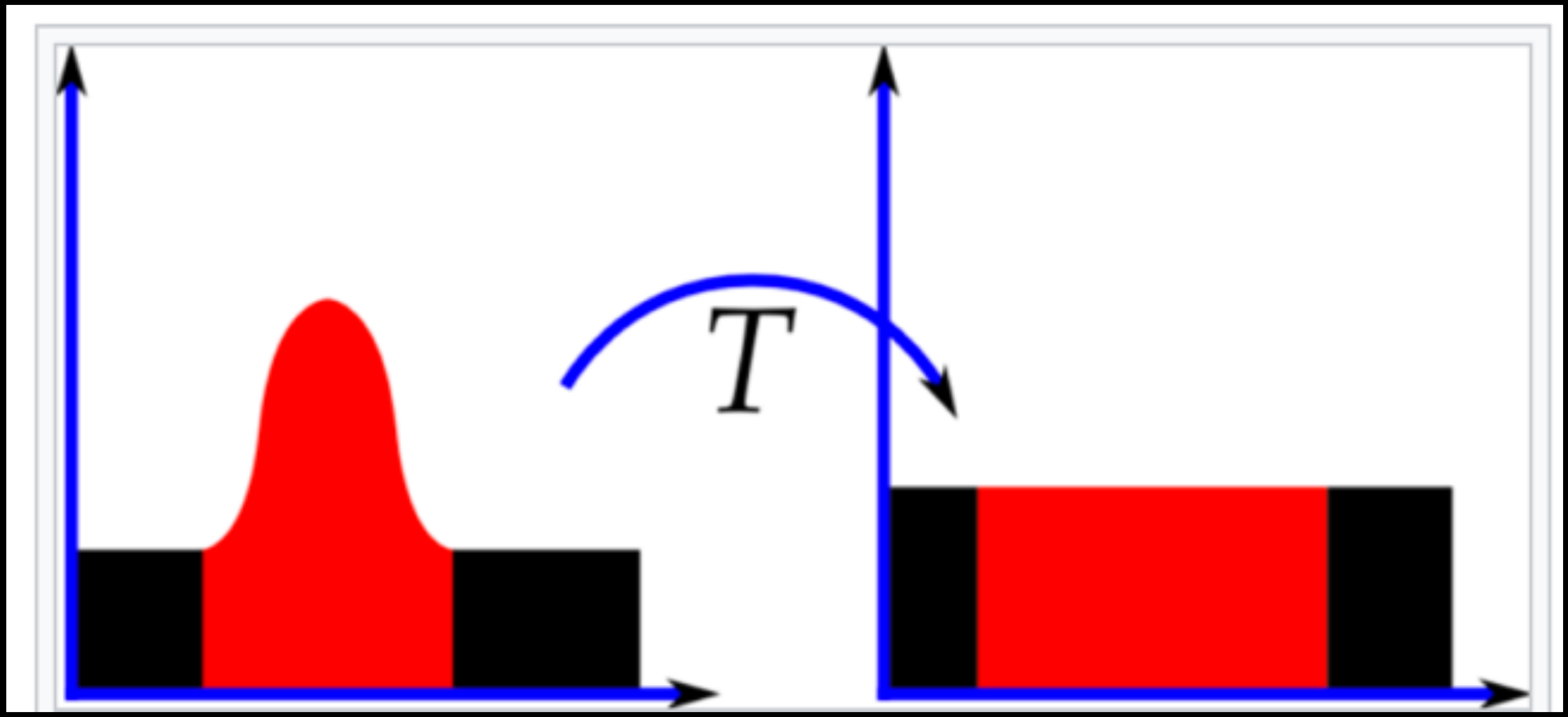
Illustration of the mean and standard deviation as functions of image contrast. (a)-(c) Images with low, medium, and high contrast, respectively. (Original image courtesy of the National Cancer Institute.)

Example:

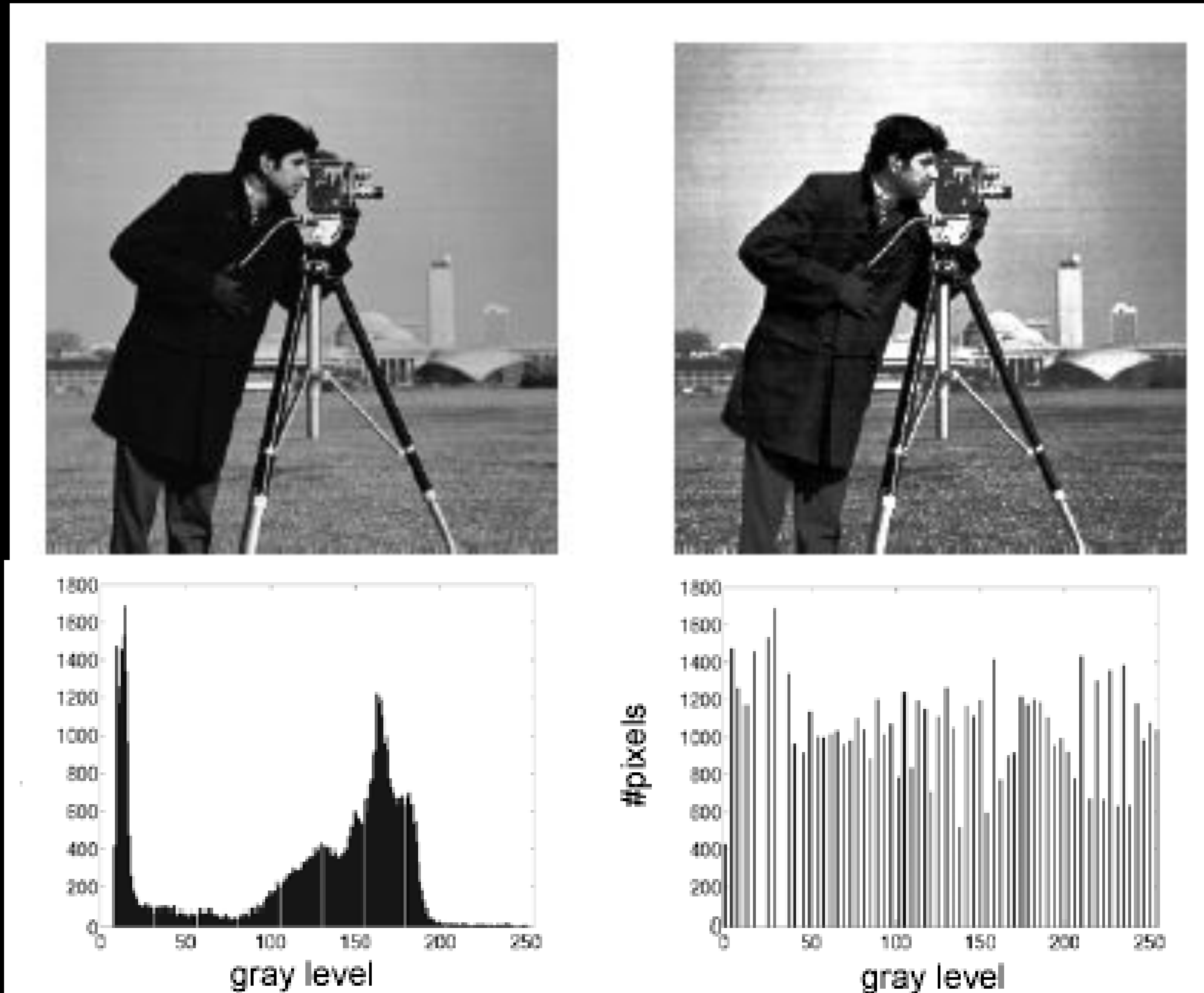


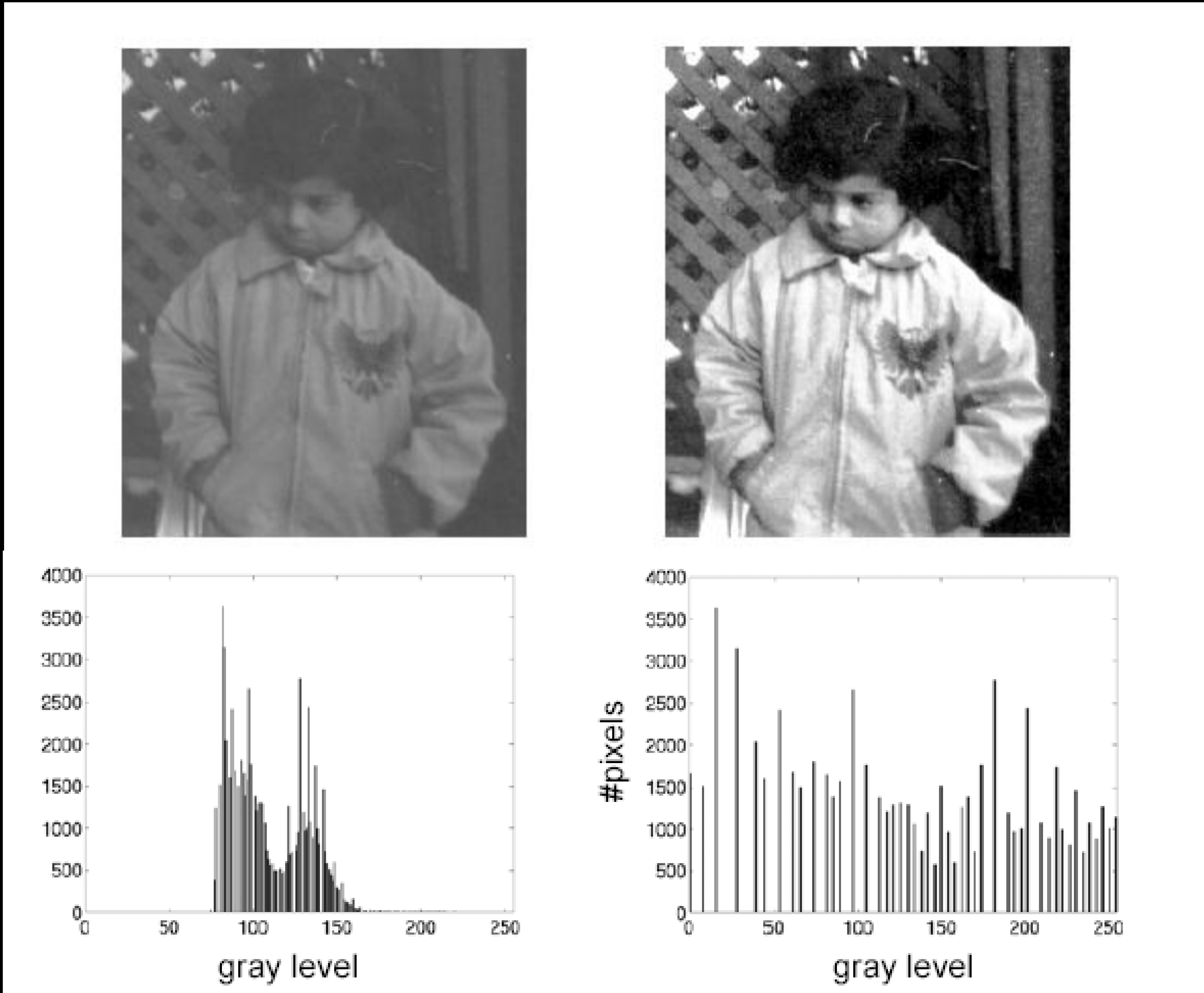
Cameraman
image

Histogram Equalization

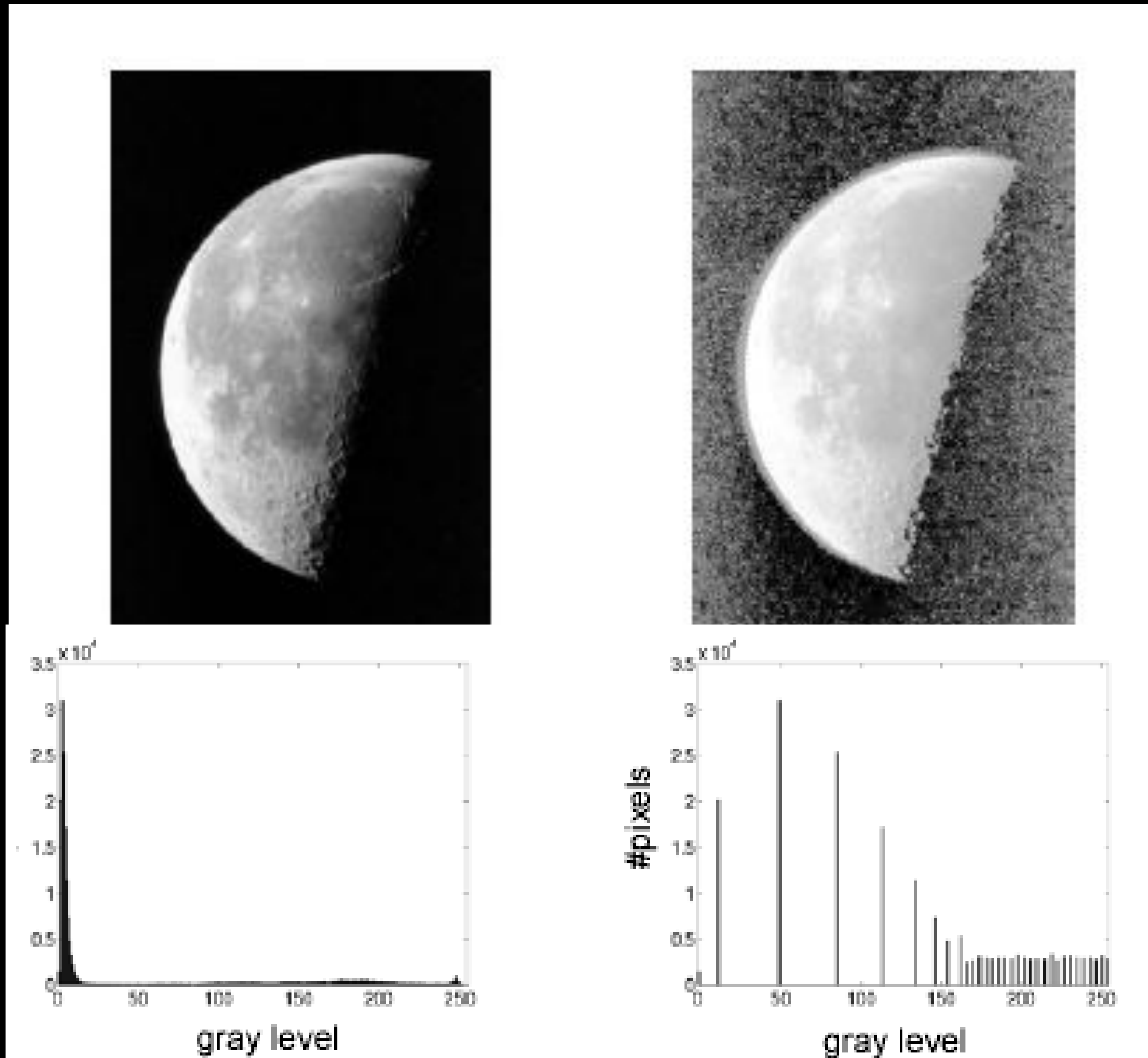


- Make it flat and spread it out
- This is a nonlinear operation


1800
1600
1400
1200
1000
800
600
400
200
0
0 50 100 150 200 250
gray level
#pixels
1800
1600
1400
1200
1000
800
600
400
200
0
0 50 100 150 200 250
gray level

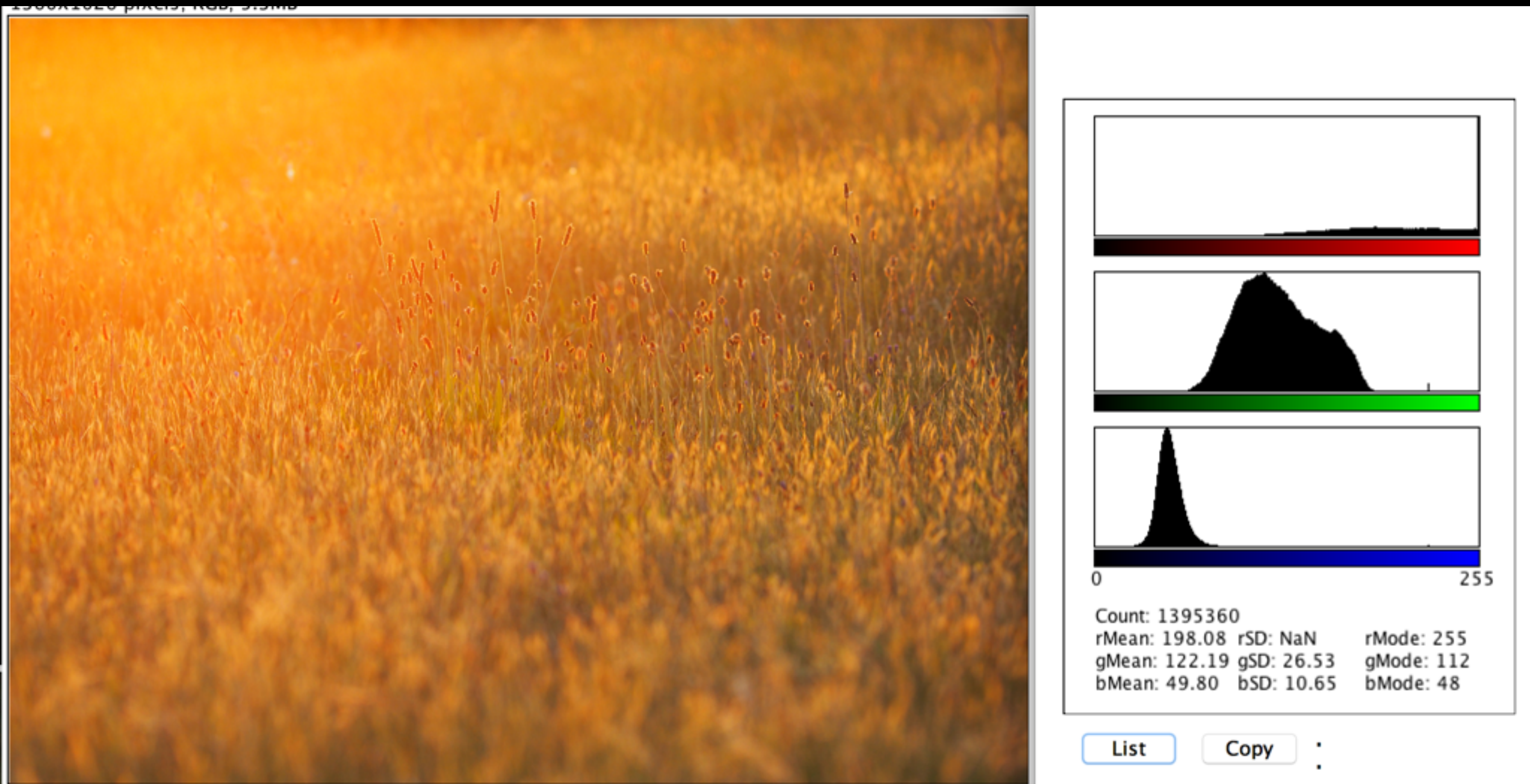


4000
3500
3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250
gray level
#pixels
4000
3500
3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250
gray level



x 10^4
3.5
3
2.5
2
1.5
1
0.5
0
0
50
100
150
200
250
gray level
#pixels
gray level

