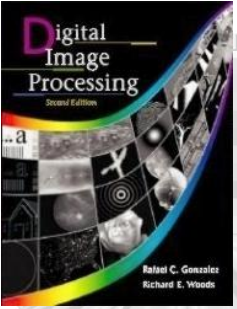


Chapter 3

Intensity Transformations and Spatial Filtering

Two principles categories of Spatial Processing

- Intensity Transformations
  - Operate on single pixels of an image, principally for the purpose of contrast manipulation & image thresholding
- Spatial Filtering
  - Deals with performing operation (like image sharpening), by working in a neighborhood (*spatial filter*, *kernel*, *template* , *window or spatial mask*) of every pixel in an image

© 2002 R. C. Gonzalez & R. E. Woods

Spatial Domain Enhancement

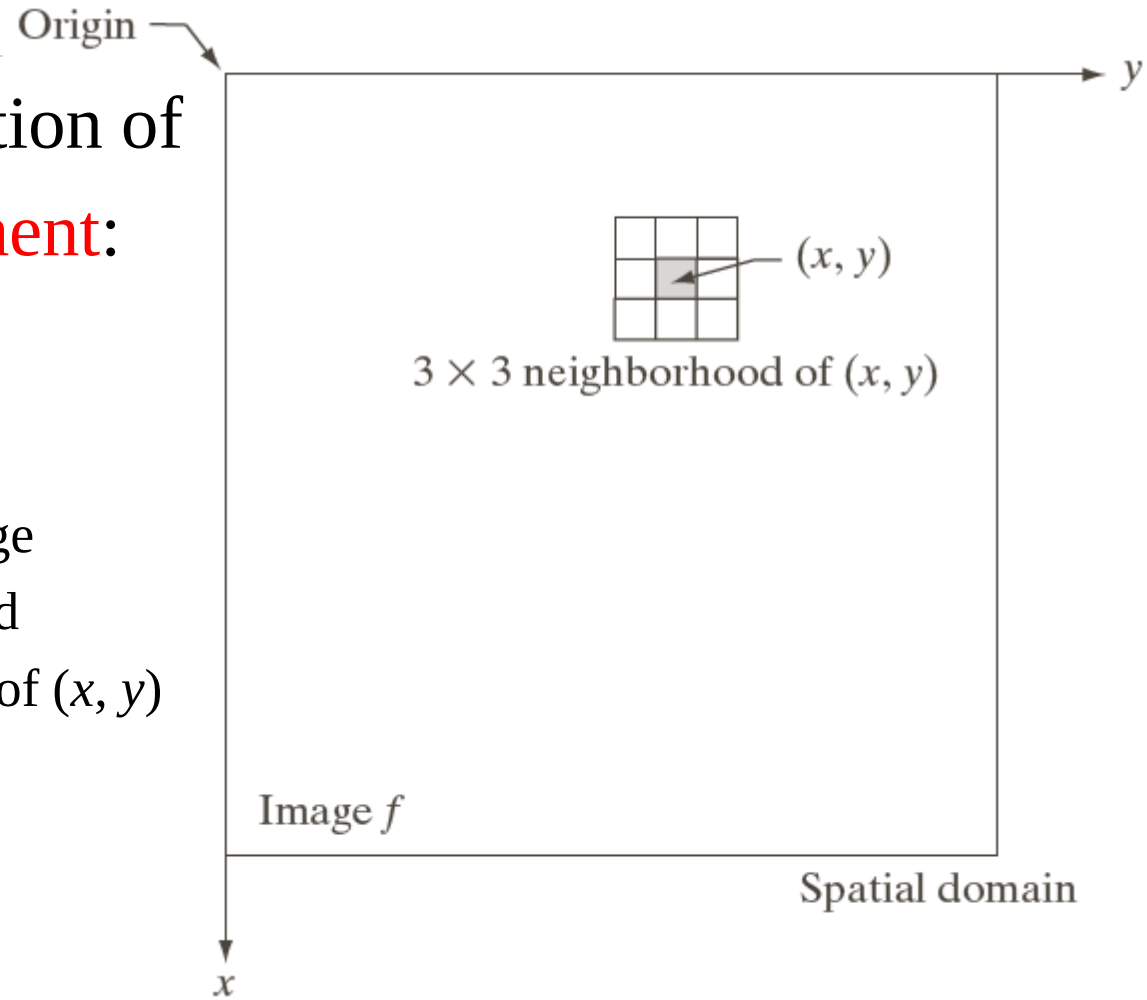
Mathematical representation of spatial domain enhancement:

$$g(x, y) = T[f(x, y)]$$

where $f(x, y)$: the input image

$g(x, y)$: the processed image

T: an operator on f, defined over some neighborhood of (x, y)



