Announcement

Homework #2 was posted online.

Homework #2 is due 2:20pm, Wednesday, Feb. 7.

Announcement

Quiz #1

Time and Date: 3:20pm – 3:35pm, Wednesday, Jan. 31

Topic: Histogram processing

Open book and open notes and you can use a calculator

Today's Agenda

- **Histogram processing**
  - Histogram equalization
  - Histogram matching
- **Spatial filtering**
  - Linear filters
    - Image Smoothing

Histogram Equalization

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw$$

$$\Rightarrow \quad p_s(s) = \frac{1}{L-1}$$

How to prove it?

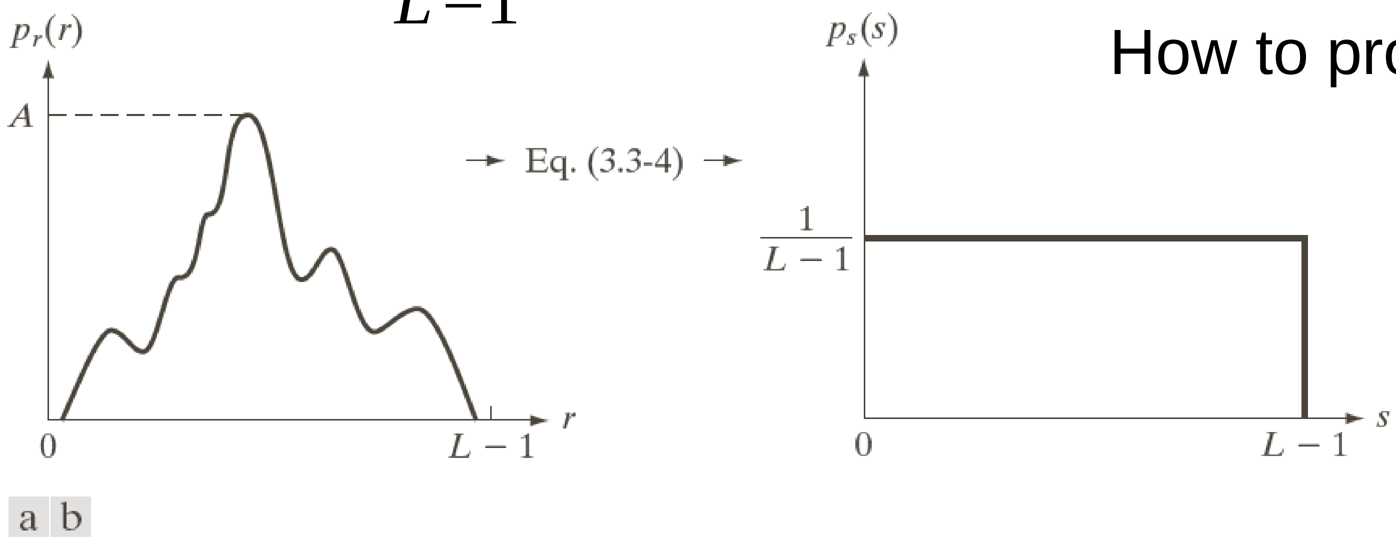


FIGURE 3.18 (a) An arbitrary PDF. (b) Result of applying the transformation in Eq. (3.3-4) to all intensity levels, r. The resulting intensities, s, have a uniform PDF, independently of the form of the PDF of the r's.

Histogram Equalization – Discrete Case

$$p_r(r_k) = n_k / MN, k = 0,1,2,..., L-1$$

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = \frac{L-1}{MN}\sum_{j=0}^{k} n_j$$

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

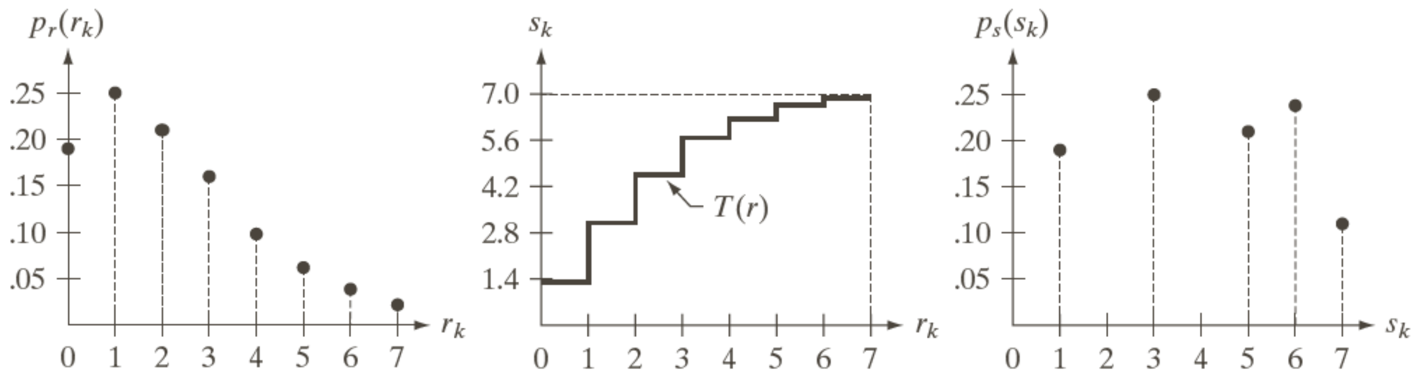
TABLE 3.1 Intensity distribution and histogram values for a 3-bit, 64×64 digital image.

S_k is a monotonic increasing function

Histogram Equalization – Discrete Case

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

TABLE 3.1 Intensity distribution and histogram values for a 3-bit, 64×64 digital image.



a b c

FIGURE 3.19 Illustration of histogram equalization of a 3-bit (8 intensity levels) image. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

Histogram equalization is not guaranteed to result in a uniform histogram.

Examples

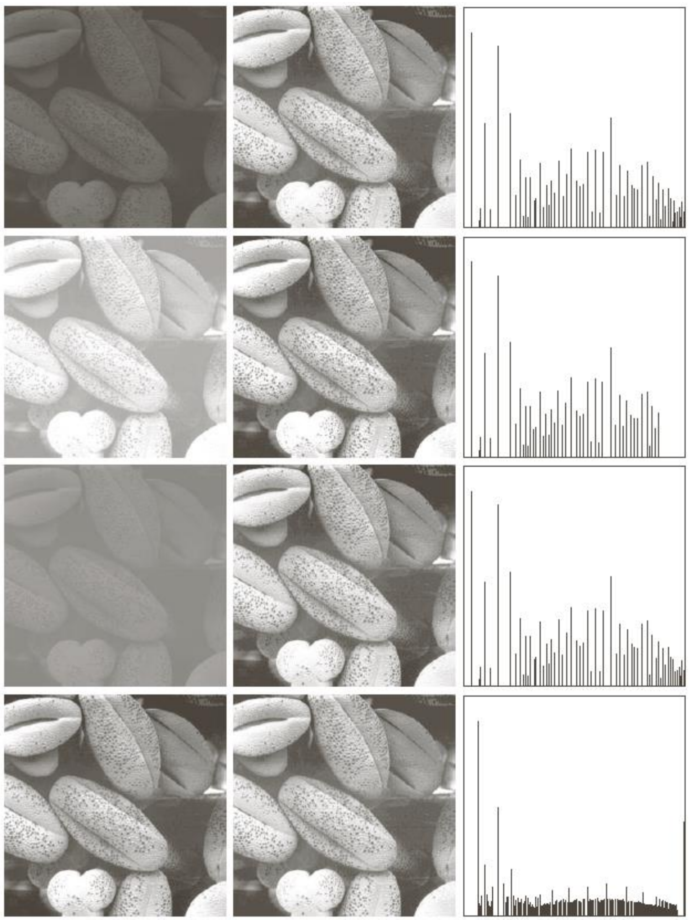


FIGURE 3.20 Left column: images from Fig. 3.16. Center column: corresponding histogram-equalized images. Right column: histograms of the images in the center column.

