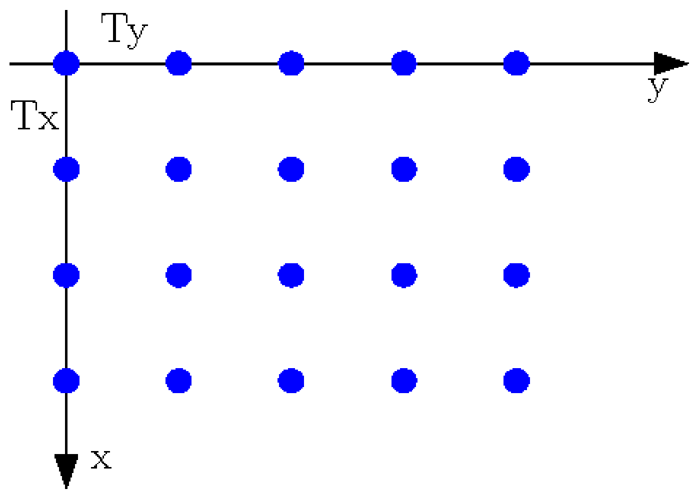
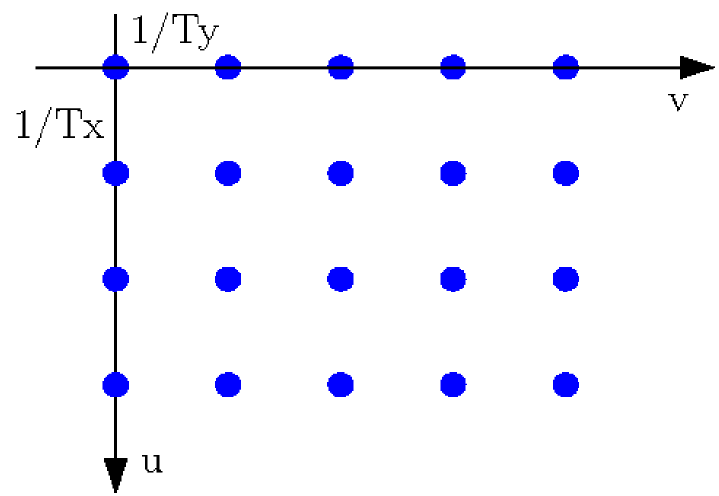
Data-domain image processing
Digital image **sampling** & re-sampling

The type of 2-D signal $f(x,y)$ we deal with is represented by a real function of two integer variables

$$f(x,y) = f(xT_x, yT_y) = f(x_a, y_a)\,|\, x_a = xT_x,\; y_a = yT_y$$



Data domain

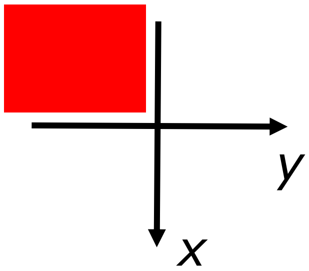


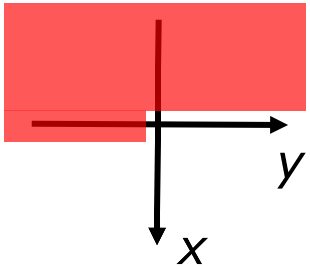
Frequency domain

Digital image sampling & re-sampling

- A 2-D **function** is **causal** if $f(x,y)=0$ for $(x<0$ and $y<0)$

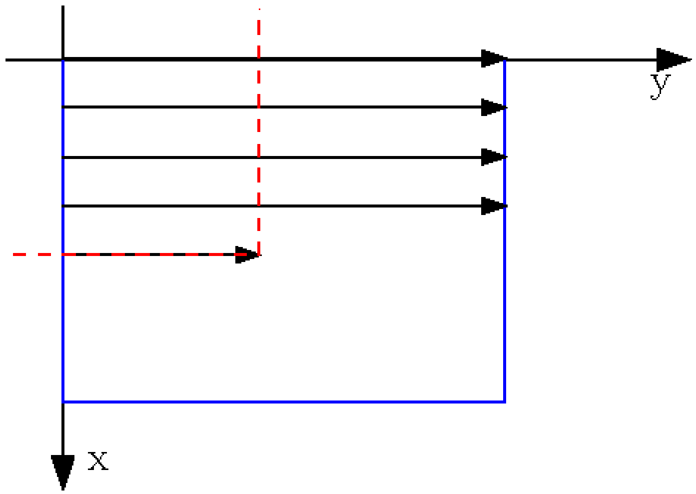
  It is **semicausal** if $f(x,y)=0$ for $x<0$, $(x=0$ and $y<0)$

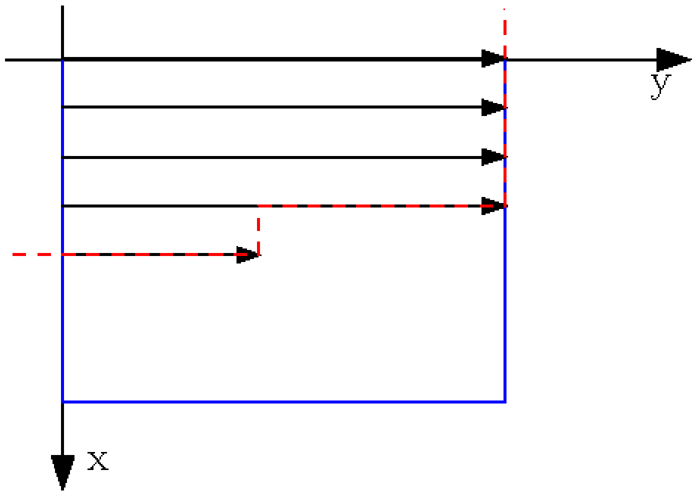




- A 2-D **processing system** is

causal or **semicausal**





if the "previous" input data are located left/above the dashed lines

Digital image sampling & re-sampling

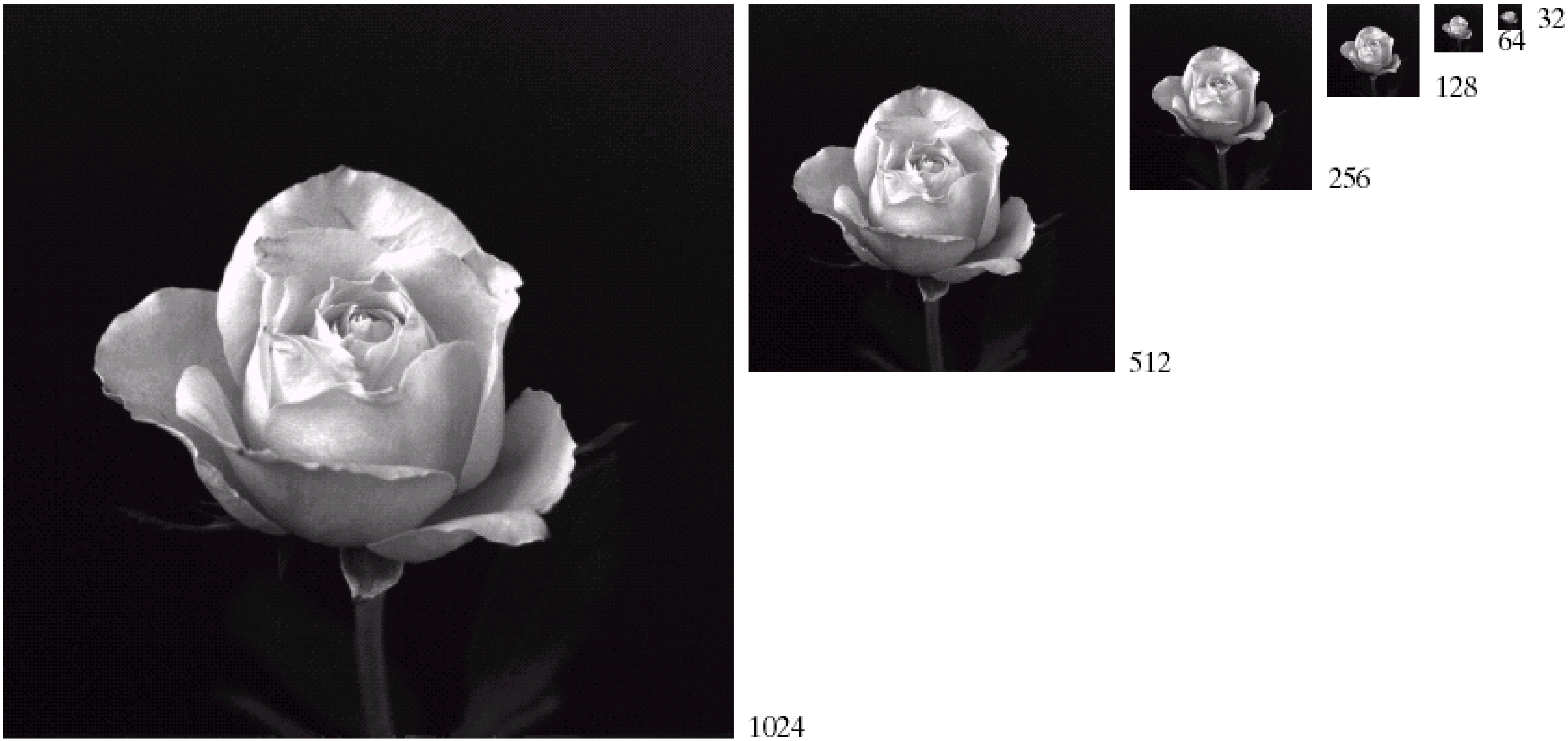
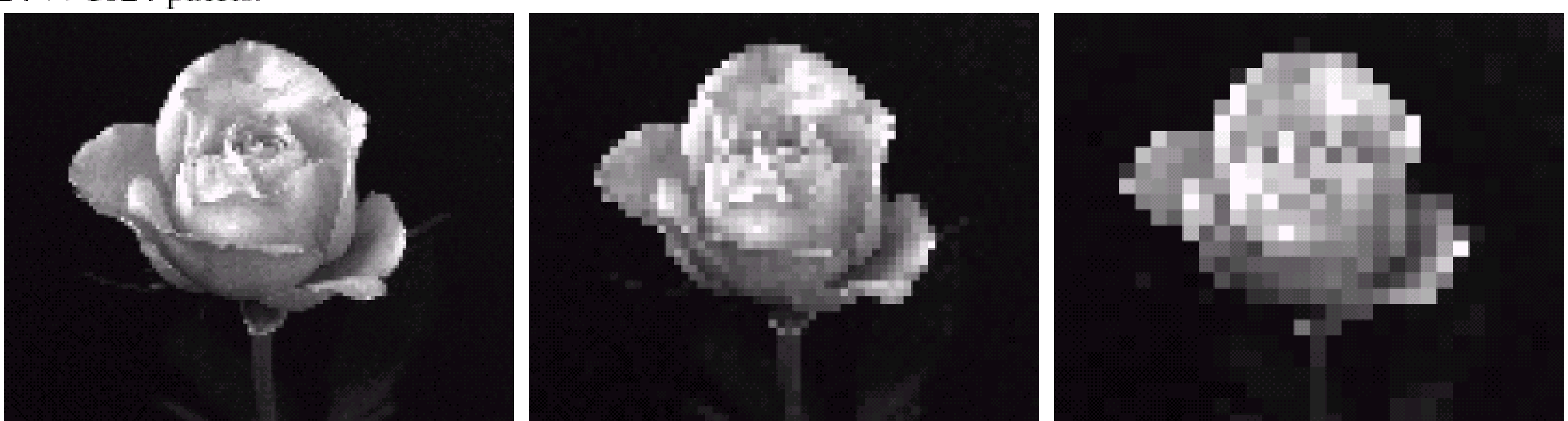


FIGURE 2.19 A 1024 × 1024, 8-bit image subsampled down to size 32 × 32 pixels. The number of allowable gray levels was kept at 256.

Digital image sampling & re-sampling

FIGURE 2.20 (a) 1024 × 1024, 8-bit image. (b) 512 × 512 image resampled into 1024 × 1024 pixels by row and column duplication. (c) through (f) 256 × 256, 128 × 128, 64 × 64, and 32 × 32 images resampled into 1024 × 1024 pixels.



Digital image sampling & re-sampling

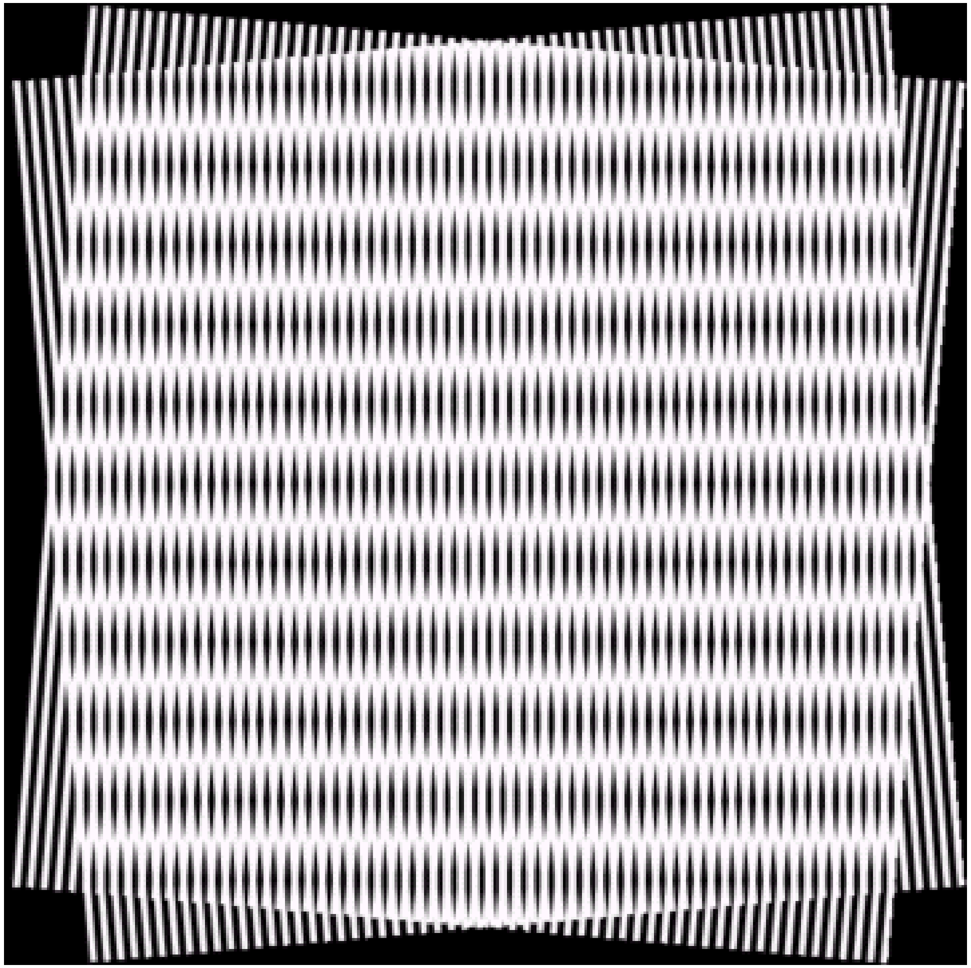
FIGURE 2.25 Top row: images zoomed from 128 × 128, 64 × 64, and 32 × 32 pixels to 1024 × 1024 pixels, using nearest neighbor gray-level interpolation. Bottom row: same sequence, but using bilinear interpolation.

Digital image sampling & re-sampling

a b c

FIGURE 4.17 Illustration of aliasing on resampled images. (a) A digital image with negligible visual aliasing. (b) Result of resizing the image to 50% of its original size by pixel deletion. Aliasing is clearly visible. (c) Result of blurring the image in (a) with a 3 × 3 averaging filter prior to resizing. The image is slightly more blurred than (b), but aliasing is not longer objectionable. (Original image courtesy of the Signal Compression Laboratory, University of California, Santa Barbara.)

Digital image sampling & re-sampling



'ation of the Moiré pattern effect.

The **Moiré pattern** is an **aliasing** effect, often visible on finely textured regions.

In this case it takes the form of a low-frequency vertically running sinusoid

Digital image **quantization**

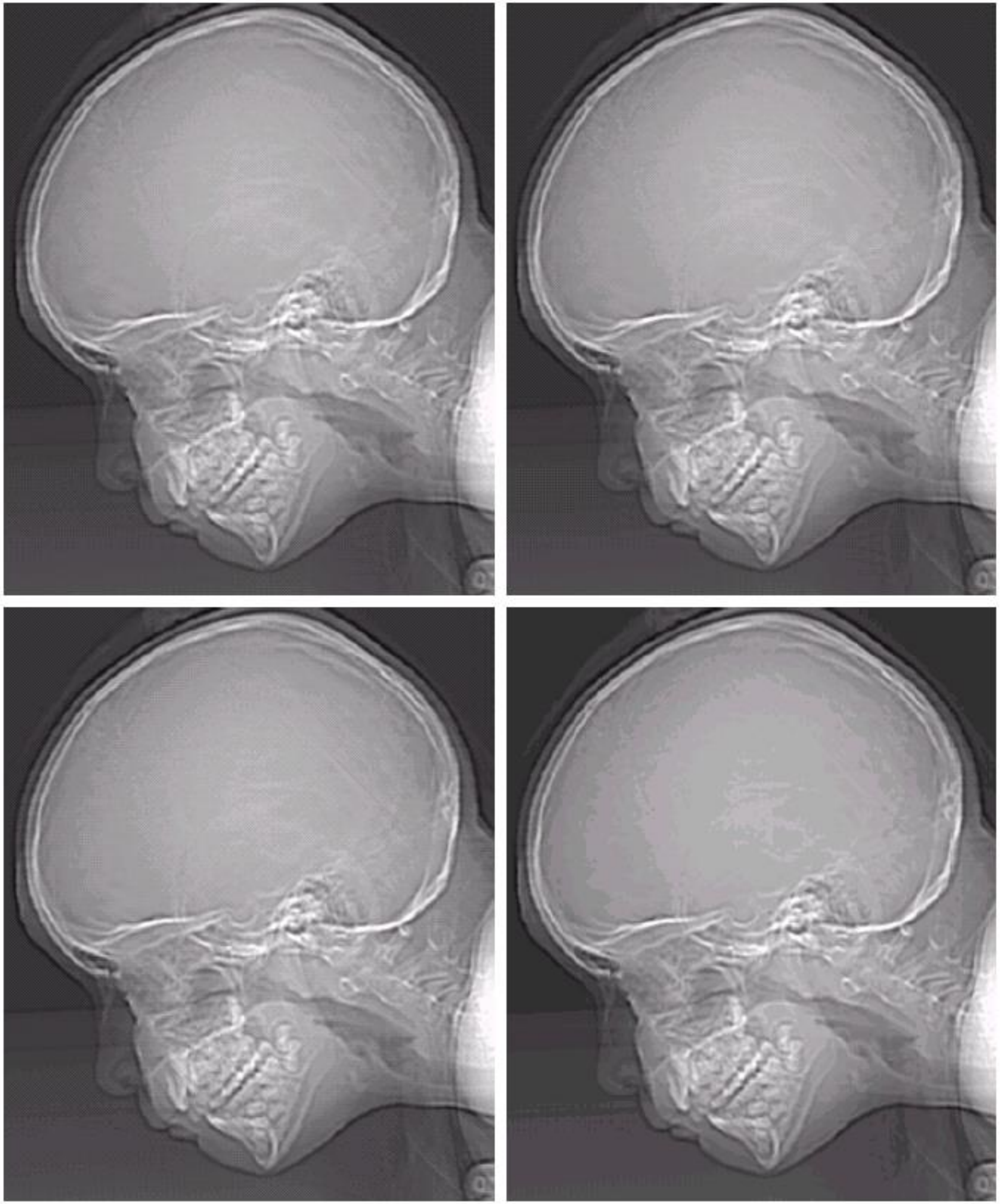


FIGURE 2.21
(a) 452 × 374, 256-level image. (b)–(d) Image displayed in 128, 64, and 32 gray levels, while keeping the spatial resolution constant.

Digital image quantization

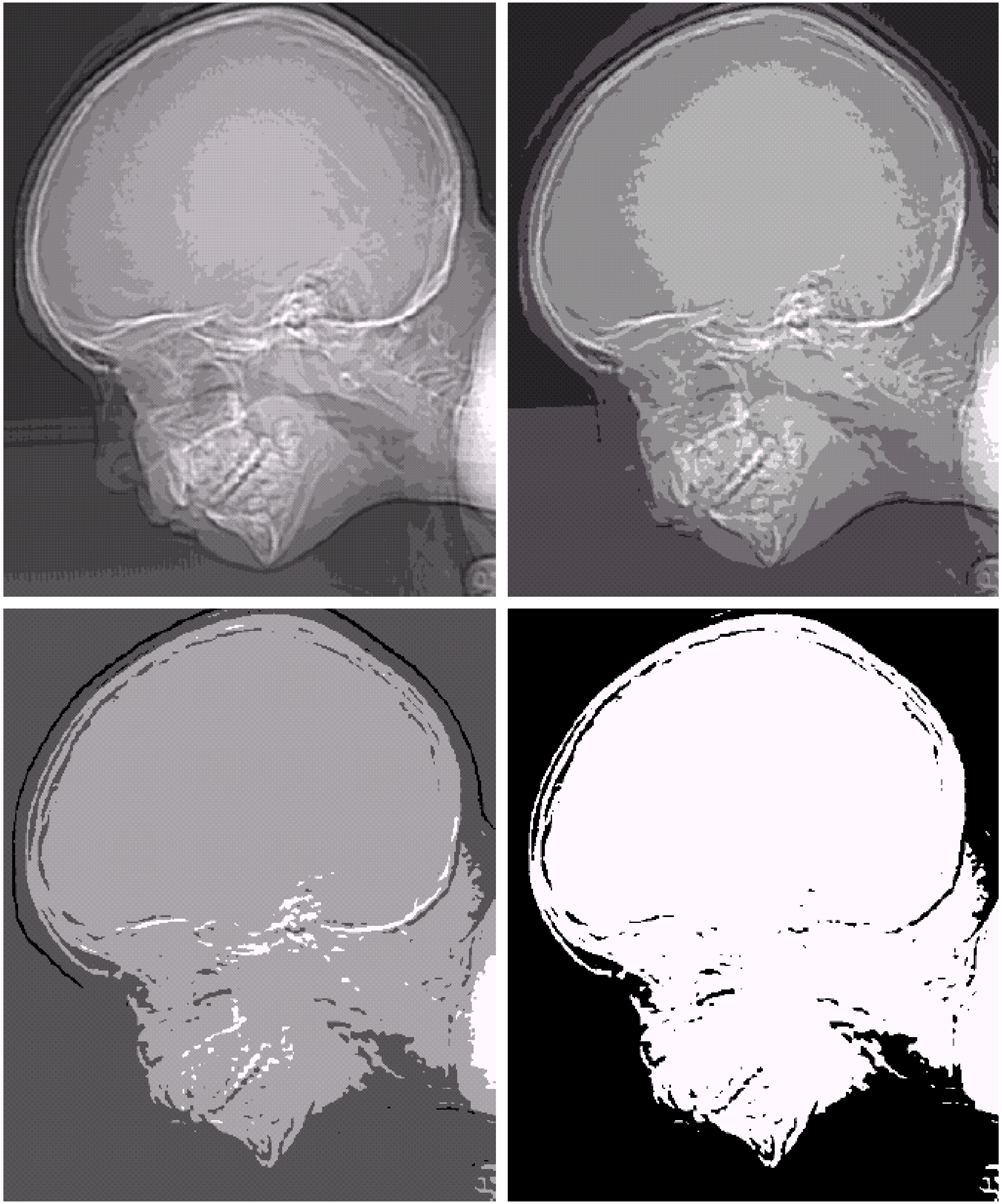


FIGURE 2.21
(Continued)
(e)–(h) Image displayed in 16, 8, 4, and 2 gray levels. (Original

Digital image quantization: dithering

Adding noise before quantization can improve the appearance of coarsely quantized images.

- The existence of pixels having amplitude close to the threshold(s) is made apparent.

Note: the binarization threshold is the same

See Matlab

- A *randomized version* of halftone printing.
- More visually effective algorithms exist
  (Floyd-Steinberg: error diffusion)

Image Enhancement in the Spatial Domain: Bit-plane slicing

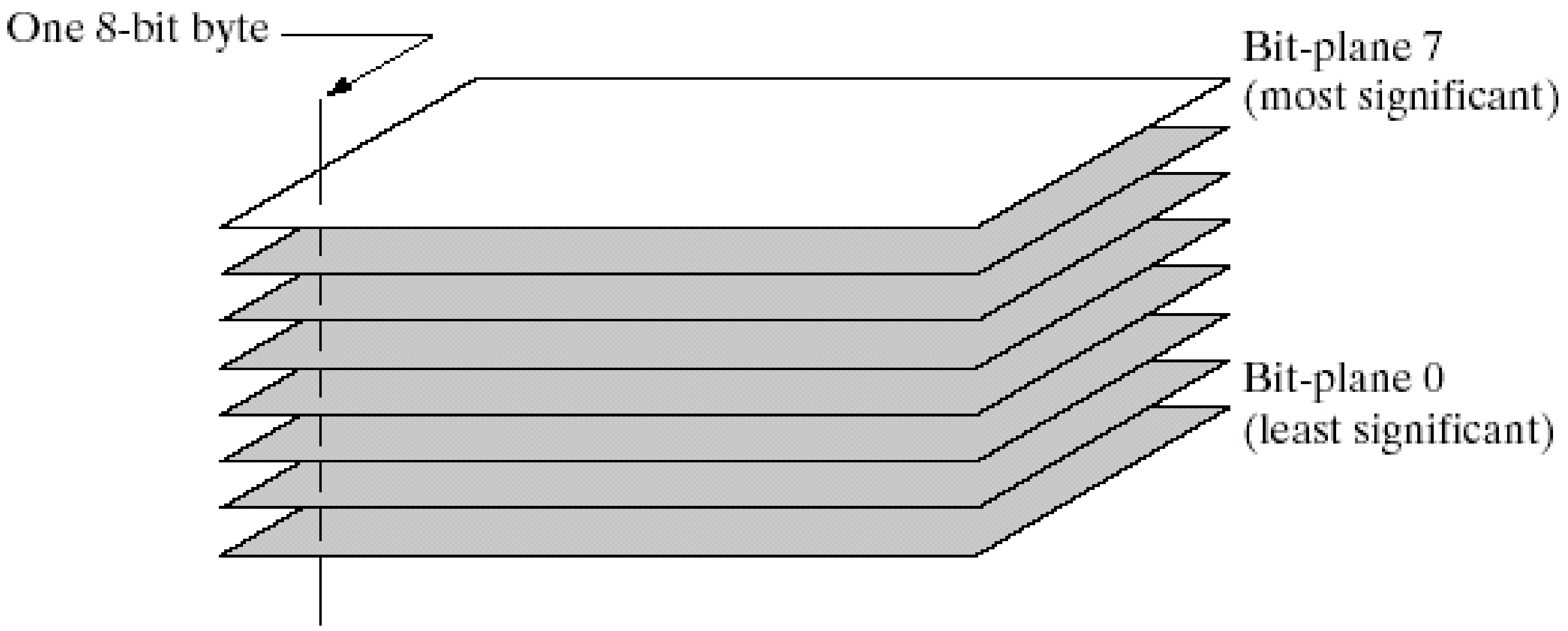


FIGURE 3.12
Bit-plane representation of an 8-bit image.

e.g.:

$$f(m,n) = 87_{10} = 01010111_2$$

Image Enhancement in the Spatial Domain: Bit-plane slicing

FIGURE 3.14 (a) An 8-bit gray-scale image of size 500 × 1192 pixels. (b) through (i) Bit planes 1 through 8, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image.