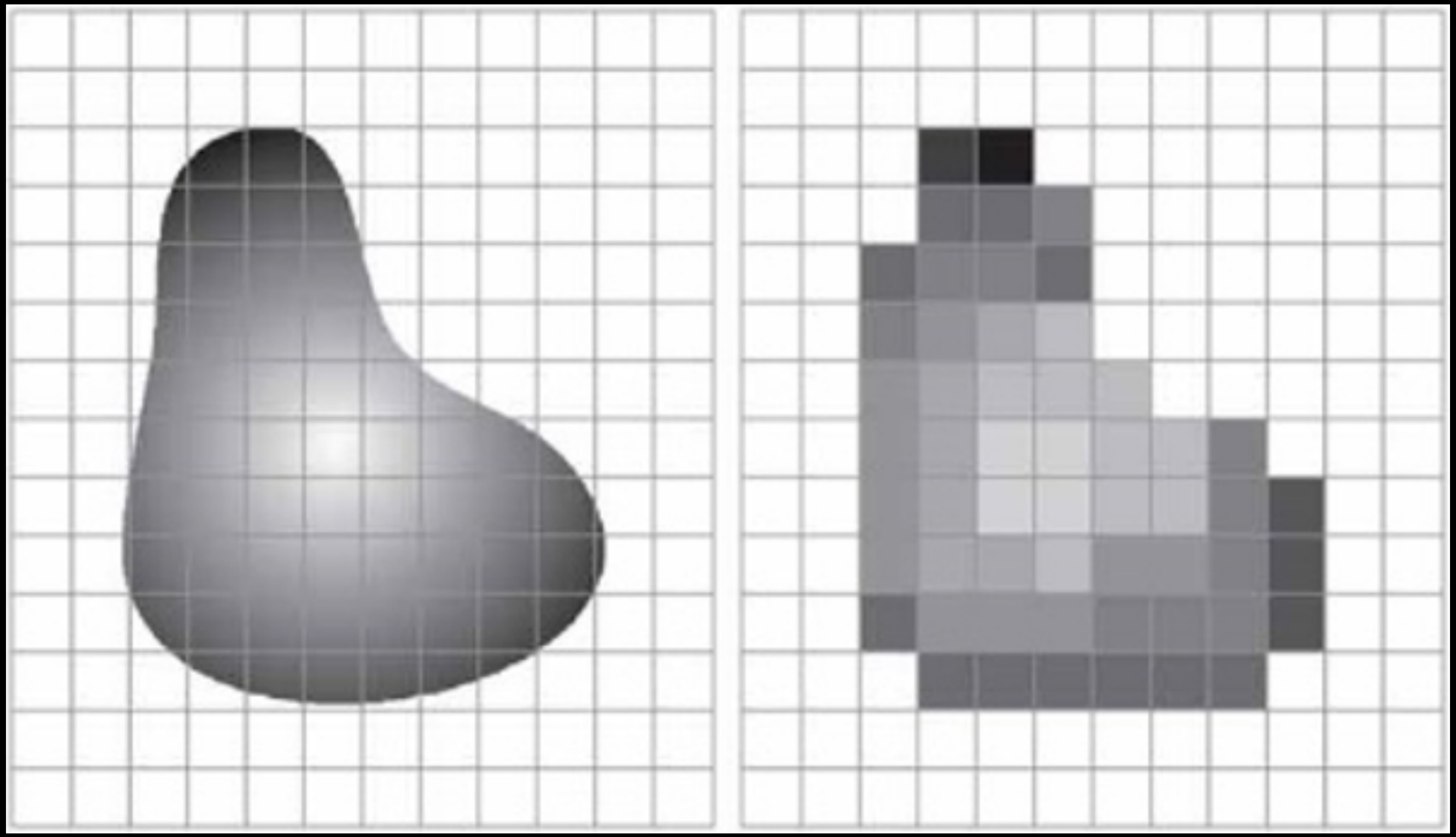
Color Histogram



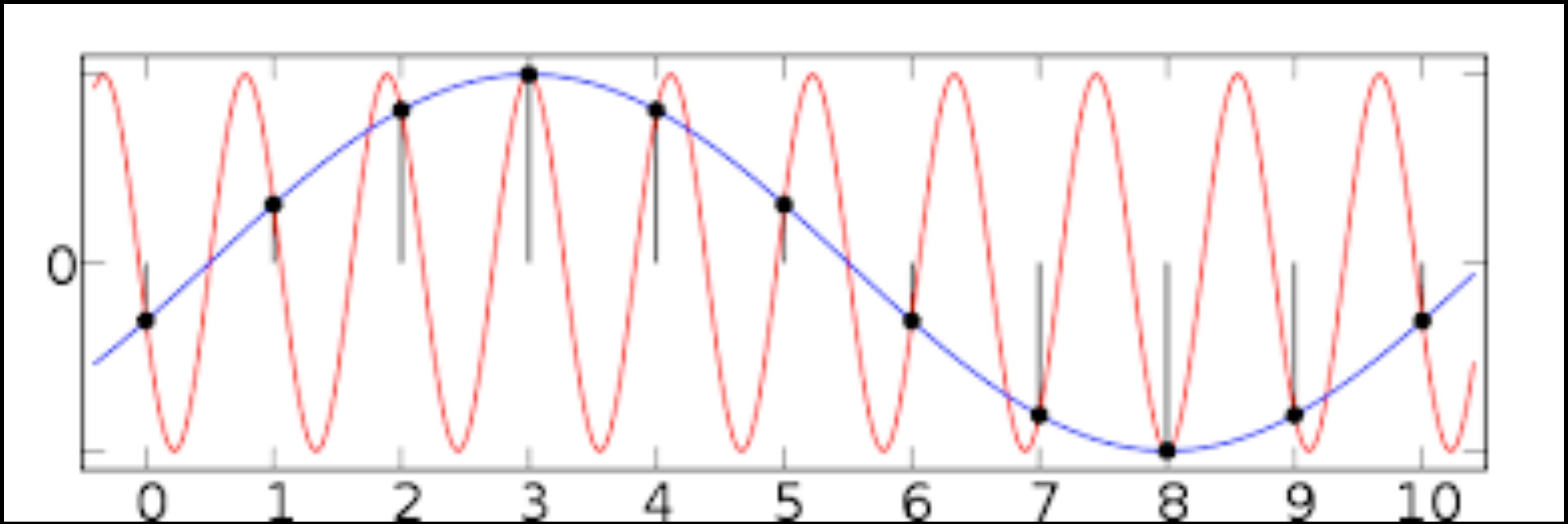
Spatial filtering

- **In image space**
- In frequency space

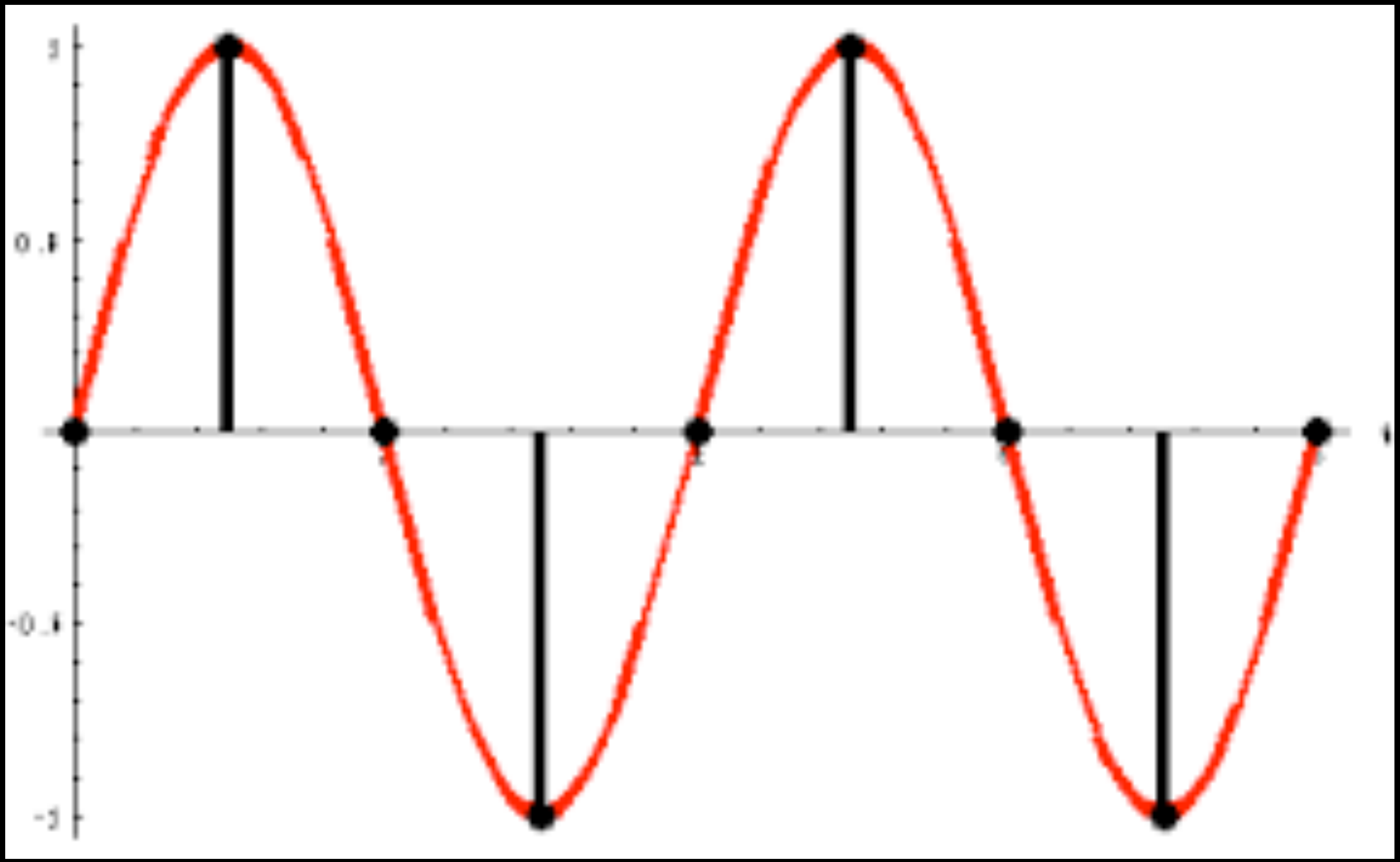
Image size / Sampling



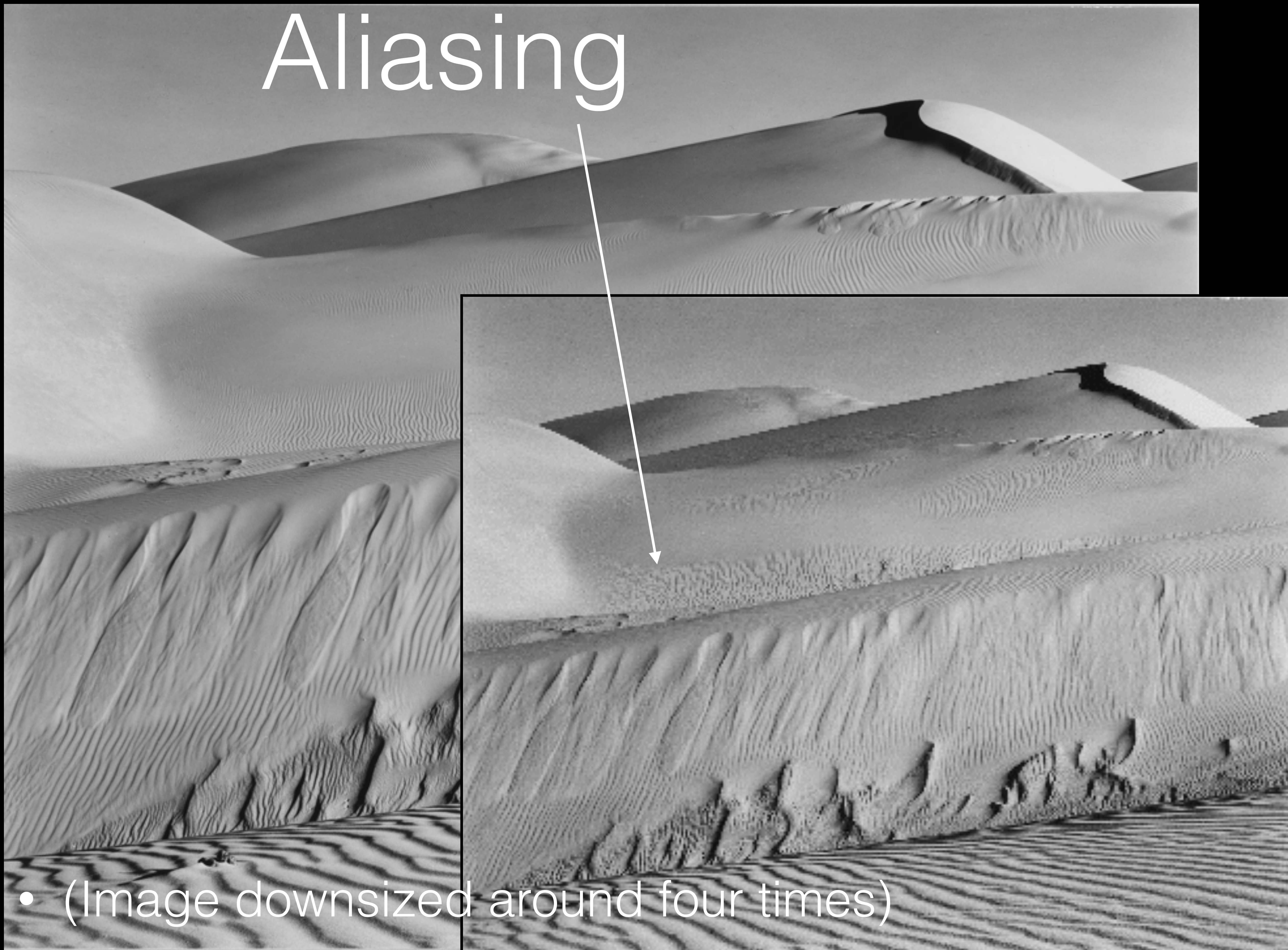
Aliasing



Nyquist sampling = twice the frequency



Aliasing

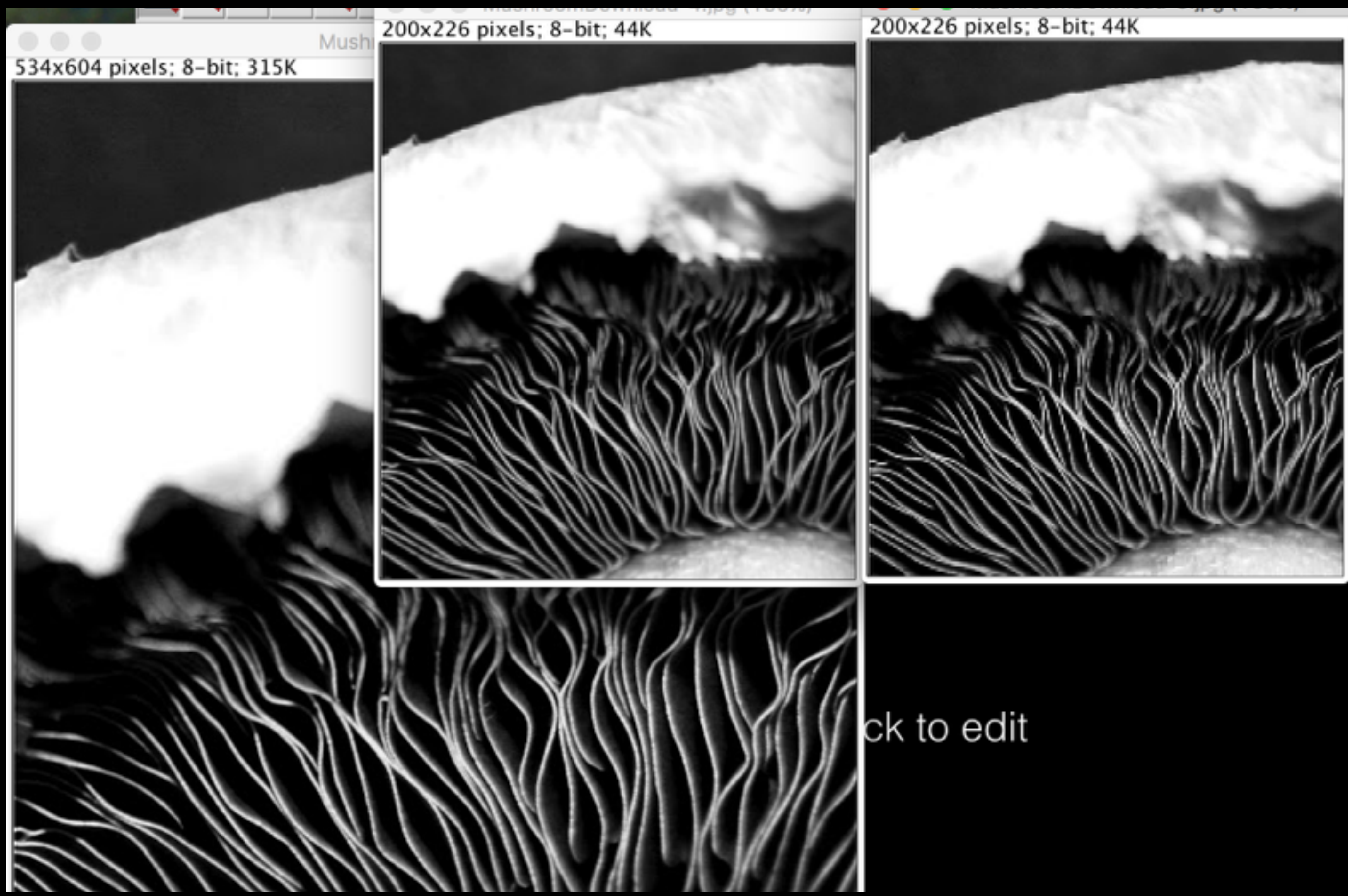


- (Image downsized around four times)

Re-sampling: Change size by interpolation

(a) Image reduced to 72 dpi and zoomed back to its original 930 dpi using nearest neighbor interpolation.
(b) Image reduced to 72 dpi and zoomed using bilinear interpolation.
(c) Same as (b) but using bicubic interpolation.