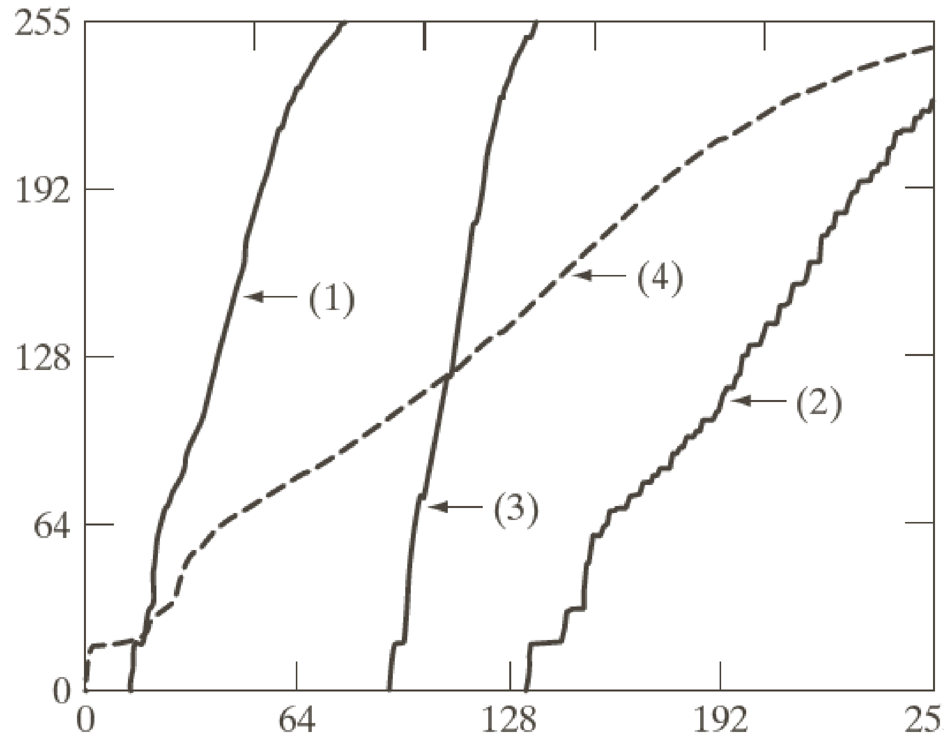
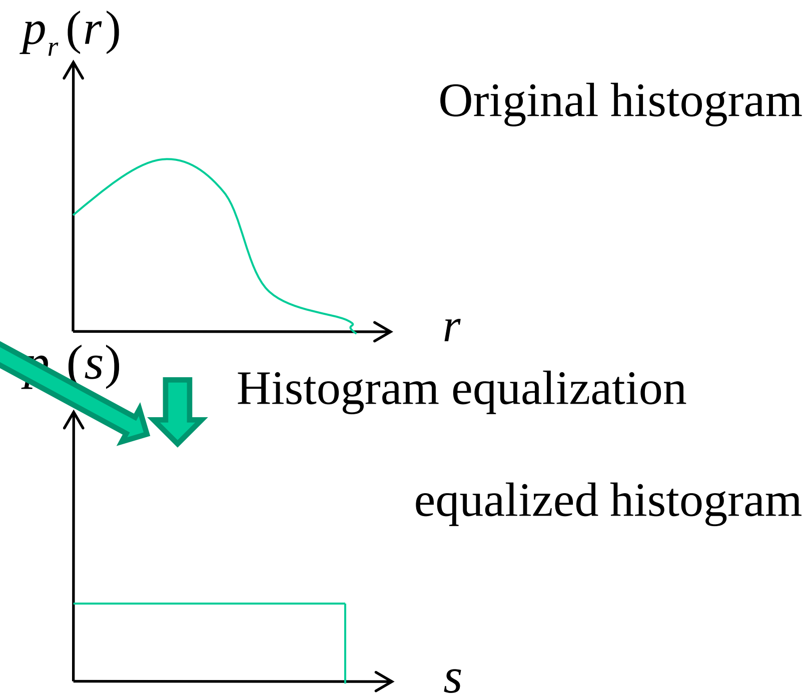


FIGURE 3.21 Transformation functions for histogram equalization. Transformations (1) through (4) were obtained from the histograms of the images (from top to bottom) in the left column of Fig. 3.20 using Eq. (3.3-8).

Histogram Matching (Specification)

Estimated pdf in the input image
↑

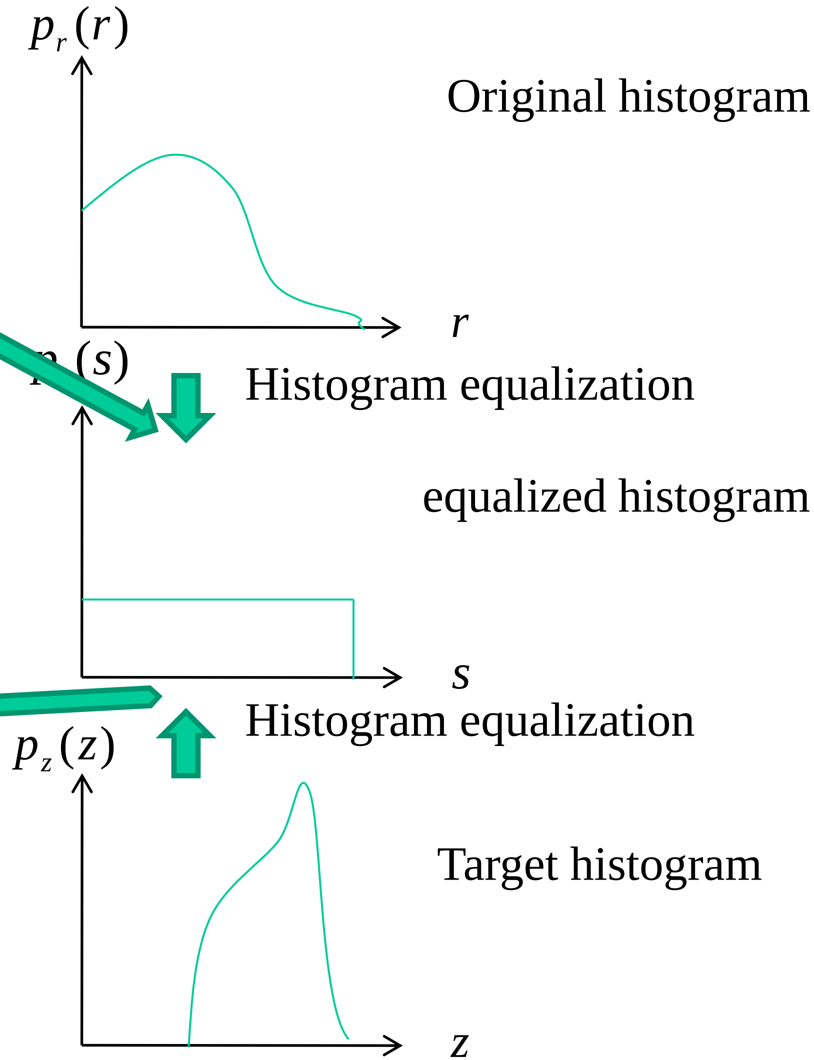
$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw$$



Histogram Matching (Specification)

$$s = T(r) = (L-1)\int_0^r p_r(w)dw$$

$$s = G(z) = (L-1)\int_0^z p_z(t)dt$$



Histogram Matching (Specification)

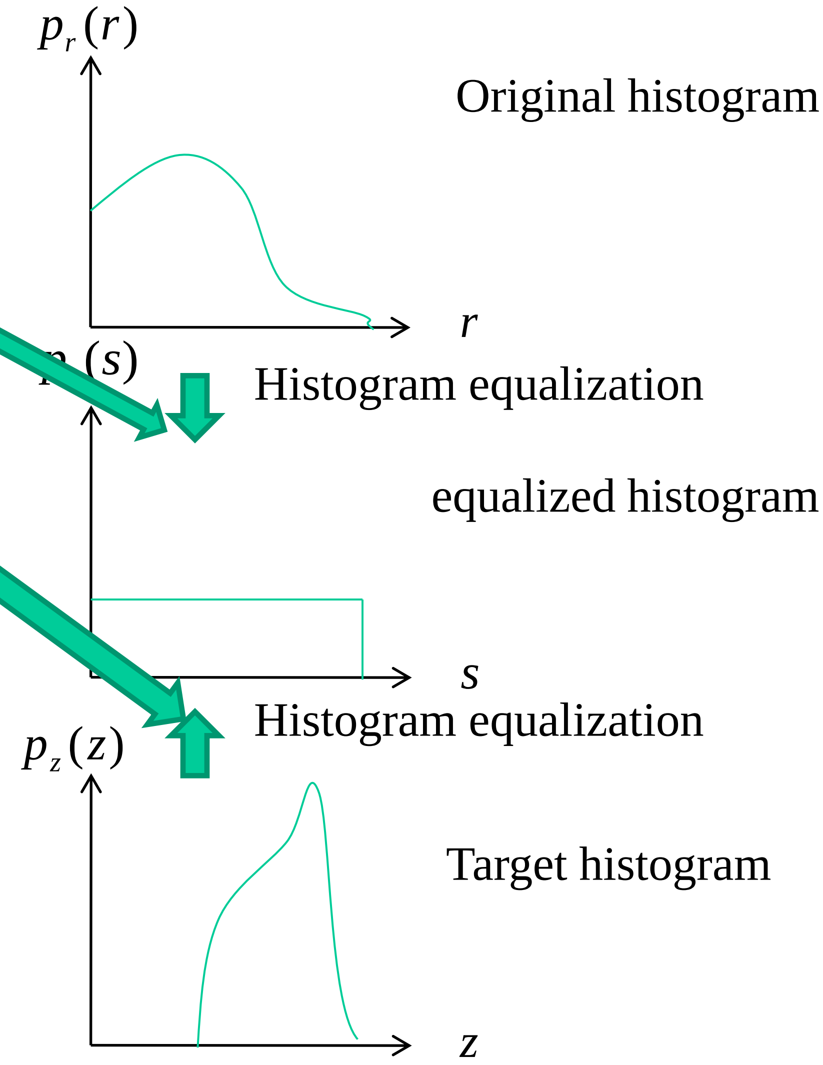
Estimated pdf in the input image

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw$$

$$s = G(z) = (L-1)\int_0^z p_z(t)\,dt$$

Desired pdf in the output image

$$z = G^{-1}(s) = G^{-1}(T(r))$$



Green circles represent known
Red circles represent unknown

Histogram Matching Algorithm for Continuous Data

Obtain the output image by:

- First compute the probability distribution function of input data $p_r(r)$
- Perform histogram equalization → $s = T(r)$
- Compute $s = G(z)$, where G is the equalization function derived from a specified histogram
- Perform the inverse mapping $z = G^{-1}(s) = G^{-1}(T(r))$
- The output image with z values is then of the specified histogram

A Continuous Example

$$p_r(r) = \begin{cases} \dfrac{2r}{(L-1)^2} & 0 \le r \le (L-1) \\ 0 & otherwise \end{cases}$$

$$p_z(z) = \begin{cases} \dfrac{3z^2}{(L-1)^3} & 0 \le z \le (L-1) \\ 0 & otherwise \end{cases}$$

Compute z?

Histogram Matching Algorithm – Discrete Image

Discrete histogram require a discretization of the output intensity values

Step1: Compute histogram of the input image $p_r(r)$ and the histogram equalized image $s = T(r)$

Step2: Compute $G(z)$ given the desired histogram $p_z(z)$

Ideally, $G(z) = s$. In practice, $G(z) \approx s$

Step3: Given the s_k value, find the value of z_q so that $G(z_q)$ is closest to s_k

Step4: form the histogram-specified image using the mapping r-z found above

A Discrete Example

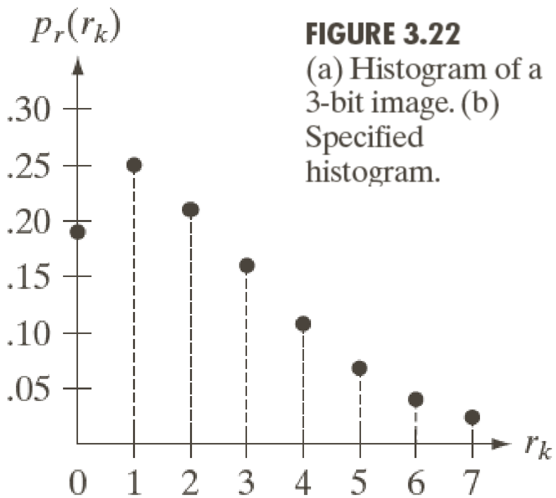
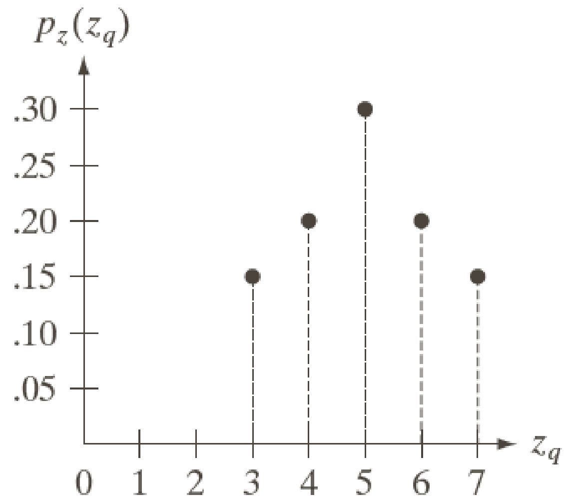


FIGURE 3.22
(a) Histogram of a 3-bit image. (b) Specified histogram.

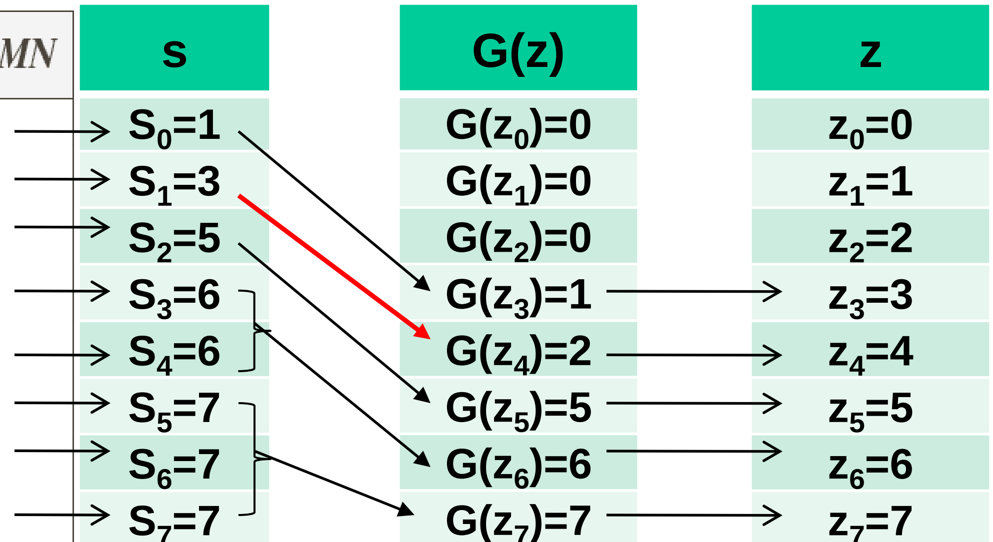


r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

z_q	Specified $p_z(z_q)$
$z_0 = 0$	0.00
$z_1 = 1$	0.00
$z_2 = 2$	0.00
$z_3 = 3$	0.15
$z_4 = 4$	0.20
$z_5 = 5$	0.30
$z_6 = 6$	0.20
$z_7 = 7$	0.15

A Discrete Example – Cont.

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02



$r_0 \rightarrow z_3$

$r_1 \rightarrow z_4$

$r_2 \rightarrow z_5$

$r_3, r_4 \rightarrow z_6$

$r_5, r_6, r_7 \rightarrow z_7$

z_q	Specified $p_z(z_q)$	Actual $p_z(z_k)$
$z_0 = 0$	0.00	0.00
$z_1 = 1$	0.00	0.00
$z_2 = 2$	0.00	0.00
$z_3 = 3$	0.15	0.19
$z_4 = 4$	0.20	0.25
$z_5 = 5$	0.30	0.21
$z_6 = 6$	0.20	0.24
$z_7 = 7$	0.15	0.11

Histogram Matching Algorithm – Discrete Image

Discrete histogram require a discretization of the output intensity values

Step1: Compute histogram of the input image $p_r(r)$ and the histogram equalized image $s = T(r)$

Step2: Compute $G(z)$ given the desired histogram $p_z(z)$

Ideally, $G(z) = s$. In practice, $G(z) \approx s$

Step3: Given the s_k value, find the value of z_q so that $G(z_q)$ is closest to s_k