Image Enhancement in the Spatial Domain: Bit-plane slicing

FIGURE 3.15 Images reconstructed using (a) bit planes 8 and 7; (b) bit planes 8, 7, and 6; and (c) bit planes 8, 7, 6, and 5. Compare (c) with Fig. 3.14(a). 4, 8, 16 gray levels respectively

Reconstruction: Sum_n [bit-plane_n * 2^(n-1)]

Can be used for

- data compression
- implementation of order-statistics filters (see later)
- LSB steganography: hides information modifying the least significant bit of every pixel

→ Matlab dip05_0

Image Enhancement in the Spatial Domain: Gray-level transforms (Tone mapping)

Generic, possibly nonlinear, pointwise operator:

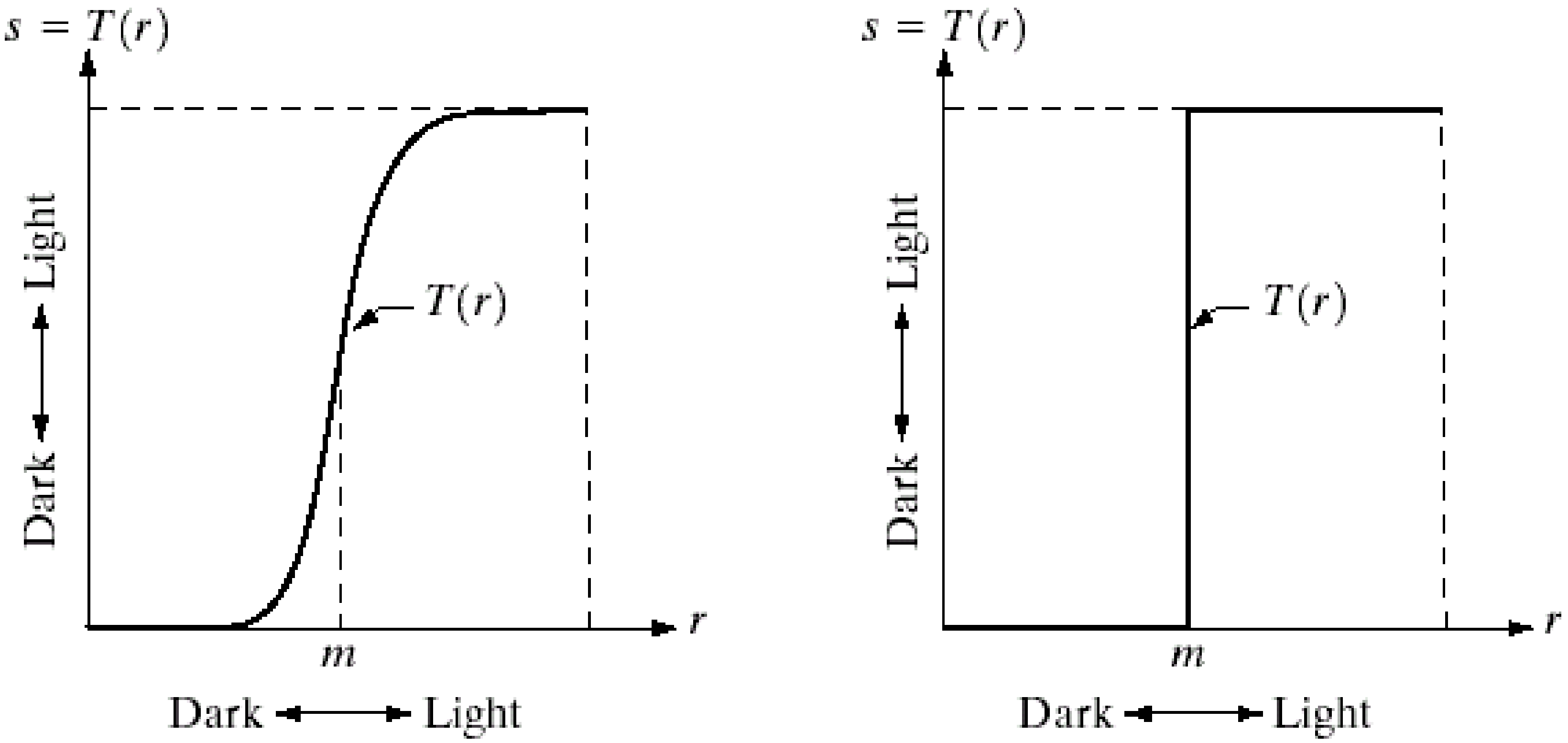
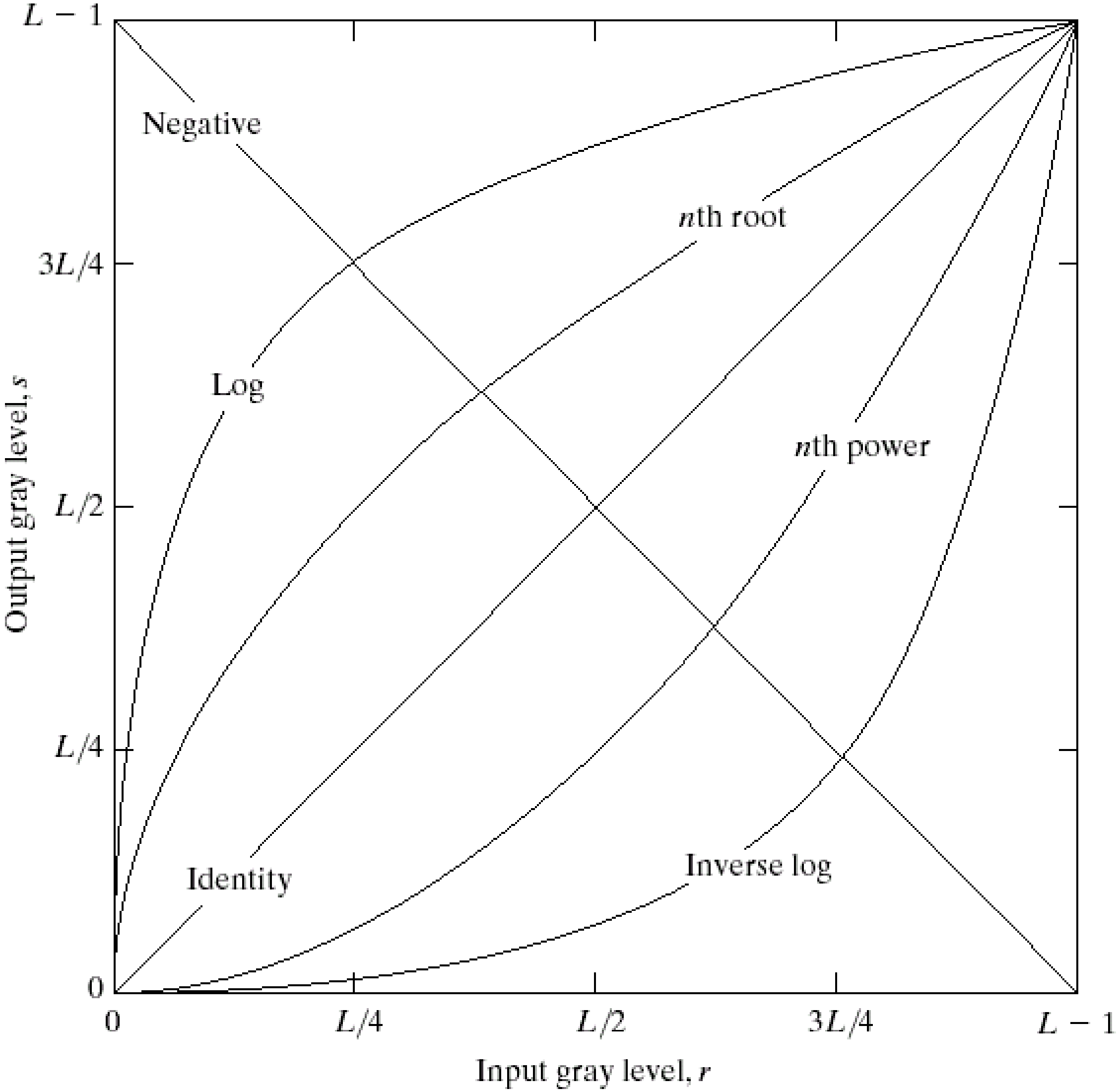


Image Enhancement in the Spatial Domain: Gray-level transforms



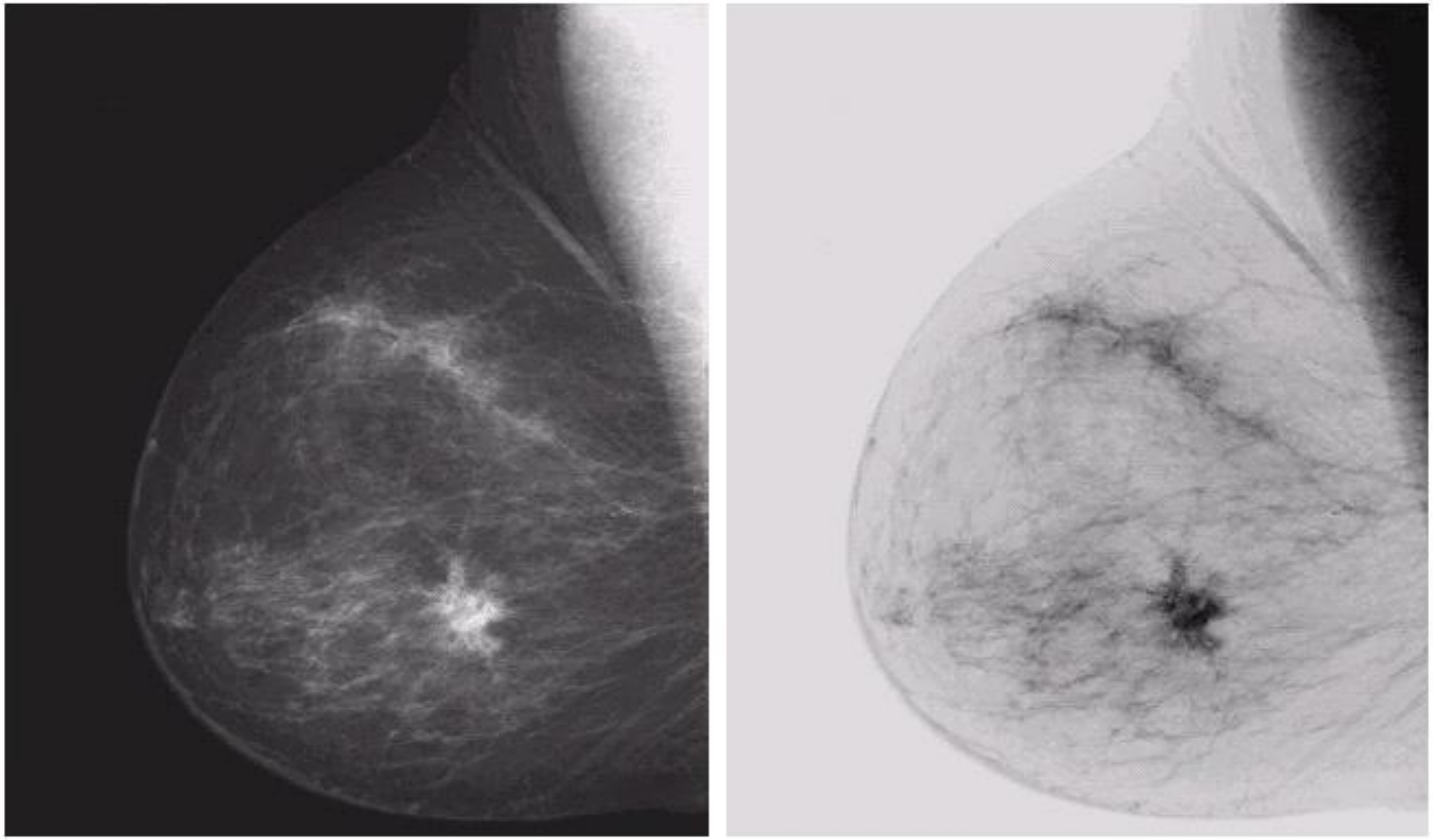
Basic gray-level transformations:

Negative: $s = L - 1 - r$

Generic log: $s = c \ln(1 + r)$

Power law: $s = c\, r^{\gamma}$

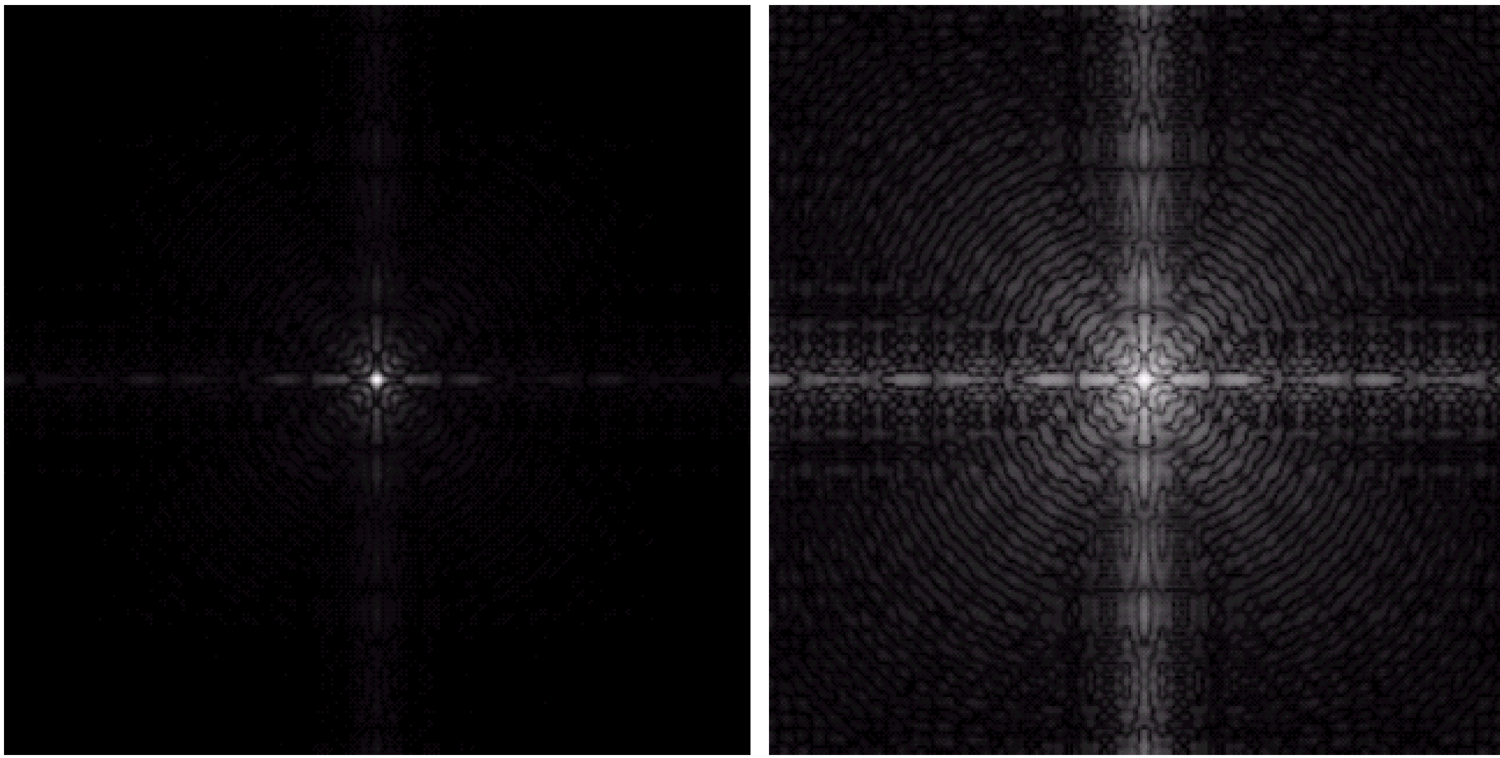
Image Enhancement in the Spatial Domain: Gray-level transforms



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.)

Image Enhancement in the Spatial Domain: Gray-level transforms



a b

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

Image Enhancement in the Spatial Domain: Gamma correction

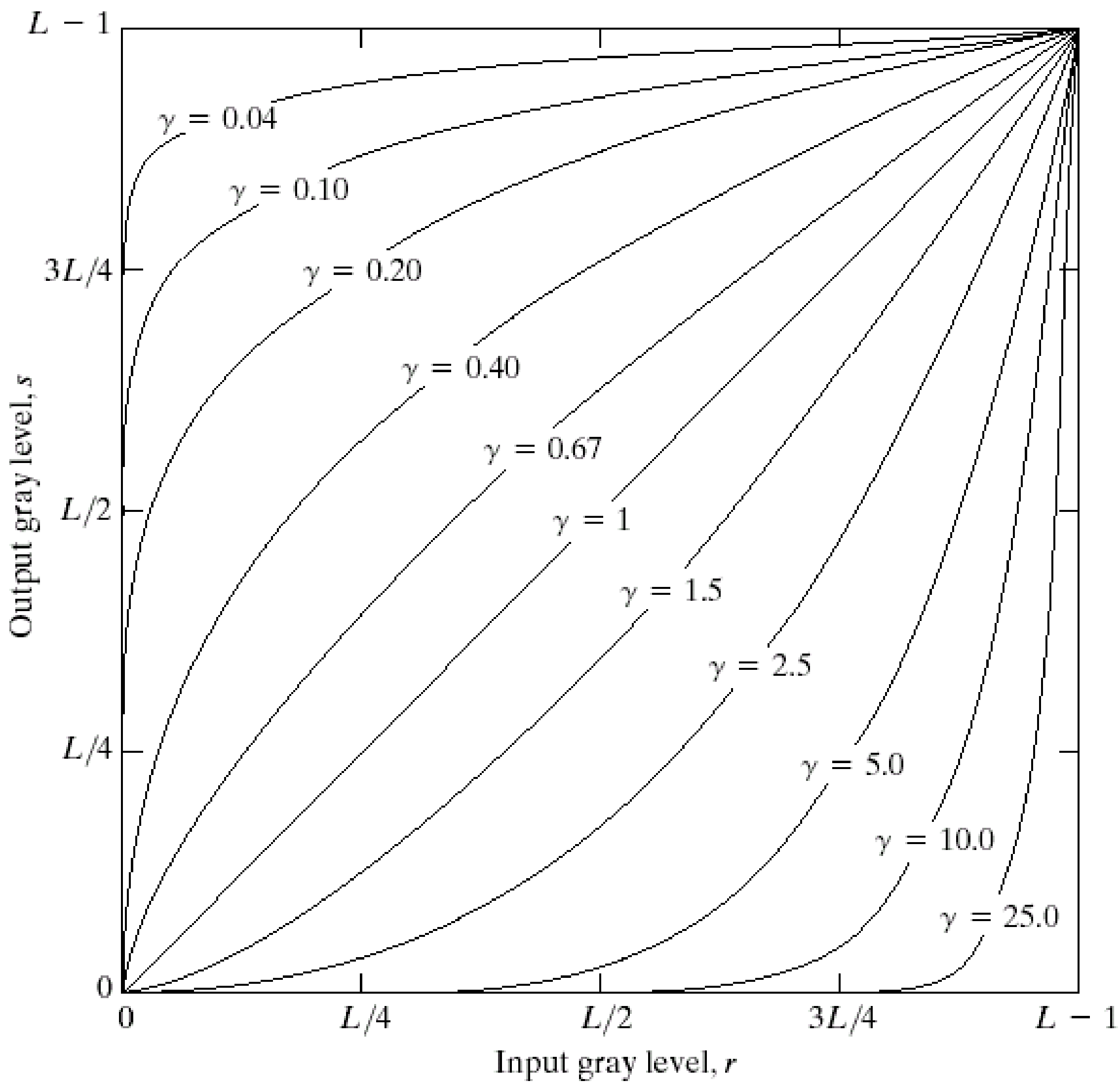
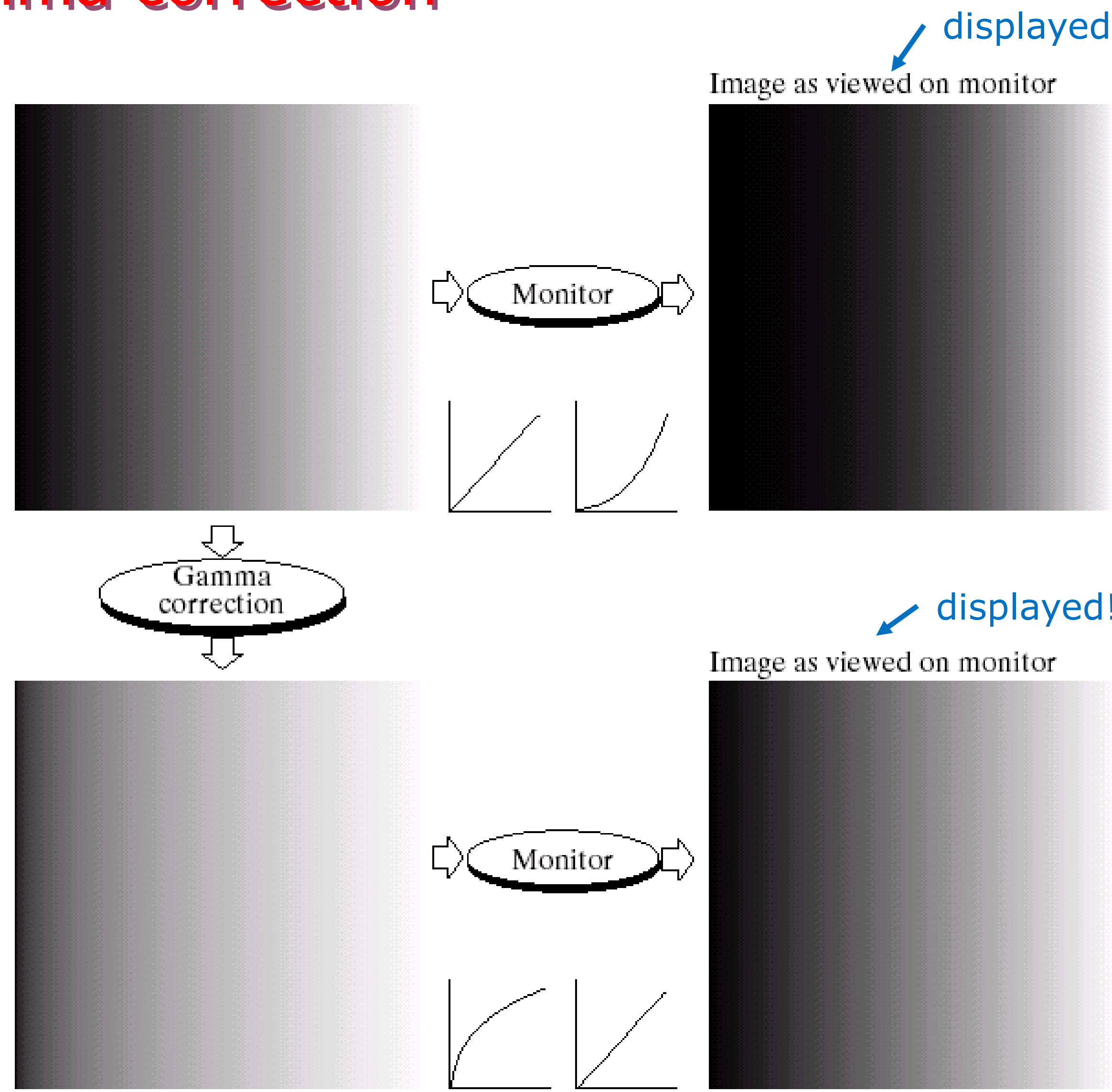


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ~~(c = 1 in all cases).~~

Image Enhancement in the Spatial Domain: Gamma correction

FIGURE 3.7
(a) Linear-wedge gray-scale image.
(b) Response of monitor to linear wedge.
(c) Gamma-corrected wedge.
(d) Output of monitor.

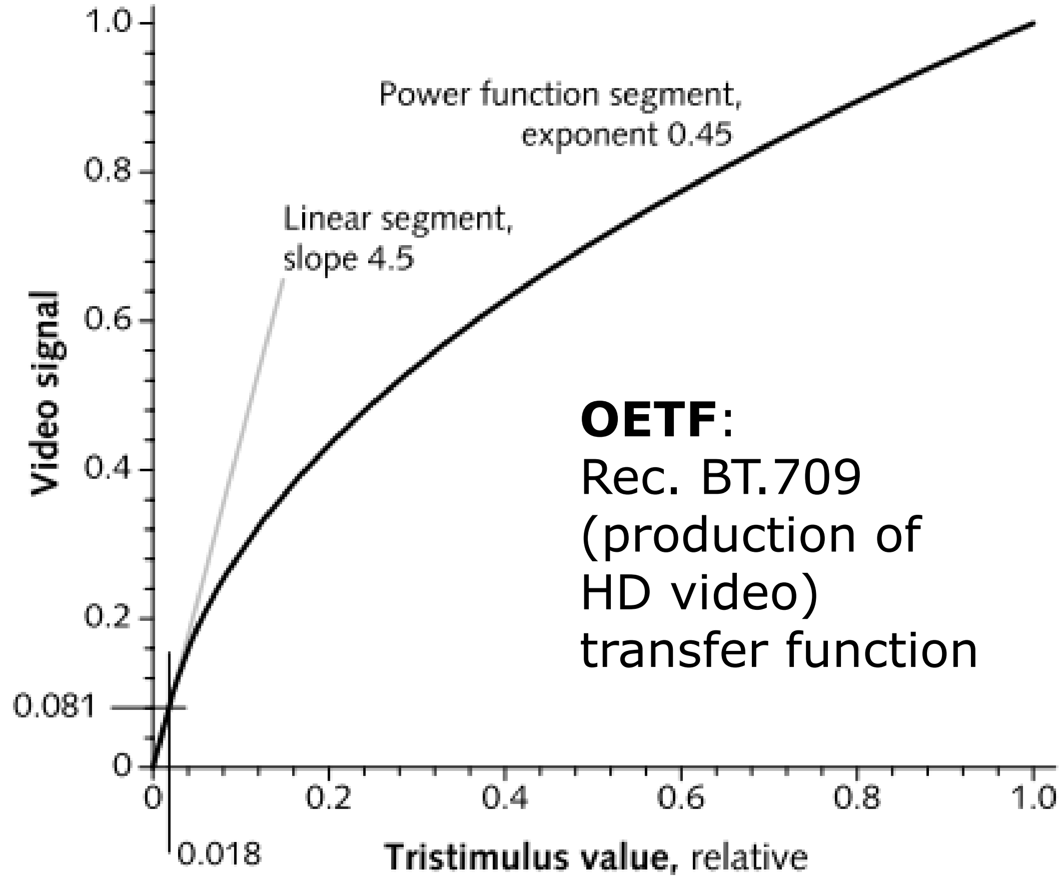


1) Monitor response can "compensate" for Weber-law sensitivity of HVS:
dp = k dL/L → p = k log(L)
higher sensit. in dark areas
→ dark transitions can be compressed with power law
L = x^gamma (e.g. 2.4)
1a) ...provided quantization errors are not incurred
2) Beware of nonlinearities that are already included in image data (e.g., JPEG)

Image Enhancement in the Spatial Domain: Gamma correction

"The fact that a CRT’s transfer function is very nearly the inverse of the lightness sensitivity of vision is an amazing, and fortunate, coincidence!"
(Charles Poynton)

Modern displays replicate the CRT's luminance response.
Rec. ITU-R BT.1886 (2011) states that 2.4-power **EOTF** shall be standard for HD content creation. Consumer displays are expected to conform.