© 2002 R. C. Gonzalez & R. E. Woods

Intensity/Graylevel Transformations

- The simplest form of ***T***, is when the neighborhood of size 1×1 (that is a single pixel). In this case, g depends only on the value of f at (x,y), and ***T*** becomes a *grey-level* (also called *intensity* or *mapping*) *transformation function* of the form:

$$s = T(r)$$

Where, for simplicity in notation, r and s are variables denoting, respectively, the grey level of f(x,y) and g(x,y) at any point (x,y)

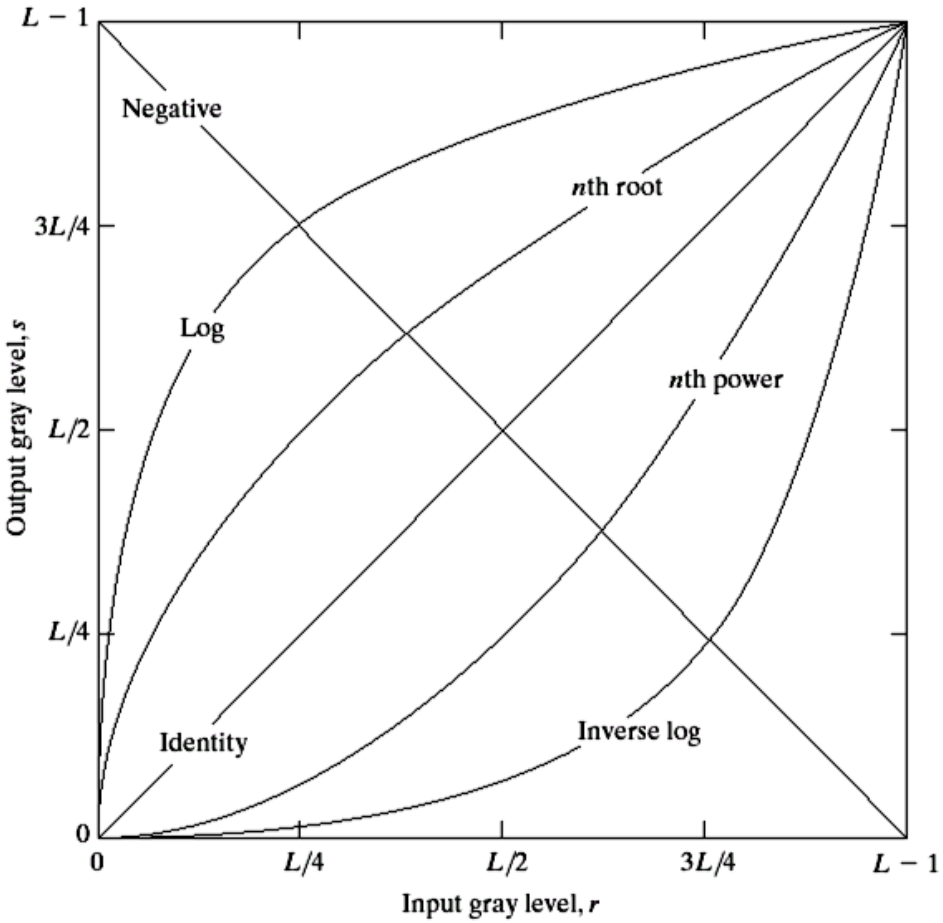
© 2002 R. C. Gonzalez & R. E. Woods

Basic Grey Level Transformations

There are many different kinds of grey level transformations

Three of the most common are shown here

- Linear
  - Negative/Identity
- Logarithmic
  - Log/Inverse log
- Power law
  - n^{th} power/n^{th} root