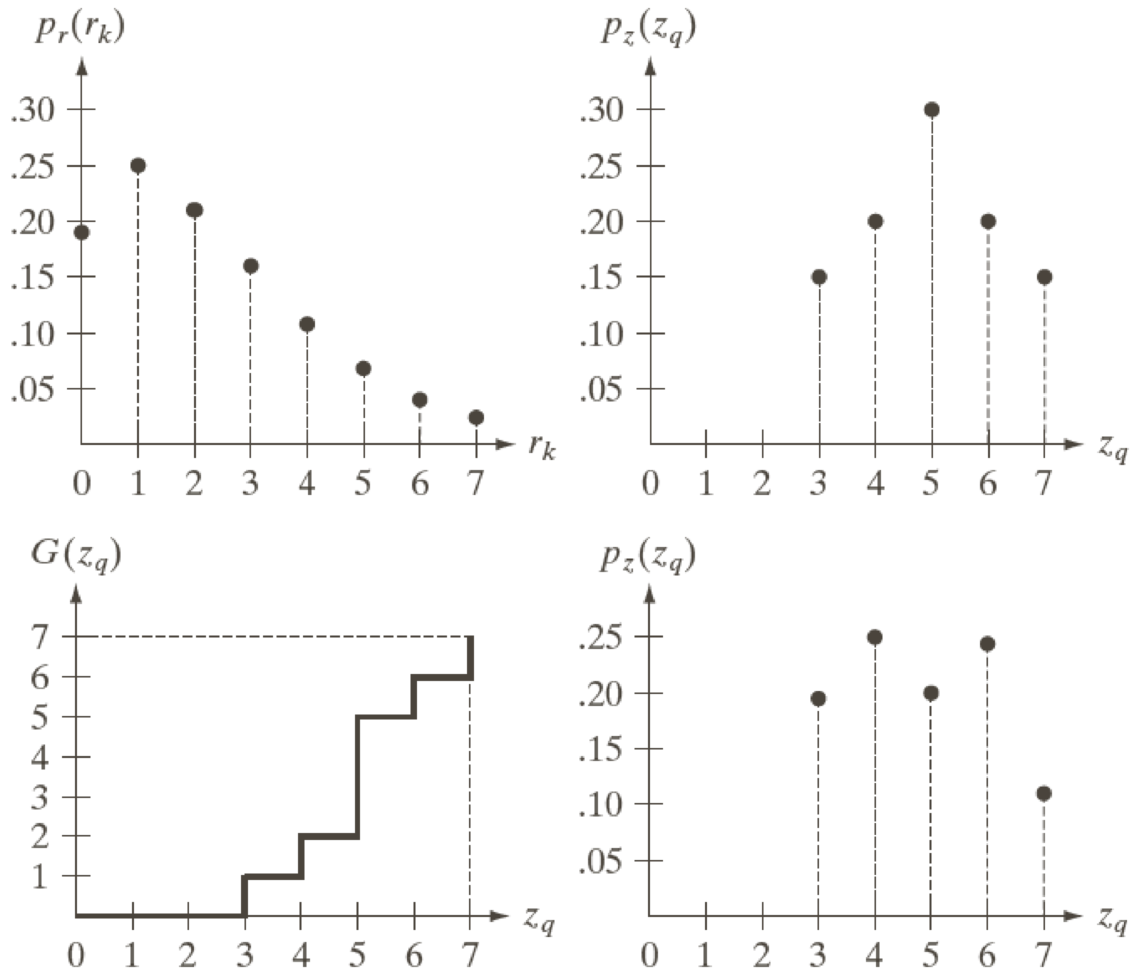
Potential issue: Cause a one-to-multiple mapping

-- multiple z_q are mapped to the same $G(z_q)$

Solution: assign the z-s pair with smallest z_q

Step4: form the histogram-specified image using the mapping r-z found above

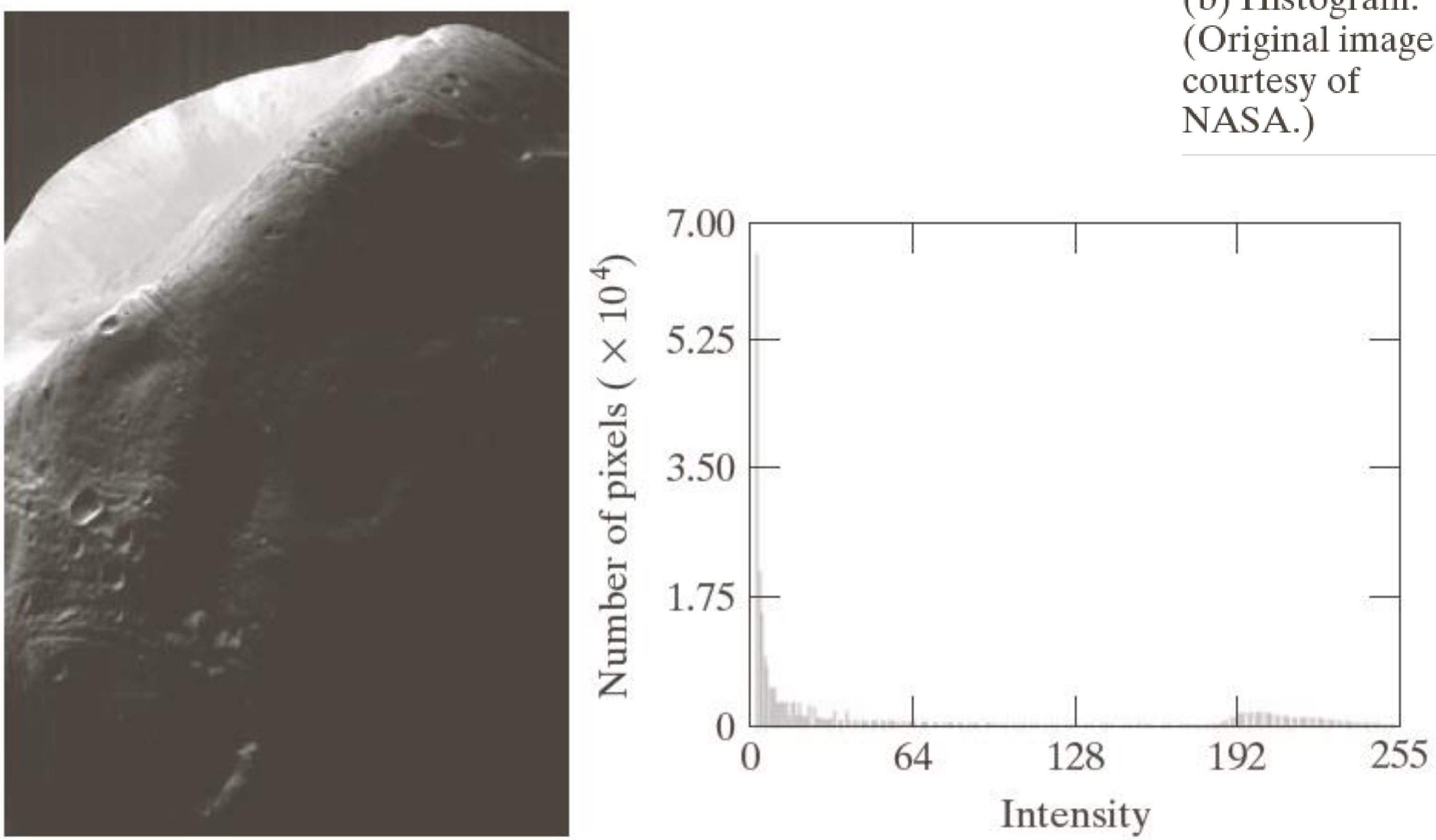
An example



a b
c d

FIGURE 3.22
(a) Histogram of a 3-bit image. (b) Specified histogram.
(c) Transformation function obtained from the specified histogram.
(d) Result of performing histogram specification. Compare (b) and (d).