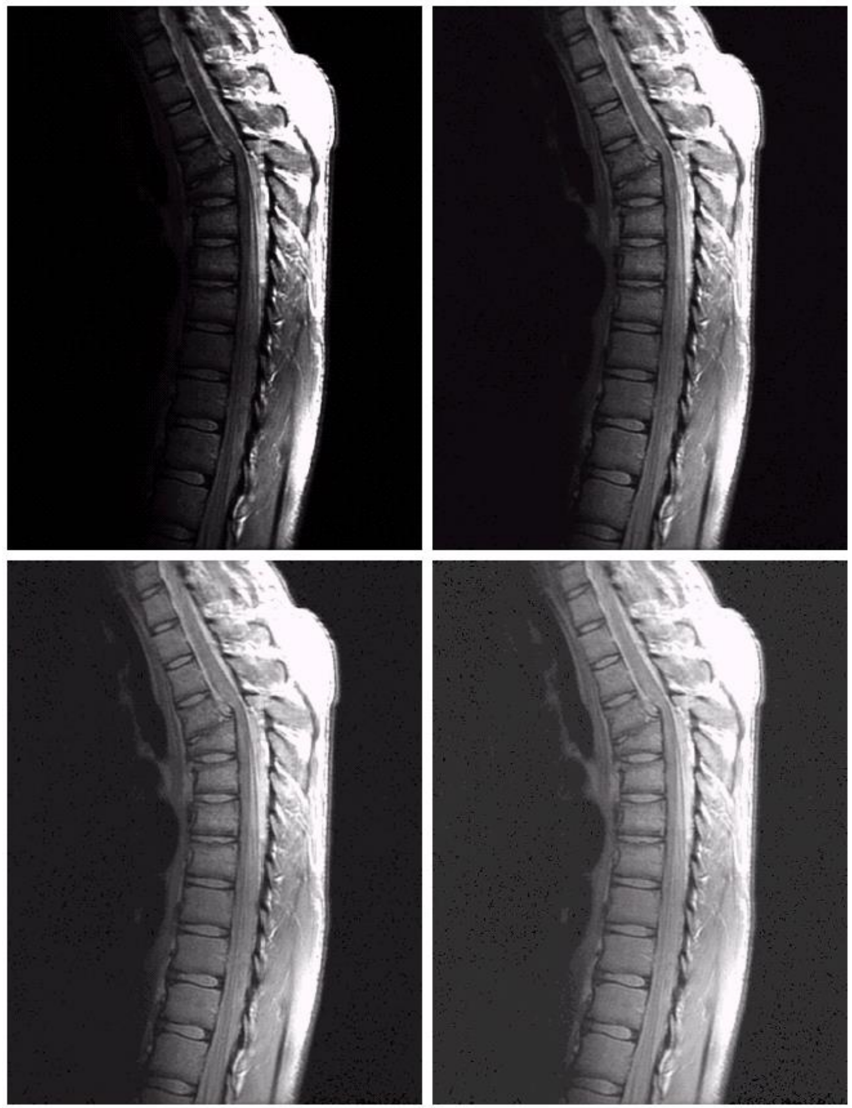


OETF:
Rec. BT.709 (production of HD video) transfer function

CIE primaries: X,Y,Z

the slope of the function is limited near zero (linear portion) in order to minimize quantization noise in the dark regions of the picture.

Image Enhancement in the Spatial Domain: Gray-level transforms



a	b
c	d

FIGURE 3.8 (a) Magnetic resonance (MR) image of a fractured human spine. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with ~~$c = 1$ and~~ $\gamma = 0.6, 0.4,$ and 0.3, respectively.

Image Enhancement in the Spatial Domain: Gray-level transforms

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.

Image Enhancement in the Spatial Domain: Piece-wise linear mapping

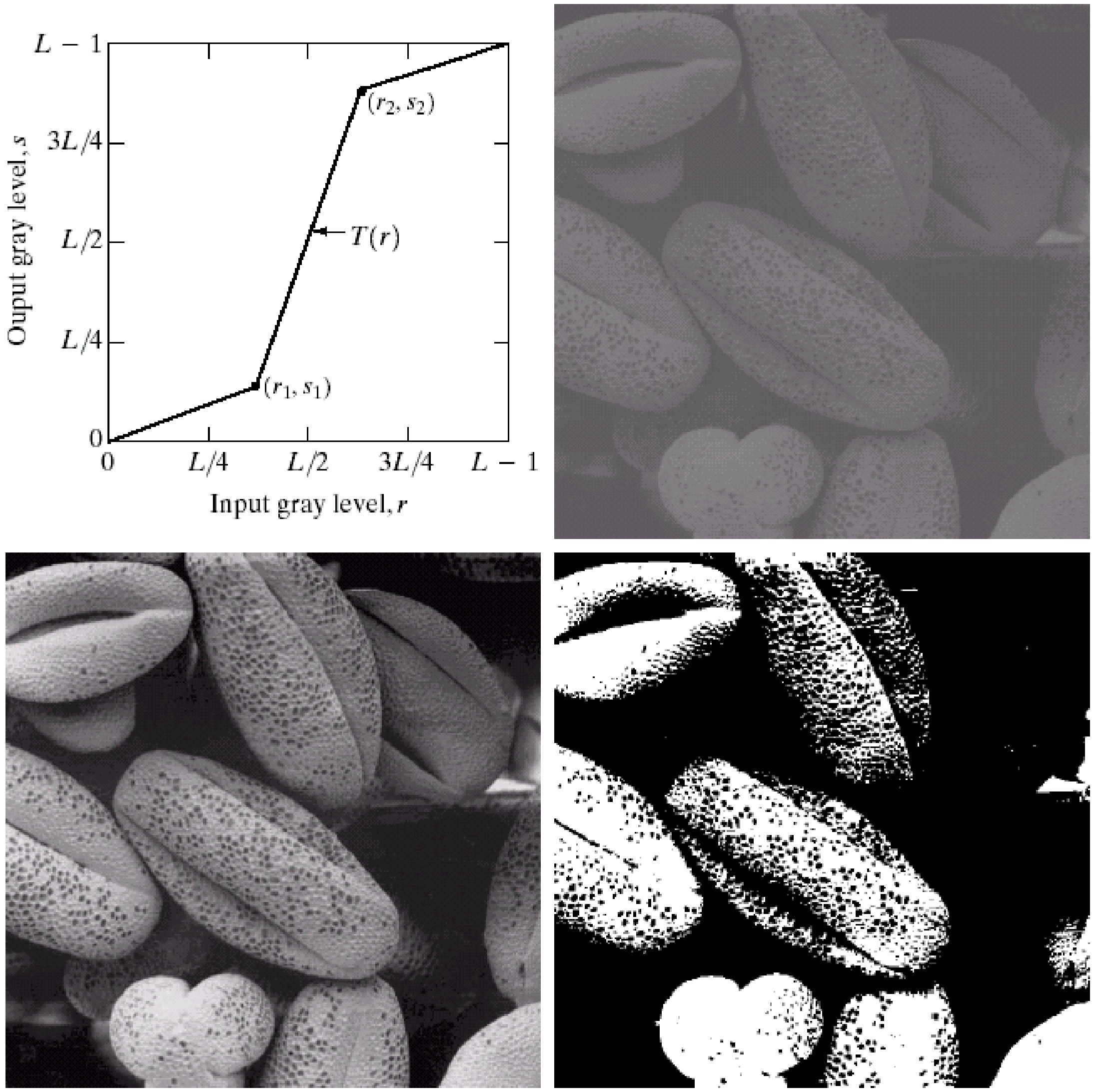


FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding.

Note: stretching is useless if the image has to be thresholded

Image Enhancement in the Spatial Domain: Gray-level slicing

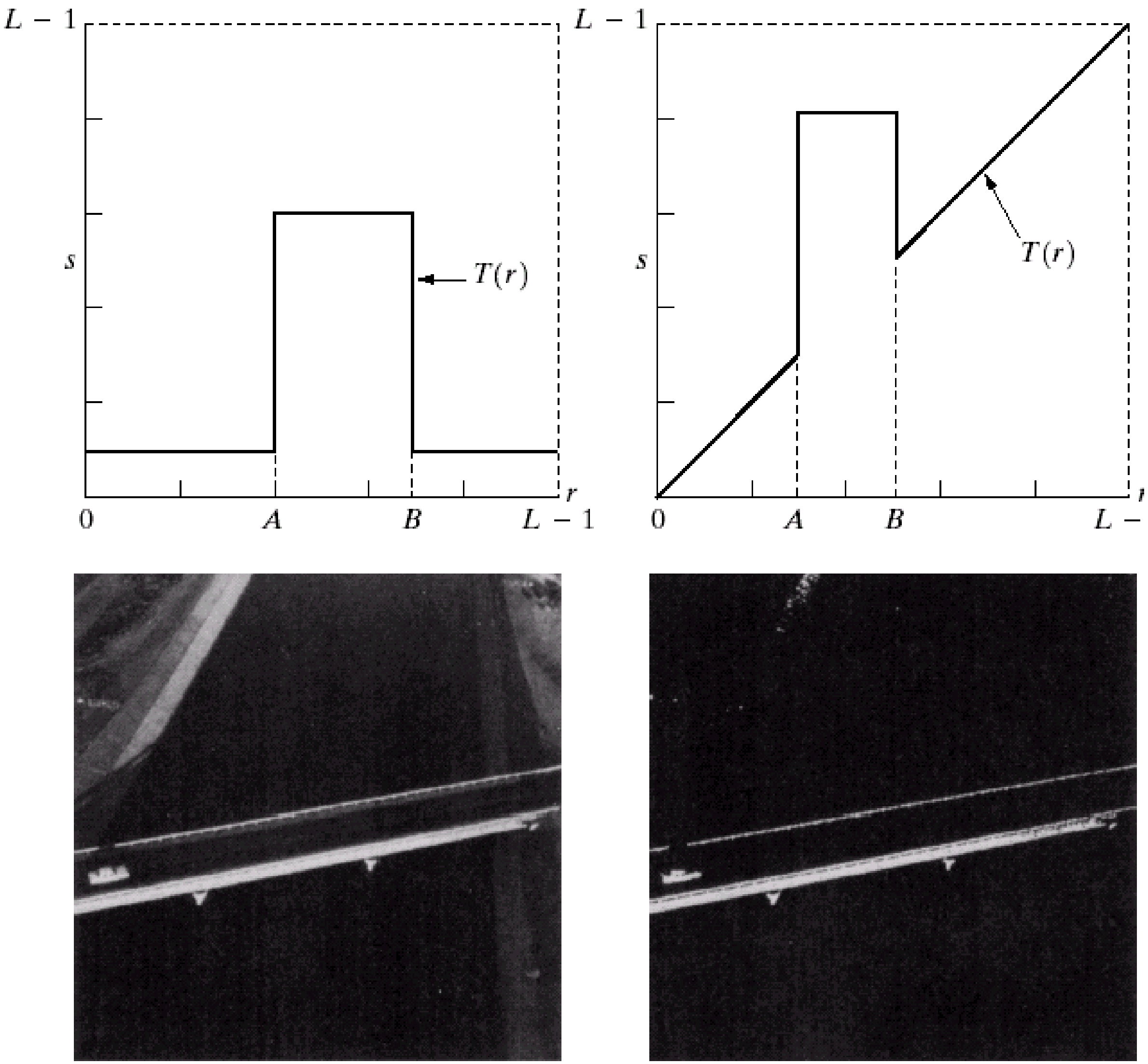


FIGURE 3.11 (a) This transformation highlights range [A, B] of gray levels and reduces all others to a constant level. (b) This transformation highlights range [A, B] but preserves all other levels. (c) An image. (d) Result of using the transformation in (a).

→ GraylevelMapping.m

Image Enhancement in the Spatial Domain: Histogram-based processing

Histogram: normalized frequency (y) of gray level values (x).

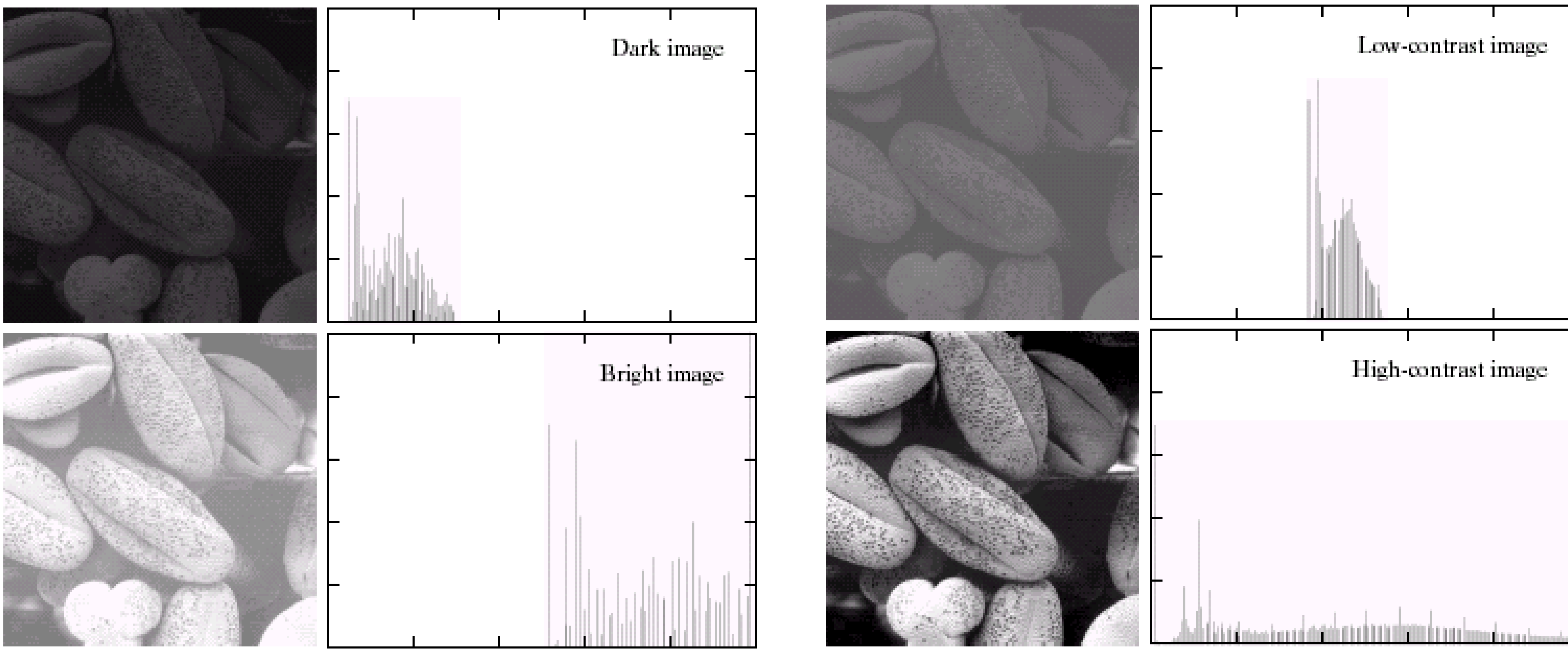


FIGURE 3.15 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms. (Original image courtesy of Dr. Roger Heady, Research School

Image Enhancement in the Spatial Domain: Histogram-based processing

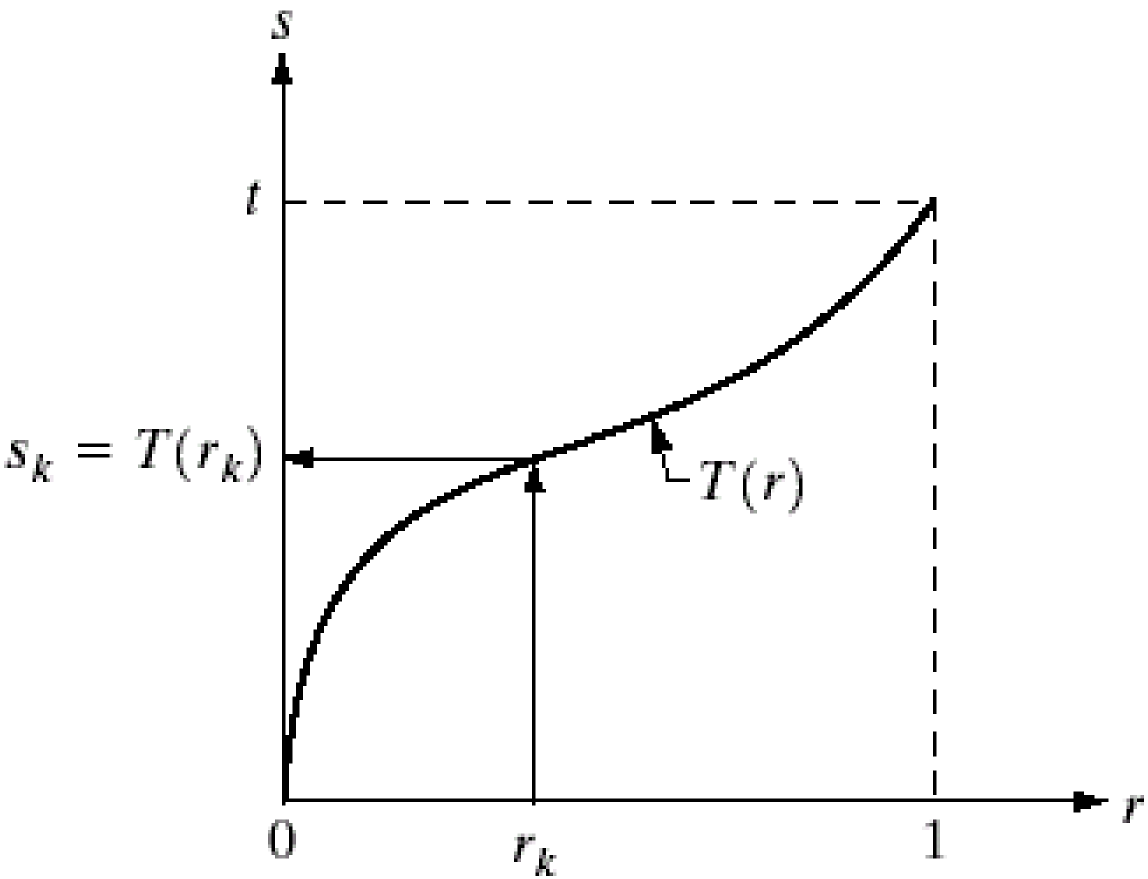
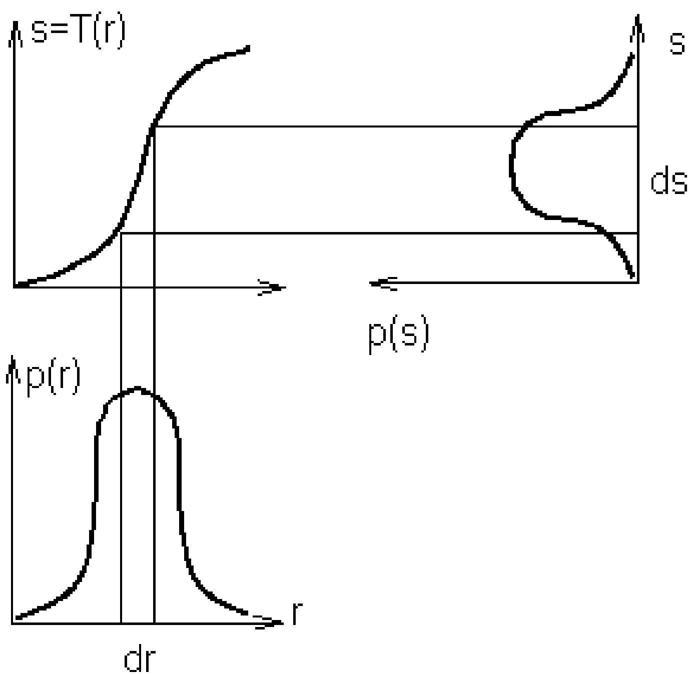


FIGURE 3.16 A gray-level transformation function that is both single valued and monotonically increasing.

(can be inverted and preserves gray-level ordering)

Image Enhancement in the Spatial Domain: Histogram **equalization**

Suppose the gray levels in an image are realizations of a random variable ***r*** in the range (0,1), with a *probability density function* (pdf): $p_r(r)$
Let $s = T(r)$ be a monotonic, invertible transformation on ***r***



All the pixels below the curve $p_r(r)$ in the interval $(r, r+dr)$ are mapped to pixels below $p_s(s)$ in $(s, s+ds)$ i.e., the two areas are equal:

$$p_s(s)\,ds = p_r(r)\,dr$$

Now, let the transformation be the *cumulative distribution function* (cdf) of ***r***

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

It is monotonic and invertible (if the pdf is nonzero for all ***r***)

Image Enhancement in the Spatial Domain: Histogram equalization

The derivative of this function is of course $ds/dr = p_r(r)$
Substituting in $p_s(s)\,ds = p_r(r)\,dr \quad \rightarrow \quad p_s(s) = 1$
i.e. the transformed variable has an ***exactly*** uniform pdf.

In a practical **discrete** case: $s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} n_j / n$

i.e., mapping each gray level r_k into the s_k value given above yields a uniform pdf for the output image.

Note: in general, only an approximately uniform distribution will be obtained in the discrete case.

Note: no parameters are needed; the processing is automatic and straightforward.

Image Enhancement in the Spatial Domain: Histogram equalization

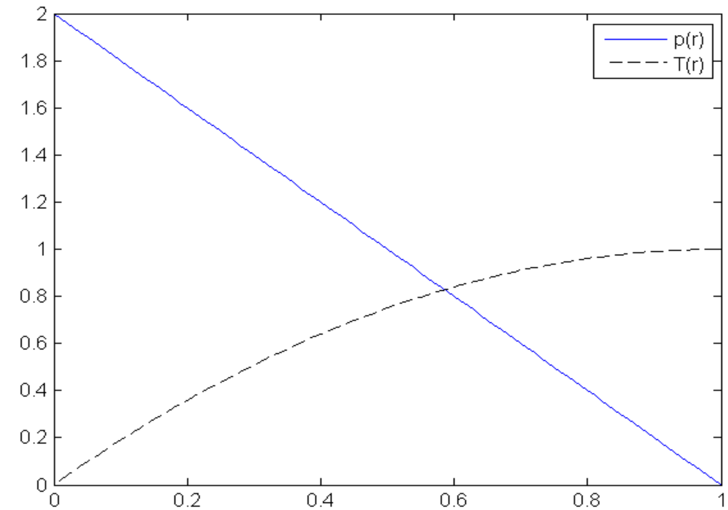
Example (continuous case):

$$p_r(r) = -2r + 2 \qquad 0 \le r \le 1$$

Equalization is obtained via the transformation:

$$s = T(r) = \int_0^r (-2w+2)\,dw = -r^2 + 2r$$

The transformed variable has a uniform pdf. Indeed:



$$r = \frac{2 \pm \sqrt{4 - 4s}}{2} = 1 \pm \sqrt{1-s}$$

$$r = 1 - \sqrt{1-s}\,, \qquad \frac{dr}{ds} = \frac{1}{2\sqrt{1-s}}$$

$$p_r(r) = -2(1-\sqrt{1-s}\,) + 2, \qquad p_r(r) = 2\sqrt{1-s}$$

$$p_s(s) = p_r(r)\frac{dr}{ds}, \qquad p_s(s) = 1 \quad \textbf{(Uniform pdf)}$$

S. Das, IIT Madras, Course on Computer Vision

Image Enhancement in the Spatial Domain: Histogram equalization

Example (discrete case):

(a) r_k	(b) n_k	(c) $p_r(r_k)$
0	790	0.19
1/7	1023	0.25
2/7	850	0.21
3/7	656	0.16
4/7	329	0.08
5/7	245	0.06
6/7	122	0.03
1	81	0.02
total	4096	1.00

(d) Cdf = s_k	(e) Quant. Values
0.19	1/7
0.44	3/7
0.65	5/7
0.81	6/7
0.89	6/7
0.95	1
0.98	1
1.00	1

(a) Quantized Gray levels; (b) a sample histogram; (c) its pdf; (d) Computed CDF and (e) approximated to the nearest gray level.