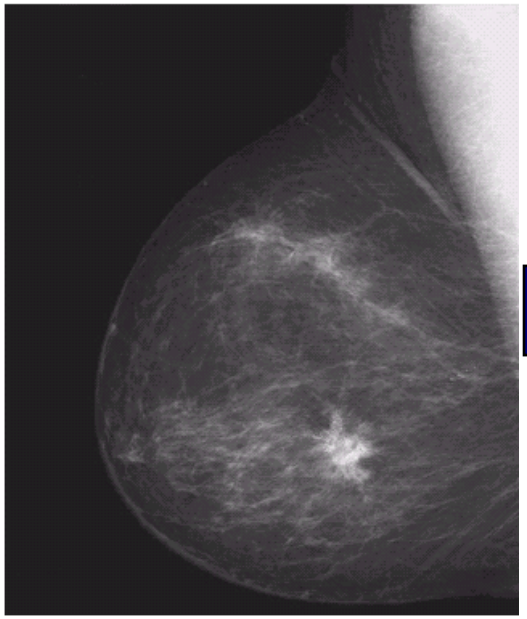
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Point Processing Example: Negative Images

Negative images are useful for enhancing white or grey detail embedded in dark regions of an image

– Note how much clearer the tissue is in the negative image of the mammogram below

Original Image

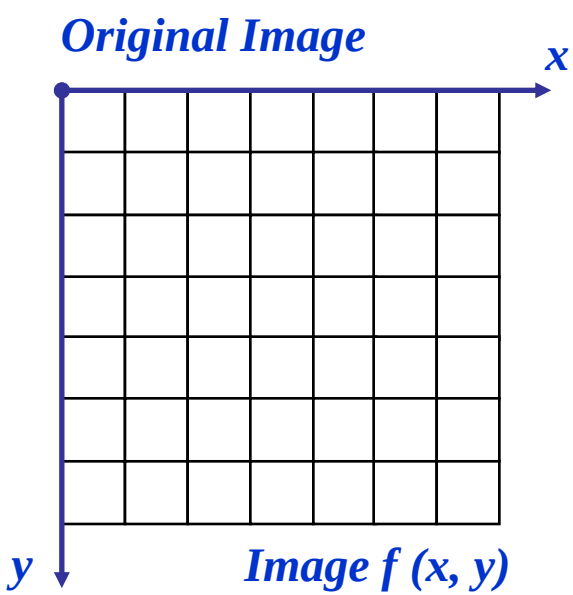


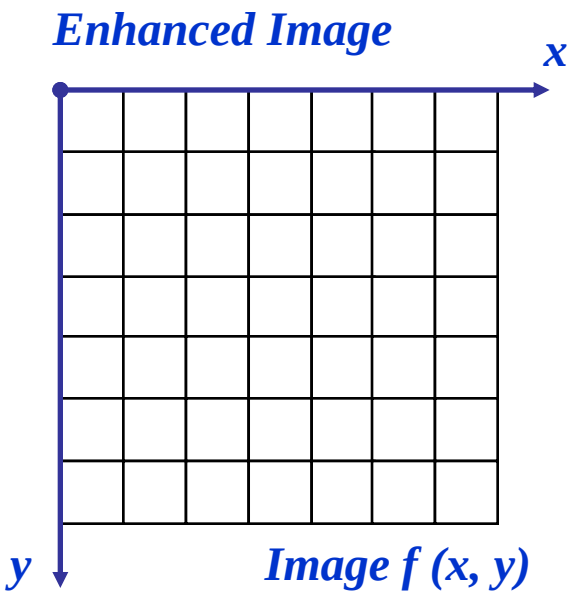
$s = 1.0 - r$

Negative Image

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Point Processing Example: Negative Images (cont…)