Average when downsizing?

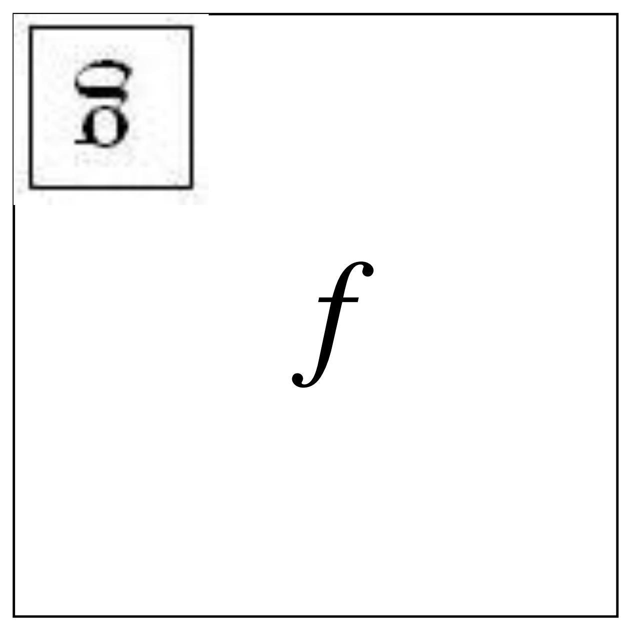


- Edges lose contrast if you average but result is smoother

Convolution

- Let *f* be the image and *g* be the kernel. The output of convolving *f* with *g* is denoted *f* * *g*.

$$(f * g)[m,n] = \sum_{k,l} f[m-k, n-l]\, g[k,l]$$



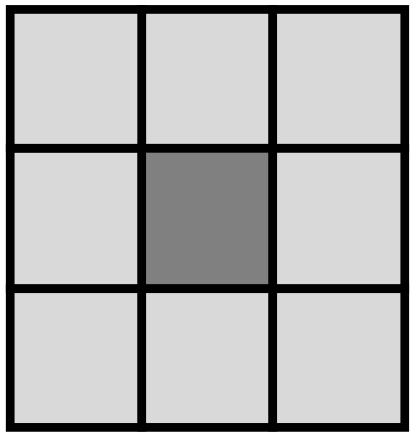
- Convention: kernel is “flipped”
- MATLAB: conv2 vs. filter2 (also imfilter)

Convolution

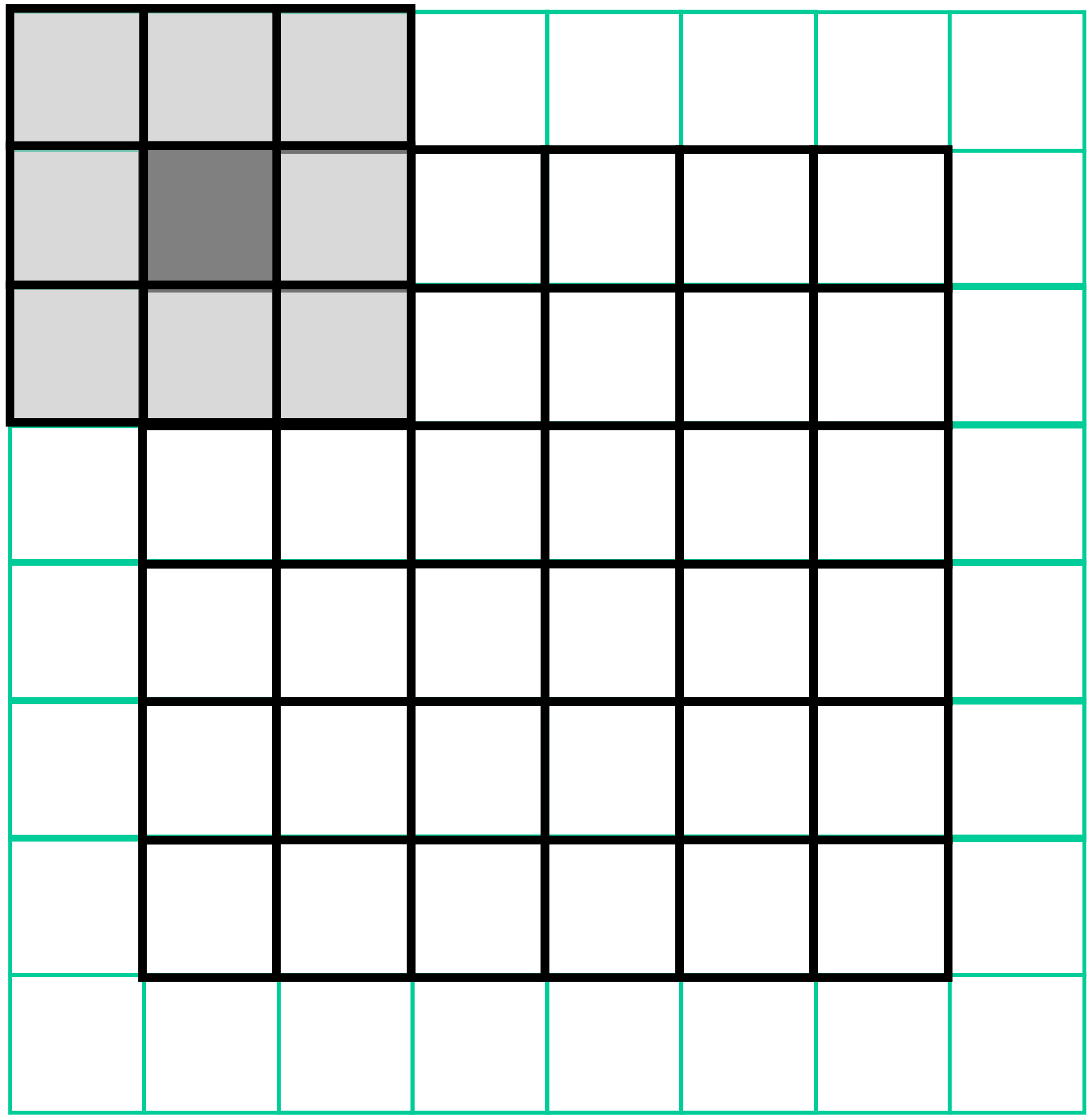
Key properties

- **Linearity:** filter(f_1 + f_2) = filter(f_1) + filter(f_2)
- **Shift invariance:** same behavior regardless of pixel location: filter(shift(f)) = shift(filter(f))
- Theoretical result: any linear shift-invariant operator can be represented as a convolution

Convolve
“Drag-and-Stamp”



Mask dimension = 2M+1



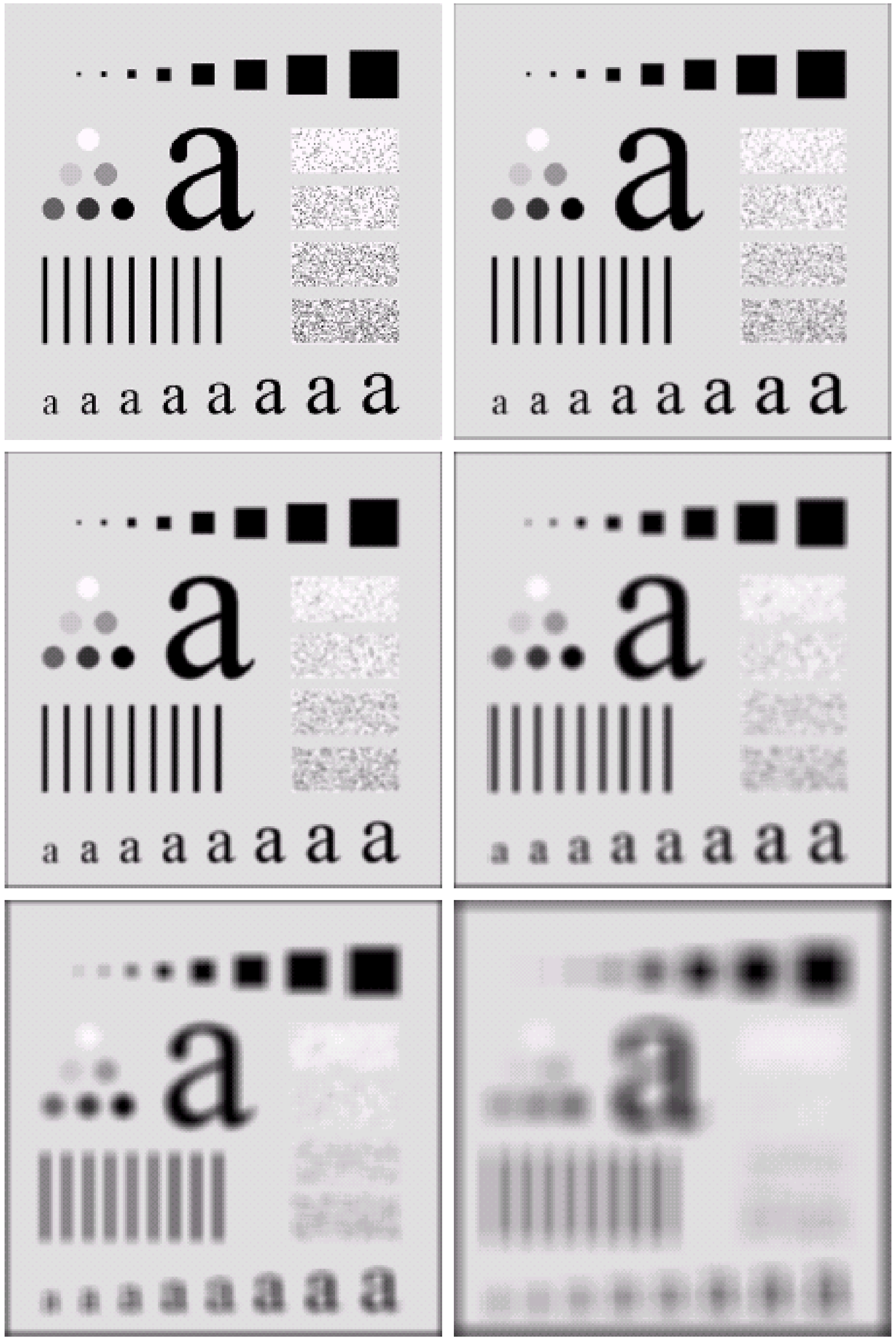
Border dimension = M

Spatial Filtering: Blurring

- Example

Averaging Mask:

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

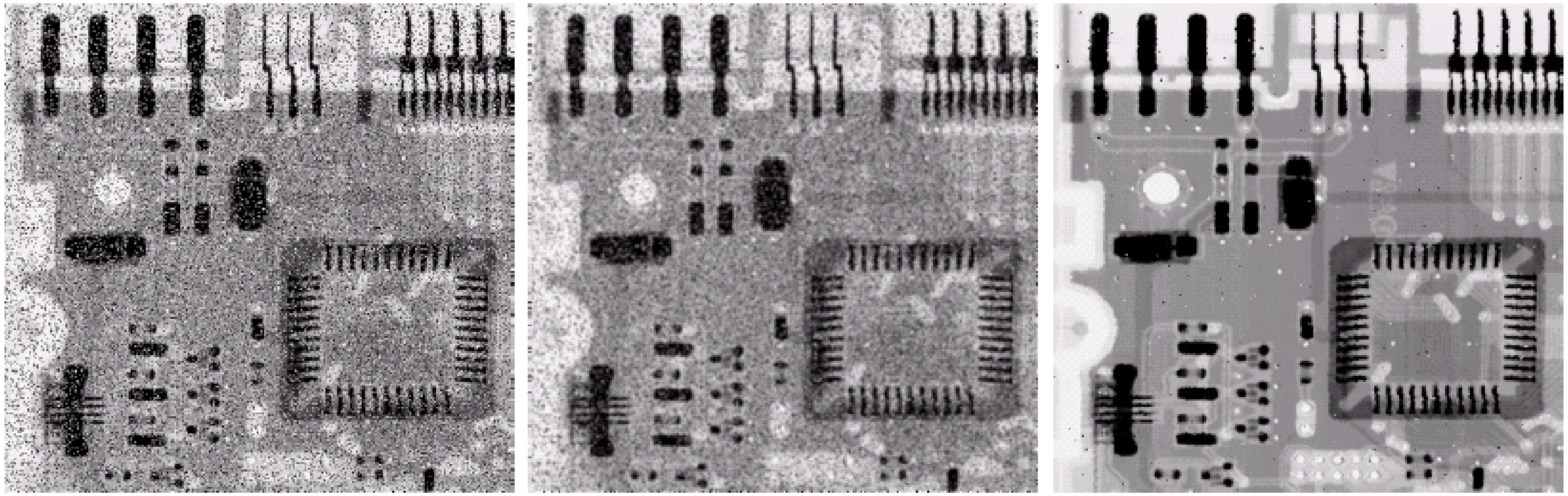


a b
c d
e f

FIGURE 3.35 (a) Original image, of size 500×500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $n = 3, 5, 9, 15$, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their gray levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50×120 pixels.

Image Enhancement:Spatial Filtering Operation

- Idea: Use a “mask” to alter pixel values according to local operation
- Aim: (De)-Emphasize some spatial frequencies in the image.

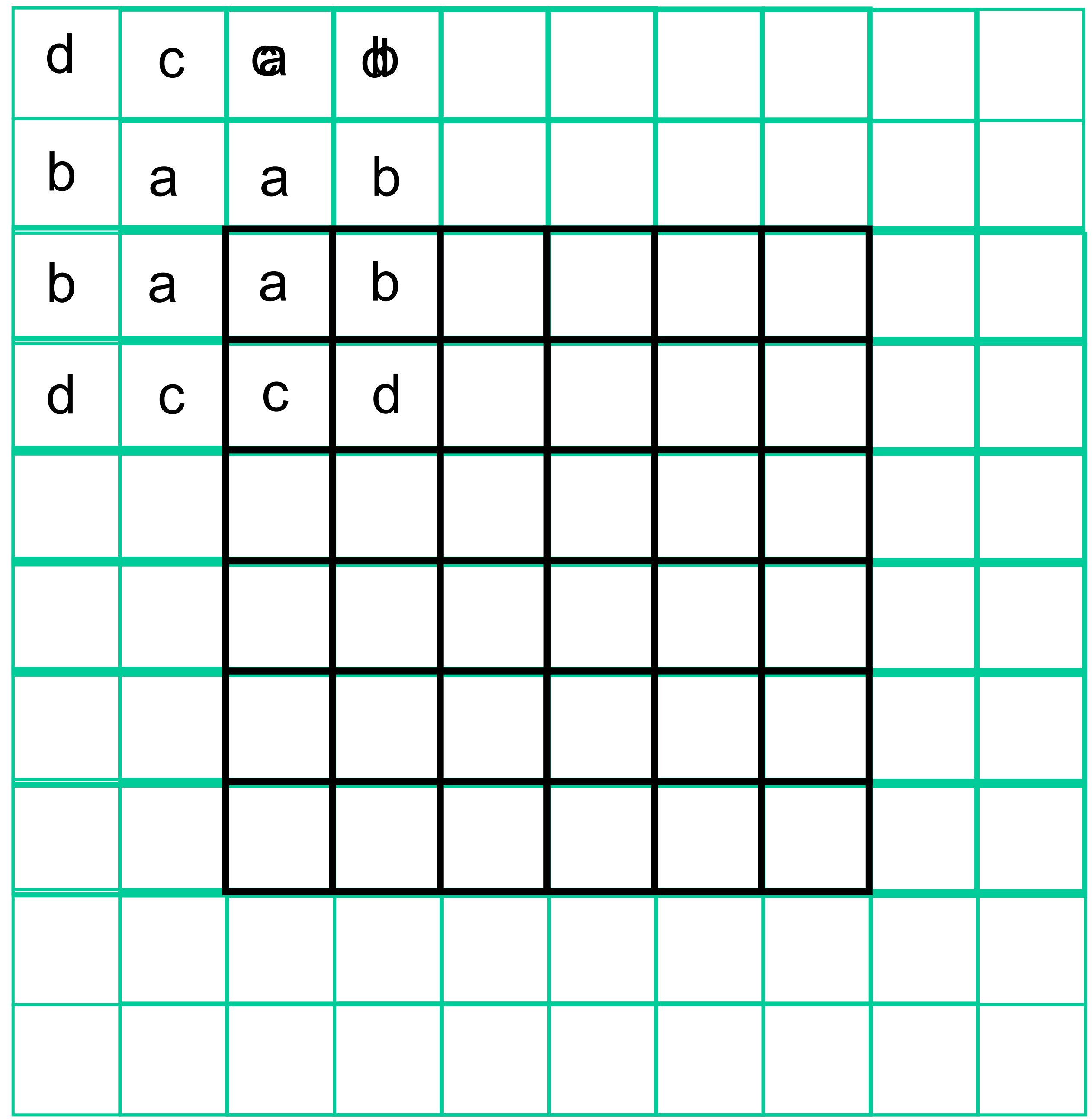


a b c

FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Image Enhancement:Spatial Filtering Operation