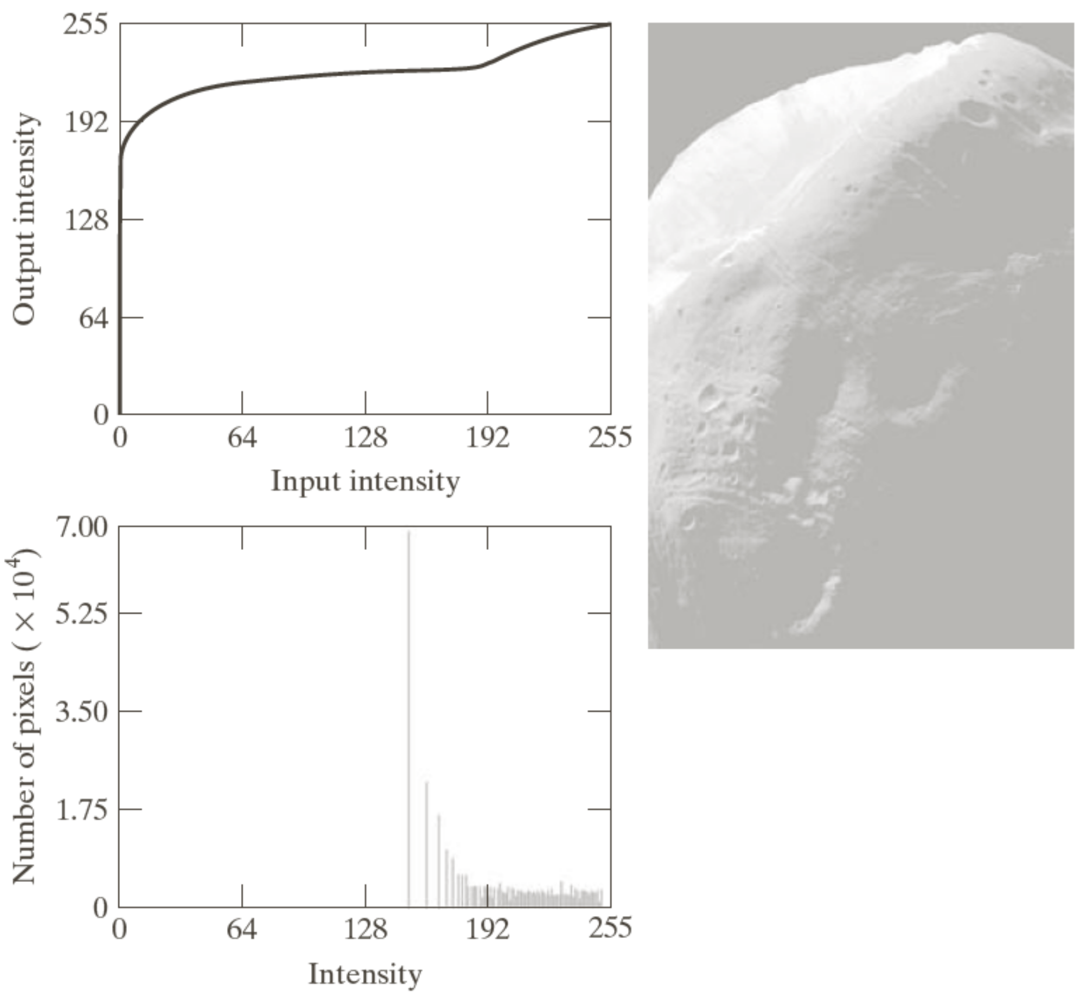
A Real Example



a b

FIGURE 3.23
(a) Image of the Mars moon Phobos taken by NASA's *Mars Global Surveyor.* (b) Histogram. (Original image courtesy of NASA.)

A Real Example – Histogram Equalization Result



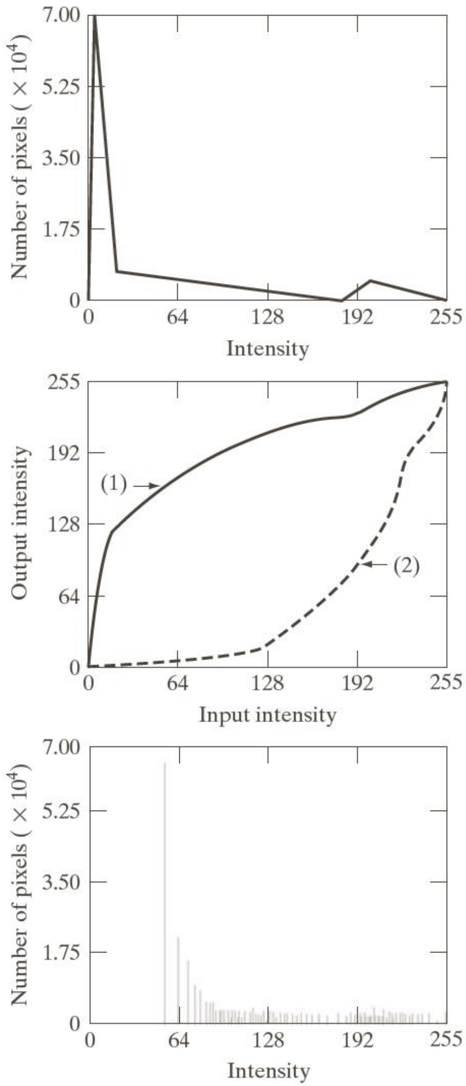
a b
c

FIGURE 3.24
(a) Transformation function for histogram equalization.
(b) Histogram-equalized image (note the washed-out appearance).
(c) Histogram of (b).

A Real Example – Histogram Matching Result

Original

HE result



a c
b
d

FIGURE 3.25
(a) Specified histogram.
(b) Transformations.
(c) Enhanced image using mappings from curve (2).
(d) Histogram of (c).

Local Histogram Processing

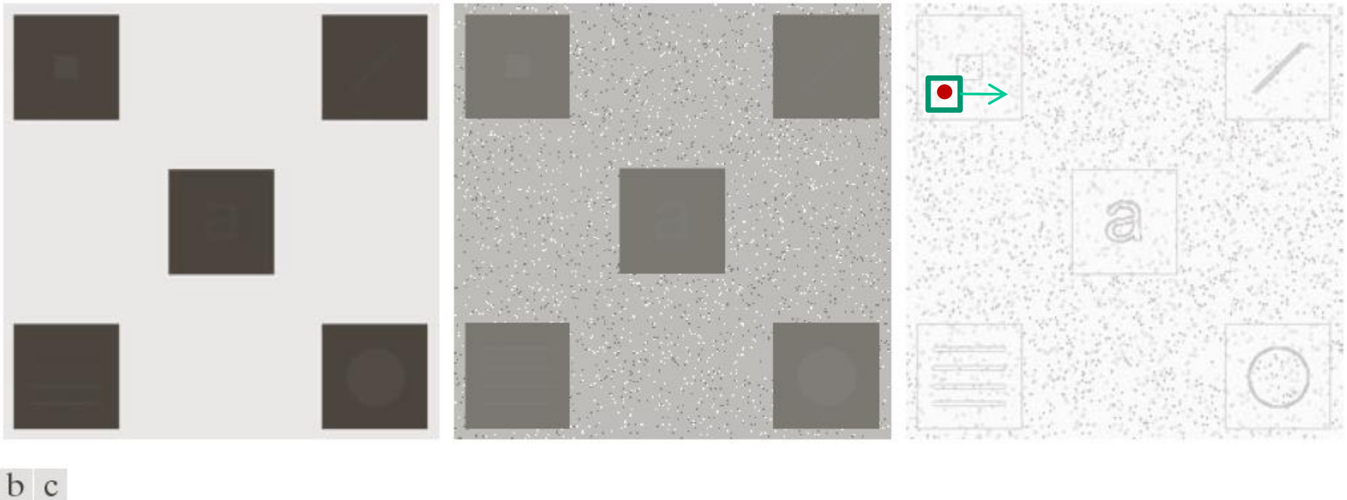
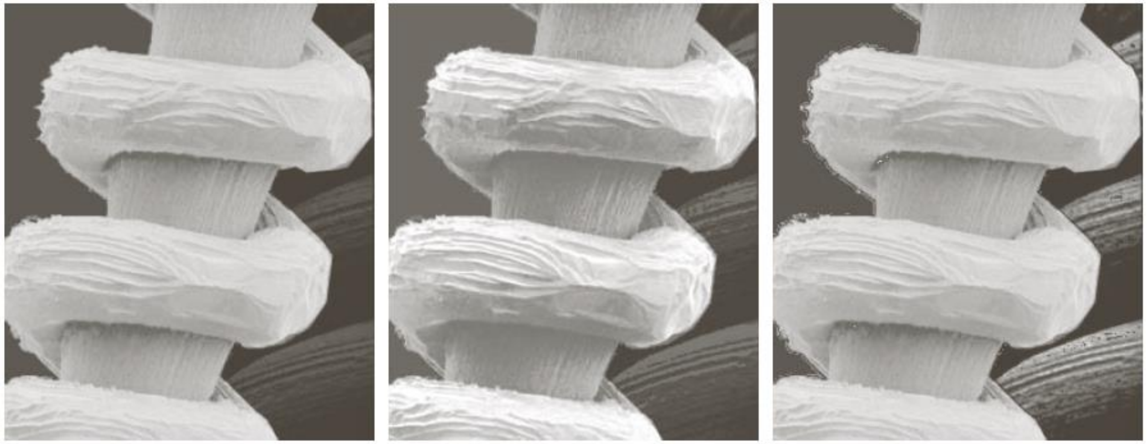


FIGURE 3.26 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization applied to (a), using a neighborhood of size 3 × 3.