$$s = intensity_{max} - r$$

Image Negative: $s = L - 1 - r$

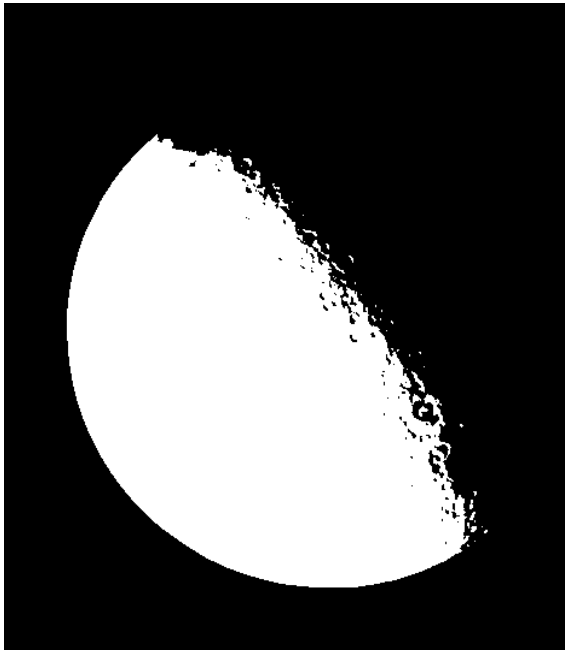
where L is # of different intensity levels

Credits: Brian Mac, Dublin Institute of Technology

Point Processing Example: Thresholding

Thresholding transformations are particularly useful for segmentation in which we want to isolate an object of interest from a background

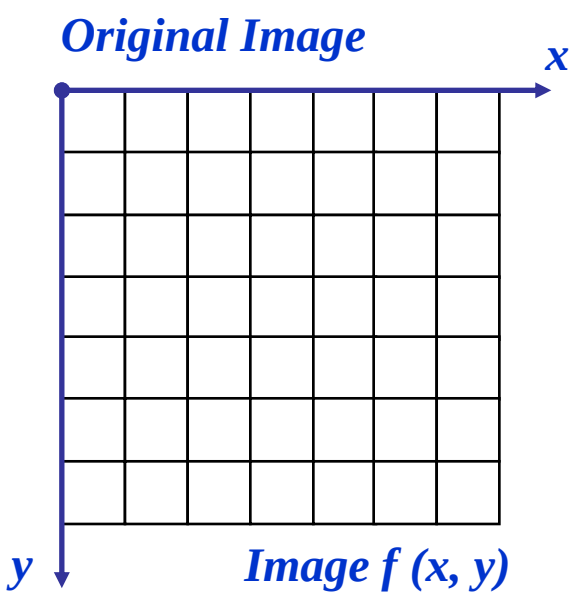
$$s = \begin{cases} 1.0 & r > \text{threshold} \\ 0.0 & r <= \text{threshold} \end{cases}$$

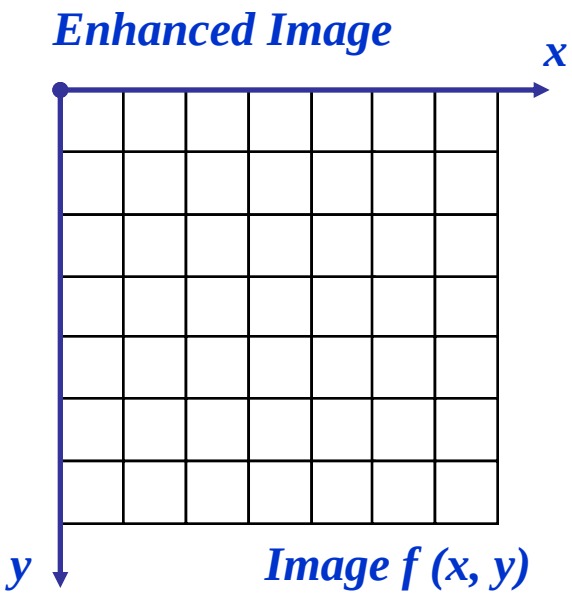


Credits: Brian Mac, Dublin Institute of Technology

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Point Processing Example: Thresholding (cont…)