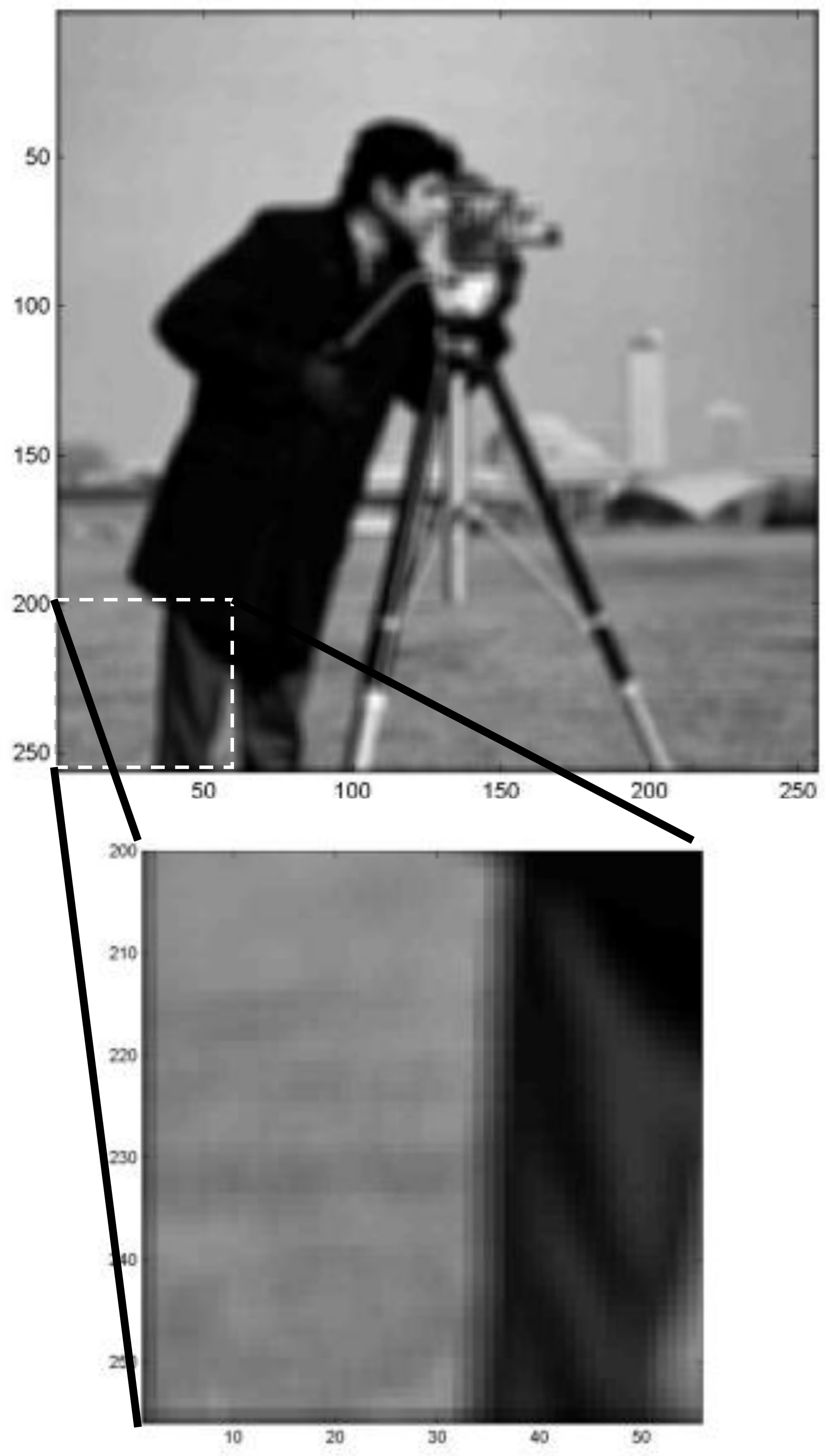
- An important point: Edge Effects (Ex.: 5x5 Mask)
  - How to fill in a “border”
    - Zeros (Ringing)
    - Replication (Better)
    - Reflection (“Best”)



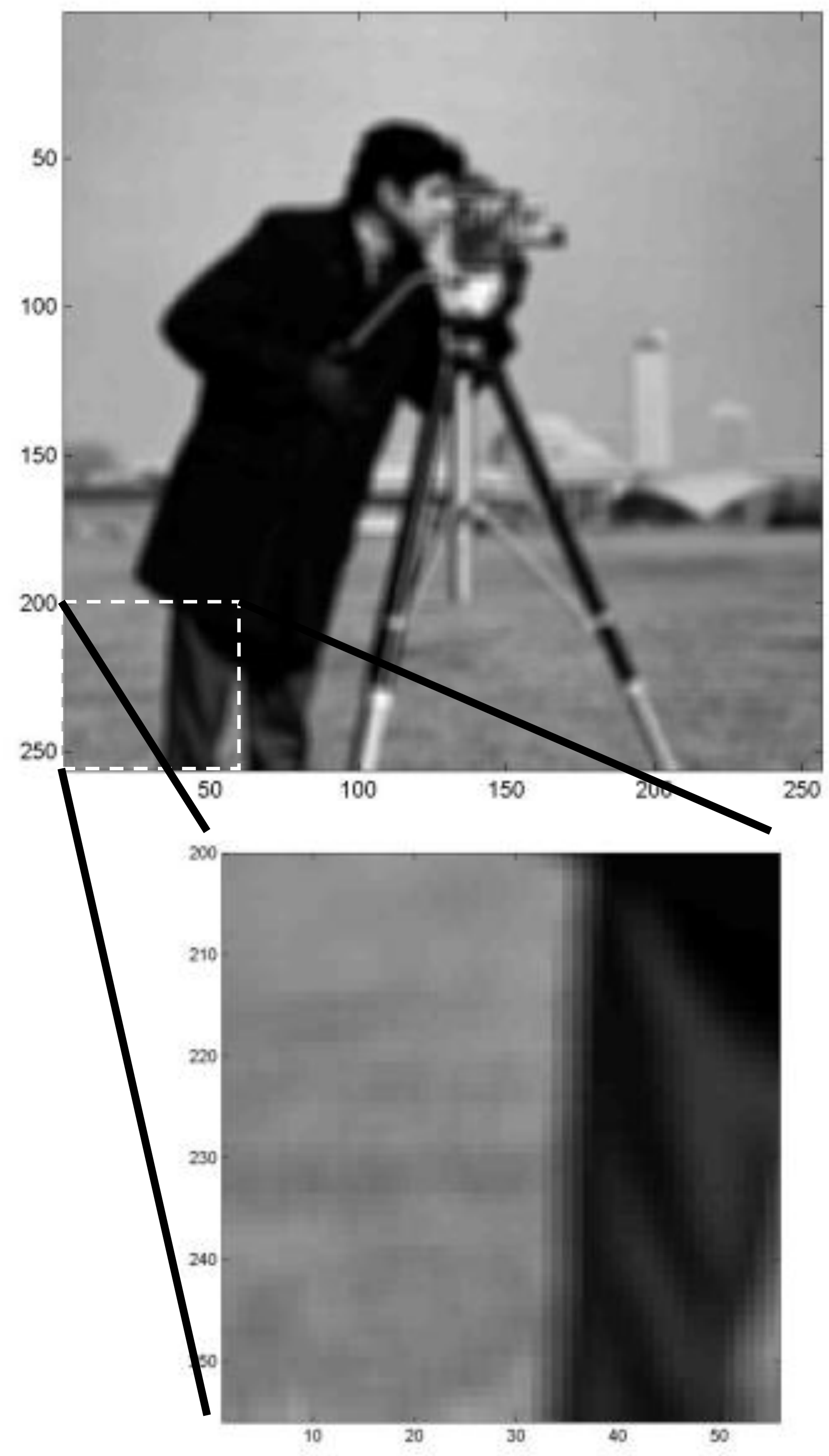
- Procedure:
  - Replicate row-wise
  - Replicate column-wise
  - Apply filtering
  - Remove borders

Image Enhancement:Spatial Filtering Operation

5x5 Blurring with 0-padding

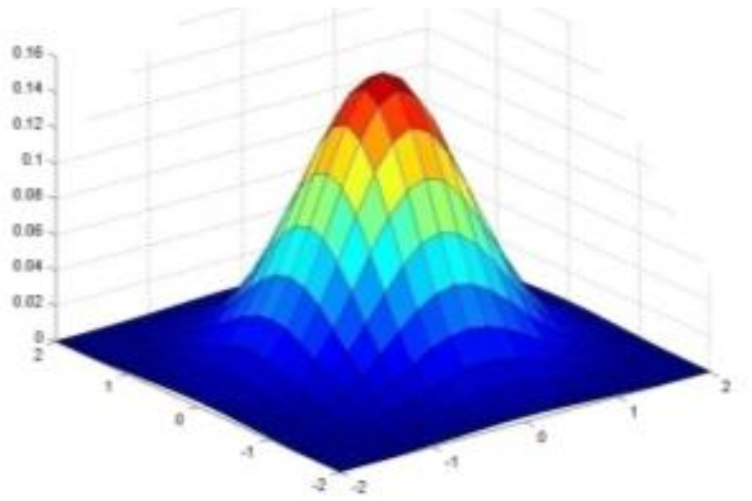


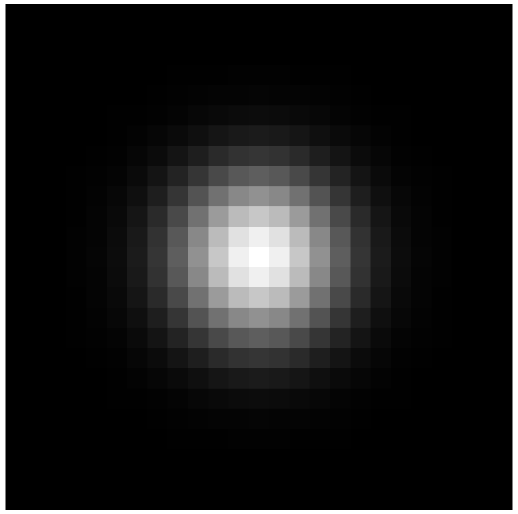
5x5 Blurring with reflected padding



Gaussian Kernel

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





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

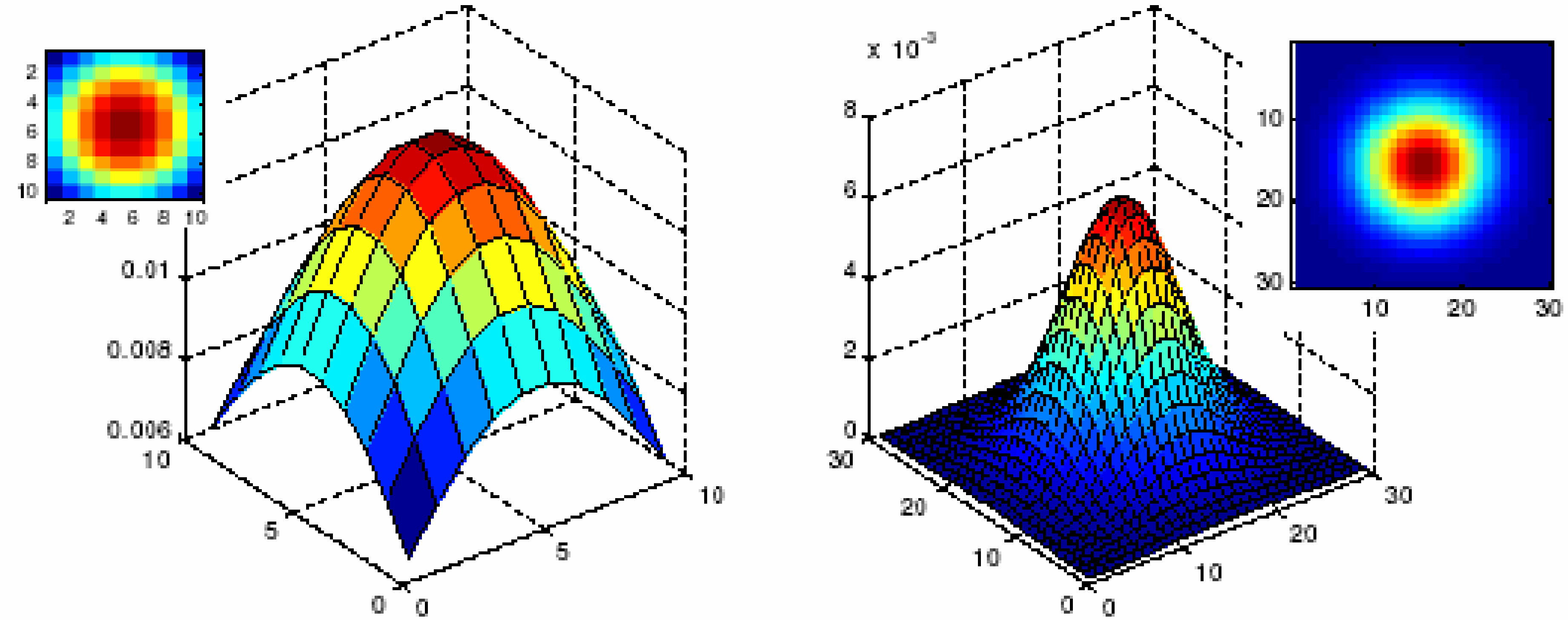
5 x 5, σ = 1

fspecial('gauss',5,1)

- Constant factor at front makes volume sum to 1 (can be ignored, as we should re-normalize weights to sum to 1 in any case)

Choosing kernel width

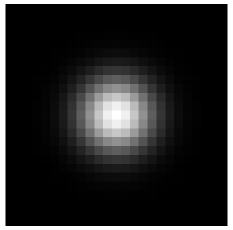
- Gaussian filters have infinite support, but discrete filters use finite kernels