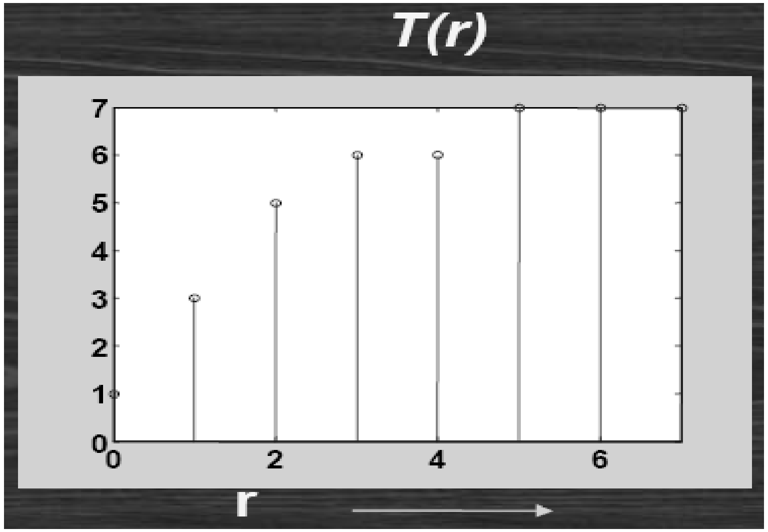
Image Enhancement in the Spatial Domain: Histogram equalization

r_k	$s_k = T(r)$
0	1/7
1/7	3/7
2/7	5/7
3/7	6/7
4/7	6/7
5/7	7/7
6/7	7/7
1	7/7

Transformation function

s_k	n_k	$p_s(s_k)$
0	0	0
1/7	790	0.19
2/7	0	0
3/7	1023	0.25
4/7	0	0
5/7	850	0.21
6/7	985	0.24
7/7=1	448	0.11

The new histogram



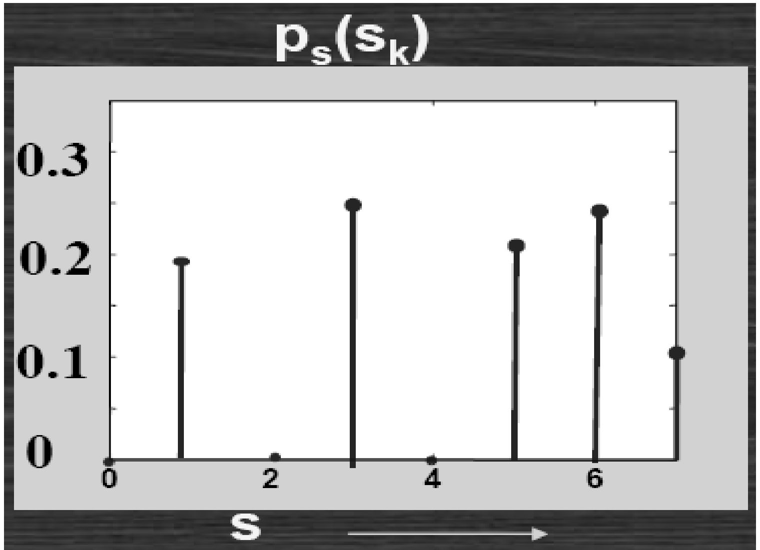


Image Enhancement in the Spatial Domain: Histogram equalization

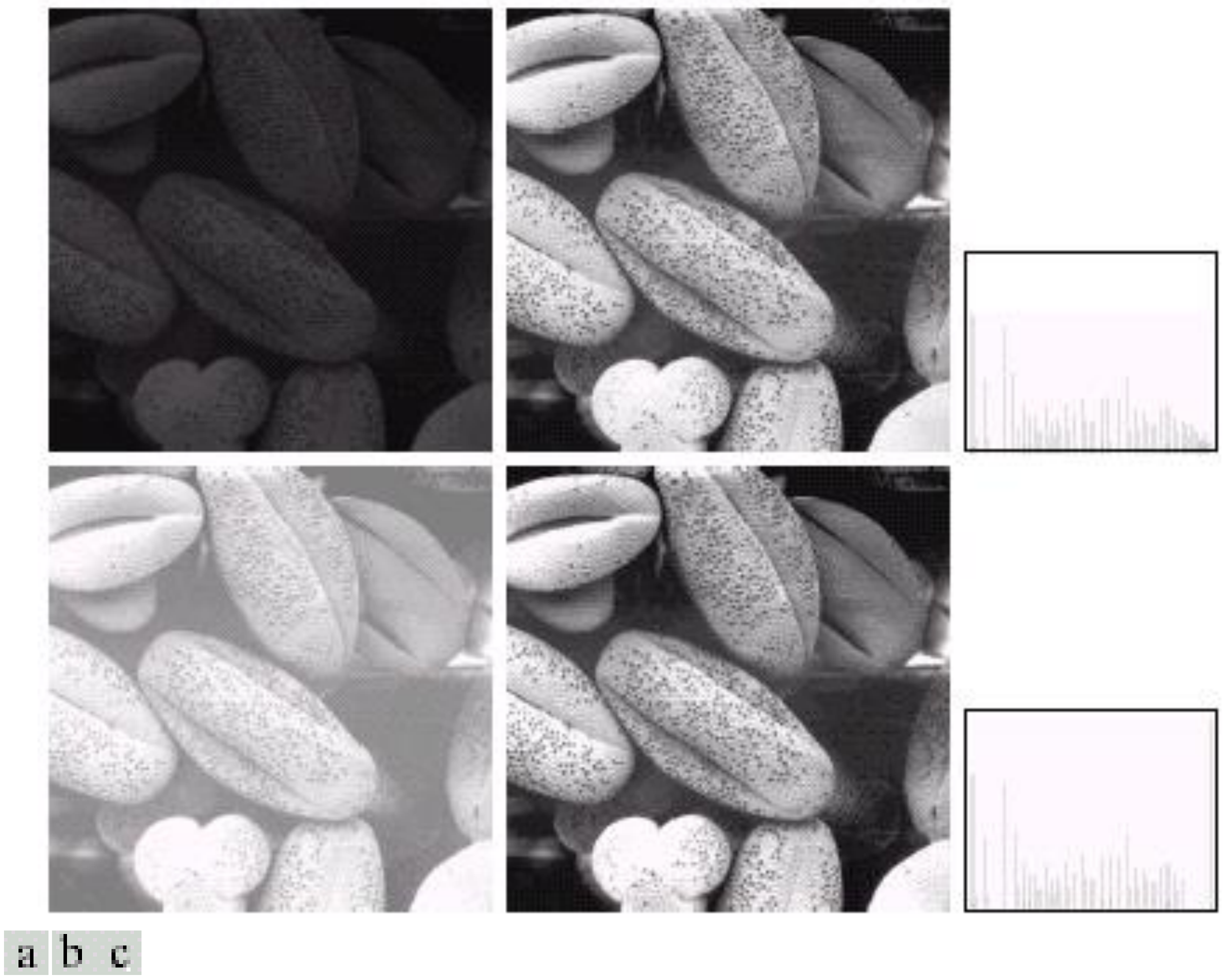


FIGURE 3.17 (a) Images from Fig. 3.15. (b) Results of histogram equalization. (c) Corresponding histograms.

Image Enhancement in the Spatial Domain: Histogram equalization

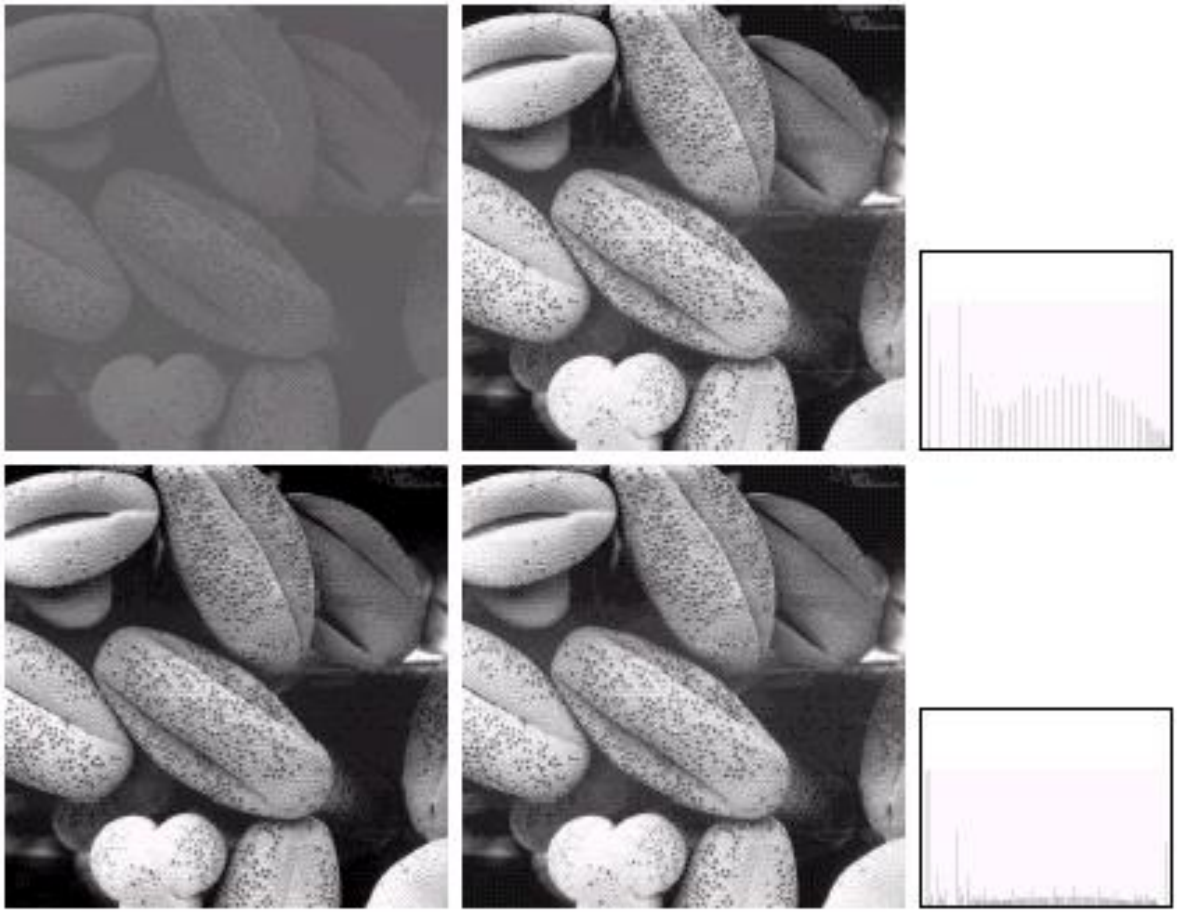


FIGURE 3.17 (a) Images from Fig. 3.15. (b) Results of histogram equalization. (c) Corresponding histograms.

Image Enhancement in the Spatial Domain: Histogram **specification**

Remember that the mapping

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} n_j / n$$

yields a (approx.) uniformly distributed output. Another variable $\boldsymbol{z}$, with a different, known and desired pdf $\boldsymbol{p_z}$, will satisfy the same equation:

$$G(z_k) = \sum_{j=0}^{k} p_z(z_j) = s_k$$

substituting:

$$z_k = G^{-1}(s_k) = G^{-1}(T(r_k))$$

i.e., mapping each gray level $\boldsymbol{r_k}$ into the $\boldsymbol{z_k}$ value given above yields the desired histogram (pdf) for the output image.

Image Enhancement in the Spatial Domain: Histogram specification

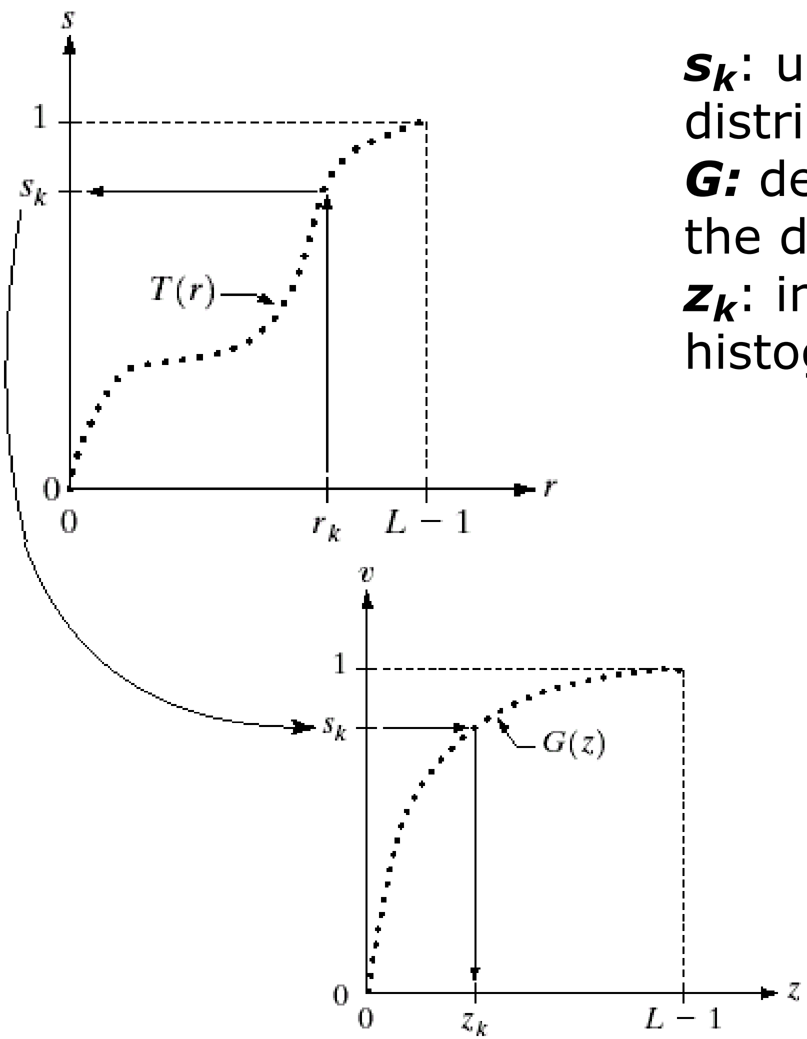


FIGURE 3.19
(a) Graphical interpretation of mapping from r_k to s_k via $T(r)$.
(c) Inverse mapping from s_k to its corresponding value of z_k.

s_k: uniformly distributed image
G: determined as cdf of the desired pdf ***p_z***
z_k: image with desired histogram

Image Enhancement in the Spatial Domain: Histogram specification

Example:

- Two histograms are given to us

r_k	n_k	p_k
0/7	790	0.19
1/7	1023	0.25
2/7	850	0.21
3/7	656	0.16
4/7	329	0.08
5/7	245	0.06
6/7	122	0.03
7/7	81	0.02

Input histogram

z_k	$p(z_k)$
0/7	0
1/7	0
2/7	0
3/7	0.15
4/7	0.2
5/7	0.3
6/7	0.2
7/7	0.15

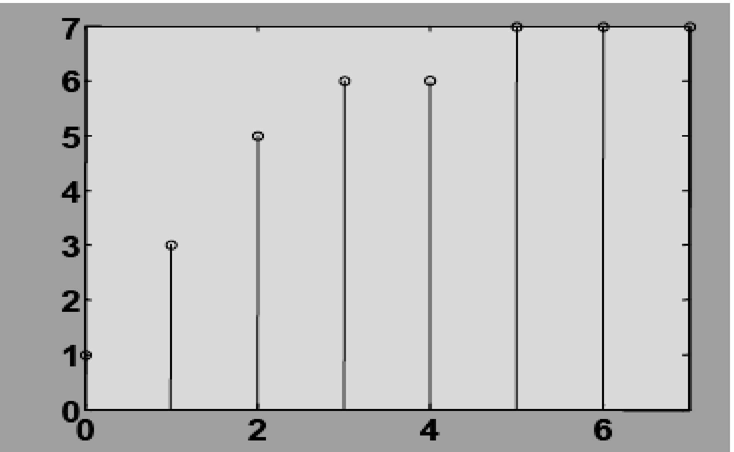
Target histogram

Then determine $T(r)$ and $G(z)$ (cdf's of the histograms):

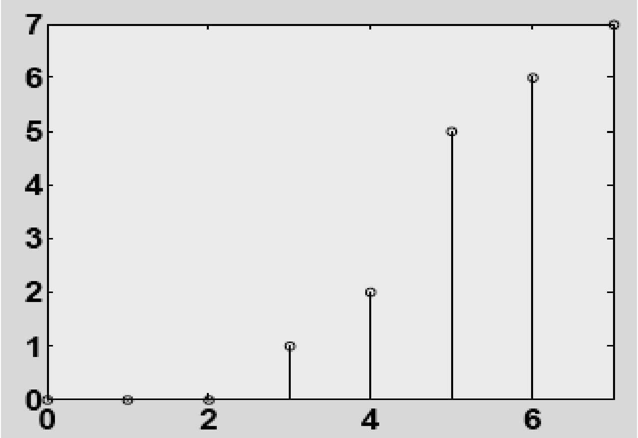
Image Enhancement in the Spatial Domain: Histogram specification

r_k	p_k	$cdf(p_k)$	Gray levels
0/7	0.19	0.19	1/7
1/7	0.25	0.44	3/7
2/7	0.21	0.65	5/7
3/7	0.16	0.81	6/7
4/7	0.08	0.89	6/7
5/7	0.06	0.95	1
6/7	0.03	0.98	1
7/7	0.02	1	1

z_k	$p(z_k)$	$cdf(z_K)$	Gray level
0/7	0	0	0/7
1/7	0	0	0/7
2/7	0	0	0/7
3/7	0.15	0.15	1/7
4/7	0.2	0.35	2/7
5/7	0.3	0.65	5/7
6/7	0.2	0.85	6/7
7/7	0.15	1	1



T(r)



G(z)

Image Enhancement in the Spatial Domain: Histogram specification

r_k	$T(r_k) \cong \ldots$
0/7	0.19
1/7	0.44
2/7	0.65
3/7	0.81
4/7	0.89
5/7	0.95
6/7	0.98
7/7	1

$\ldots \cong G(z_k)$	$z_k = G^{-1}(T(r_k))$
0	0/7
0	1/7
0	2/7
0.15	3/7
0.35	4/7
0.65	5/7
0.85	6/7
1	7/7

Mapping function

$0 \rightarrow 3$

$1 \rightarrow 4$

$2 \rightarrow 5$

$3,4 \rightarrow 6$

$5,6,7 \rightarrow 7$

Image Enhancement in the Spatial Domain: Histogram specification

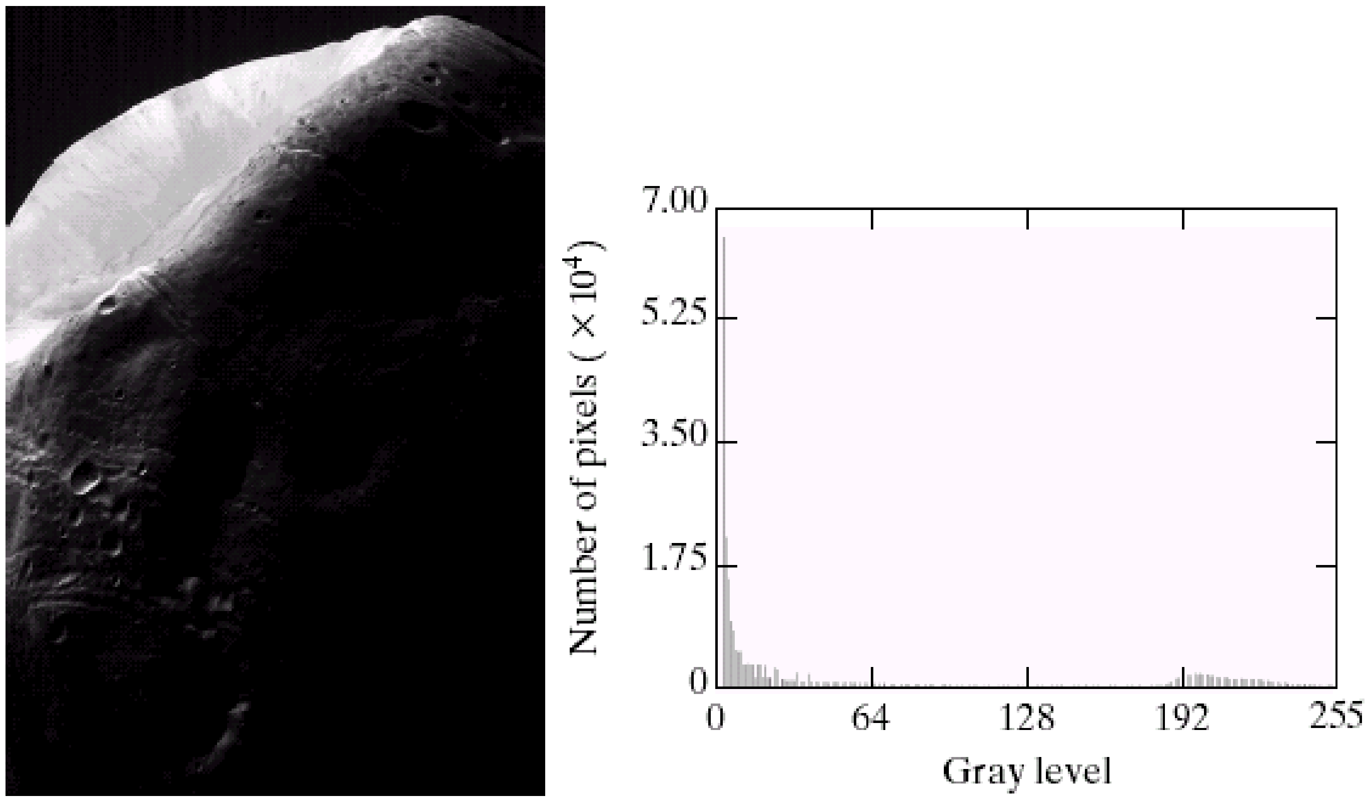
distributions: original target obtained

r_k	n_k	p_k
0/7	790	0.19
1/7	1023	0.25
2/7	850	0.21
3/7	656	0.16
4/7	329	0.08
5/7	245	0.06
6/7	122	0.03
7/7	81	0.02

$p(z_k)$
0
0
0
0.15
0.2
0.3
0.2
0.15

n'_k	$p'(z_k)$
0	0
0	0
0	0
790	0.19
1023	0.25
850	0.21
656+329	0.24
245+122+81	0.11

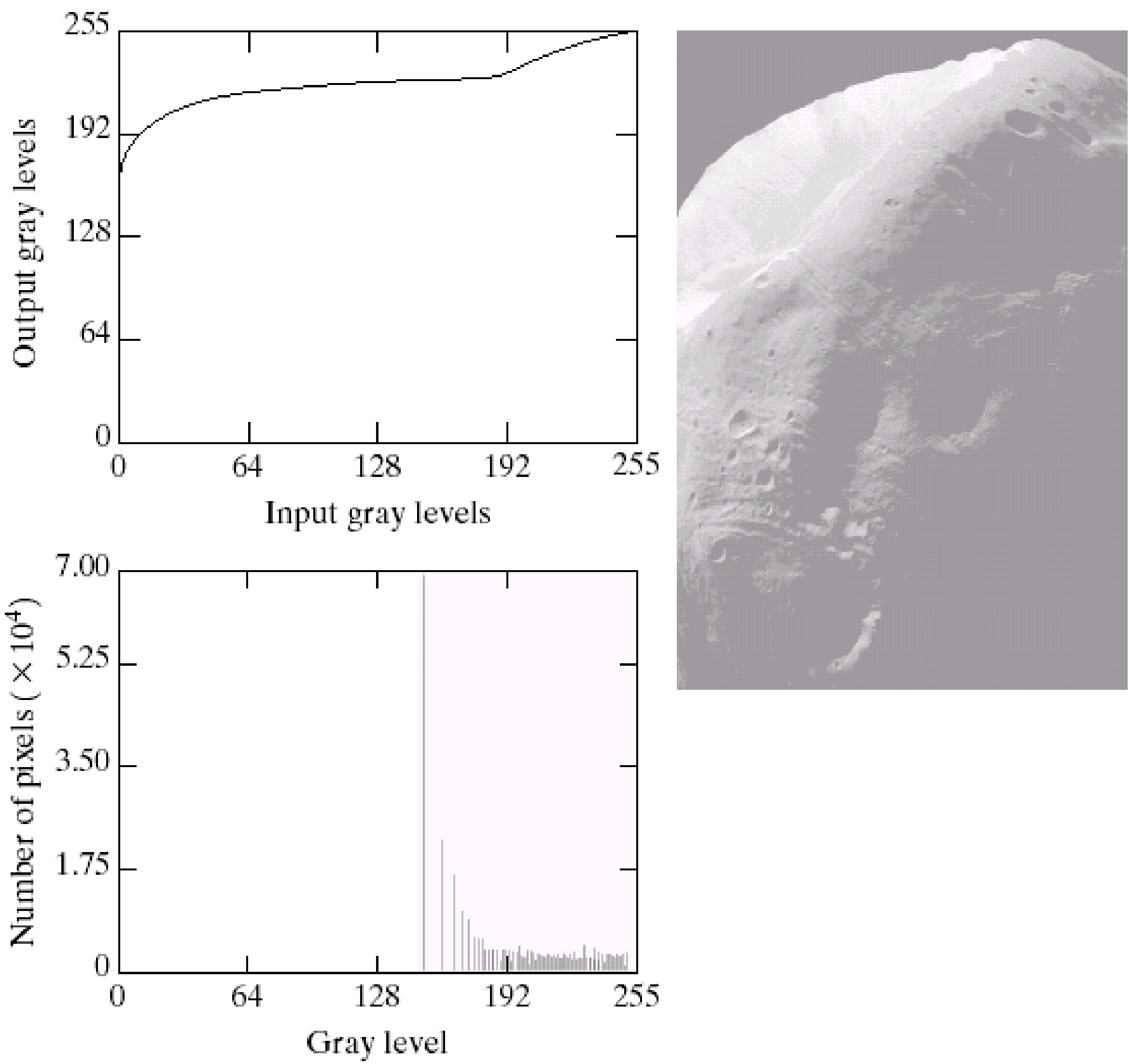
Image Enhancement in the Spatial Domain: Histogram specification



a b

FIGURE 3.20 (a) Image of the Mars moon Photos taken by NASA's *Mars Global Surveyor*. (b) Histogram. (Original image courtesy of NASA.)

Image Enhancement in the Spatial Domain: Histogram specification



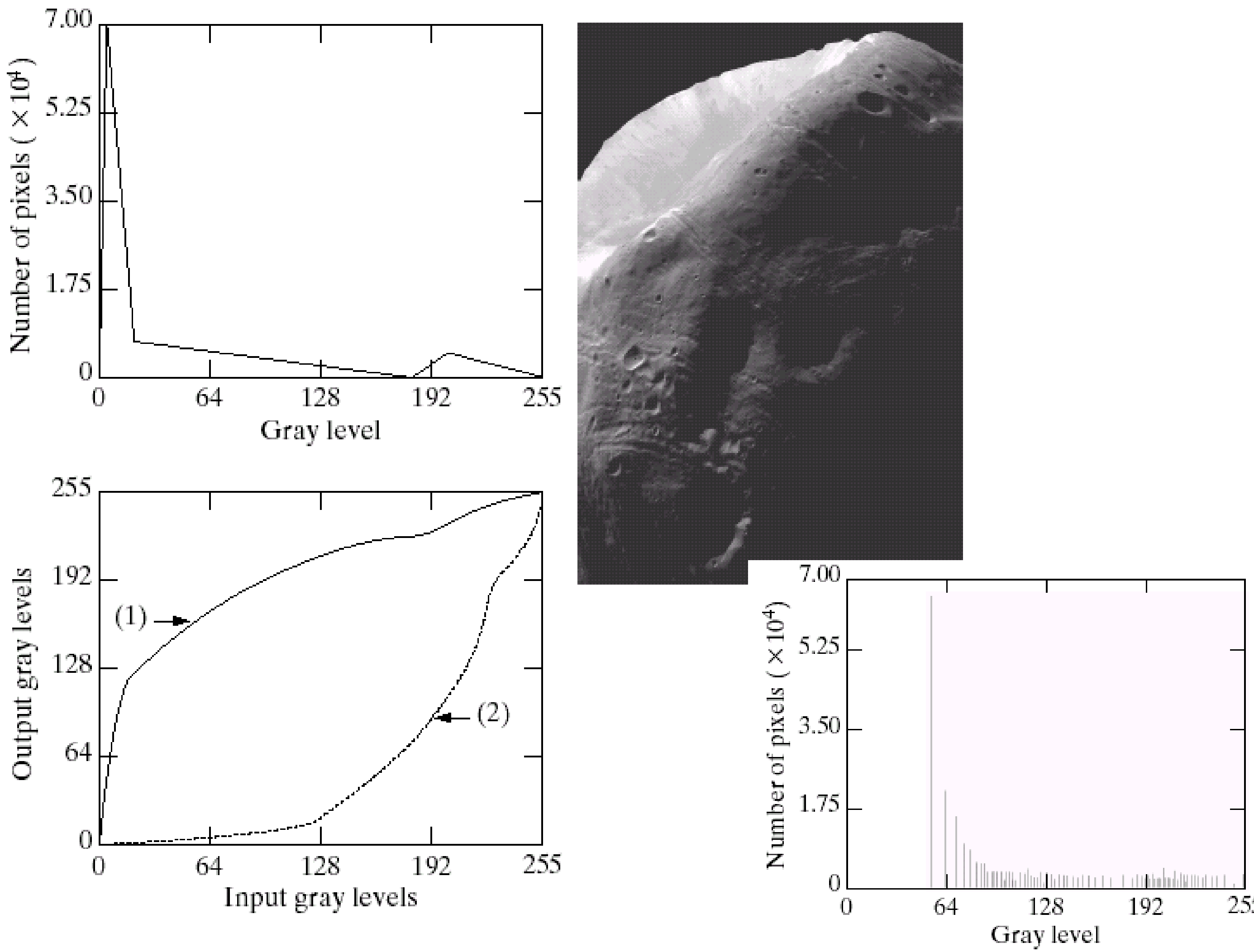
a b
c

FIGURE 3.21
(a) Transformation function for histogram equalization.
(b) Histogram-equalized image (note the washed-out appearance).
(c) Histogram of (b).