$$s = \begin{cases} 1.0 & r > threshold \\ 0.0 & r <= threshold \end{cases}$$

Credits: Brian Mac, Dublin Institute of Technology

Gray-Level Mapping

$$g(x, y) = f(x, y) + b \qquad g(x, y) = a \cdot f(x, y) + b$$

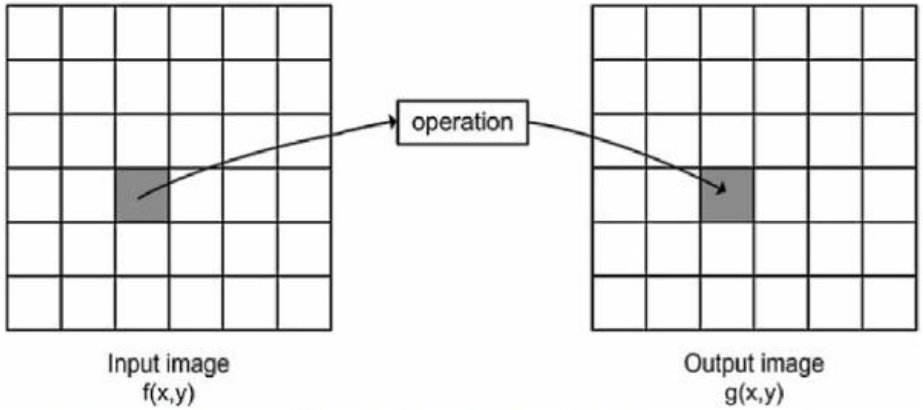
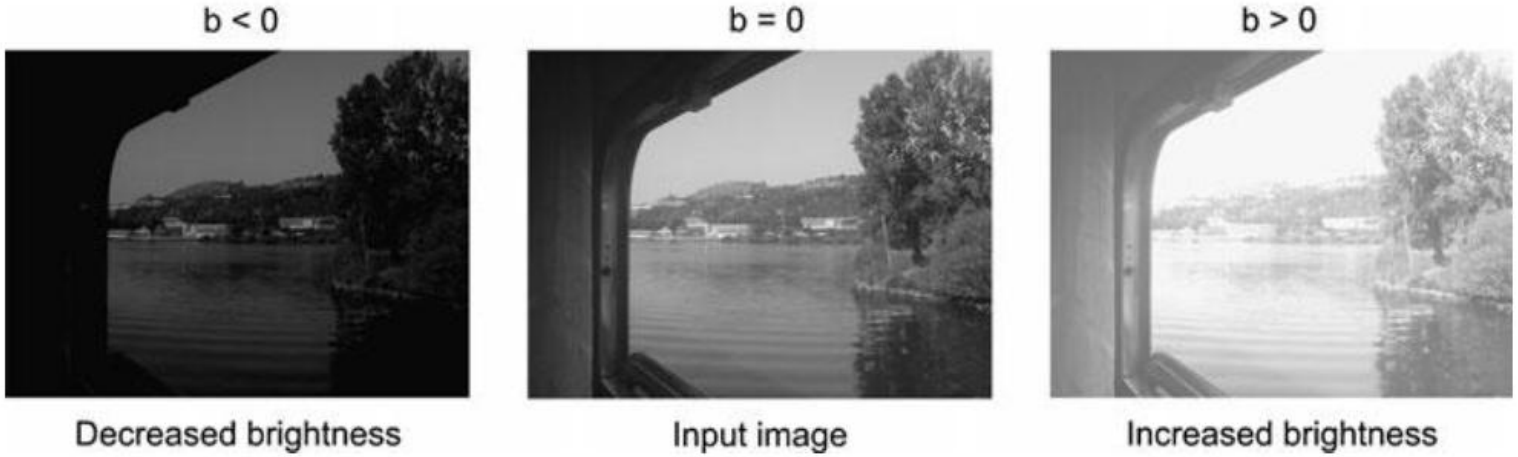


Fig. The principle of point processing.