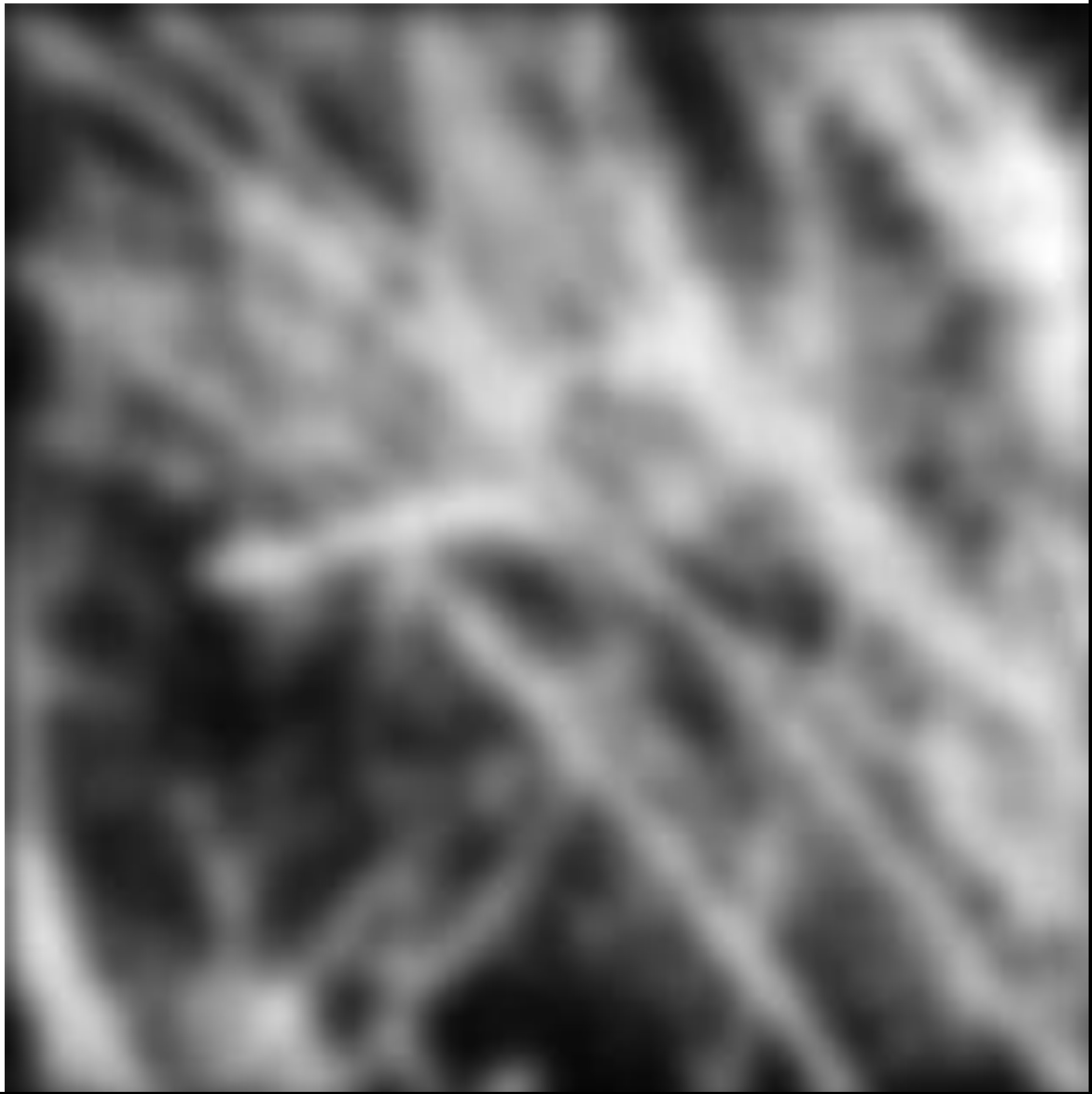


σ = 5 with 10x10 kernel

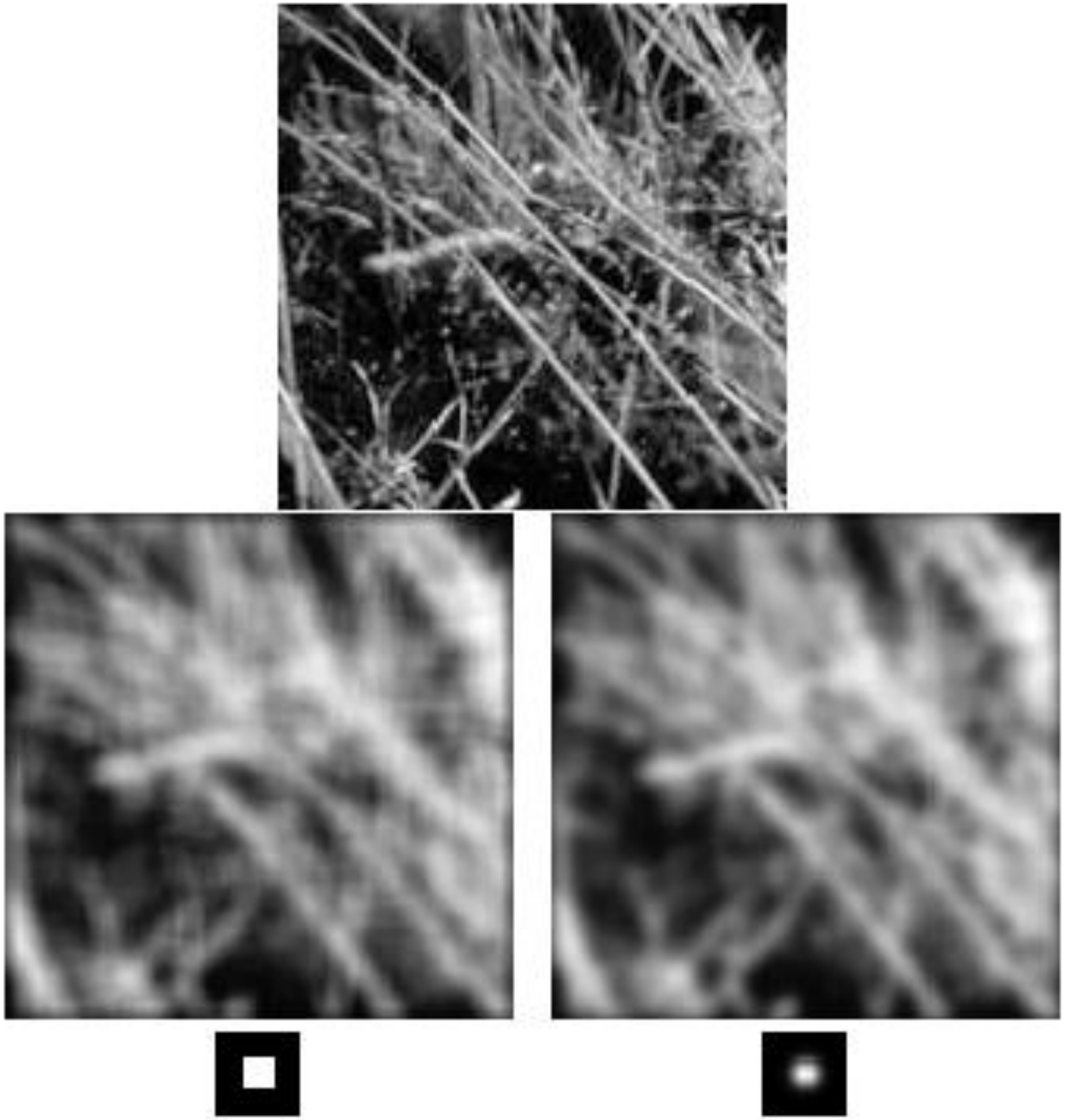
σ = 5 with 30x30 kernel

Example: Smoothing with a Gaussian





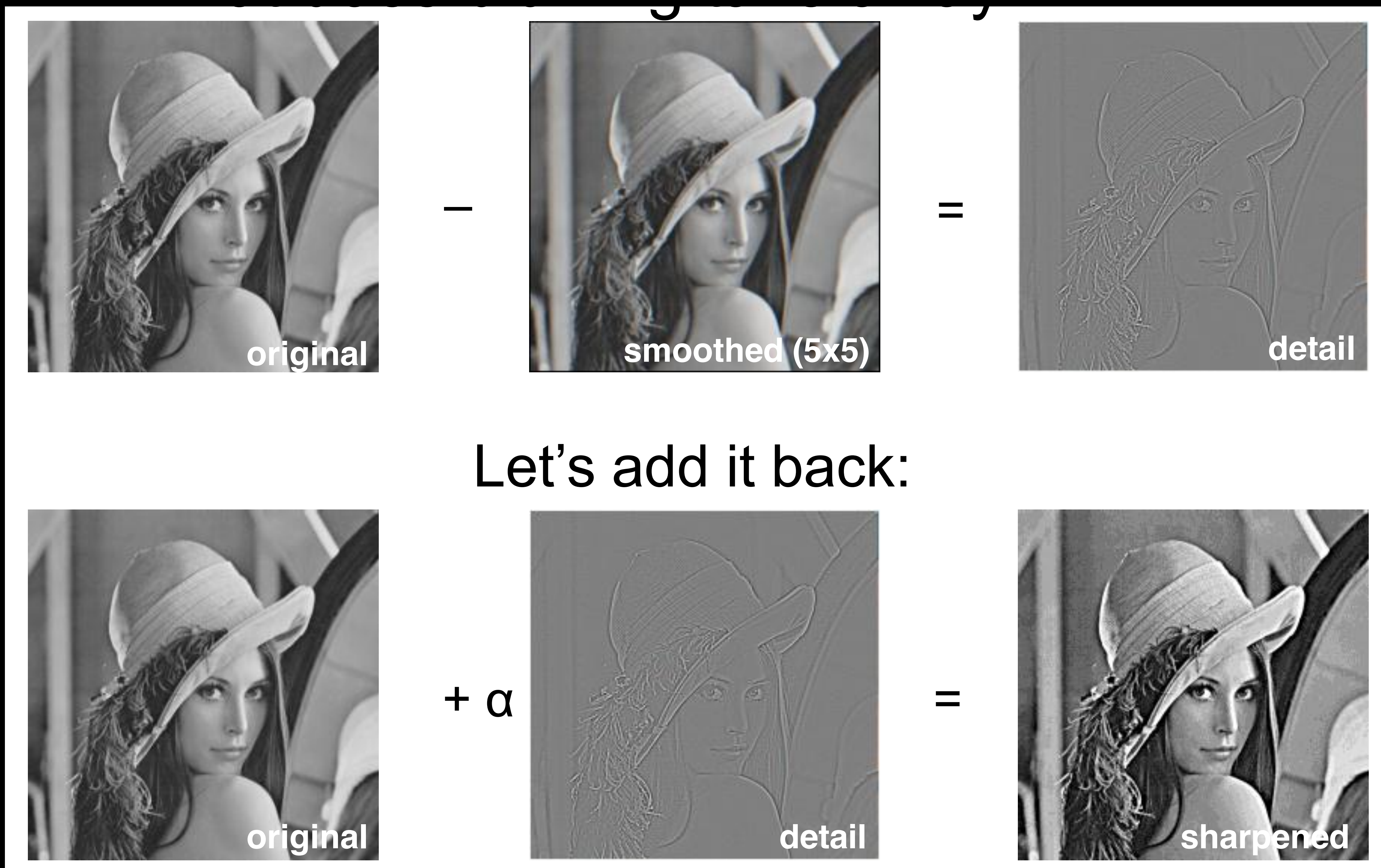
Mean vs. Gaussian filtering



Gaussian filters

- Remove “high-frequency” components from the image (low-pass filter)
- Convolution with self is another Gaussian
  - So can smooth with small-width kernel, repeat, and get same result as larger-width kernel would have
  - Convolving two times with Gaussian kernel of width σ is same as convolving once with kernel of width $\sigma\sqrt{2}$
- *Separable* kernel
  - Factors into product of two 1D Gaussians

Use this to sharpen!



More on Linear Operations: Sharpening Filters

- Sharpening filters use masks that typically have + and – numbers in them.
- They are useful for highlighting or enhancing details and high-frequency information (e.g. edges)
- They can (and often are) based on derivative-type operations in the image (whereas smoothing operations were based on “integral” type operations)

Derivatives

Differentiation and convolution

- Recall, for 2D function, f(x,y):

$$\frac{\partial f}{\partial x} = \lim_{\varepsilon \to 0}\left(\frac{f(x+\varepsilon, y)}{\varepsilon} - \frac{f(x,y)}{\varepsilon}\right)$$

- This is linear and shift invariant, so must be the result of a convolution.

- We could approximate this as

$$\frac{\partial f}{\partial x} \approx \frac{f(x_{n+1}, y) - f(x_n, y)}{\Delta x}$$

- which is obviously a convolution with kernel

-1	**1**

Derivative-type Filters

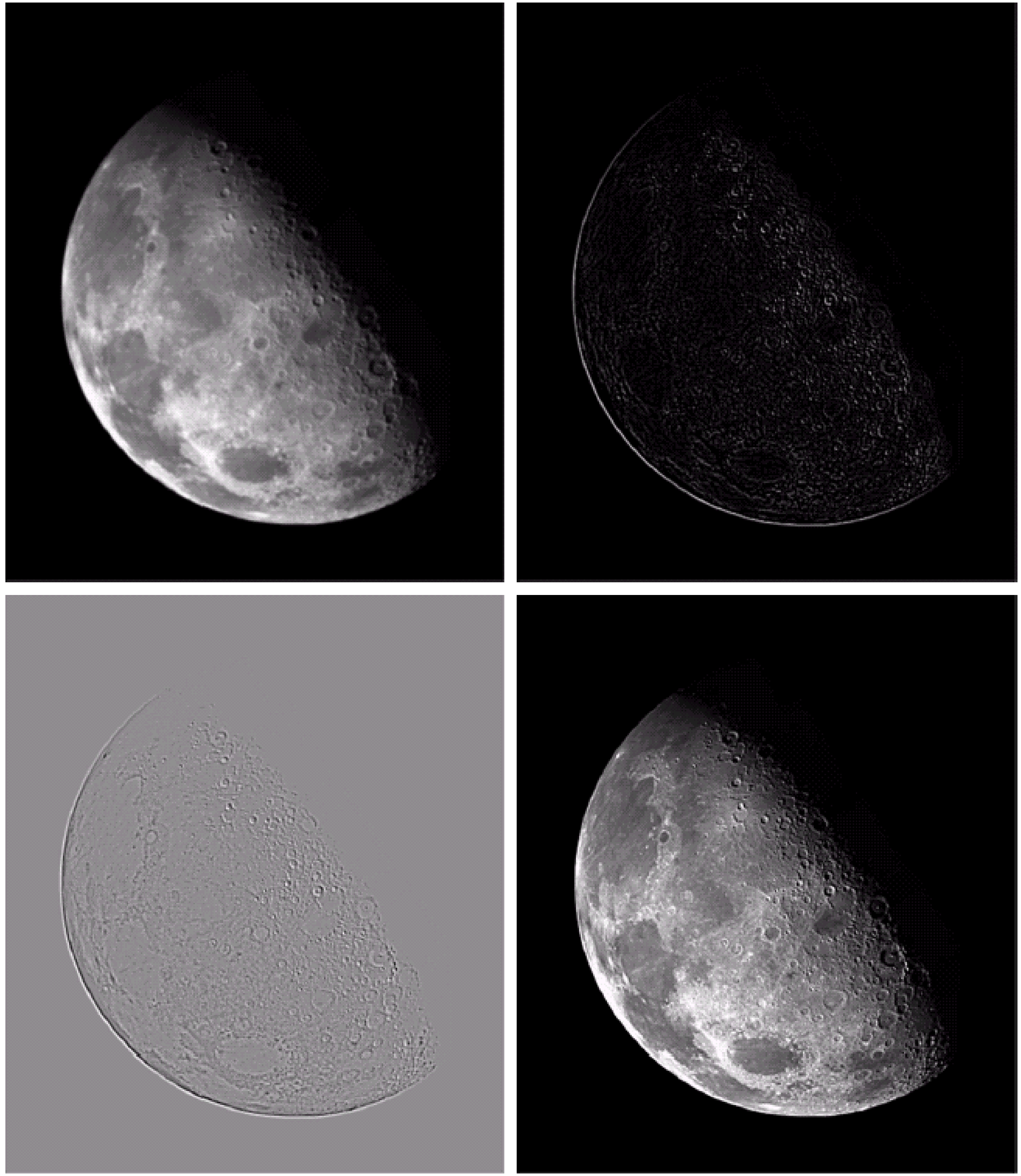
$$\begin{bmatrix} -1 & 1 \end{bmatrix} \longrightarrow \frac{\partial f}{\partial \mathrm{x}} \approx f(x+1, y) - f(x, y)$$

$$\begin{bmatrix} -1 \\ 1 \end{bmatrix} \longrightarrow \frac{\partial f}{\partial \mathrm{y}} \approx f(x, y+1) - f(x, y)$$

$$\begin{bmatrix} 1 & -2 & 1 \end{bmatrix} \longrightarrow \frac{\partial^2 f}{\partial \mathrm{x}^2} \approx f(x+1, y) - 2f(x, y) + f(x-1, y)$$

$$\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \longrightarrow \frac{\partial^2 f}{\partial \mathrm{y}^2} \approx f(x, y+1) - 2f(x, y) + f(x, y-1)$$

Laplacian: $$\nabla^2 f = \frac{\partial^2 f}{\partial \mathrm{x}^2} + \frac{\partial^2 f}{\partial \mathrm{y}^2} \Rightarrow \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} + \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

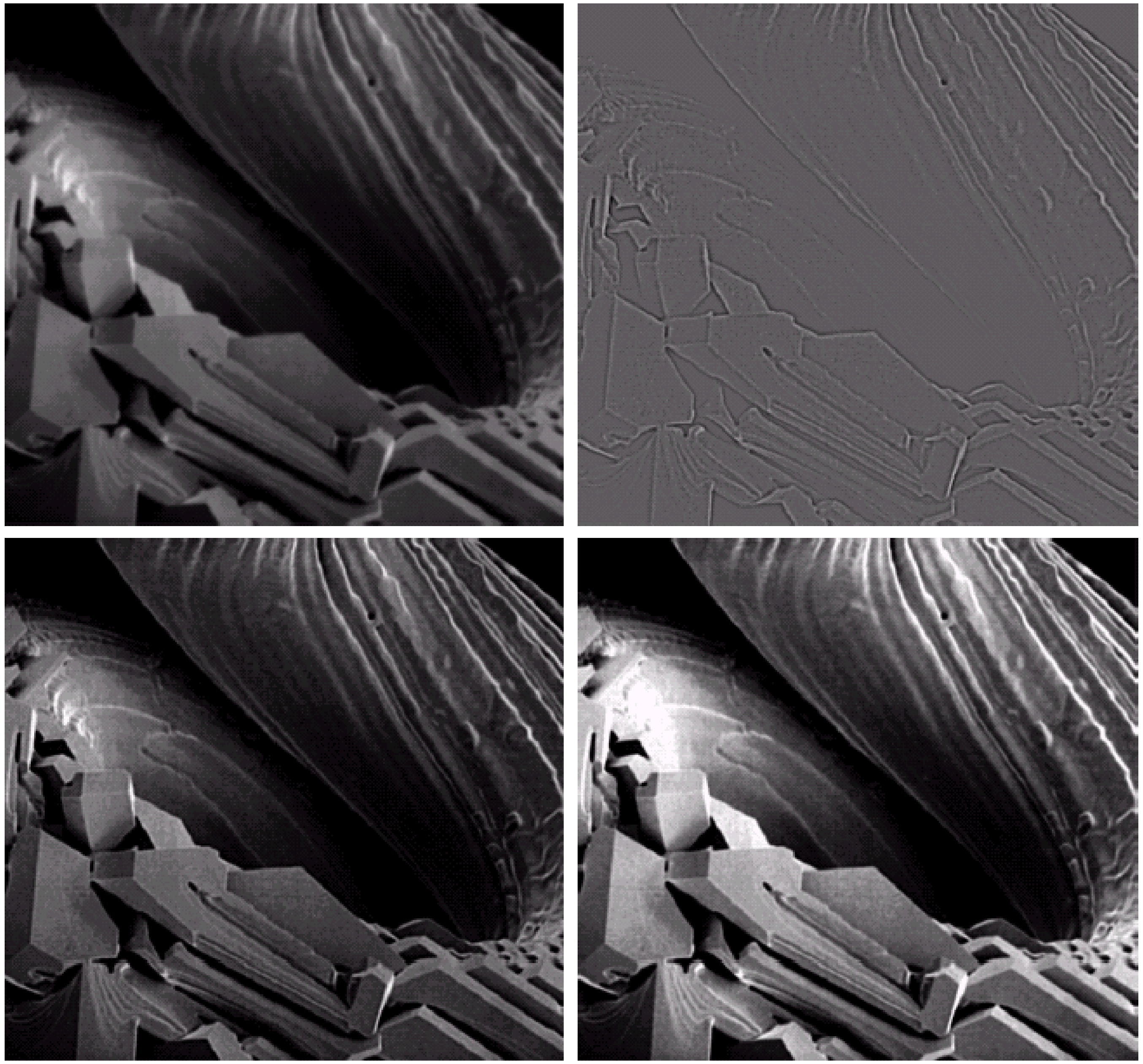


a b
c d

FIGURE 3.40 (a) Image of the North Pole of the moon. (b) Laplacian-filtered image. (c) Laplacian image scaled for display purposes. (d) Image enhanced by using Eq. (3.7-5). (Original image courtesy of NASA.)

Sharpening Using the Laplacian Filter

$$g(x,y) = A\,f(x,y) - \nabla^2 f(x,y) \longrightarrow \begin{bmatrix} -1 & -1 & -1 \\ -1 & A+8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



a b
c d

FIGURE 3.43
(a) Same as Fig. 3.41(c), but darker.
(a) Laplacian of (a) computed with the mask in Fig. 3.42(b) using $A = 0$.
(c) Laplacian enhanced image using the mask in Fig. 3.42(b) with $A = 1$. (d) Same as (c), but using $A = 1.7$.