Using Histogram Statistics for Image Enhancement



a b c

FIGURE 3.27 (a) SEM image of a tungsten filament magnified approximately 130×. (b) Result of global histogram equalization. (c) Image enhanced using local histogram statistics. (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

$$m_G = \sum_{i=0}^{L-1} r_i p(r_i),$$

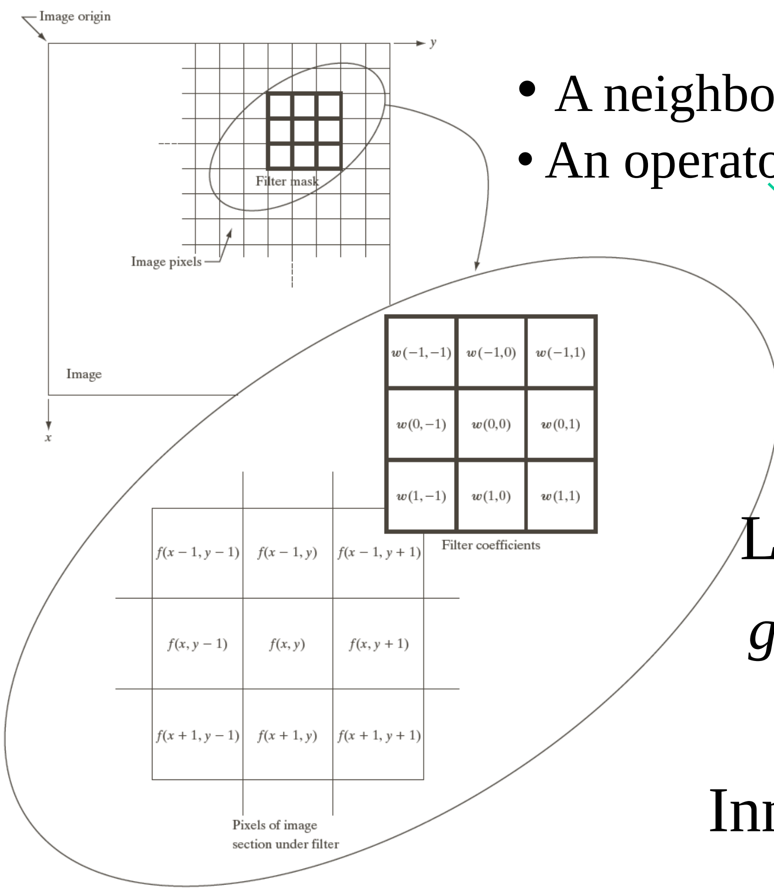
$$\sigma_G^2 = \sum_{i=0}^{L-1} (r_i - m)^2 p(r_i)$$

$$m_{S_{xy}} = \sum_{i=0}^{L-1} r_i p_{S_{xy}}(r_i),$$

$$\sigma_{S_{xy}}^2 = \sum_{i=0}^{L-1} (r_i - m_{S_{xy}})^2 p_{S_{xy}}(r_i)$$

$$g(x,y) = \begin{cases} 4f(x,y) & \text{if } m_{S_{xy}} \le 0.4 m_G \text{ AND } 0.02\sigma_G \le \sigma_{S_{xy}} \le 0.4\sigma_G \\ f(x,y) & \text{otherwise} \end{cases}$$

Fundamentals of Spatial Filtering



- A neighborhood
- An operator with the same size: linear/nonlinear

Note: Each element in w will visit every pixel in the image just once.

Linear spatial filtering:

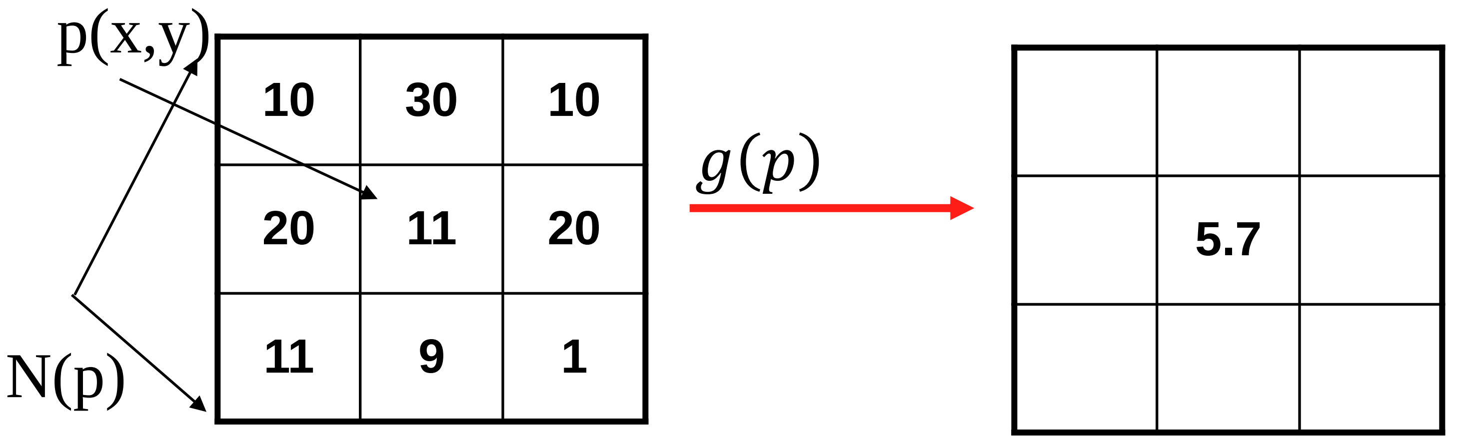
$$g(x,y)=\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)\, f(x+s, y+t)$$

Inner product $g(x,y)=\mathbf{w}\bullet\mathbf{f}=\mathbf{w}^T\mathbf{f}$

FIGURE 3.28 The mechanics of linear spatial filtering using a 3 × 3 filter mask. The form chosen to denote the coordinates of the filter mask coefficients simplifies writing expressions for linear filtering.

Fundamentals of Spatial Filtering

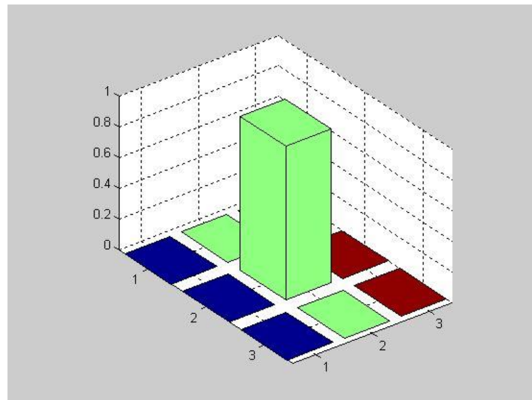
Modifying the pixels in an image based on some function of a local neighborhood of the pixels



g(p):

- Linear function
  - Correlation
  - Convolution
- Nonlinear function
  - Order statistic (median)

Linear Filtering



*

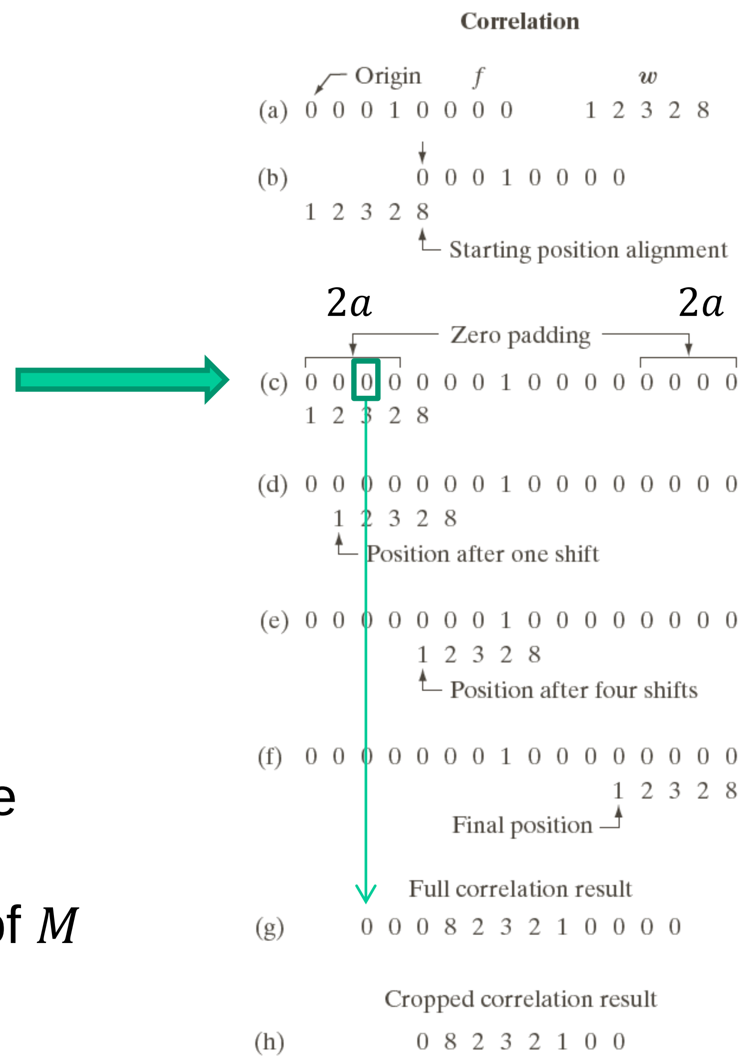
0	0	0
0	1	0
0	0	0

=

Spatial Correlation: 1D Signal

1D correlation $\sum_{s=-a}^{a} w(s)\, f(x+s)$

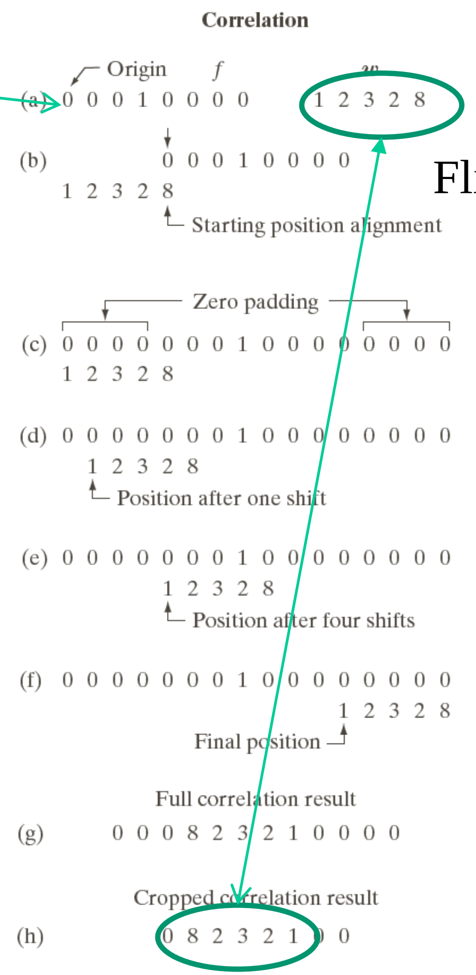
Zero-padding: add zeros on the left and right margin, respectively