Credits: S. Murala, EEL484 Lecture notes/slides, IIT Ropar

Logarithmic Transformations

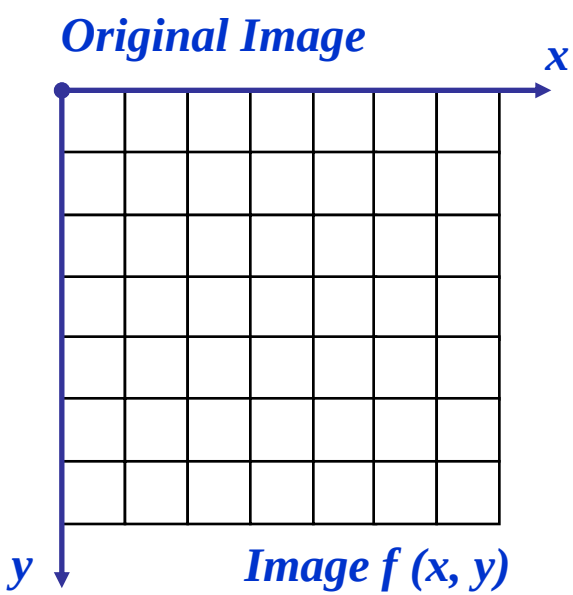
The general form of the log transformation is

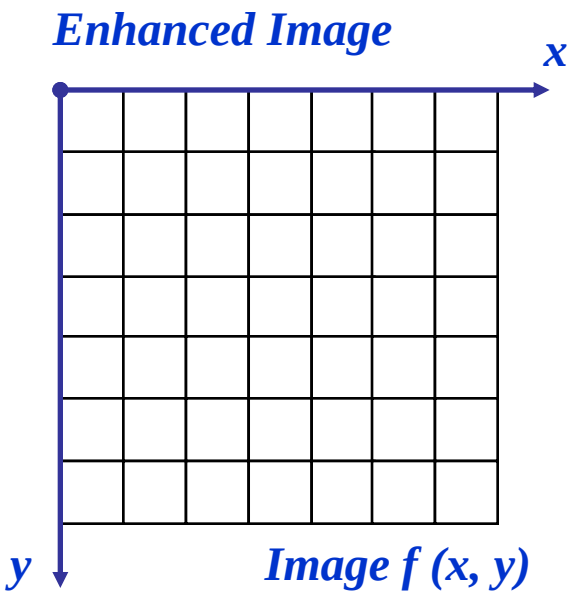
$$s = c * log(1 + r)$$

The log transformation maps a narrow range of low input grey level values into a wider range of output values

The inverse log transformation performs the opposite transformation

Logarithmic Transformations (cont…)





$$s = log(1 + r)$$

We usually set c to 1

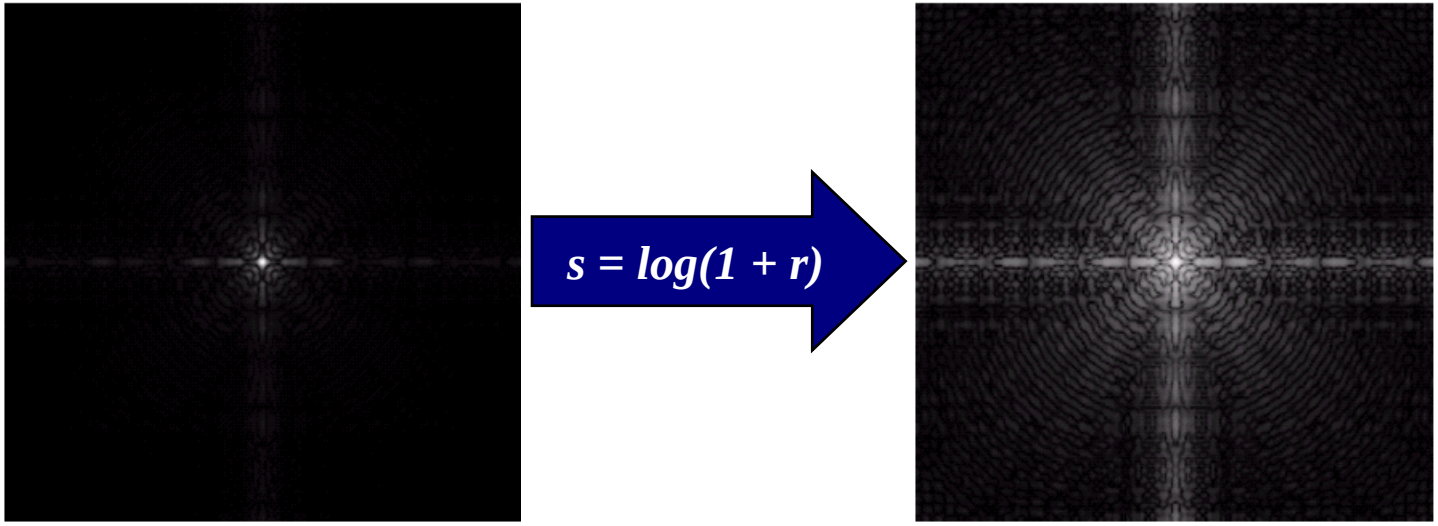
Grey levels must be >= 0.0

Credits: Brian Mac, Dublin Institute of Technology

Logarithmic Transformations (cont…)