Boosting High Frequencies

Laplacian of Gaussian

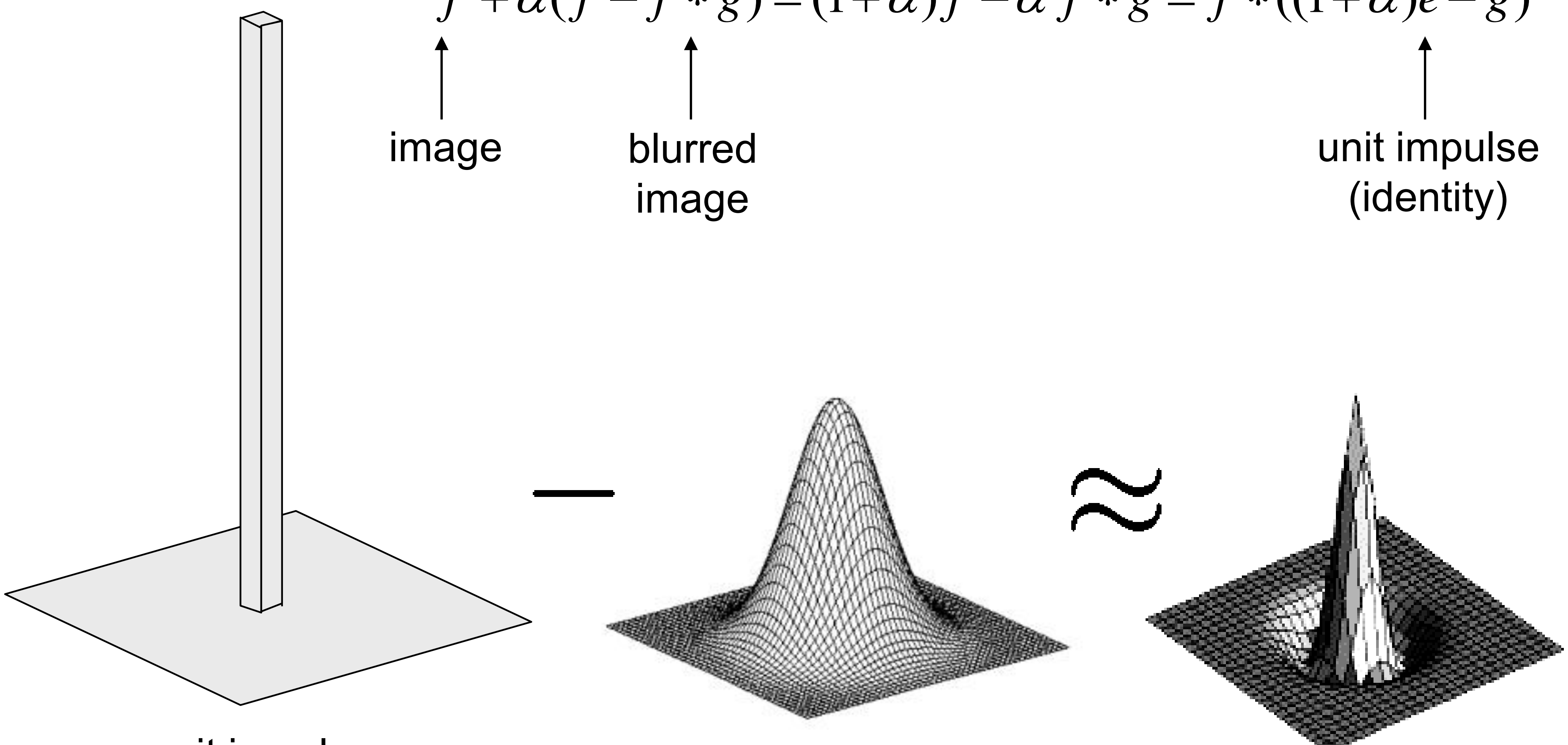
Gaussian Unsharp Mask Filter

$$f+\alpha(f-f*g)=(1+\alpha)f-\alpha f*g=f*((1+\alpha)e-g)$$

image

blurred image

unit impulse (identity)