Image Enhancement in the Spatial Domain: Histogram specification

a	c
b	
d	

FIGURE 3.22
(a) Specified histogram.
(b) Curve (1) is from Eq. (3.3-14), using the histogram in (a); curve (2) was obtained using the iterative procedure in Eq. (3.3-17).
(c) Enhanced image using mappings from curve (2).
(d) Histogram of (c).



Note: Histogram manipulation of color images can be performed in the HSV color space → Matlab

Image Enhancement in the Spatial Domain: Histogram manipulation of **color** images

https://www.mathworks.com/help/images/understanding-color-spaces-and-color-space-conversion.html

- Colors typically represented as RGB numeric values
- Other models exist, referred to as *color spaces* because most of them can be mapped into a 3-D coordinate system
- They make certain calculations more convenient
- They provide a way to identify colors that is more intuitive

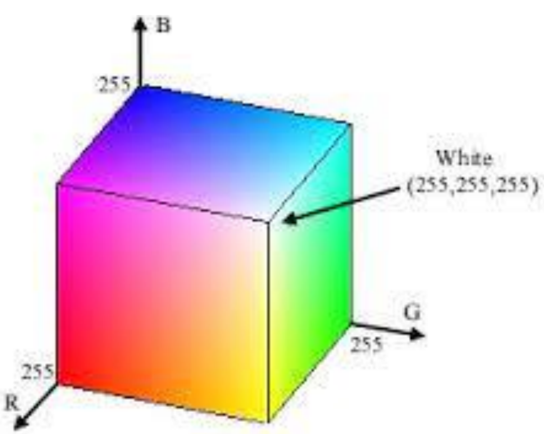
- They may avoid (reduce) color alterations!

Matlab provides commands to convert data among different color spaces

Image Enhancement in the Spatial Domain: Histogram manipulation of **color** images

RGB

Linear RGB	Raw data obtained from a camera sensor. R, G, and B are directly proportional to the amount of light that illuminates the sensor. Preprocessing of raw image data, such as white balance, color balance, and chromatic aberration compensation, are performed on linear RGB values
sRGB	sRGB values apply gamma correction to linear RGB values. Images are frequently displayed in the sRGB color space because they appear brighter and colors are easier to distinguish



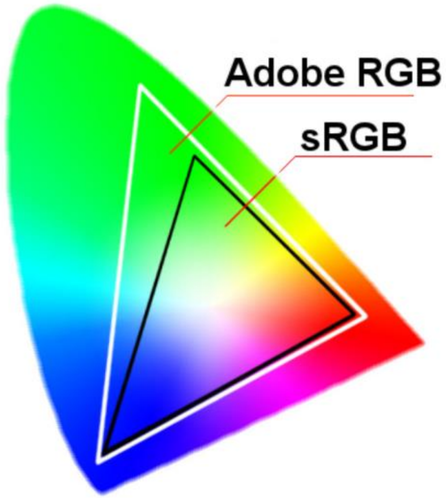


Image Enhancement in the Spatial Domain: Histogram manipulation of **color** images

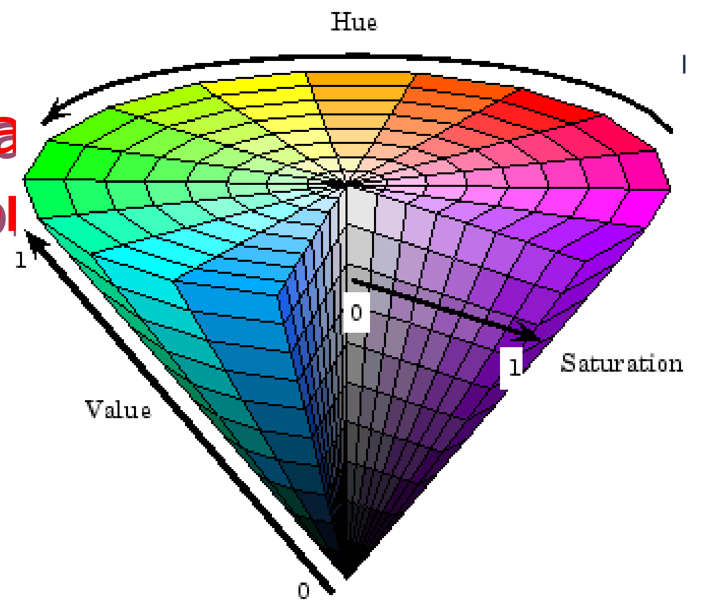
YCbCr is widely used for digital video

Y	Luminance or brightness of the image. Colors increase in brightness as *Y* increases
Cb	Chrominance value that indicates the difference between the blue component and a reference value
Cr	Chrominance value that indicates the difference between the red component and a reference value

YCbCr does not use the full range of the image data type so that the video stream can include additional (non-image) information:

- float: *Y* is in [16/255, 235/255] and *Cb* and *Cr* are in [16/255, 240/255]
- uint8: *Y* is in [16, 235] and *Cb* and *Cr* are in [16, 240]
- uint16: *Y* is in [4112, 60395] and *Cb* and *Cr* are in [4112, 61680]

Image Enhancement in the Spatia
Histogram manipulation of **colo**



HSV corresponds better to how people experience color than RGB does

H Hue corresponds to the color’s position on a color wheel. *H* is in the range [0, 1]. As *H* increases, colors transition from red to orange, yellow, green, cyan, blue, magenta, and finally back to red. Both 0 and 1 indicate red.

S Saturation is the amount of hue or departure from neutral. *S* is in the range [0, 1]. As *S* increases, colors vary from unsaturated (shades of gray) to fully saturated (no white component).

V Value is the maximum value among the red, green, and blue components of a specific color. *V* is in the range [0, 1]. As *V* increases, the corresponding colors become increasingly brighter.

Histogram manipulation in the HSV color space → Matlab

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Image Enhancement in the Spatial Domain: Exact Histogram specification

→ Paper (z05_Exact_HS) & Matlab (demoHS.m)

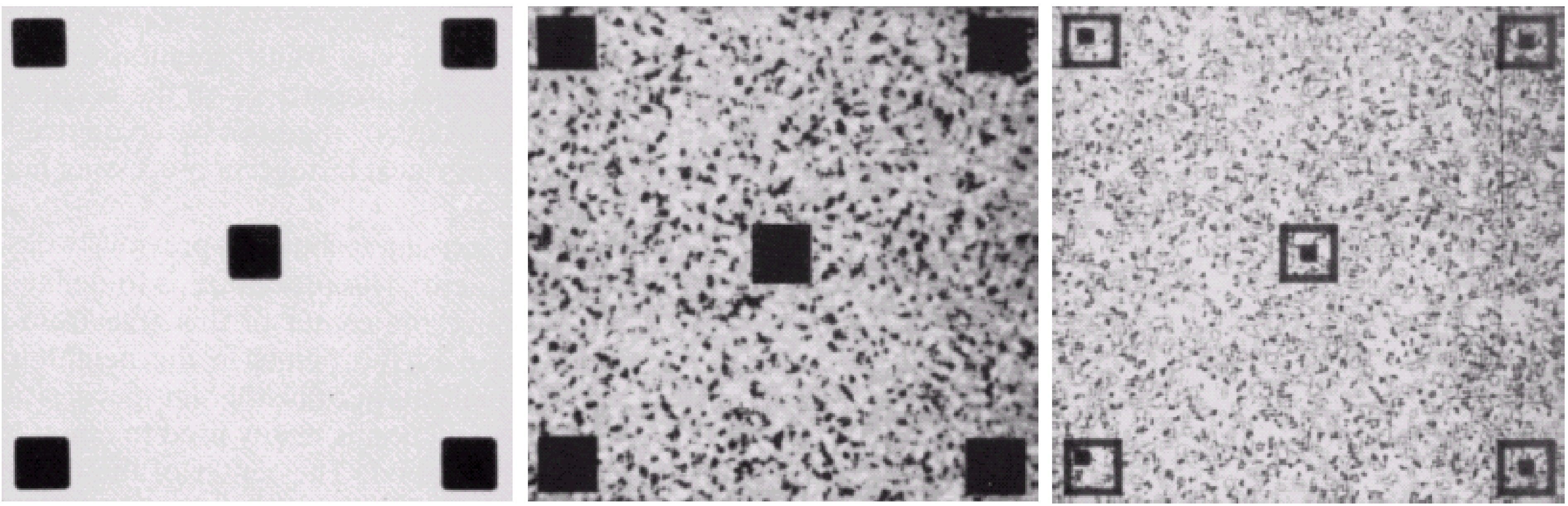
Image Enhancement in the Spatial Domain: 2-D Histogram specification

→ Papers (z05_2D_HistEq) & (z05_HVS_ToneMapping)

Image Enhancement in the Spatial Domain: **Local** histogram modification

At each location the **local histogram** is computed, the required mapping is determined, and the pixel is mapped.
(At the next step, just *update* the histogram)

→ (CL)AHE paper



a b c

FIGURE 3.23 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization using a 7 × 7 neighborhood about each pixel.

Image Enhancement in the Spatial Domain: Enhancement based on **local statistics**

Local values can be estimated for different image statistics, and used to locally control a gray-level modification function.
E.g.: local mean and variance in the neighborhood *Sxy*:

$$m_{Sxy} = \sum_{s,t \in Sxy} r(s,t)\, p[r(s,t)] \qquad \sigma^2_{Sxy} = \sum_{s,t \in Sxy} [r(s,t) - m_{Sxy}]^2\, p[r(s,t)]$$

Enhancement example: increase by a factor $A>1$ the luminance of pixels in medium-variance, low-mean areas:

$$g(x,y) = \begin{cases} A\, f(x,y) & \text{if} \quad m_{Sxy} < k_0 M_f \quad \& \quad k_1 D_f < \sigma^2_{Sxy} < k_2 D_f \\ f(x,y) & \text{otherwise} \end{cases}$$

M_f and D_f respectively are the ***global*** average and s.d. of the image; they are used to make the operator more robust.

Image Enhancement in the Spatial Domain: Enhancement based on local statistics

FIGURE 3.24 SEM image of a tungsten filament and support, magnified approximately 130×. (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene).

Image Enhancement in the Spatial Domain: Enhancement based on local statistics

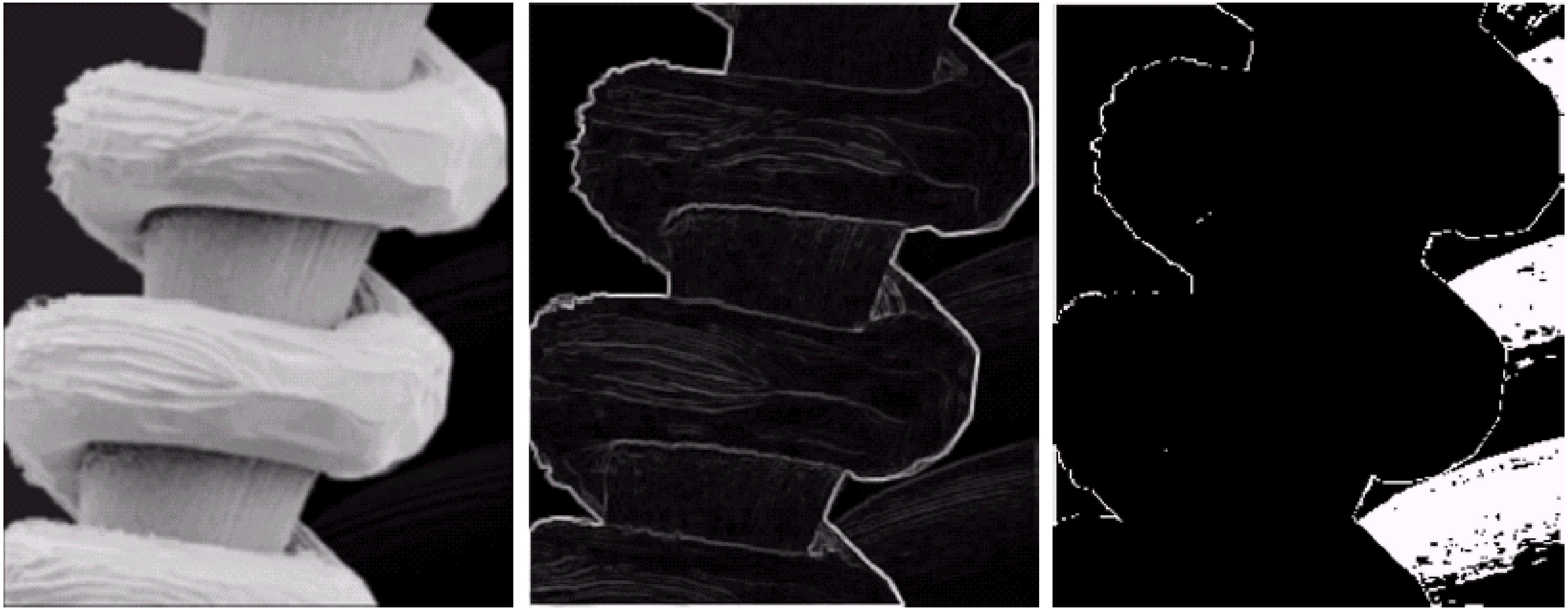


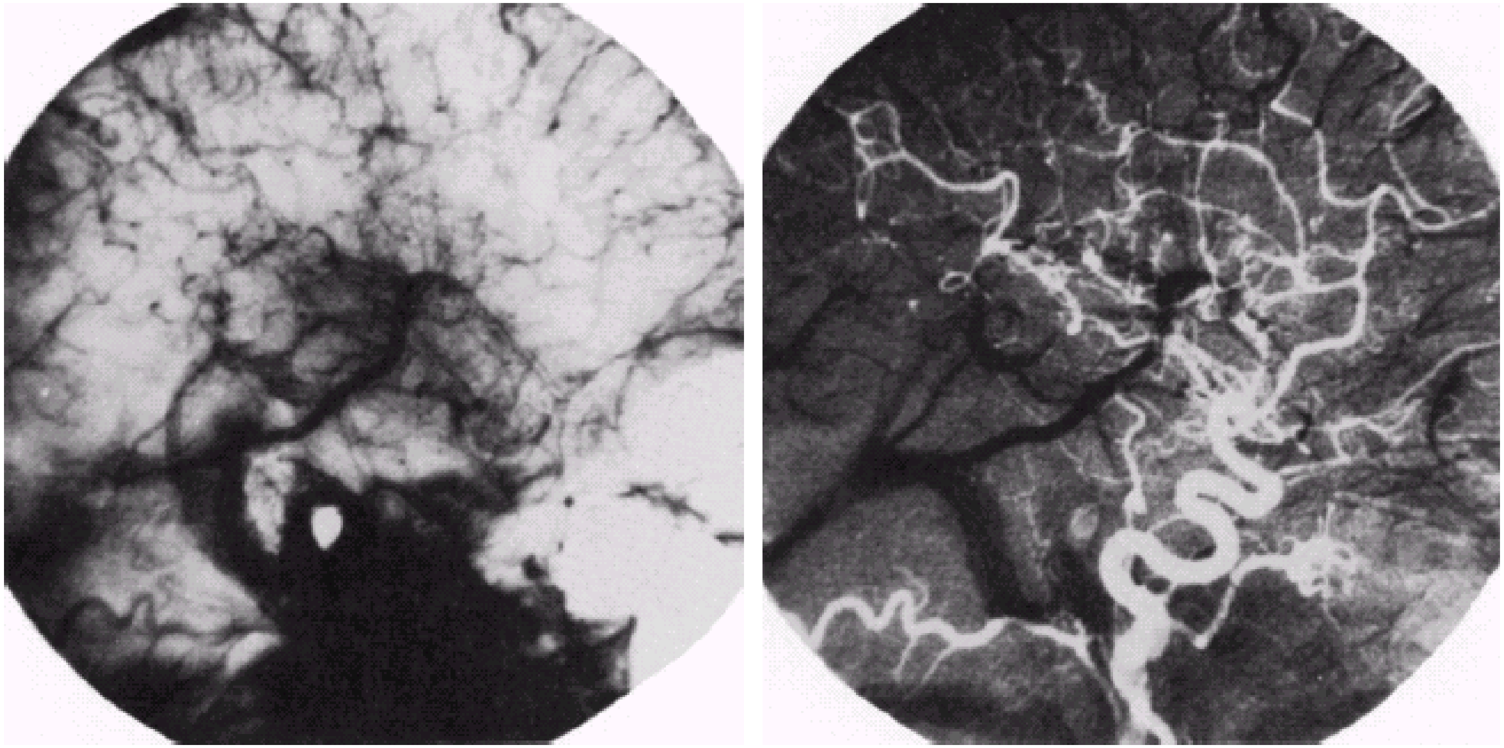
FIGURE 3.25 (a) Image formed from all local means obtained from Fig. 3.24 using Eq. (3.3-21). (b) Image formed from all local standard deviations obtained from Fig. 3.24 using Eq. (3.3-22). (c) Image formed from all multiplication constants used to produce the enhanced image shown in Fig. 3.26.

Image Enhancement in the Spatial Domain: Enhancement based on local statistics

FIGURE 3.26 Enhanced SEM image. Compare with Fig. 3.24. Note in particular the enhanced area on the right side of the image.

Image Enhancement in the Spatial Domain: Using multiple images: **subtraction**

Using a **«local-along-time»** neighborhood



a b

FIGURE 3.29 Enhancement by image subtraction. (a) Mask image. (b) An image (taken after injection of a contrast medium into the bloodstream) with mask subtracted out.

Image Enhancement in the Spatial Domain: Using multiple images: **averaging**

Using a **«local-along-time»** neighborhood

Assume an image is formed as:

$$g(x, y) = f(x, y) + n(x, y)$$

where $n(x,y)$ is i.i.d. zero-mean noise. If we can average K acquisitions of the image, the variance of the noise is reduced by the factor K:

$$\bar{g}(x, y) = \frac{1}{K}\sum_{k=1}^{K} g_k(x, y) = f(x, y) + \frac{1}{K}\sum_{k=1}^{K} n_k(x, y)$$

This approach is useful when the sensor noise is relatively high: poorly illuminated (static) scenes, astronomical images, …

Image Enhancement in the Spatial Domain: Using multiple images: averaging

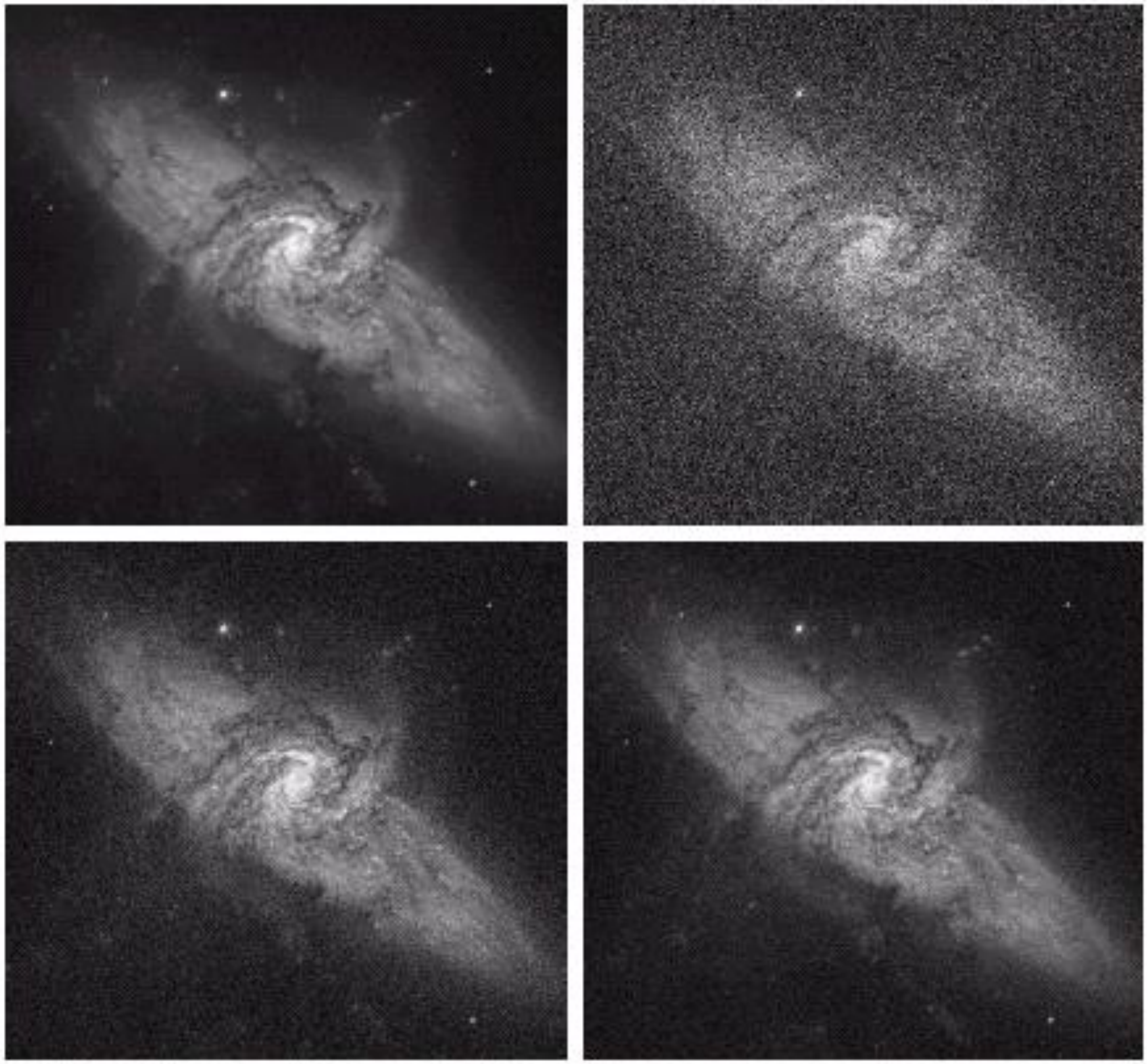


Fig.3.30

A) Ideal

B) Noise added (s.d.=64)

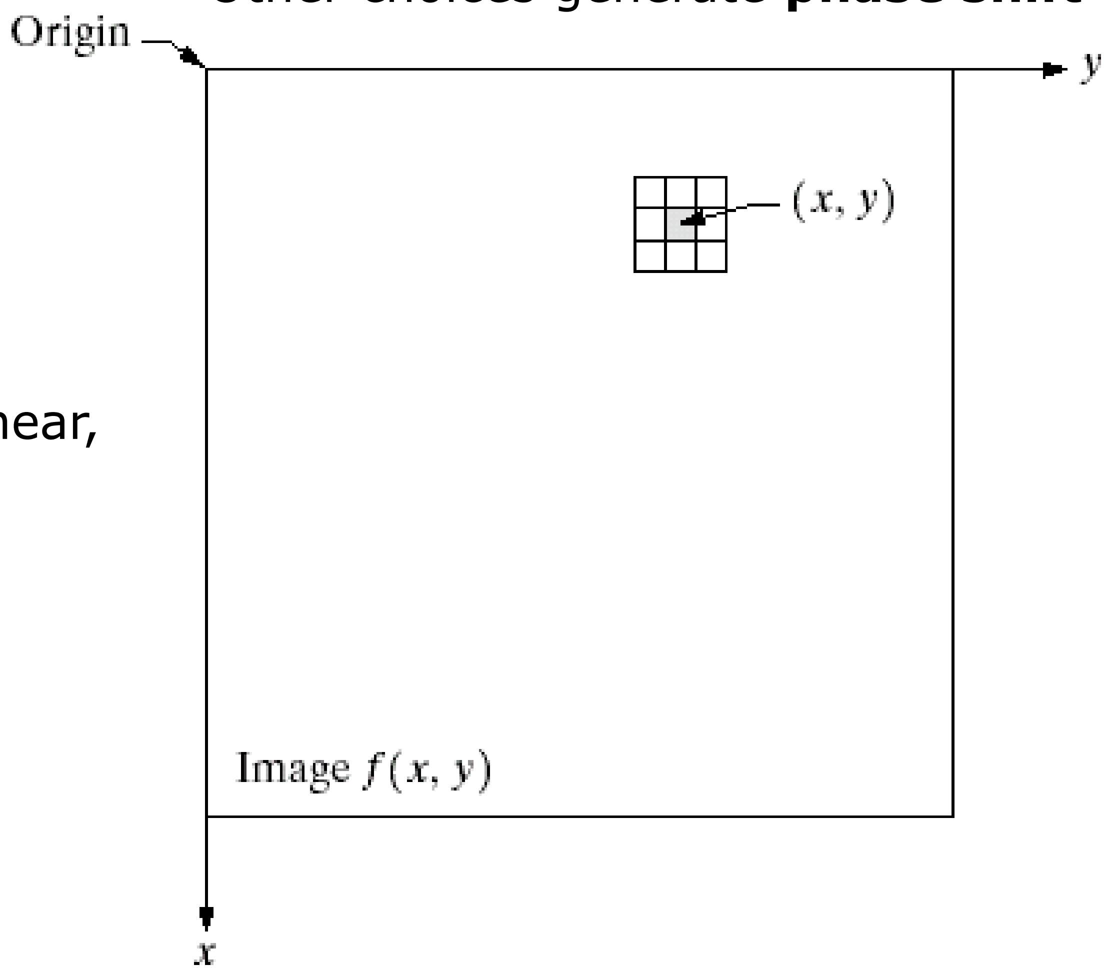
C) K=8

D) K=16

Image Enhancement in the Spatial Domain: Local operators

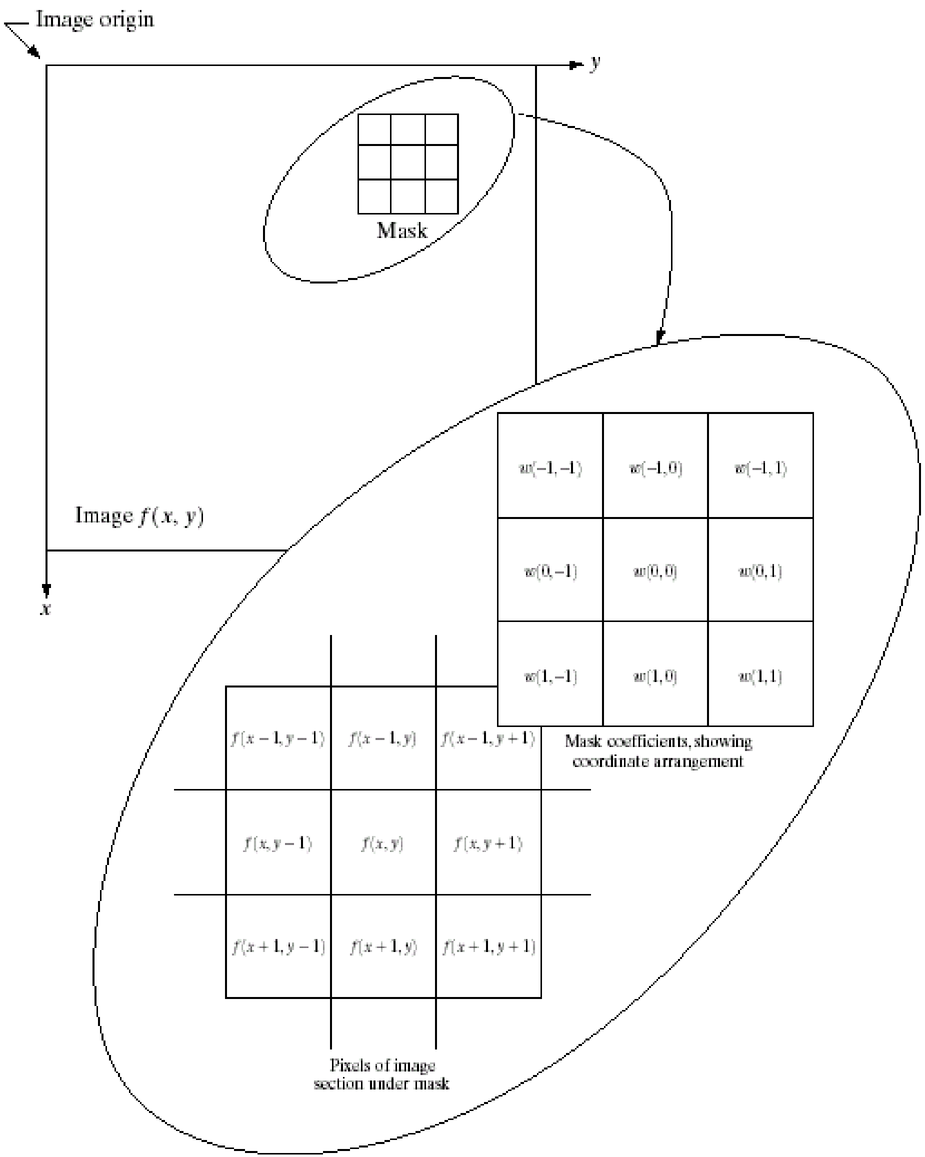
Note: this mask is centered in (x,y), i.e. its origin is its center. → **noncausal operator**
Other choices generate **phase shift**

FIGURE 3.1 A 3 × 3 neighborhood about a point (x, y) in an image.