Log functions are particularly useful when the input grey level values may have an extremely large range of values

In the following example the Fourier transform of an image is put through a log transform to reveal more detail



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

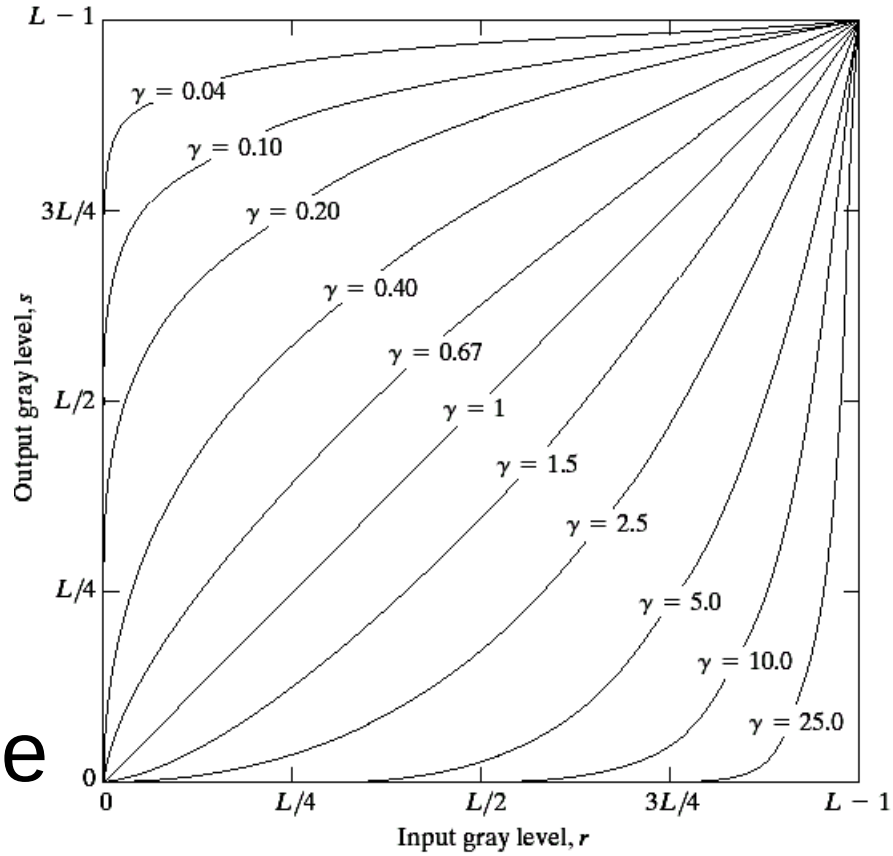
Power-Law (Gamma) Transformations

Power law transformations have the following form

$$s = c * r^{\gamma}$$

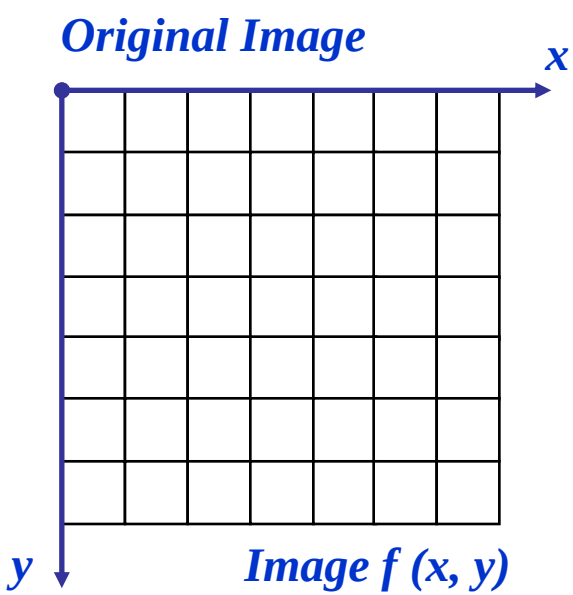
Map a narrow range of dark input values into a wider range of output values or vice versa.