unit impulse

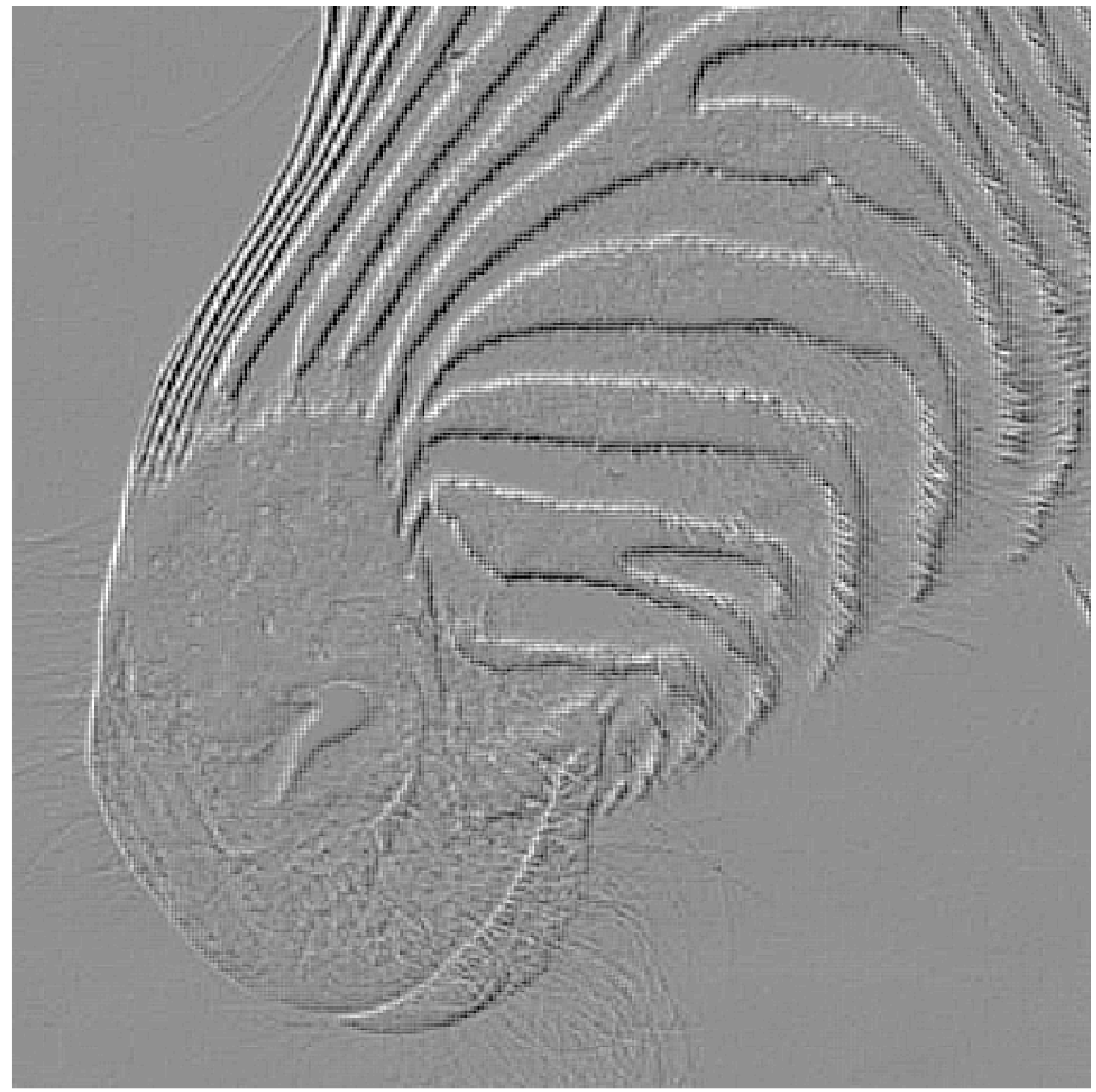
Gaussian

Laplacian of Gaussian

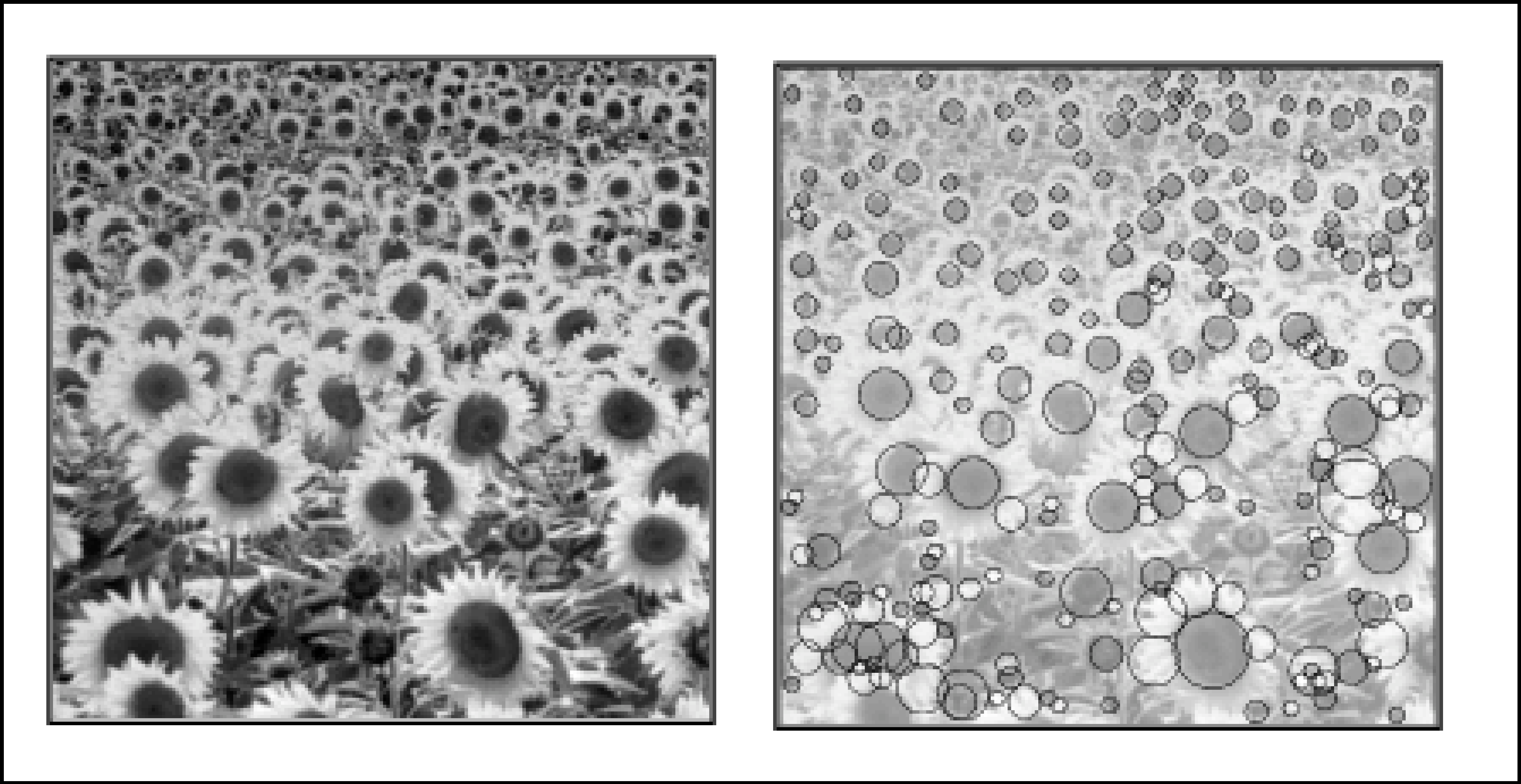
Edge detection

Original

LoG-filtered



Blob detection

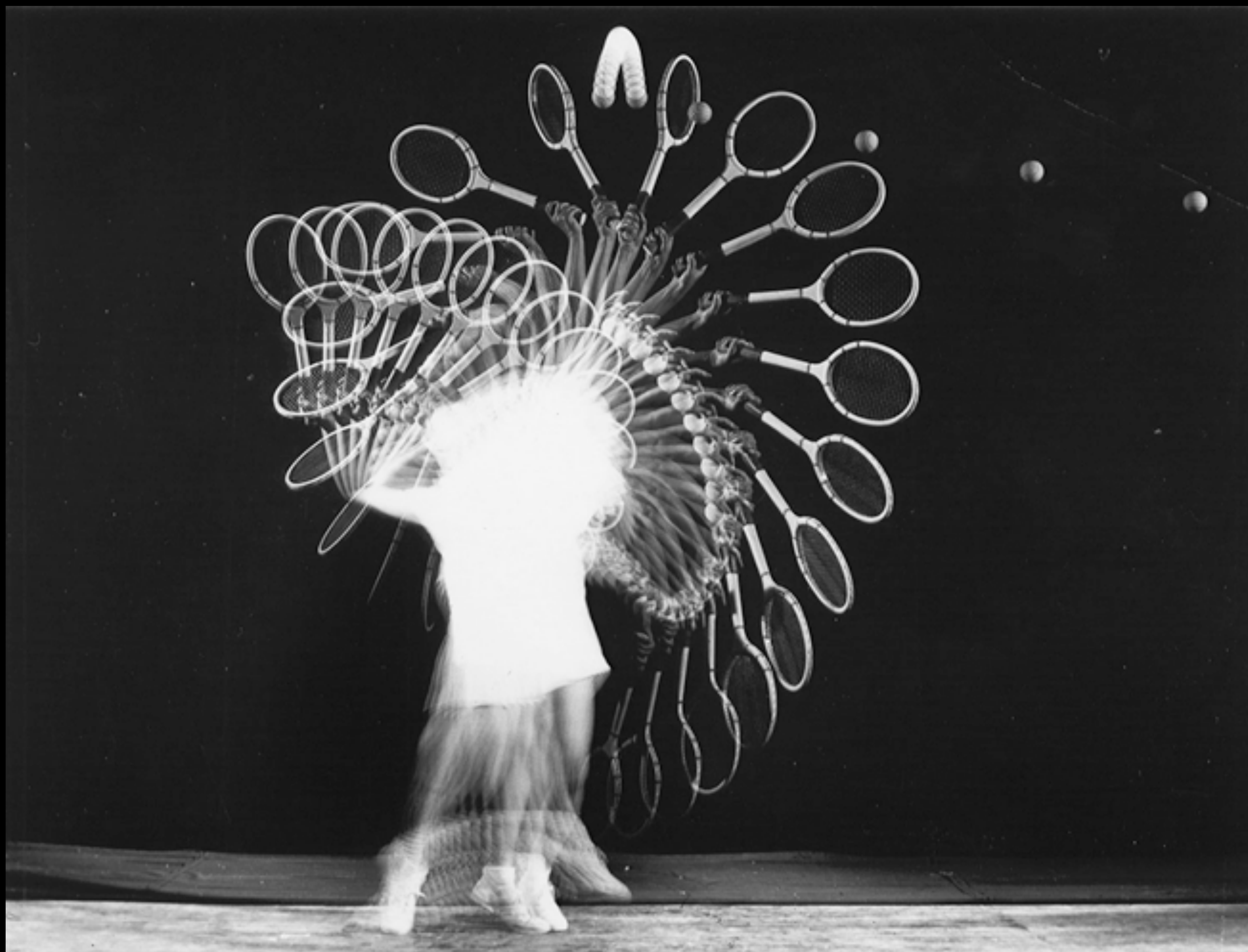


Sampling in time

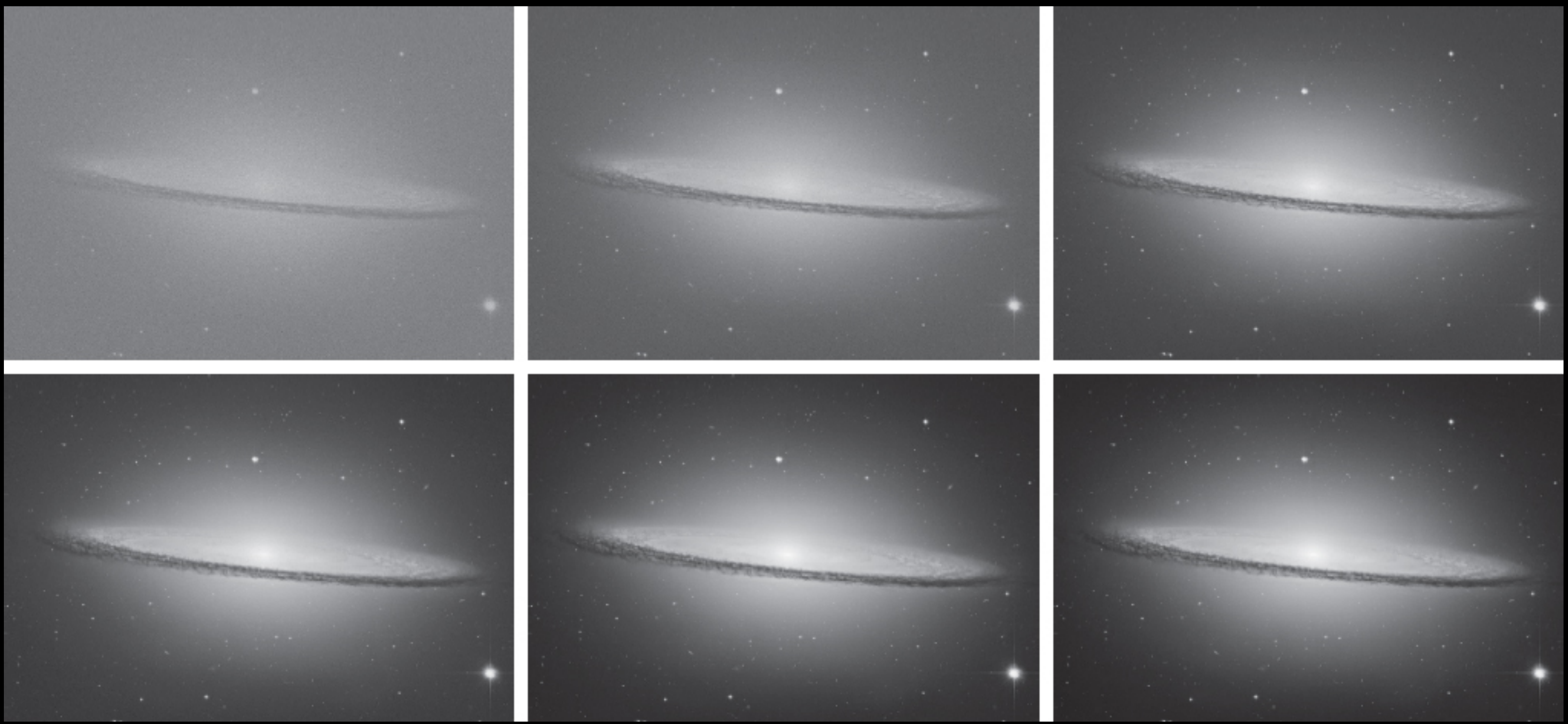
SPEED

Rolling Shutter/Global Shutter and Artifacts

Flash Strobe

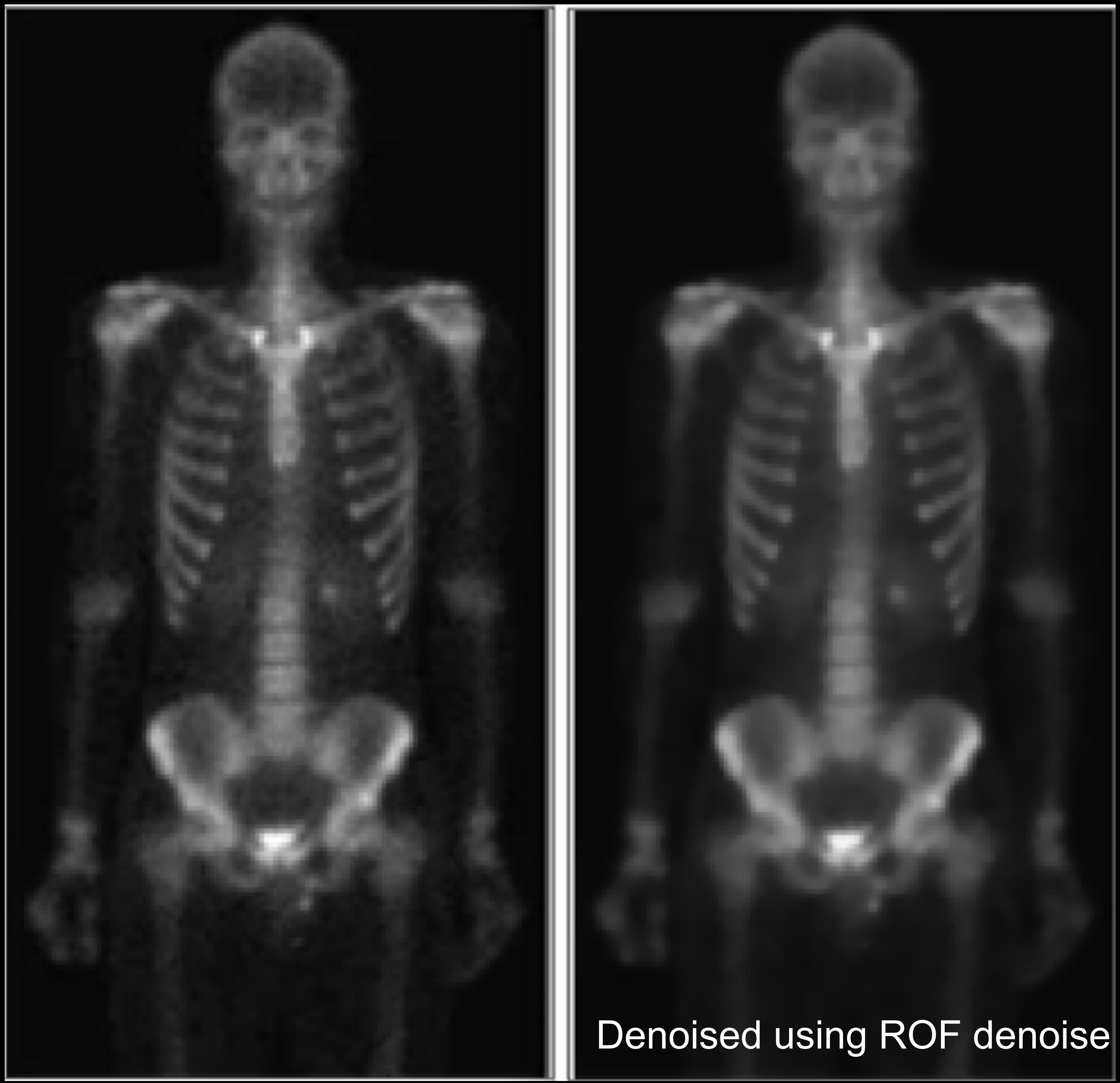


Average multiple images

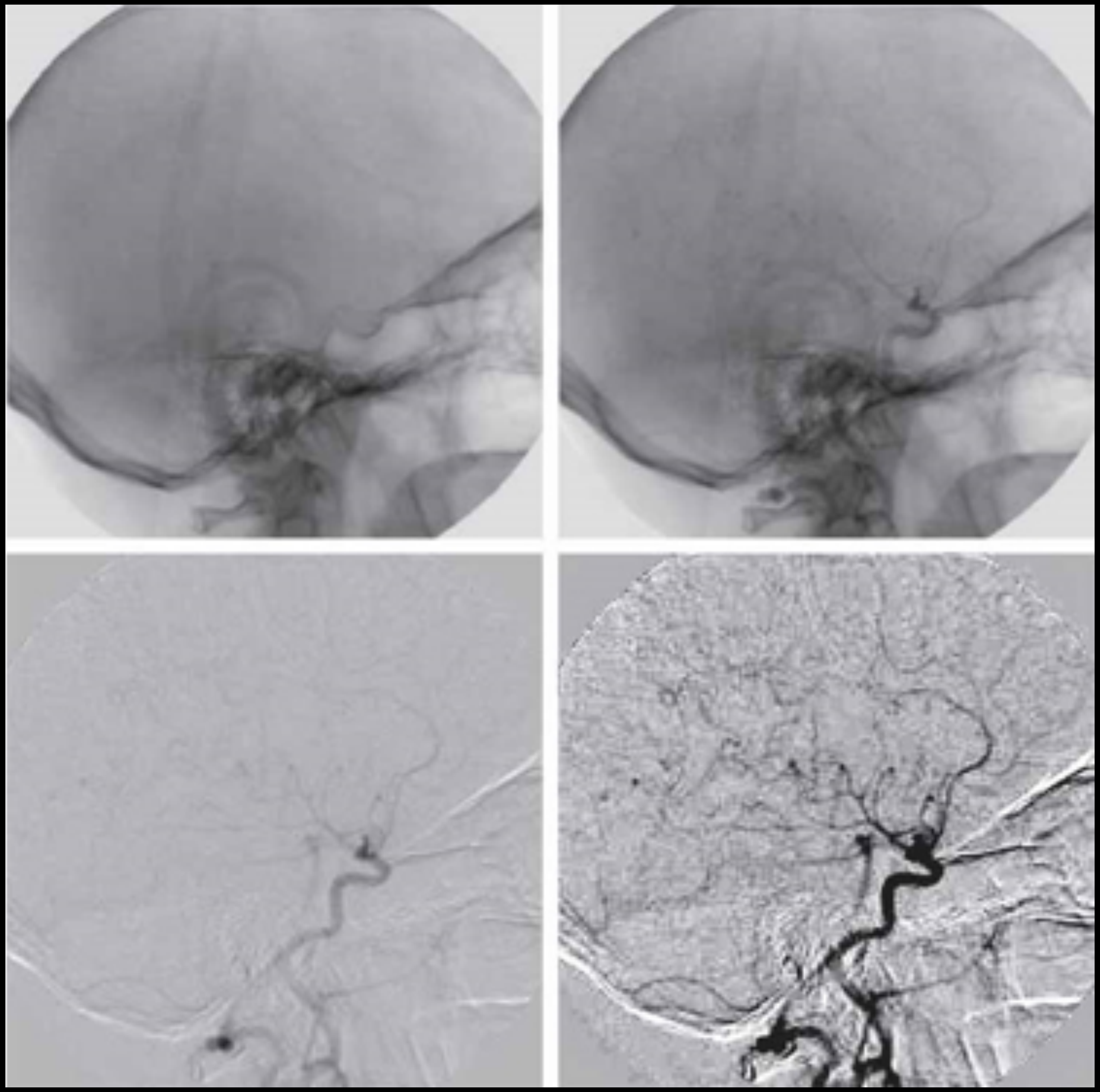


(a) Noisy image of the Sombrero Galaxy. (b)-(f) Result of averaging 10, 50, 100, 500, and 1,000 noisy images, respectively. All images are size 1548x2238 pixels and all scaled so intensities span the full [0, 255] intensity scale.

Computational Denoising



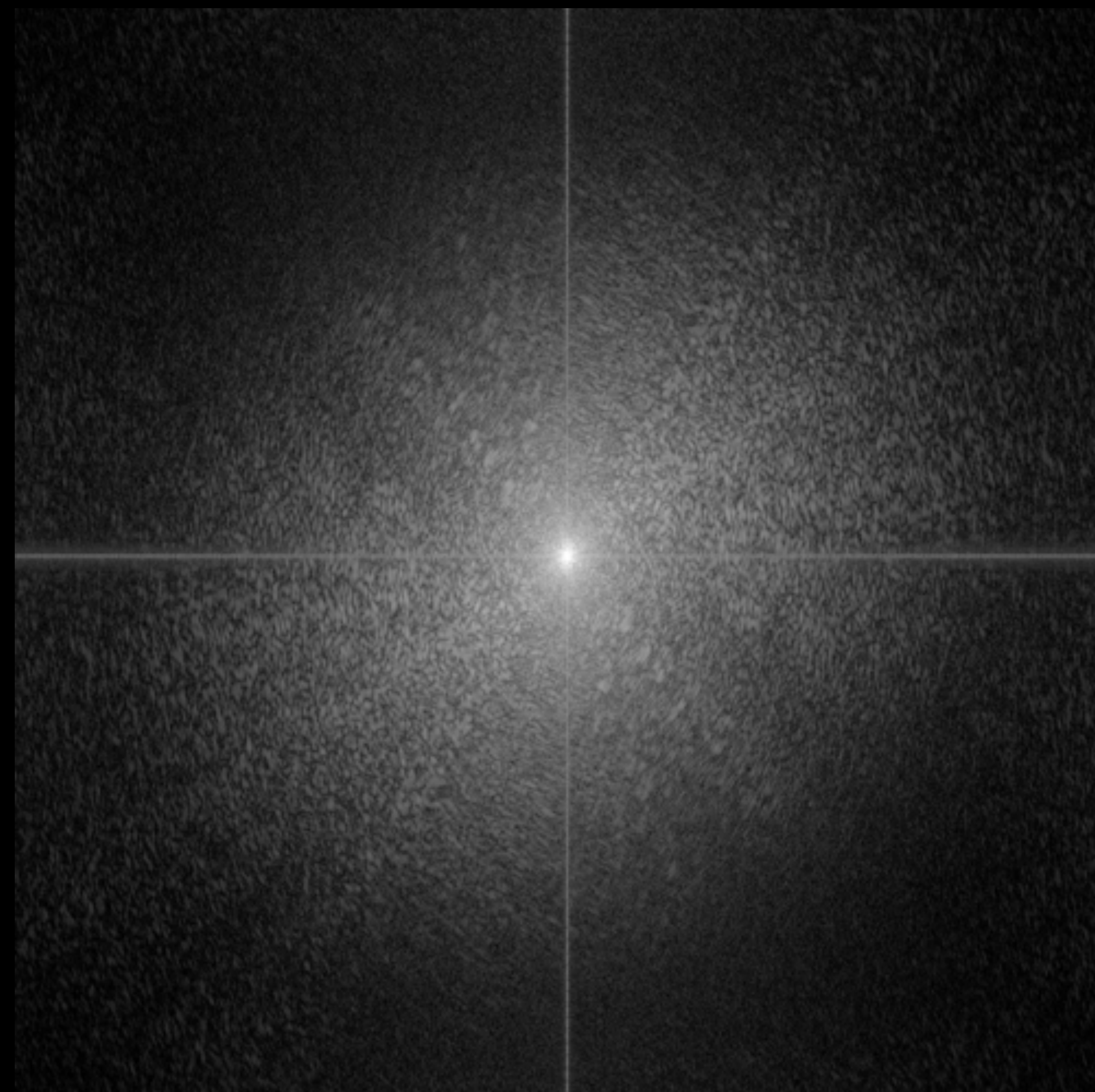
Digital Subtraction Angiography



Digital subtraction angiography.
(a) Mask image. (b) A live image.
(c) Difference between (a) and (b).
(d) Enhanced difference image.

Image courtesy of the Image Sciences Institute, University Medical Center, Netherlands (from our textbook: Digital Image Processing in Matlab)

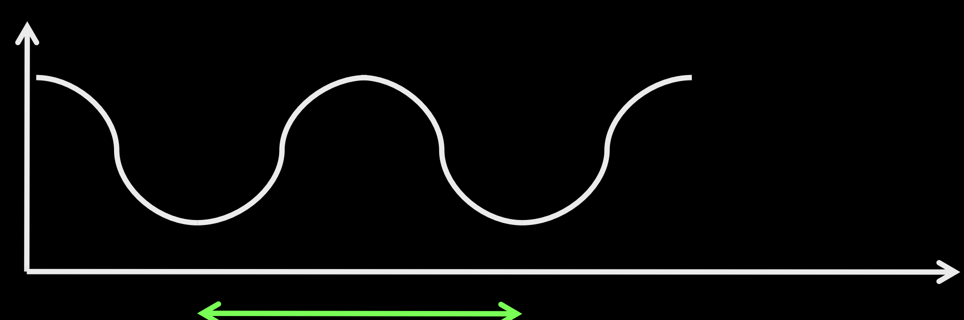
Filtering in Frequency (Fourier) Space

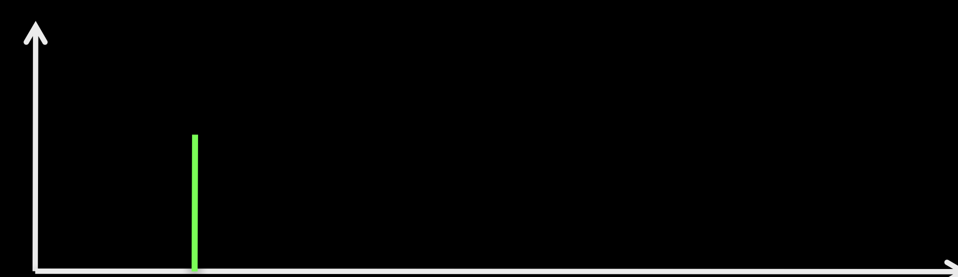


$$\mathcal{F}\{g(t)\} = G(f) = \int_{-\infty}^{\infty} g(t)e^{-i2\pi ft}\,dt$$

$$\mathcal{F}^{-1}\{G(f)\} = g(t) = \int_{-\infty}^{\infty} G(f)e^{i2\pi ft}\,df$$