Generic, possibly nonlinear, neighborhood-based operator:
$g(x,y)=T[f(x,y)]$

Image Enhancement in the Spatial Domain: Local operators



The mask entries are coefficients that can be used in different ways.

The simplest is ***linear filtering*** via the normalized ***convolution sum***:

$$g(x,y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)}$$

Note: other scaling factors are used if the coeffs. sum is zero

Note: if the output image is required to be the same size as the input image, the latter must be suitably ***padded***.

Image Enhancement in the Spatial Domain: Local operators

FIGURE 3.33 Another representation of a general 3×3 spatial filter mask.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Image Enhancement in the Spatial Domain: Local operators

Options	Description
Filtering Mode	
`'corr'`	Filtering is done using correlation (see Figs. 3.13 and 3.14). This is the default.
`'conv'`	Filtering is done using convolution (see Figs. 3.13 and 3.14).
Boundary Options	
`P`	The boundaries of the input image are extended by padding with a value, `P` (written without quotes). This is the default, with value 0.
`'replicate'`	The size of the image is extended by replicating the values in its outer border.
`'symmetric'`	The size of the image is extended by mirror-reflecting it across its border.
`'circular'`	The size of the image is extended by treating the image as one period a 2-D periodic function.
Size Options	
`'full'`	The output is of the same size as the extended (padded) image (see Figs. 3.13 and 3.14).
`'same'`	The output is of the same size as the input. This is achieved by limiting the excursions of the center of the filter mask to points contained in the original image (see Figs. 3.13 and 3.14). This is the default.

Matlab implementation using ‘imfilter’

TABLE 3.2 Options for function `imfilter`.

Image Enhancement in the Spatial Domain: Local operators

Matlab: correlation or convolution

Origin f

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

w

1	2	3
4	5	6
7	8	9

(a)

Padded f

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(b)

Initial position for w

1	2	3	0	0	0	0
4	5	6	0	0	0	0
7	8	9	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(c)

Correlation result

0	0	0	0	0
0	9	8	7	0
0	6	5	4	0
0	3	2	1	0
0	0	0	0	0

(d)

Full correlation result

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	9	8	7	0	0
0	0	6	5	4	0	0
0	0	3	2	1	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(e)

Rotated w

9	8	7	0	0	0	0
6	5	4	0	0	0	0
3	2	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(f)

Convolution result

0	0	0	0	0
0	1	2	3	0
0	4	5	6	0
0	7	8	9	0
0	0	0	0	0

(g)

Full convolution result

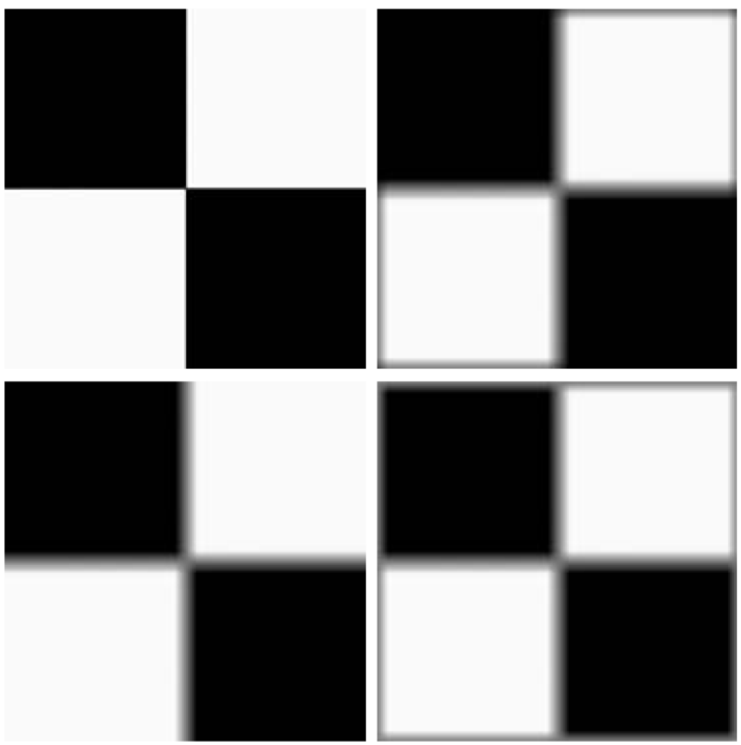
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	1	2	3	0	0
0	0	4	5	6	0	0
0	0	7	8	9	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(h)

FIGURE 3.14 Illustration of two-dimensional correlation and convolution. The 0s are shown in gray to simplify viewing.

Image Enhancement in the Spatial Domain: Local operators

Matlab: image padding + filtering
[lowpass, w = ones(31,31)]



(a) Original image. (b) Result of using `imfilter` with default zero padding. (c) Result with the `'replicate'` option. (d) Result with the `'symmetric'` option. (e) Result with the `'circular'`

Image Enhancement in the Spatial Domain: Linear **lowpass filters**

$$\frac{1}{9} \times \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16} \times \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

3×3 smoothing (averaging) filter masks. The constant multipli er in front of each mask is equal to the sum of the values of its coefficients, as is

Both masks have power-of-two coefficients, which are simple to implement. In the second one even the sum of the coefficients is a power of two.

Note: a uniform input image is not changed
Note: for an image having amplitude 1 and frequency π in both hor. and vert. directions:

$$\cdots \begin{array}{|c|c|c|c|}\hline 1 & -1 & 1 & -1 \\ \hline -1 & 1 & -1 & 1 \\ \hline 1 & -1 & 1 & -1 \\ \hline \end{array} \cdots$$

the output amplitude is respectively:

$G_1 = (5x1 + 4x(-1)) / 9 = 1/9$

$G_2 = (8x1 + 8x(-1)) / 16 = 0$

Image Enhancement in the Spatial Domain:
Linear lowpass filters

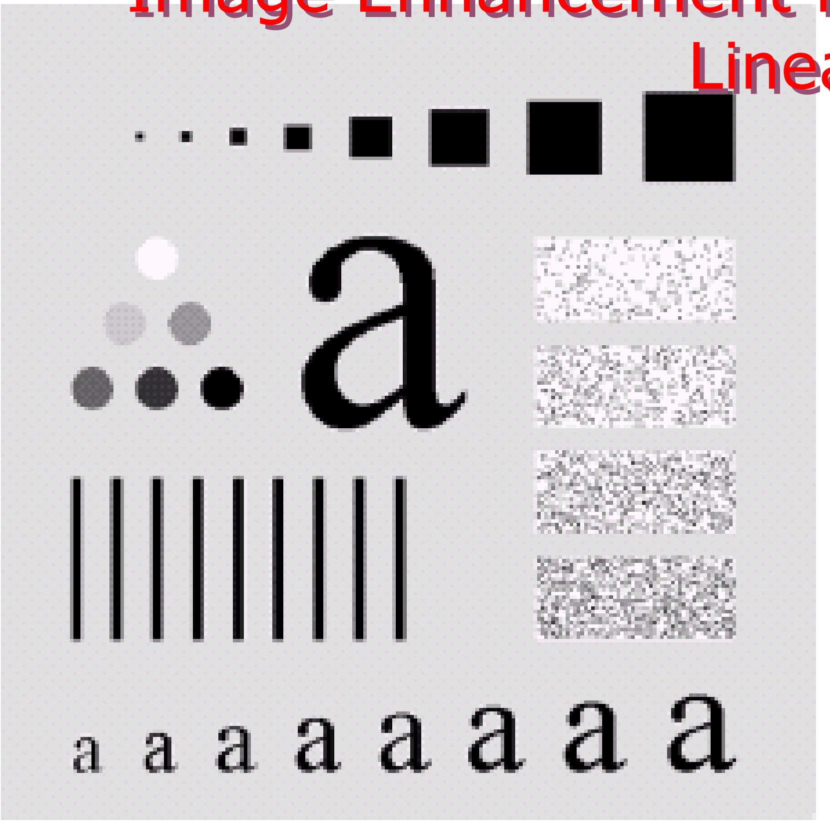
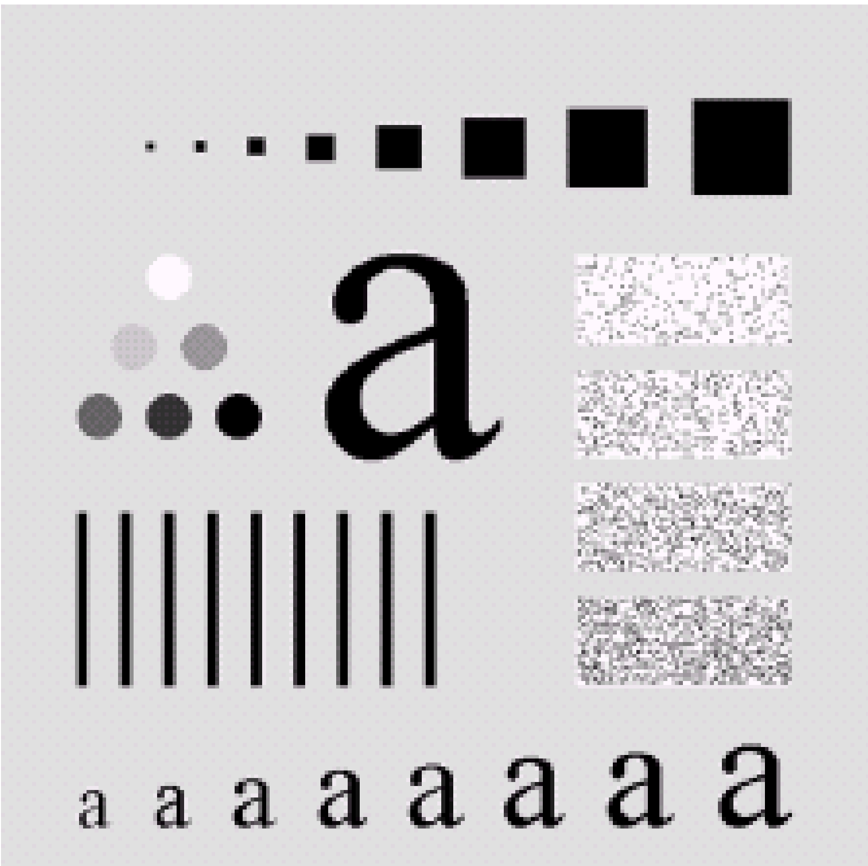
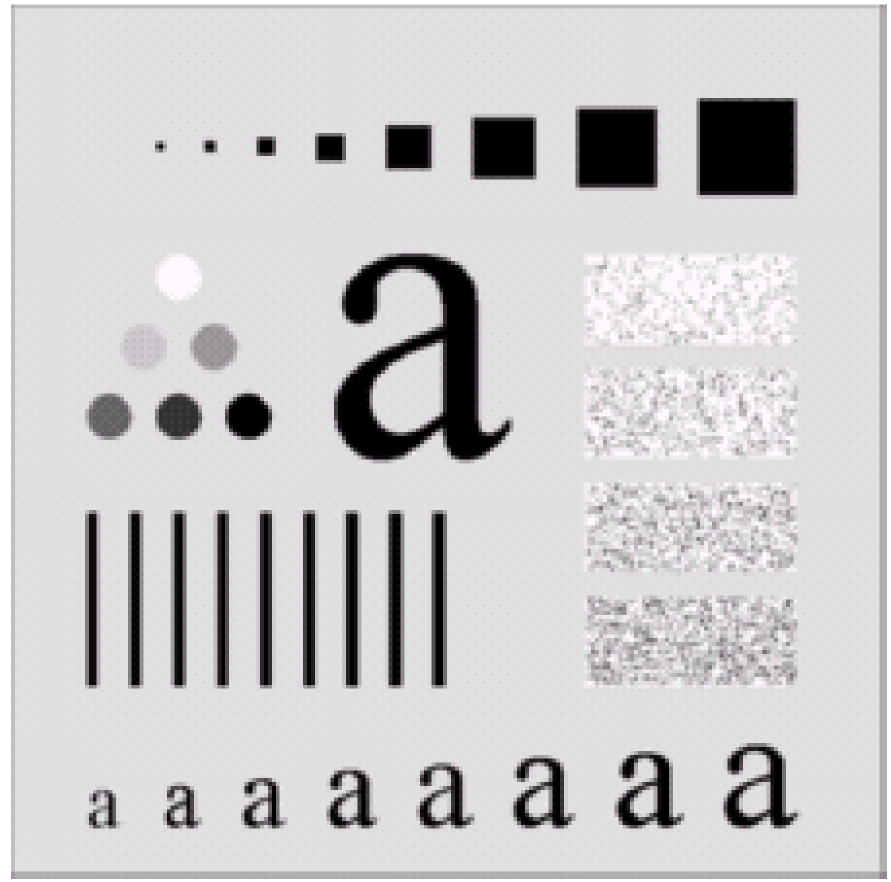


FIGURE 3.35 (a) Original image, of size 500×500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $n = 3, 5, 9, 15,$ and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. 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 rec-

Image Enhancement in the Spatial Domain: Linear lowpass filters

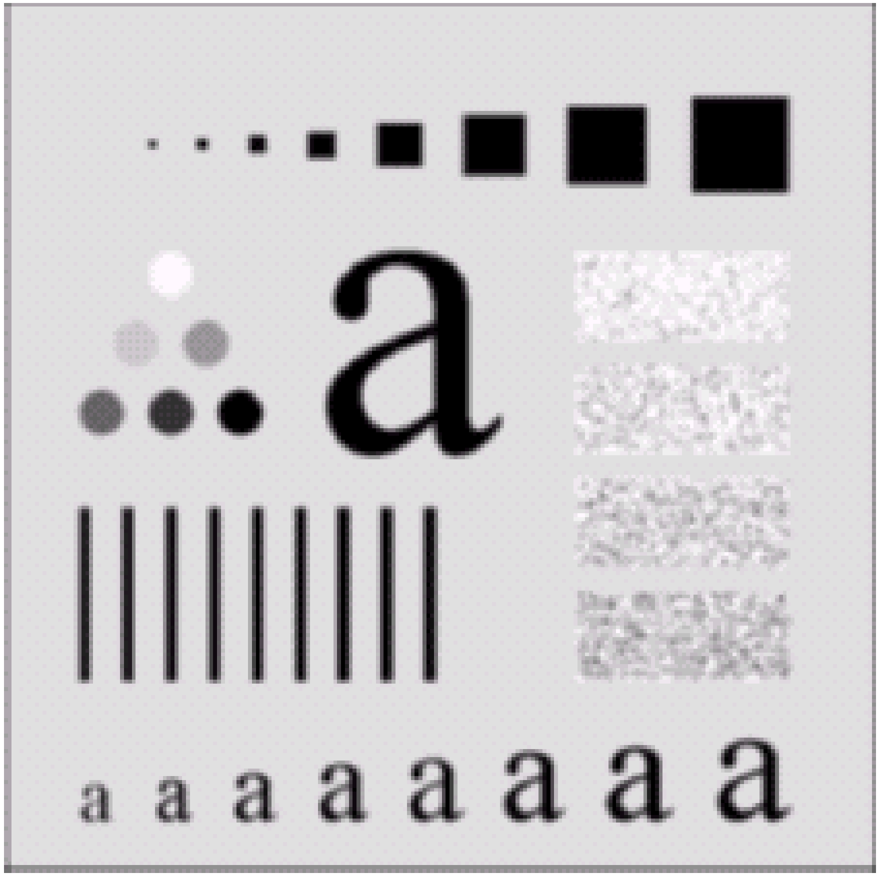


Original

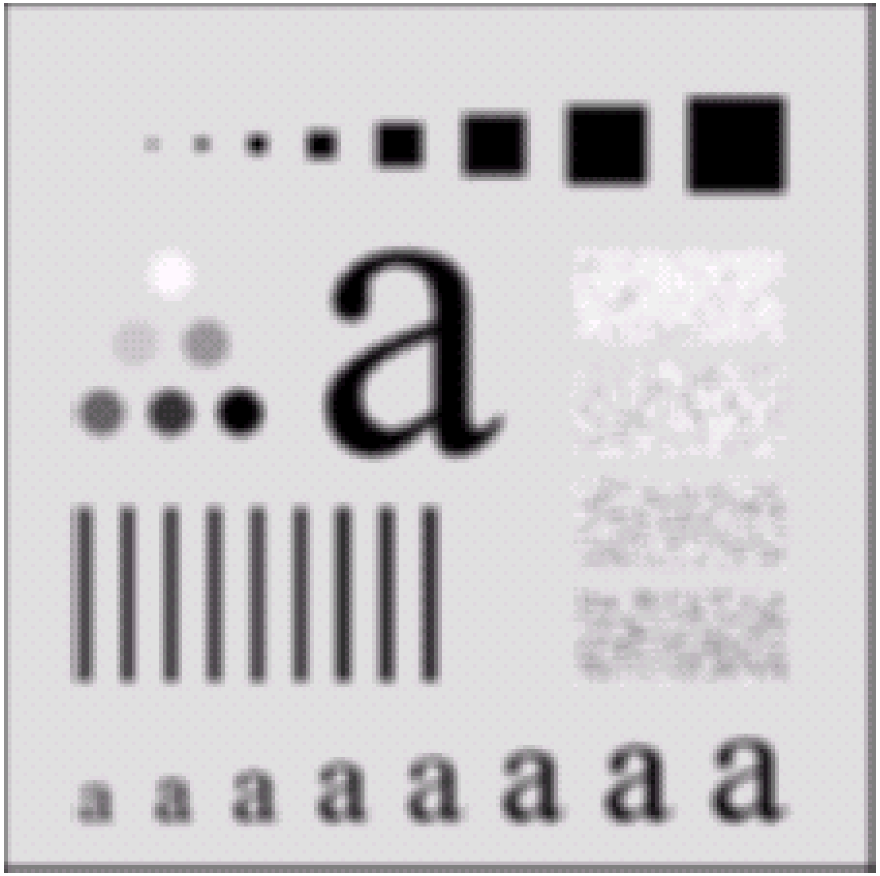


3x3

Image Enhancement in the Spatial Domain: Linear lowpass filters

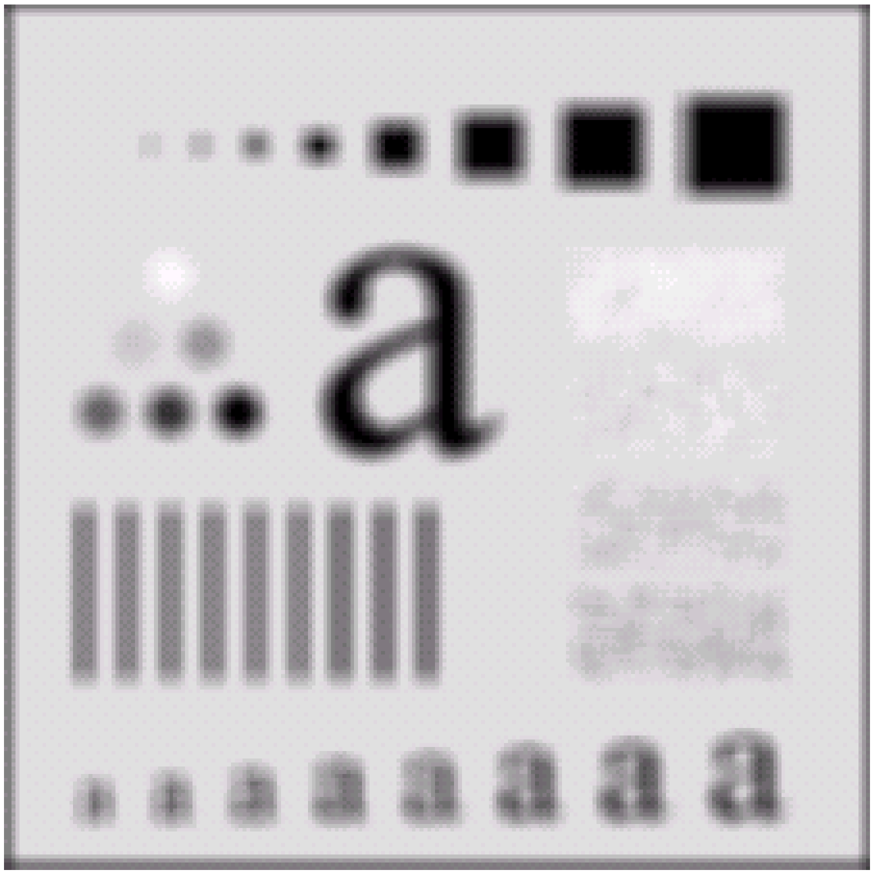


5x5

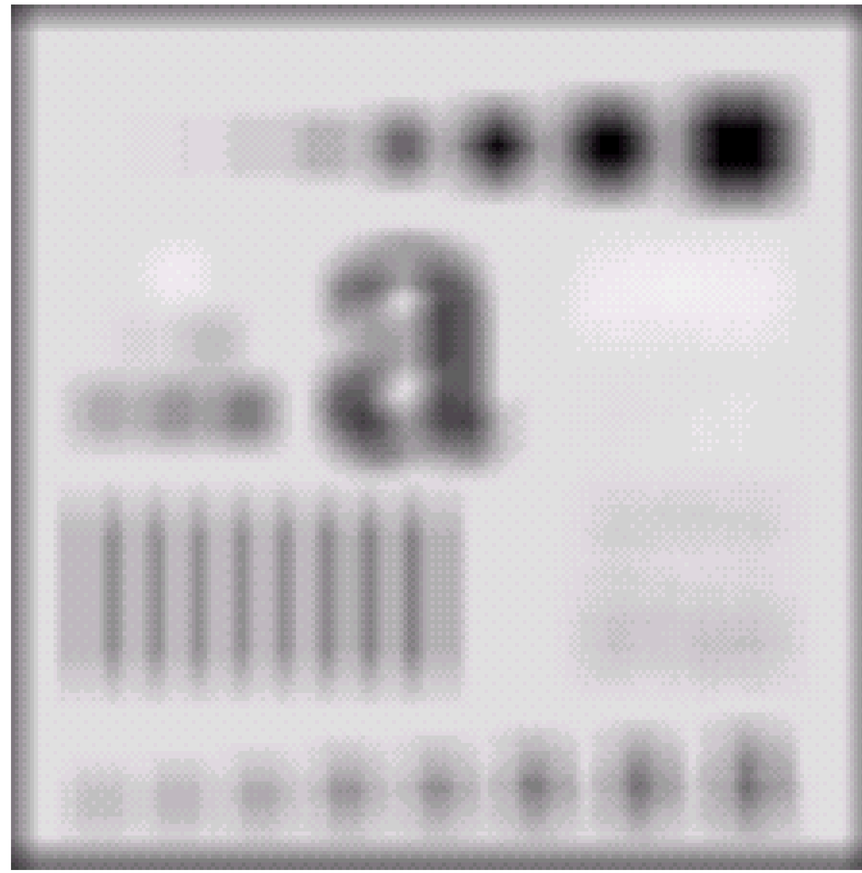


9x9

Image Enhancement in the Spatial Domain: Linear lowpass filters



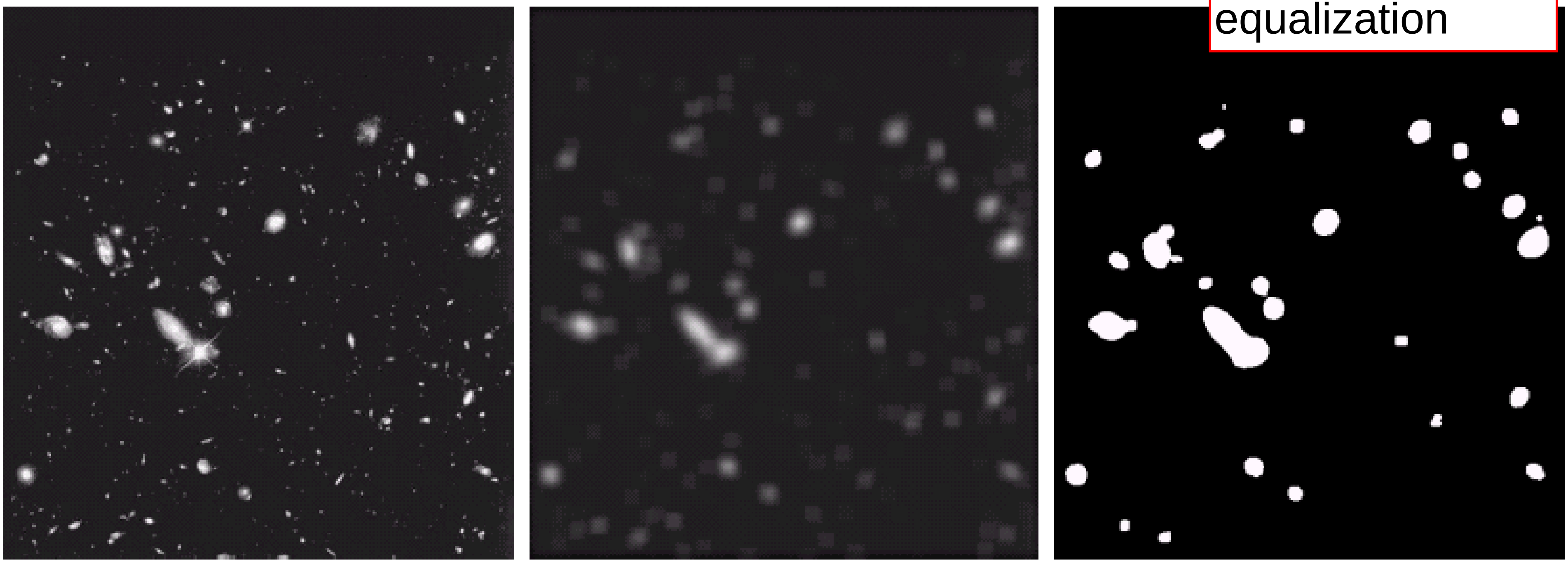
15x15



35x35

Image Enhancement in the Spatial Domain: Linear lowpass filters

another usage example:
exact histogram equalization



a b c

FIGURE 3.36 (a) Image from the Hubble Space Telescope. (b) Image processed by a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

… A first elementary result in image segmentation!