- **Full correlation result** has the size of $M + 2a$
- **Cropped result** has the size of M – the size of the original signal

Spatial Correlation: 1D Signal

Discrete impulse



Flipped

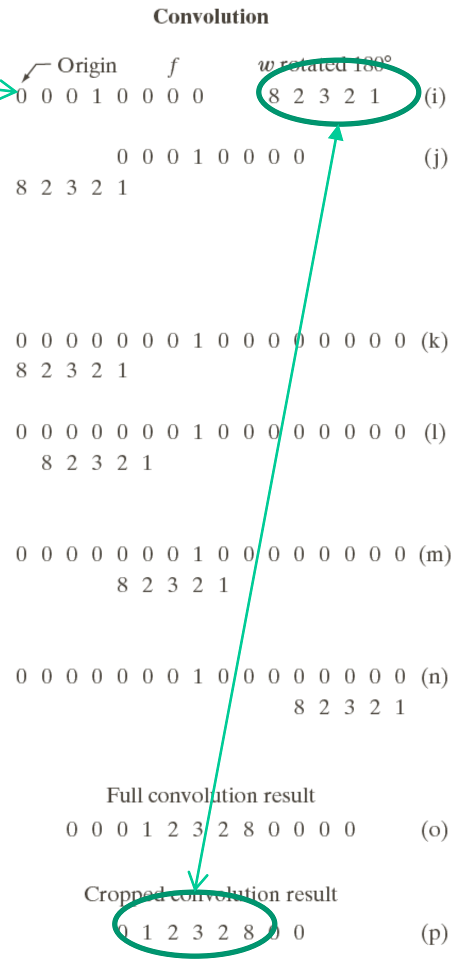
1D correlation $$\sum_{s=-a}^{a} w(s)\, f(x+s)$$

- Full correlation result has the size of $M + 2a$
- Cropped result has the size of M – the size of the original signal

The impulse response is a rotation of the filter by 180 degree

Spatial Convolution: 1D Signal

Discrete impulse

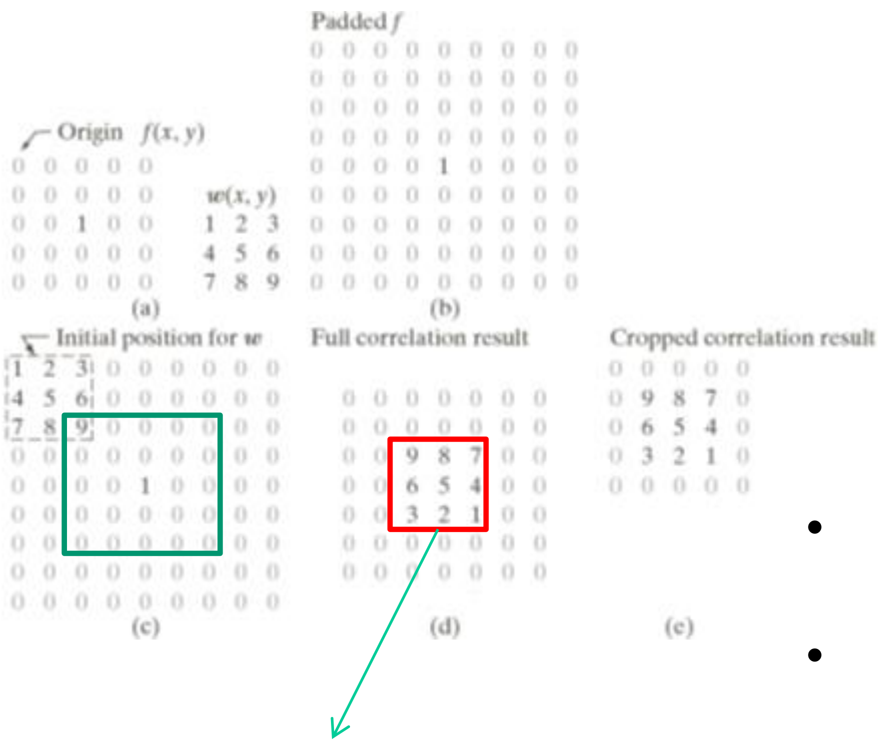


Flipped filter

1D convolution $\sum_{s=-a}^{a} w(s) f(x-s)$

The impulse response is the same as the filter

Extend to 2D Image: 2D Image Correlation

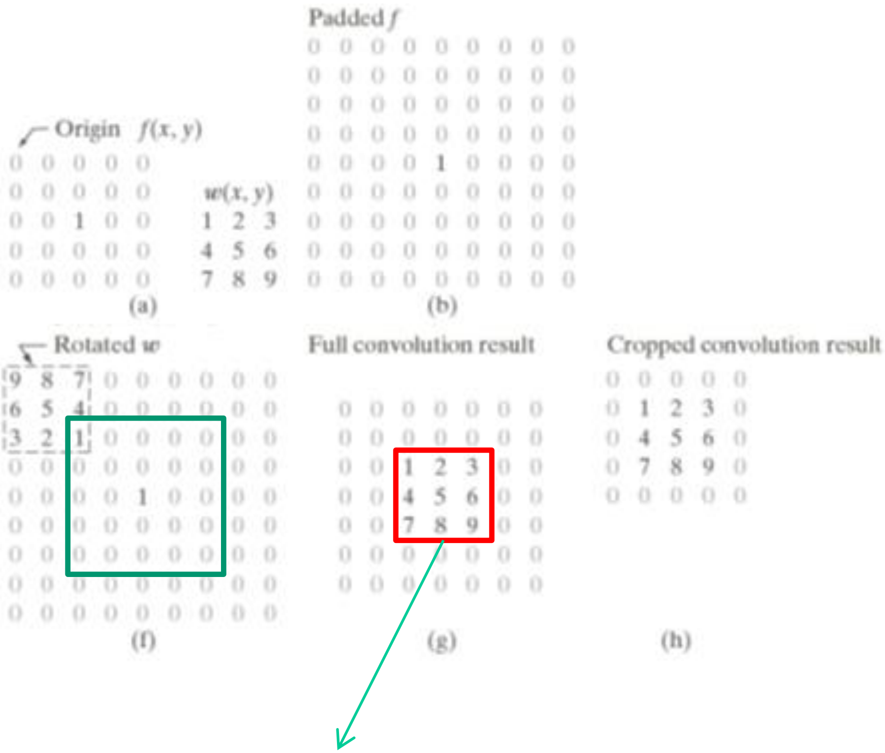


$$\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)\, f(x+s, y+t)$$

- Full correlation result has the size of $(M + 2a, N + 2b)$
- Cropped result has the size of (M, N) – the size of the original image

The 2D impulse response of image correlation is a rotation of the filter by 180 degree

Extend to 2D Image: 2D Image Convolution



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

The 2D impulse response of image convolution is the same as the filter

- Flip in both horizontal and vertical directions (rotate 180 degree) -> same if the filter is symmetric
- Convolution filter/mask/kernel
- Full convolution result has the size of $(M+2a, N+2b)$
- Cropped result has the size of (M, N) – the size of the original image

Linear Filters

General process:

- Form new image whose pixels are a weighted sum of original pixel values, using the same set of weights at each point.

Properties

- Output is a linear function of the input
- Output is a shift-invariant function of the input (i.e. shift the input image two pixels to the left, the output is shifted two pixels to the left)

Example: smoothing by averaging

- form the average of pixels in a neighborhood

Example: smoothing with a Gaussian

- form a weighted average of pixels in a neighborhood

Example: finding an edge

Smoothing Spatial Filter – Low Pass Filters

Weighted average

$\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

$\frac{1}{16} \times$

1	2	1
2	4	2
1	2	1

a b

FIGURE 3.32 Two 3 × 3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to 1 divided by the sum of the values of its coefficients, as is required to compute an average.