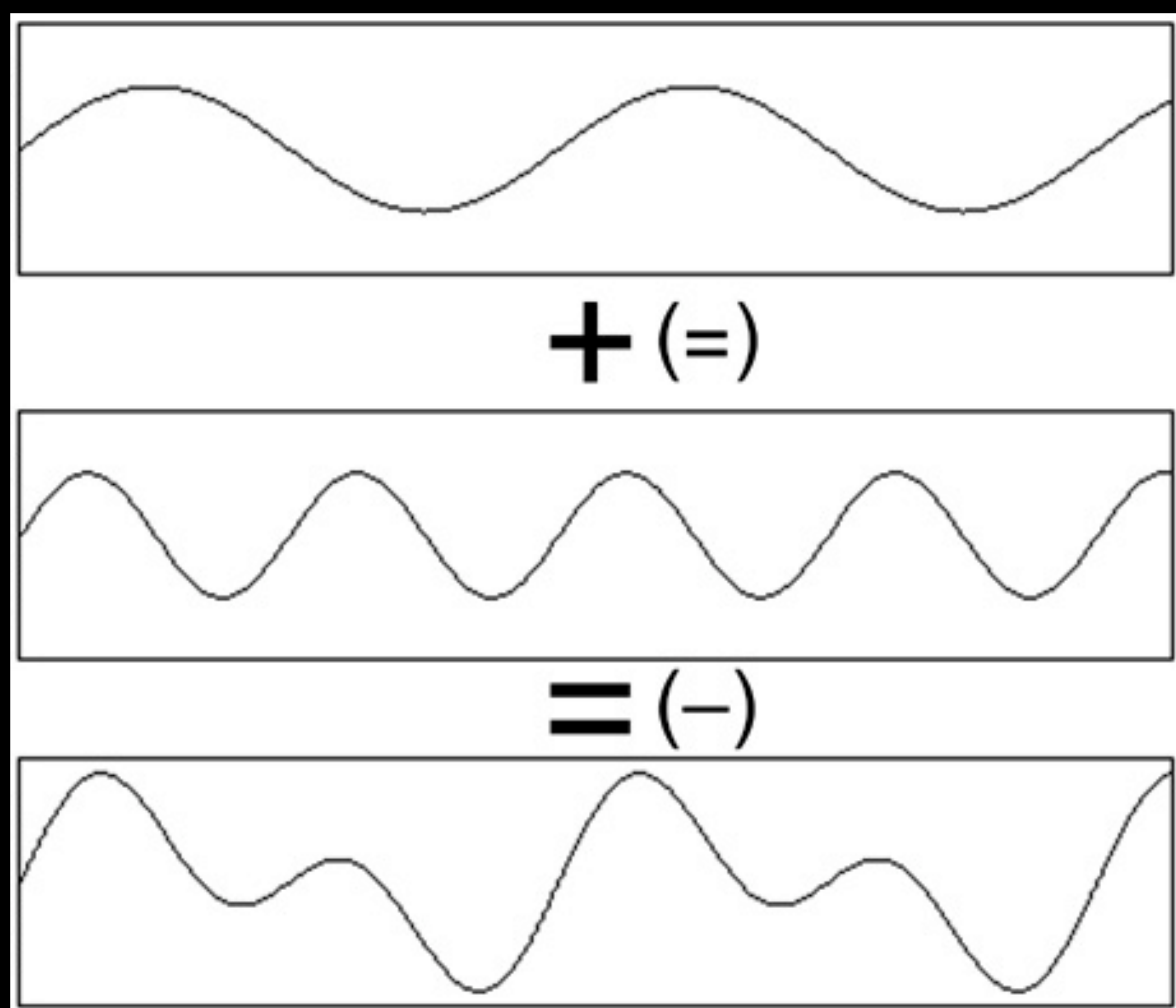
Time and space



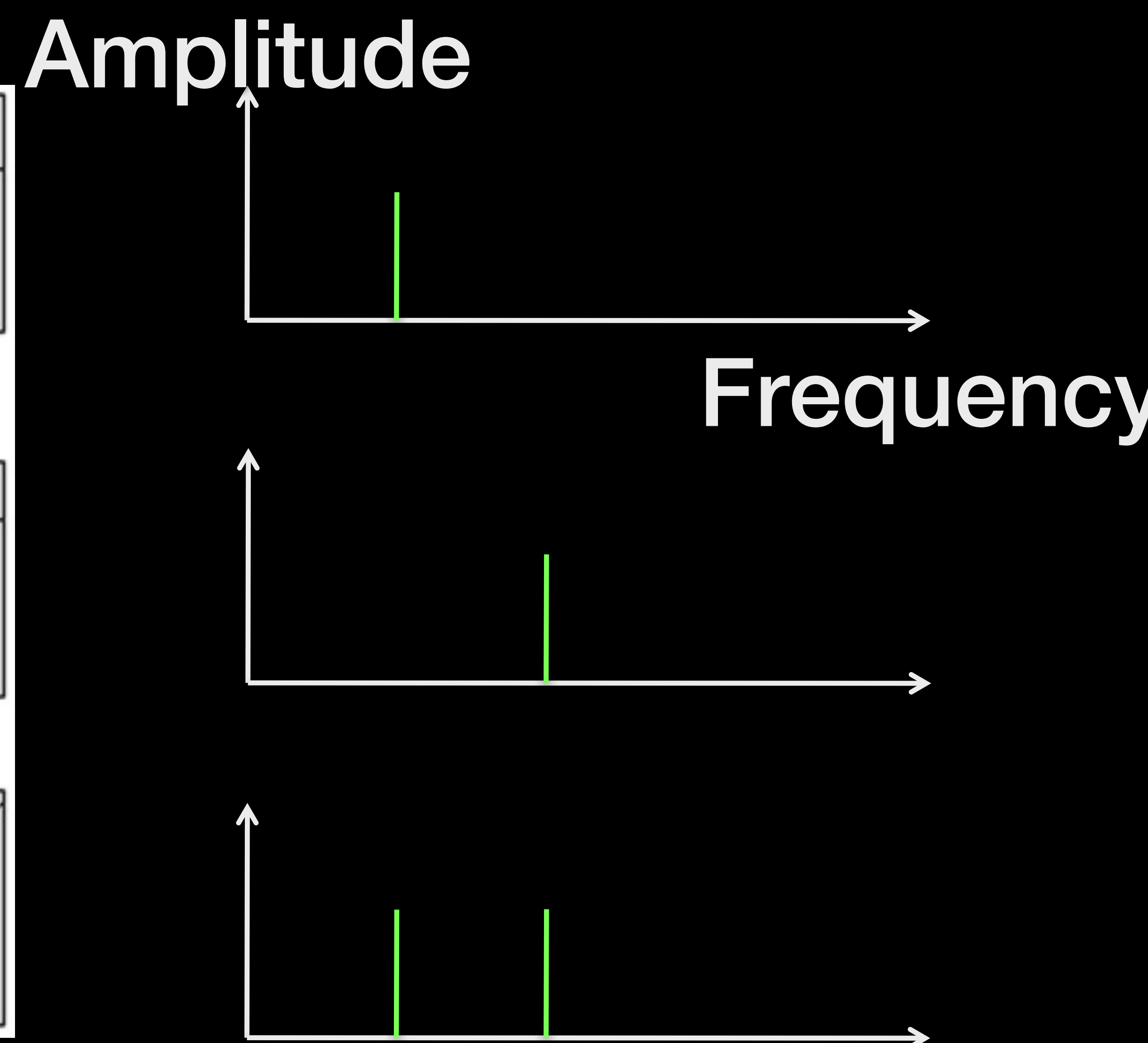


We can Fourier Transform back and forth

Fourier series

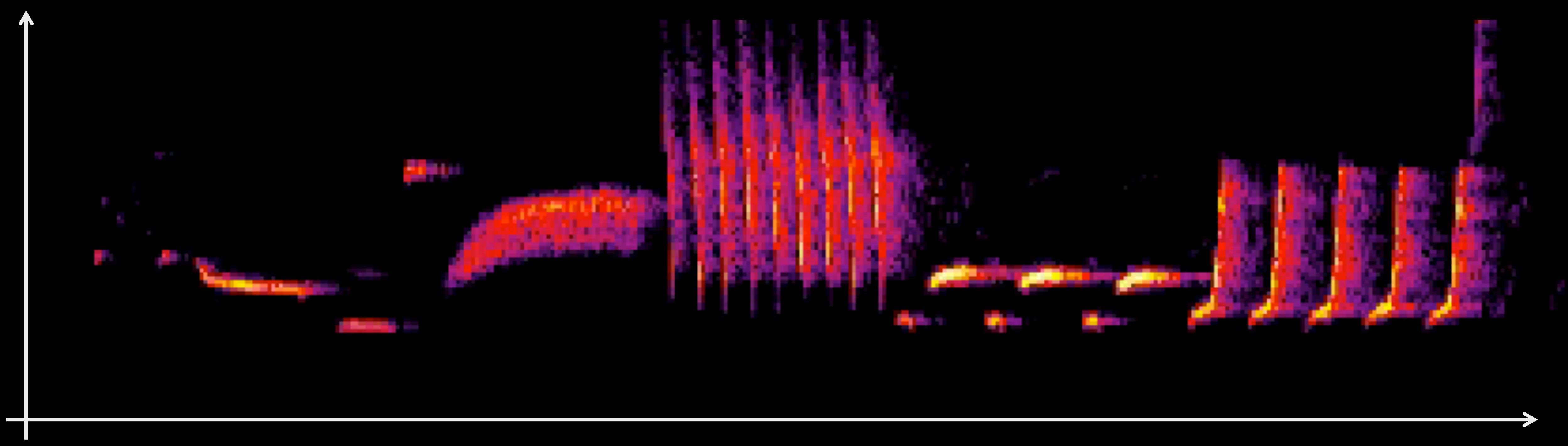


Pressure *vs* Time

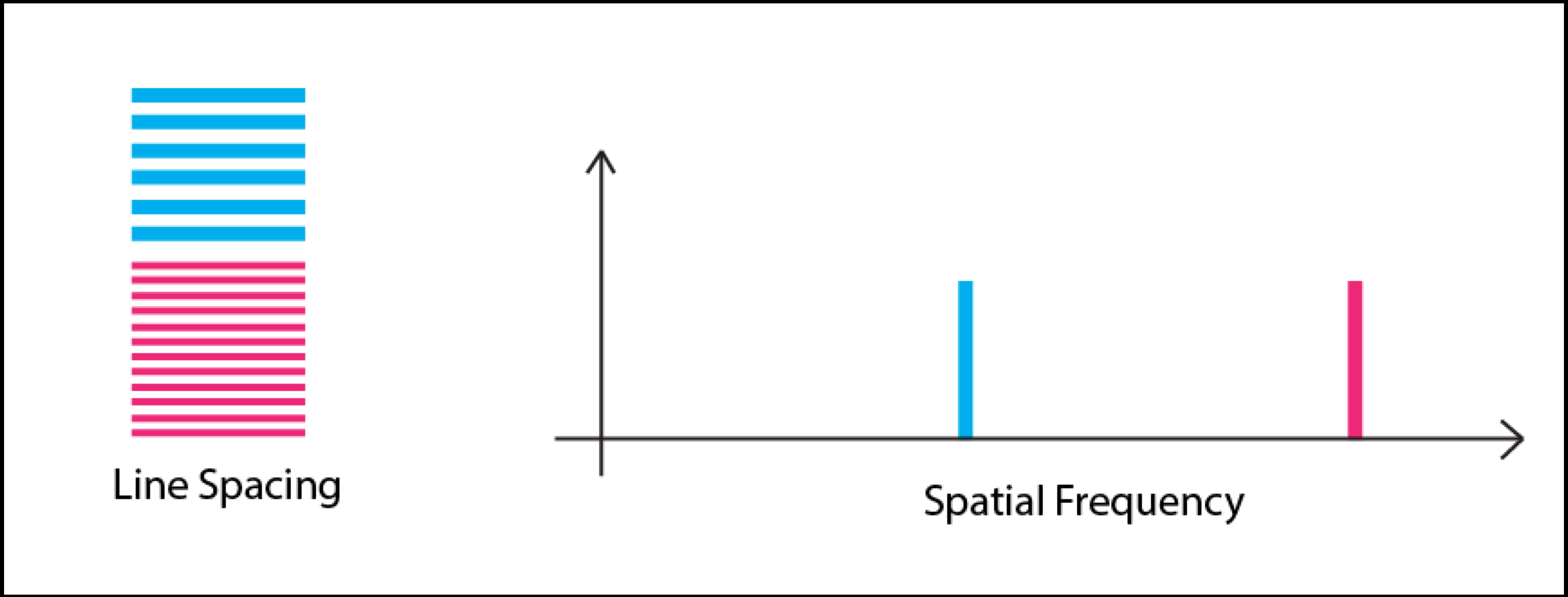


Amplitude *vs* Frequency

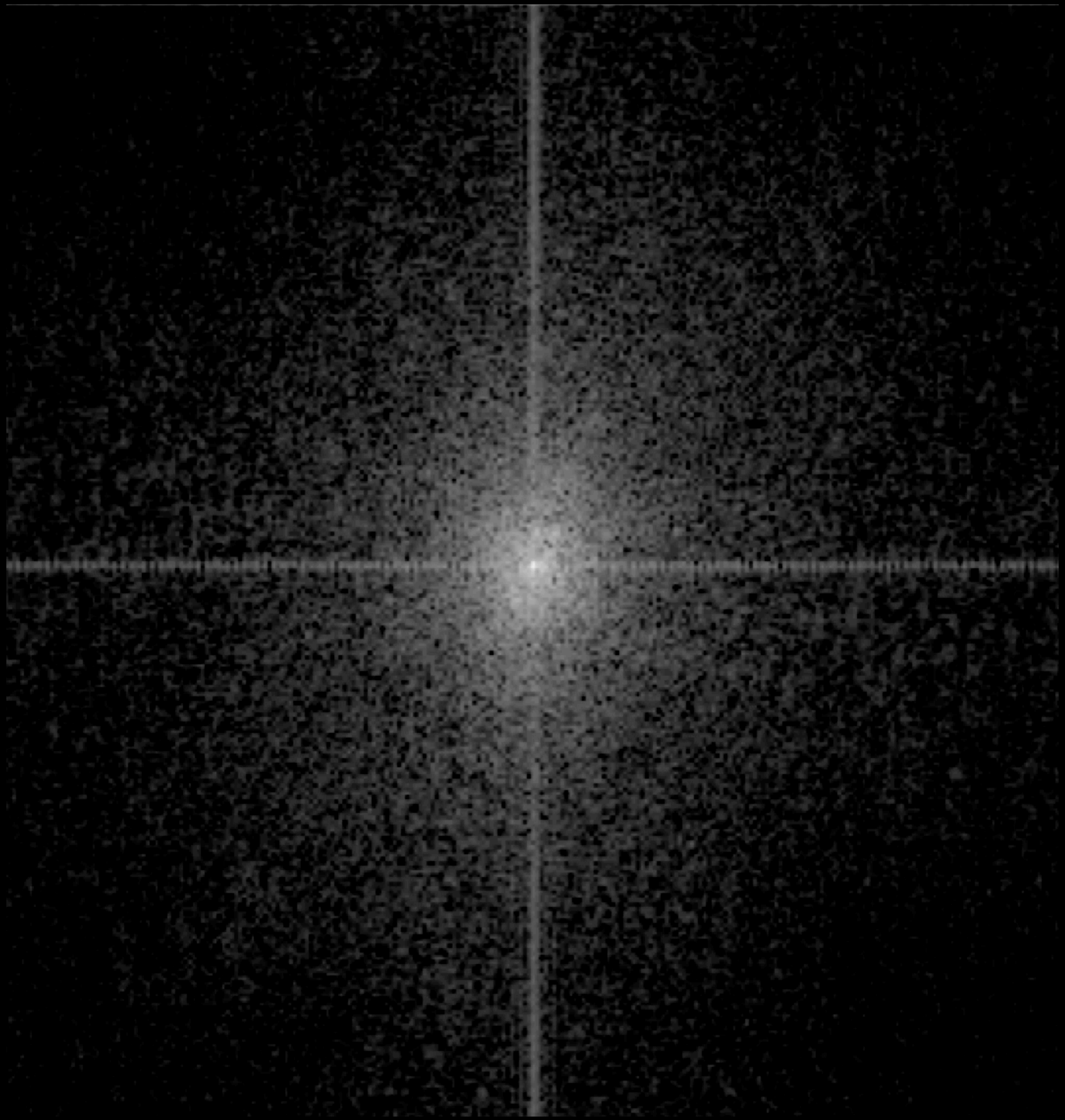
Short-time Fourier spectrogram



1D spatial frequency

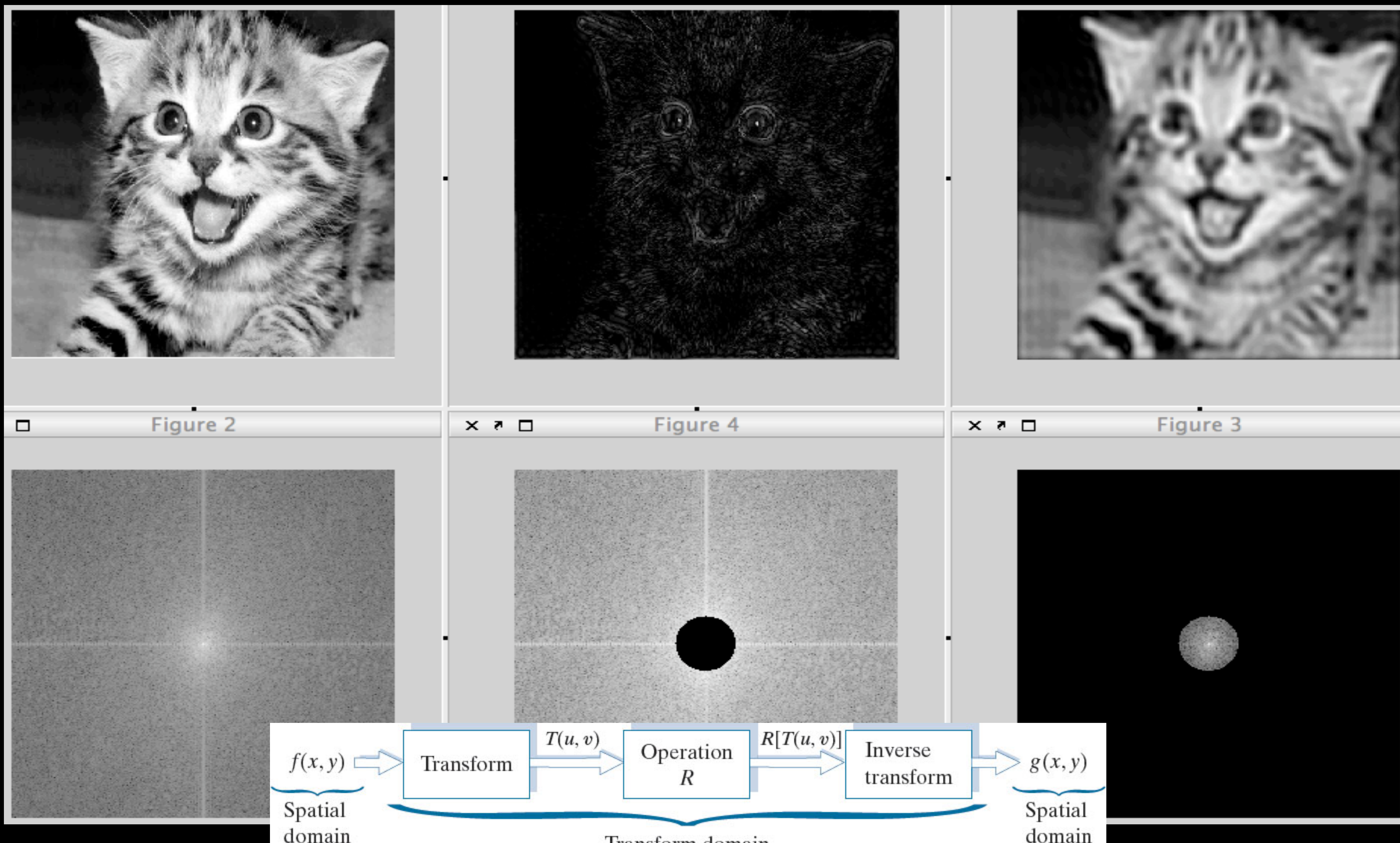


2D spatial frequency



$$y(k_1, k_2) = \iint f(x_1, x_2) \mathrm{e}^{-i2\pi(k_1 x_1 + k_2 x_2)} \, dx_1 dx_2$$

High and low-pass



Filtering out a single spatial frequency