Image Enhancement in the Spatial Domain: Nonlinear "lowpass" filters

Let *Sxy* be an *mxn* neighborhood of *(x,y)*; define the **Median filter**:

$$\hat{f}(x,y) = \underset{(s,t)\in Sxy}{\mathrm{median}}\{g(s,t)\}$$

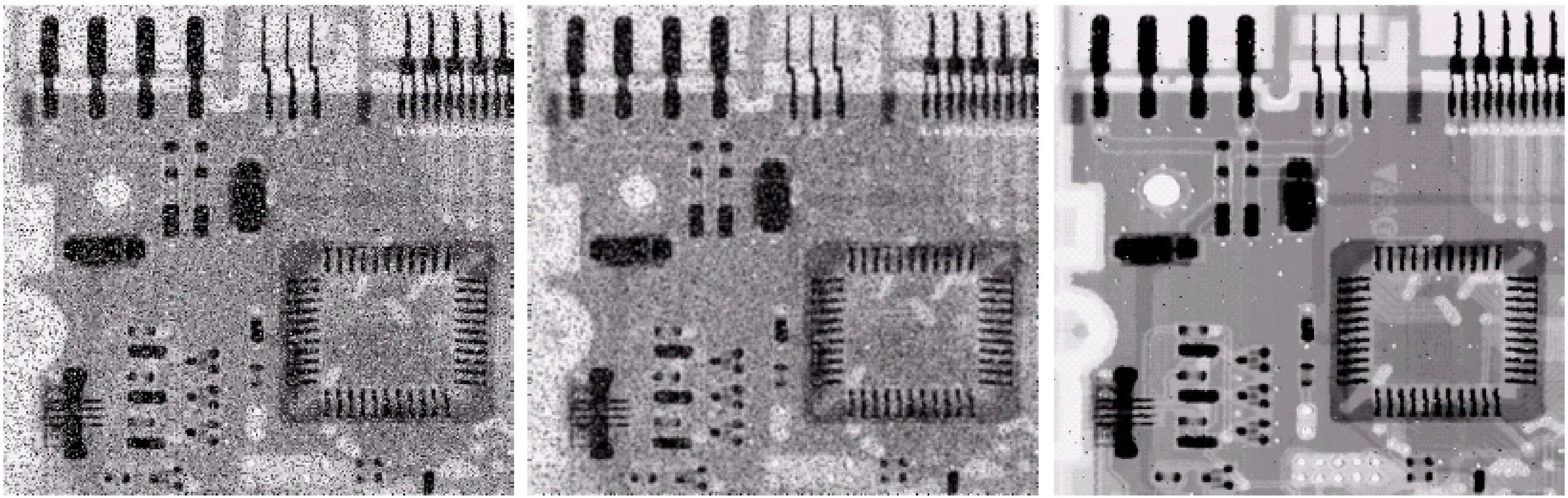
Sort the pixel values in *Sxy* and take the one in position (*mn*+1)/2

The filter can be *iteratively* applied to the data, possibly until convergence ("root signal")

Note: *mn* should be odd; if it is even one can take as output the average of the values in positions *mn*/2 and *mn*/2+1. The formal statistical properties of the filter change.

[More about order statistics later, when dealing with image restoration]

Image Enhancement in the Spatial Domain: Nonlinear "lowpass" filters



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 in the Spatial Domain: Linear **highpass** filters

Define a **1-D digital derivative** (other definitions are possible):

First-order: $$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$ Note: phase response is not zero

Second-order: $$\frac{\partial^2 f}{\partial x^2} = [f(x+1) - f(x)] - [f(x) - f(x-1)] = f(x+1) + f(x-1) - 2f(x)$$

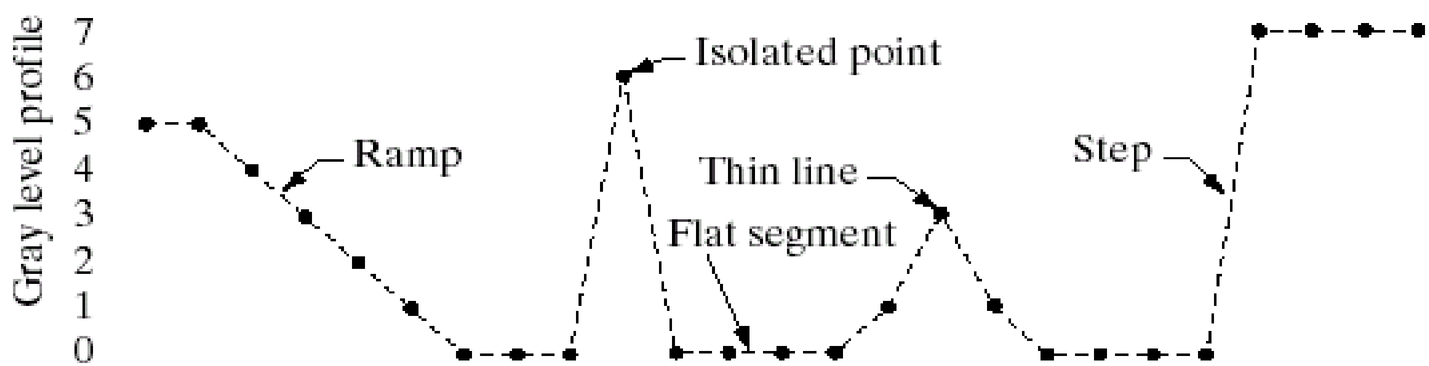


Image strip	5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7	•	•
First Derivative		−1	−1	−1	−1	−1	0	0	6	−6	0	0	0	1	2	−2	−1	0	0	0	7	0	0	0			
Second Derivative		−1	0	0	0	0	1	0	6	−12	6	0	0	1	1	−4	1	1	0	0	7	−7	0	0			

Image Enhancement in the Spatial Domain: Linear **highpass** filters

2-D case:

Gradient: $$\nabla \mathbf{f} = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right] \qquad |\nabla \mathbf{f}| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \; ; \quad \alpha = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

Laplacian: $$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$= f(x+1, y) + f(x-1, y) - 2 f(x, y) + f(x, y+1) + f(x, y-1) - 2 f(x, y)$$

Image Enhancement in the Spatial Domain: Linear highpass filters

0	1	0
1	–4	1
0	1	0

1	1	1
1	–8	1
1	1	1

0	–1	0
–1	4	–1
0	–1	0

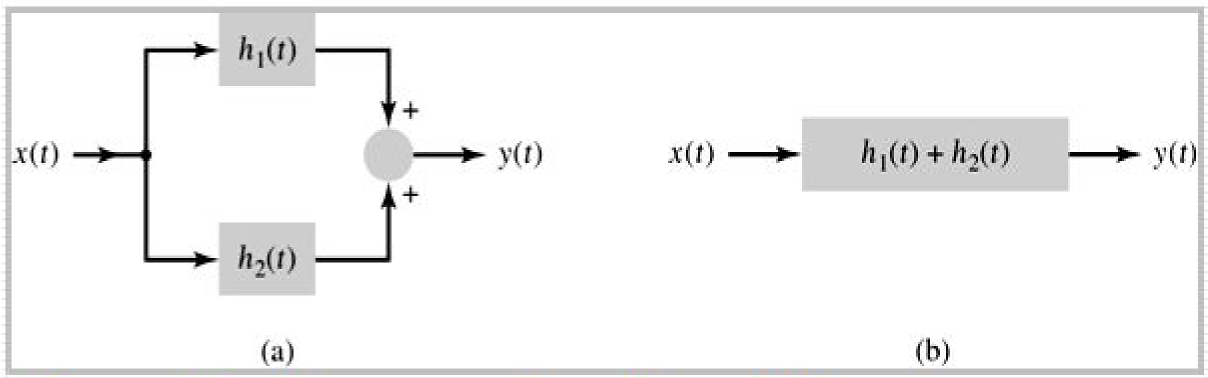
–1	–1	–1
–1	8	–1
–1	–1	–1

FIGURE 3.39 (a) Filter mask used to implement the digital Laplacian, as defined in Eq. (3.7-4). (b) Mask used to implement an extension of this equation that includes the diagonal neighbors. (c) and (d) Two other implementations of the Laplacian.

Beware: all such definitions can be found in the literature

Image Enhancement in the Spatial Domain: Linear highpass filters

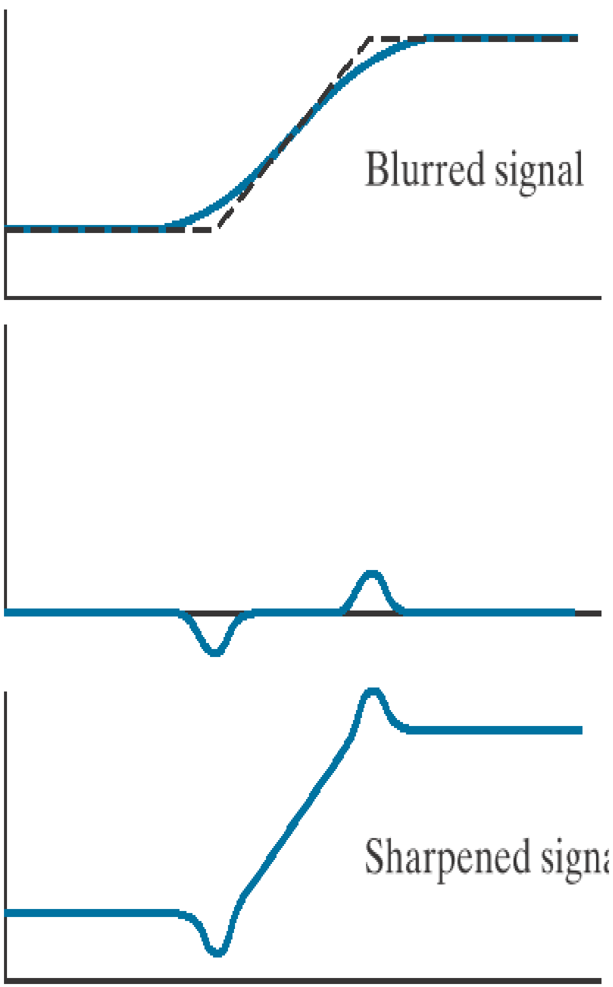
NOTE: the superposition principle always applies to linear operators



$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 \\ 0 & -2 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Image Enhancement in the Spatial Domain: Laplacian-based image **sharpening**



Add a fraction of the Laplacian of an image to the image itself:

$$g(x,y) = f(x,y) - \lambda \nabla^2 f(x,y)$$

(use '+' sign if masks in Fig.3.39 c or d are used)

This is usually named "**Unsharp Masking**":

$$g(x,y) = f(x,y) - k\, f_{LP}(x,y)$$

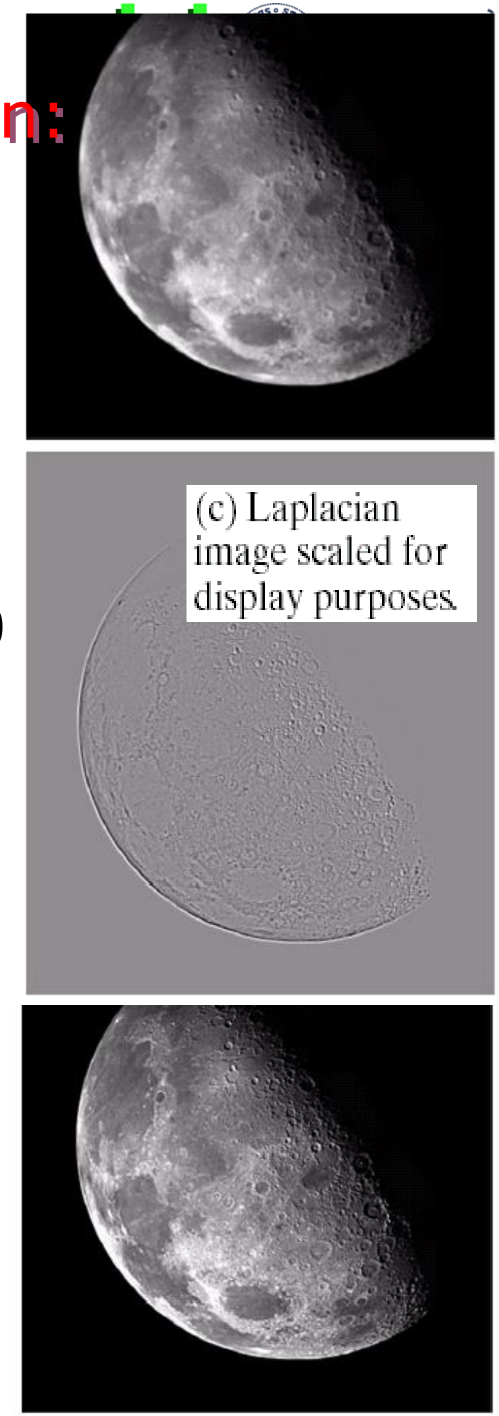
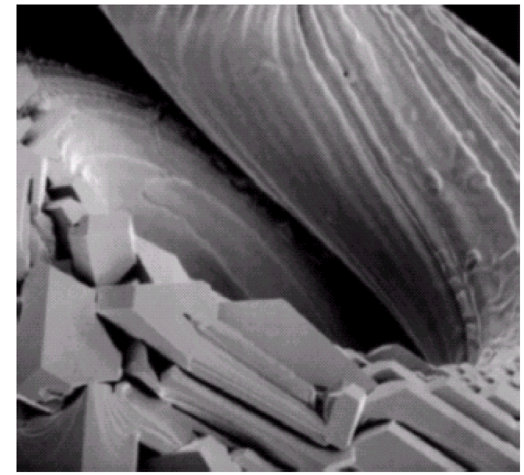


Image Enhancement in the Spatial Domain: Laplacian-based image sharpening

Note: a uniform input image remains unchanged (sum of the coeffs. is 1)

0	–1	0
–1	5	–1
0	–1	0

–1	–1	–1
–1	9	–1
–1	–1	–1



Note: for an image having amplitude 1 and frequency π in both hor. and vert. directions:

1	-1	1	-1	
-1	1	-1	1	….
1	-1	1	-1	

and if λ=1, the output amplitude is respectively:

G_1 = 5x1 + (-4)x(-1) = 9

G_2 = (9-4)x1 + (-4)x(-1) = 9

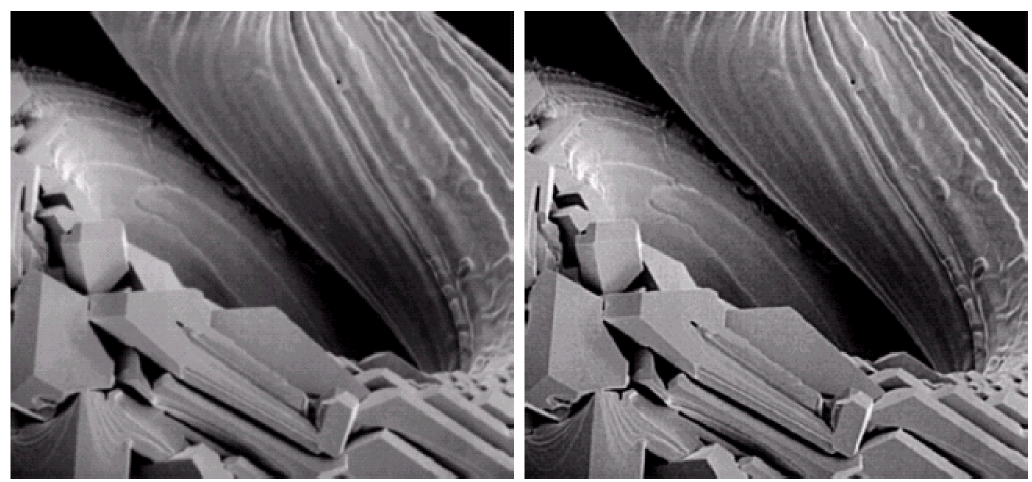


FIGURE 3.41 (a) Composite Laplacian mask. (b) A second composite mask. (c) Scanning electron microscope image. (d) and (e) Results of filtering with the masks in (a) and (b), respectively. Note how much sharper (e) is than (d). (Original image courtesy of Mr. Michael

Image Enhancement in the Spatial Domain: Laplacian-based image sharpening

High-boost filtering
Generalization of the sharpening filter
(beware: the average gray level changes!):

0	−1	0
−1	$A+4$	−1
0	−1	0

−1	−1	−1
−1	$A+8$	−1
−1	−1	−1

FIGURE 3.42 The high-boost filtering technique can be implemented with either one of these masks, with $A \geq 1$.

Image Enhancement in the Spatial Domain: Laplacian-based image sharpening

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 $A = 1.7$.

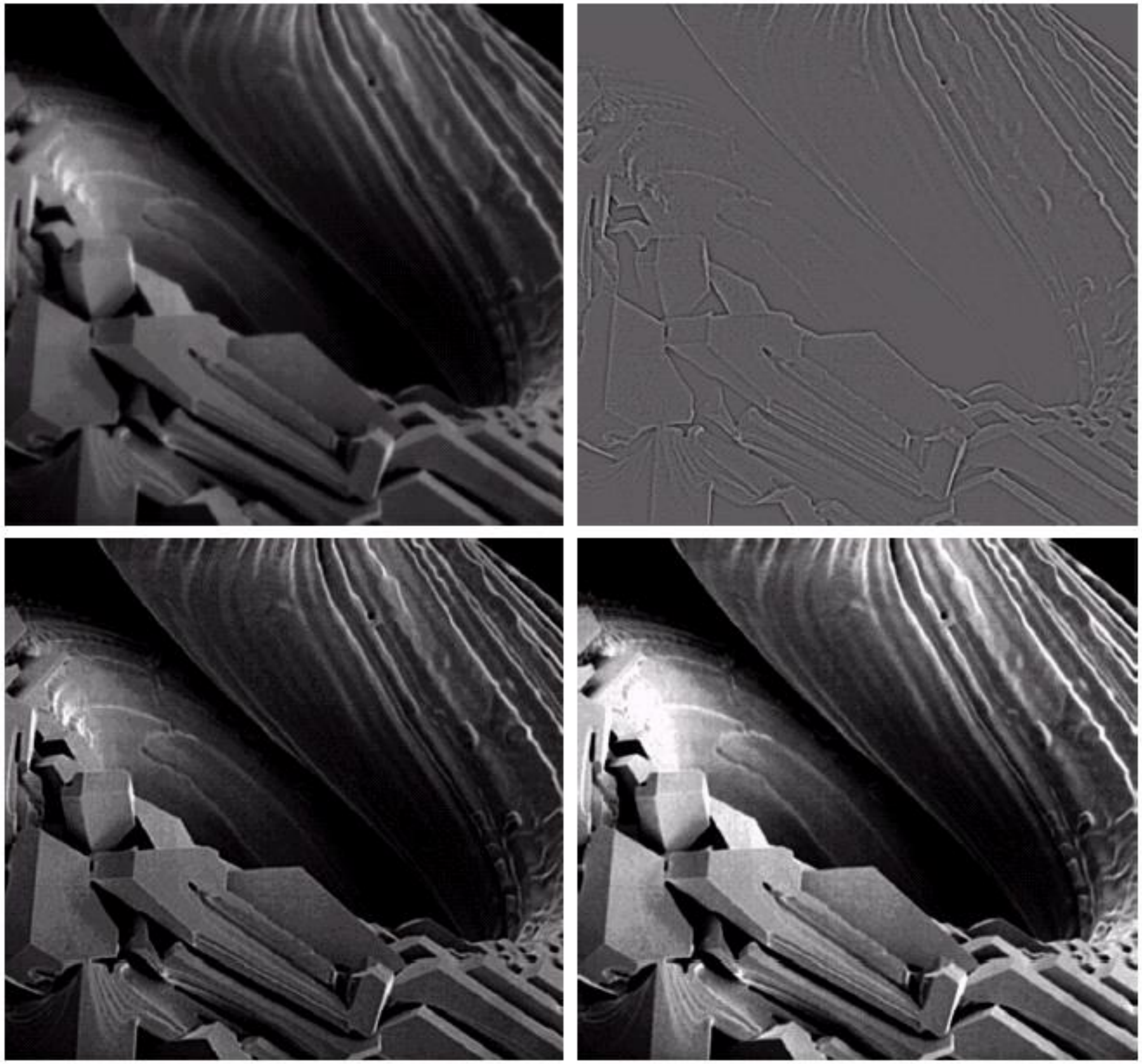


Image Enhancement in the Spatial Domain: Gradient-based image sharpening

Estimation of the gradient $|\nabla \mathbf{f}| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$

Wrong, but useful solutions

Roberts

−1	0
0	1

0	−1
1	0

Sobel

−1	−2	−1
0	0	0
1	2	1

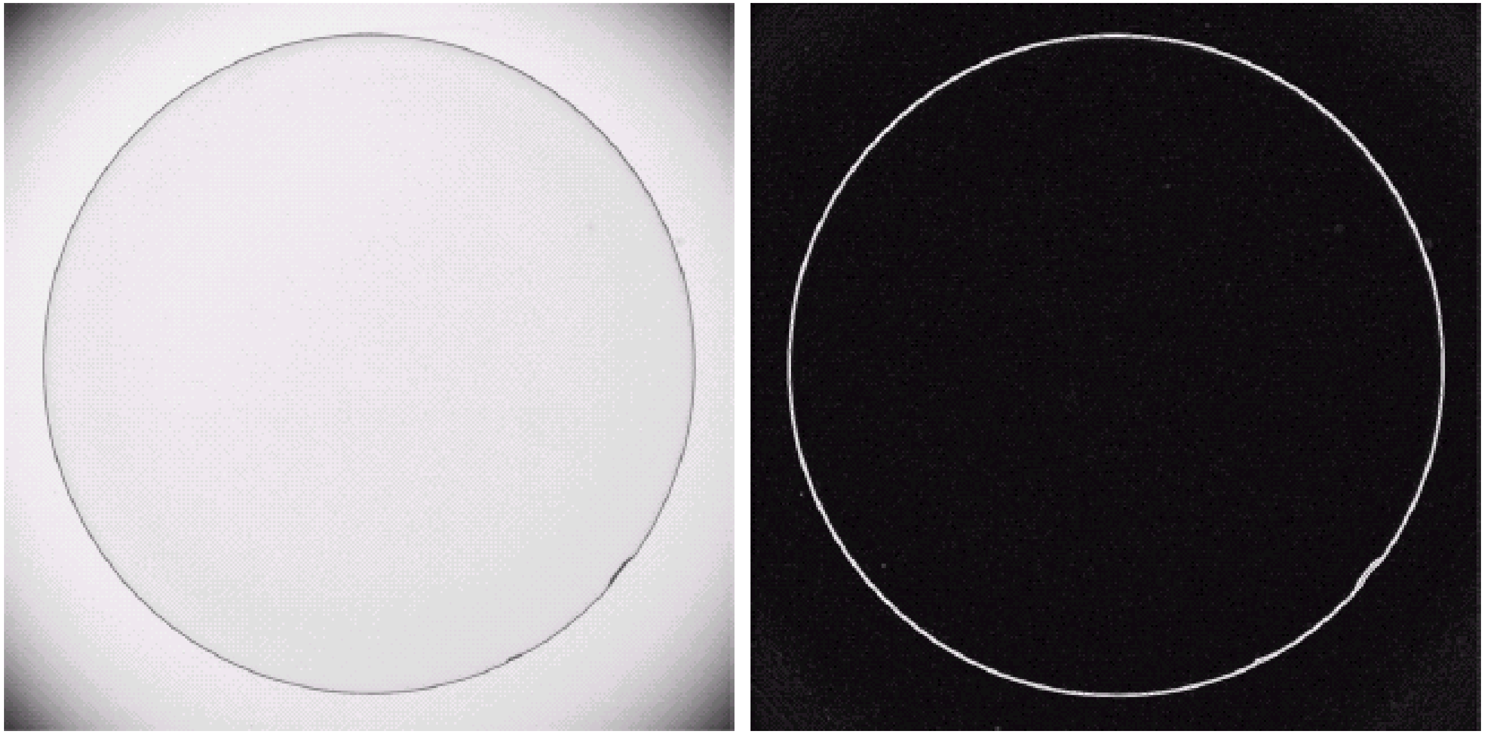
−1	0	1
−2	0	2
−1	0	1

Note: a uniform input image yields 0 (sum of the coeffs. is 0)

1	-1	1	-1
-1	1	-1	1
1	-1	1	-1

Note: response to frequency π is 0 too

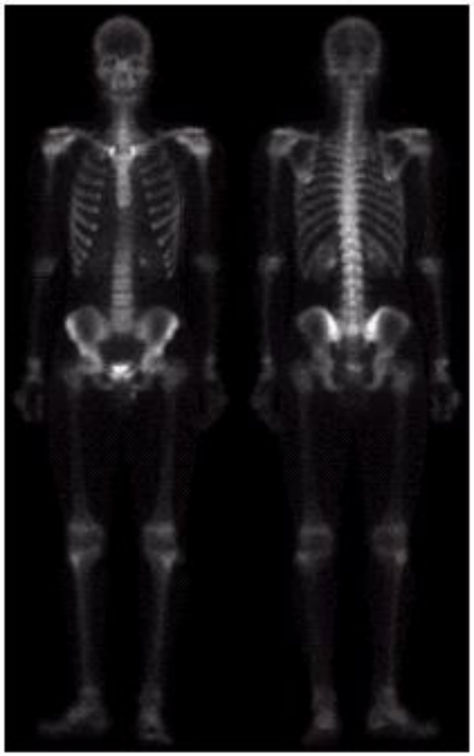
Image Enhancement in the Spatial Domain: Gradient-based image sharpening



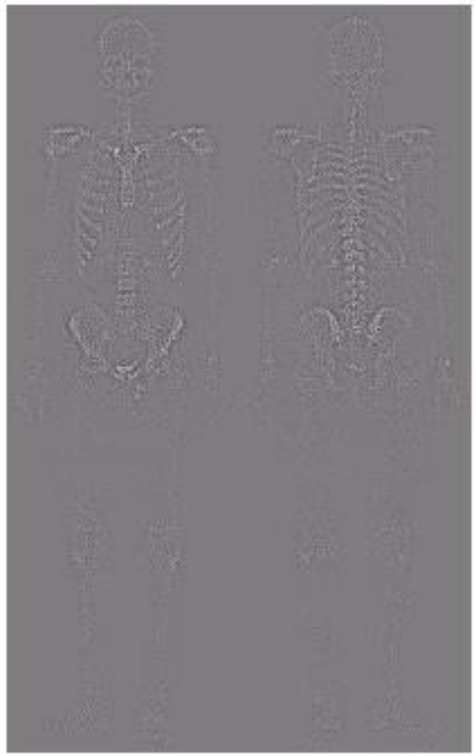
a b

FIGURE 3.45
Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock). (b) Sobel gradient. (Original image courtesy of Mr. Pete Sites, Perceptics Corporation.)