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

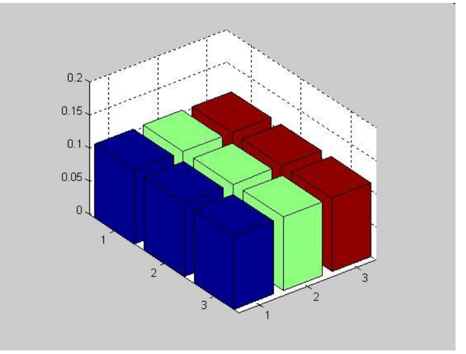
- Noise deduction
- reduction of "irrelevant details"
- edge blurred

Normalization factor

Smoothing Spatial Filter

Image averaging $R = \frac{1}{9}\sum_{i=1}^{9} z_i$

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9



$* \frac{1}{9}$

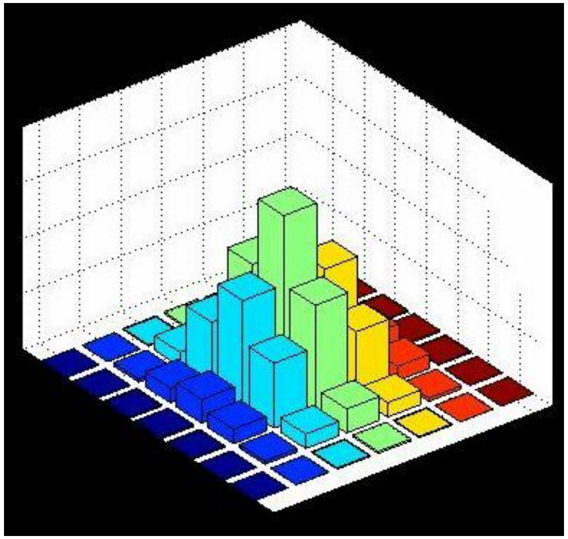
1	1	1
1	1	1
1	1	1

$=$

Smoothing Spatial Filter

2D Gaussian filter $$h(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

*



=

Comparison using Different Smoothing Filters – Different Kernels

Average

Gaussian

Comparison using Different Smoothing Filters: Different Size

Filter size: 3, 5, 9, 15, 35

Square size: 3, 5, 9, 15, 25, 35, 45, 55 with a spacing of 25

Bar: 5x100 with a spacing of 20

Letter size: 10, 12, 14, 16, 18, 20, 24

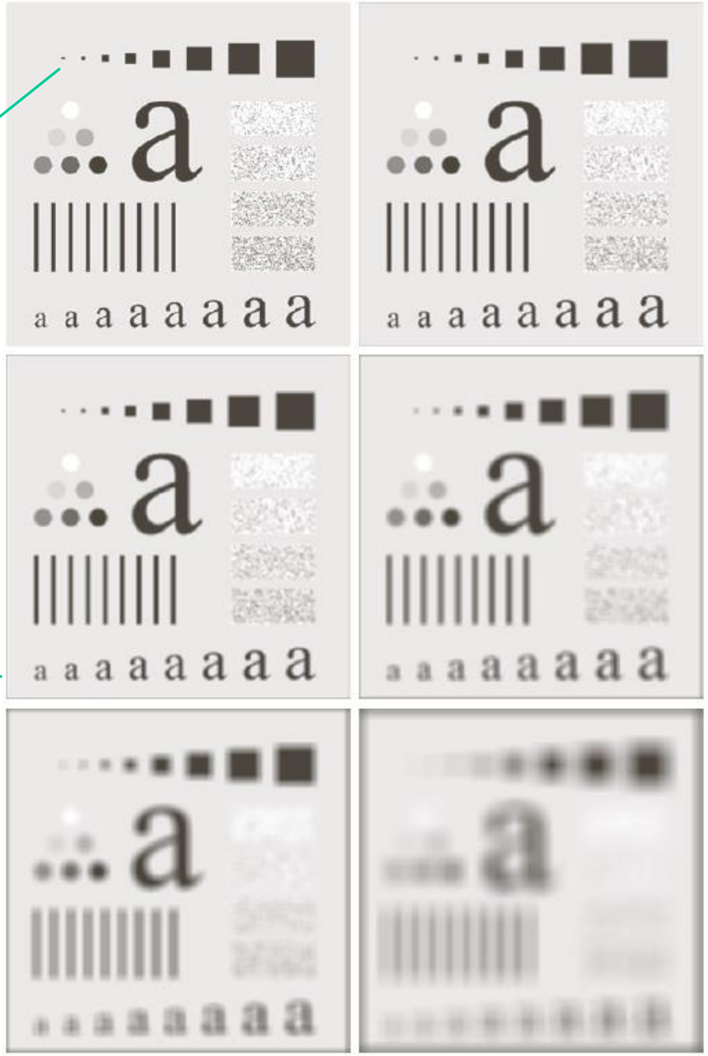


FIGURE 3.33 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $m = 3, 5, 9, 15$, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. 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 intensity 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.

a b
c d
e f

Image Smoothing and Thresholding

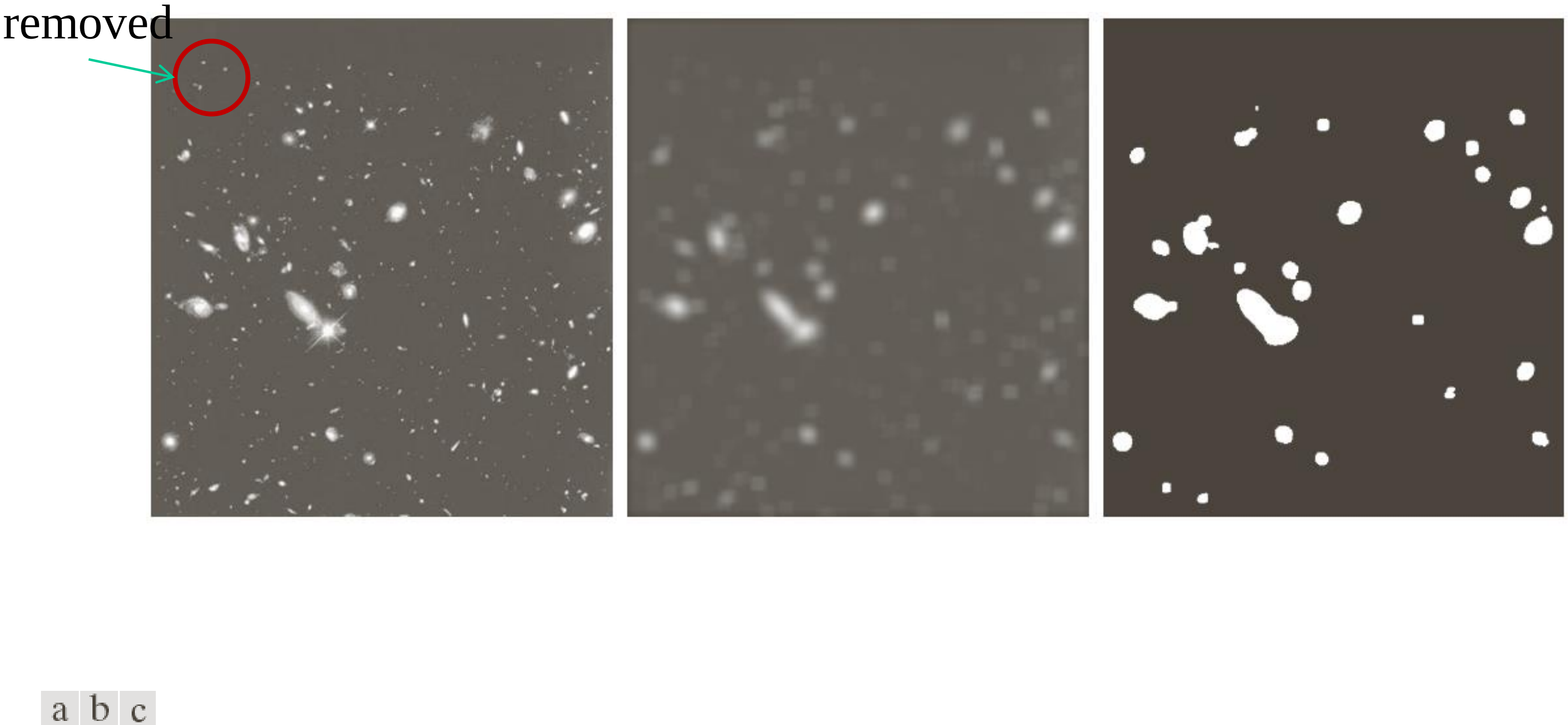


FIGURE 3.34 (a) Image of size 528 × 485 pixels from the Hubble Space Telescope. (b) Image filtered with a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)