Image Enhancement in the Spatial Domain: Gradient- and Laplacian-based image sharpening



a =
Original (NMR)

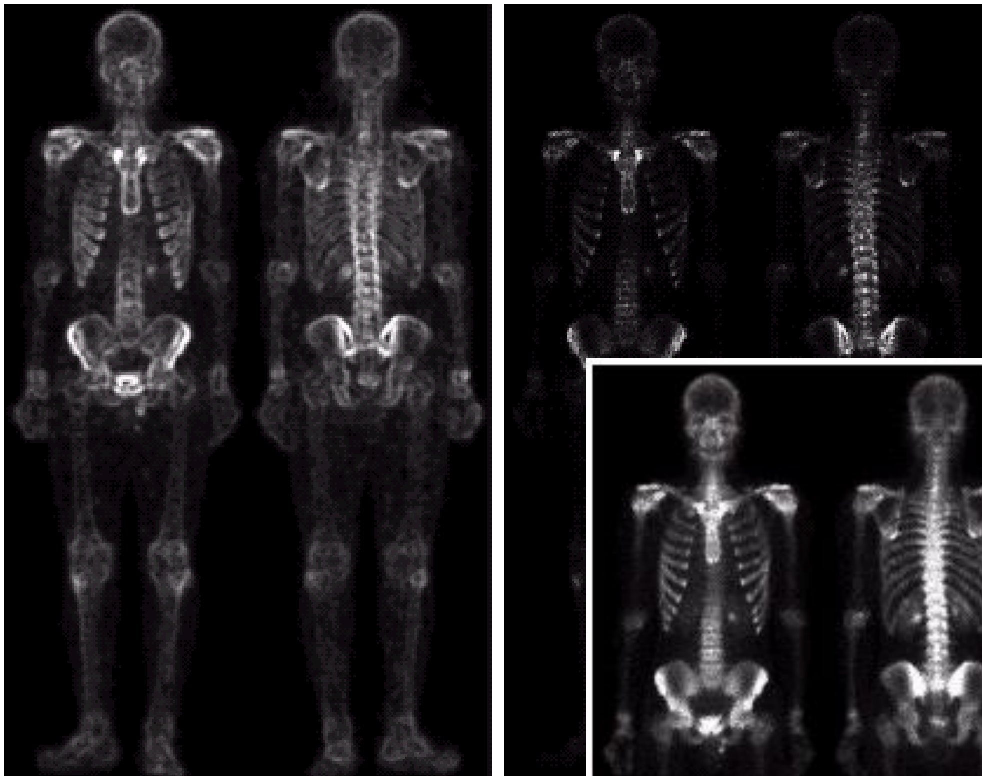


b =
Laplacian(a)

c =
a+b

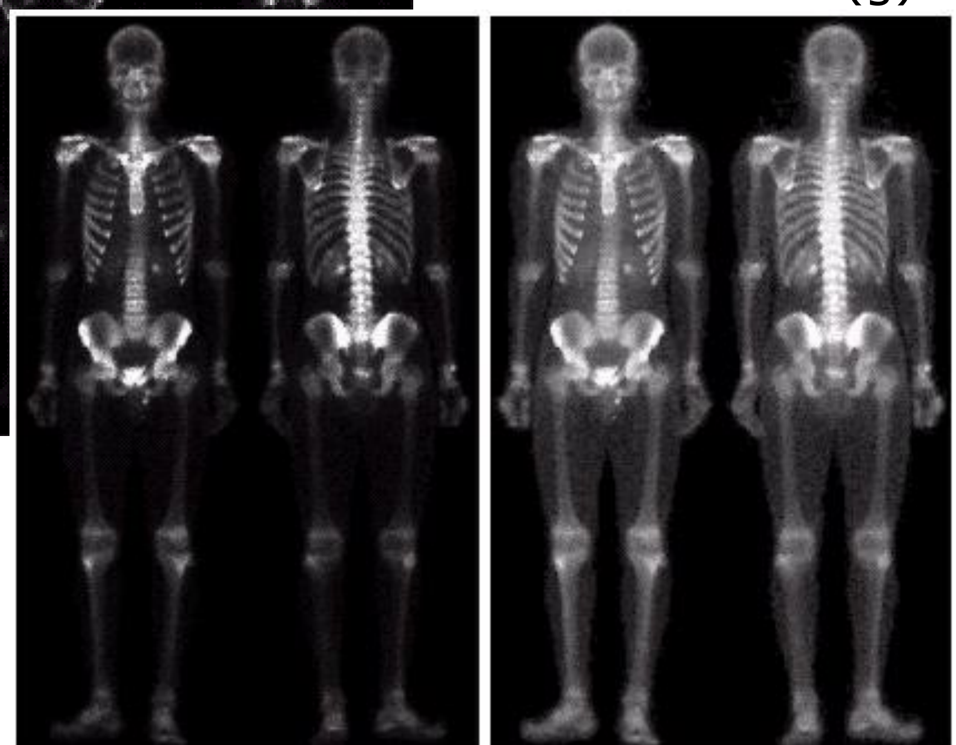
d =
Sobel(a)

Image Enhancement in the Spatial Domain: Gradient- and Laplacian-based image sharpening



f = b . e

e = LP(d)



h = T(g)

g = a + f

FIGURE 3.46 *(Continued)* (e) Sobel image smoothed with a 5 × 5 averaging filter. (f) Mask image formed by the product of (b) and (e). (g) Sharpened image obtained by the sum of (a) and (f). (h) Final result obtained by applying a power-law transformation to (g). Compare (g) and (h) with (a).

Image Enhancement in the Spatial Domain: **Rational** unsharp masking

	d	
a	*b*	*c*
	e	

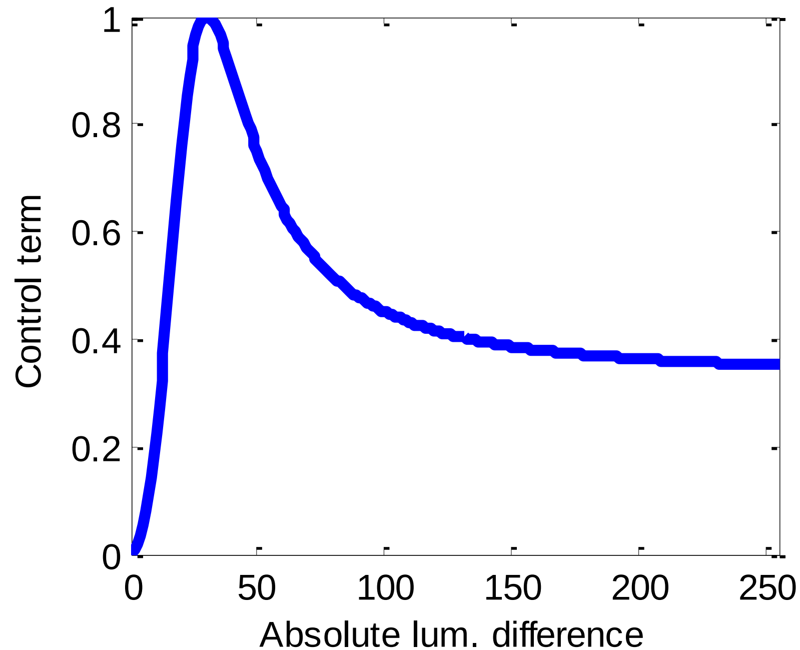
$$z_x = (a-c)^2/\mu^2 \qquad z_y = (d-e)^2/\mu^2$$

$$b' = b + \lambda\left(\frac{(2b-a-c)(z_x^2+4z_x)}{3z_x^2+2} + \frac{(2b-d-e)(z_y^2+4z_y)}{3z_y^2+2}\right)$$

A control term avoids noise amplification and excessive amplification of sharp and large edges.

Peak amplification is 1 for $z_x = z_y = 1$

peak position is controlled by μ



See also: DL_BUM, Blur-guided UM

Retinex-based algorithms

A specific, simple version of a multi-faceted approach

→ Retinex_at_50_McCann17.pdf

Problem: image enhancement

- Poorly illuminated images
- Backlit objects
- Light sources within the image

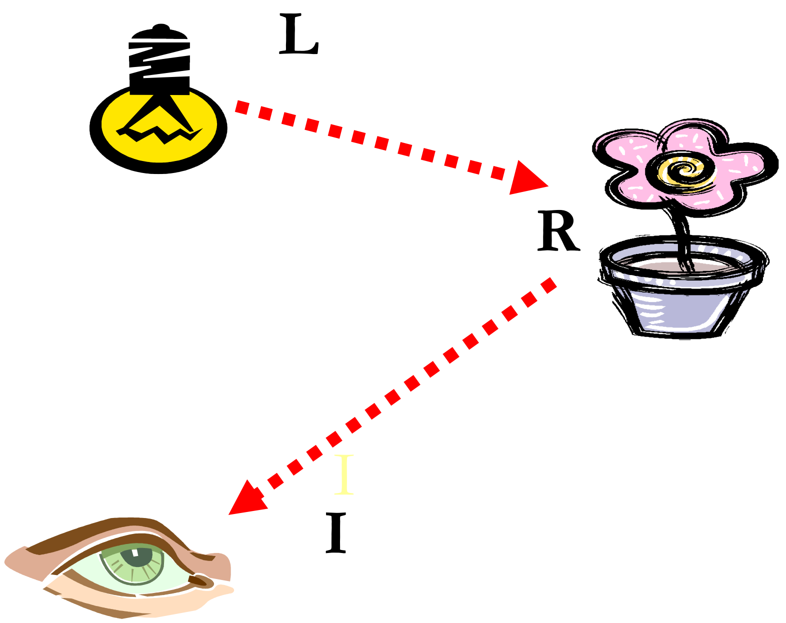
Solution:

- Retinex-based approaches increase the brightness in dark areas and emphasize the details

Courtesy Dr. Stefano Marsi

Retinex-based algorithms

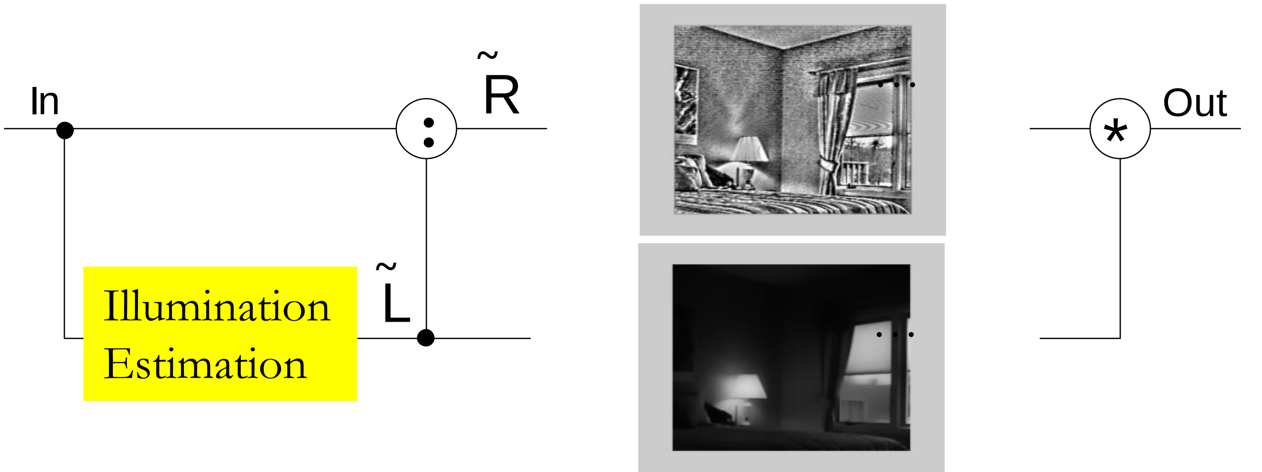
- Retinex = Retina + Cortex
- An image **I** can be considered as the product of the scene illumination **L** and the objects reflectance **R**
- The *perception* of an object is, to a certain extent, independent of the illumination conditions



$$I(x,y) = L(x,y)\,R(x,y)$$

Retinex-based algorithms

- Through an estimation of the illumination it is possible to obtain an approximation of the reflectance.



- The components are separately processed and then recombined
  - The reflectance (details) is suitably emphasized
  - The illumination is increased in dark areas
- Logarithmic data are typically used (as provided by HDR sensors, or deliberately transformed):
  - The product is replaced by an addition block
  - The division is replaced by a subtraction block

Retinex-based algorithms: Estimation of the illumination

Illumination characteristics:

- The illumination typically changes very smoothly among contiguous pixels
- Abrupt transitions can also appear
  - in presence of light sources in the image
  - in presence of different illumination systems (e.g. backlights)

The illumination should be estimated using a
narrow band and edge-preserving lowpass filter

Linear narrow band filters are realized:

- via very large masks
- via recursive filtering
- via multi-resolution decomposition

Retinex-based algorithms: Estimation of the illumination

We can use a (nonlinear) *Recursive Rational Filter* (RRF)

A simple first order IIR filter is

$$y(n) = (1-a)x(n) + ay(n-1)$$

a can be used to trim the bandwidth:

-- $a \rightarrow 1$: very narrow passband

-- $a = 0$: the filter is switched off

Let's make this filter nonlinear using a ratio of suitable terms

Retinex-based algorithms: Estimation of the illumination

$$y(n,m)=\frac{\alpha[y(n-1,m)S_v+y(n,m-1)S_h]+[(S_v+S_h)(1-\alpha)+1]x(n,m)}{S_v+S_h+1}$$

where

$$S_h=\frac{A}{\left(\log\frac{1+x(n,m-1)}{1+x(n,m+1)}\right)^2+\delta} \qquad S_v=\frac{A}{\left(\log\frac{1+x(n-1,m)}{1+x(n+1,m)}\right)^2+\delta}$$

- *Sh* and *Sv* resp. are the horizontal and vertical *uniformity sensors*
- Note: filtering has to be applied forward and backward to get zero-phase overall response

Retinex-based algorithms: Estimation of the illumination

- It is manifest that the output can include a sharp “map” of bright objects in the image, not necessarily a map of light sources

Retinex-based algorithms: Processing the illumination

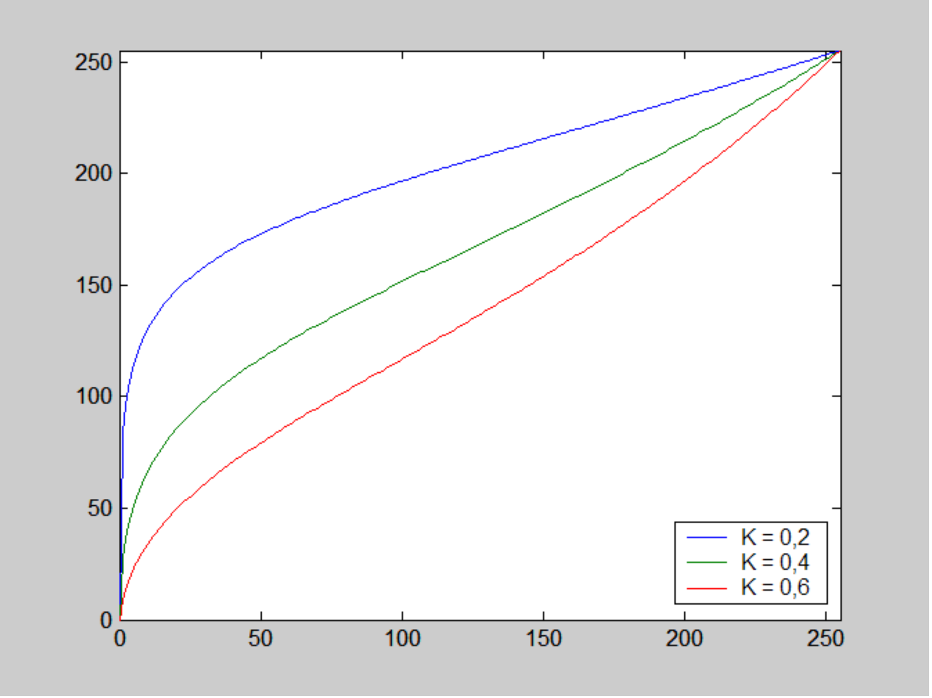
A modified gamma function is used …

- To increase the brightness in dark zones
- To avoid excessive dynamics compression in bright areas

… possibly followed by histogram stretching

- To better exploit the system dynamics

$$\Gamma(\tilde{L}) = 255 \left(\tilde{L} / 255 \right)^{K(\tilde{L}/255)+K}$$



Retinex-based algorithms: Processing the reflectance

In the log domain, the derived reflectance is a zero-mean image of local differences

A sigmoid-like function is used to process the reflectance component, in order to:

- emphasize the details when these are poorly defined
- limit the emphasis when the details are already well defined, to avoid artifacts generation
- reduce the signal when it is extremely weak (the information is superseded by the noise)

Retinex-based algorithms: Processing the reflectance

The adopted function is:

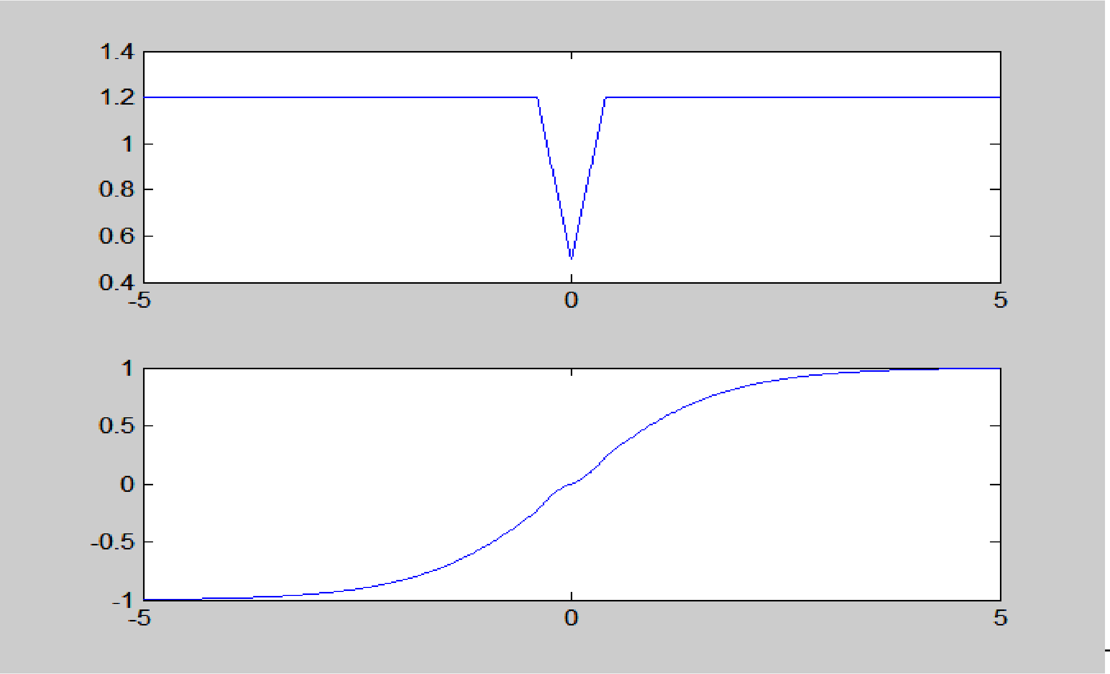
$$R_2 = 2K\left(\frac{1}{1+\exp(-c(R_1)R_1)} - \frac{1}{2}\right)$$

where c is a suitable coefficient which controls the slope of the sigmoid and creates the central *dead zone*

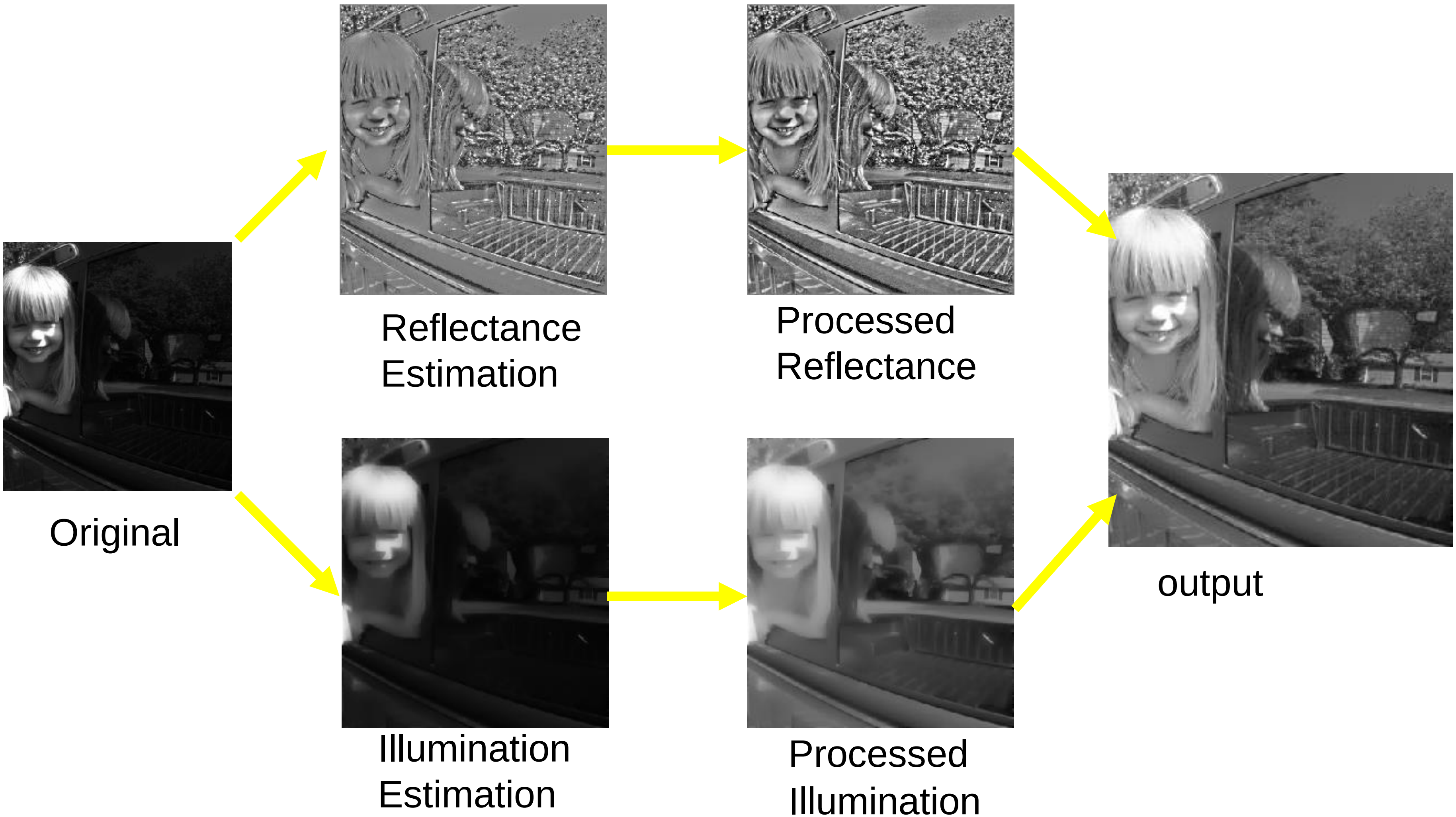
Dead zone term

$c(R_1)$

Overall reflectance mapping $R_2(R_1)$



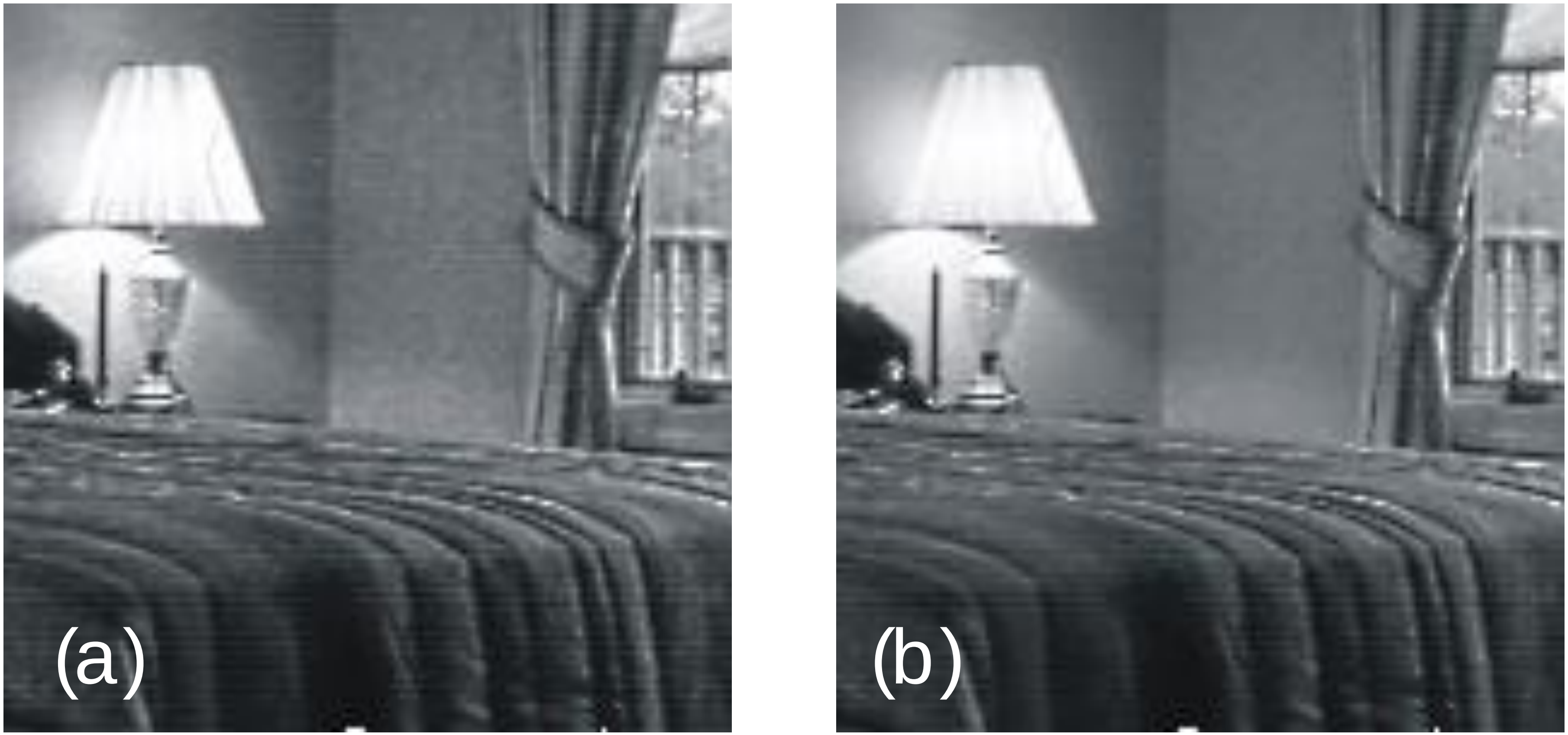
Retinex-based algorithms: Processing example



Retinex-based algorithms: a comparison

1. Original
2. Histogram equalization
3. Gamma correction
4. Retinex-based method

Retinex-based algorithms: Noise control



without (a) and with (b) the dead zone

Retinex-based algorithms: More images…

Retinex-based algorithms: More images...

See also: Blur-guided UM