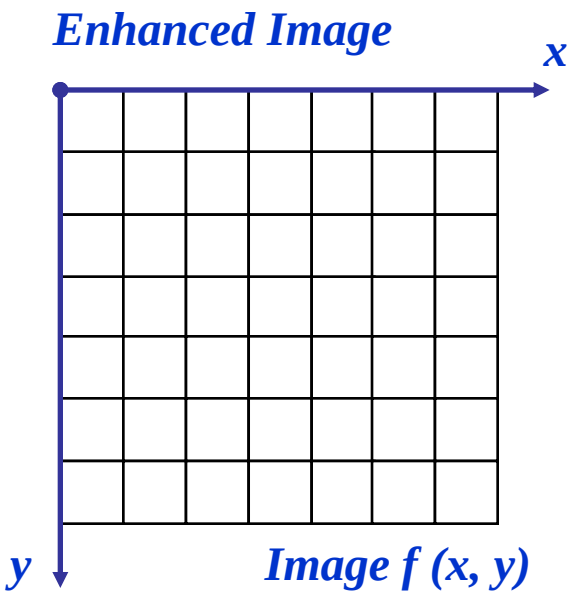
Varying γ gives a whole family of curves



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Power-Law (Gamma) Transformations





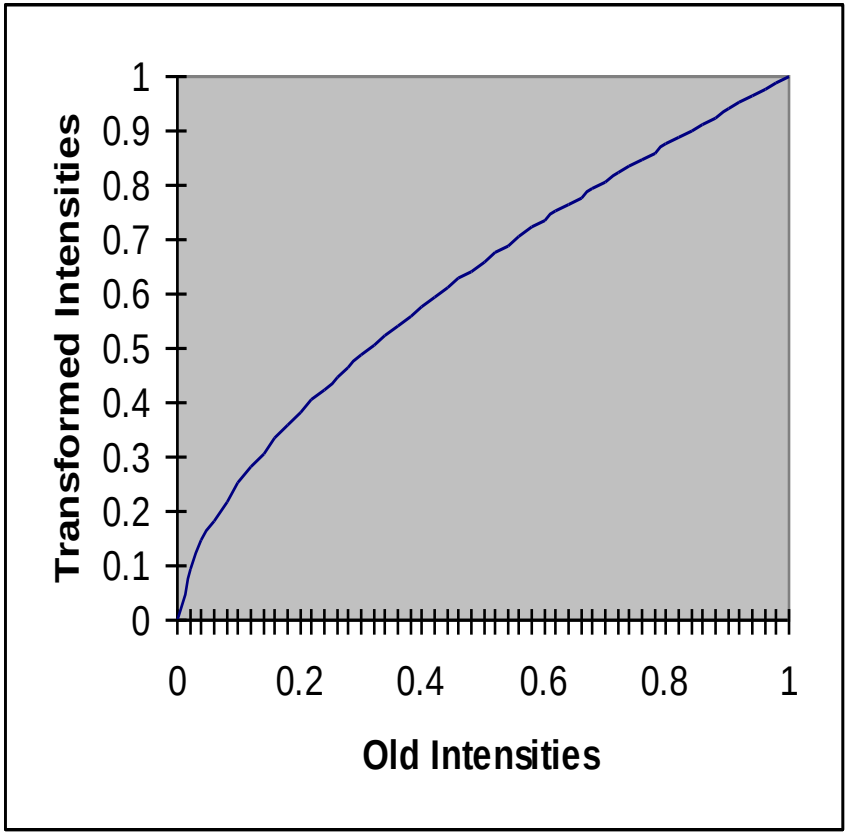
$$s = r^{\gamma}$$

We usually set c to 1.

Power Law Example

Power Law Example (cont…)

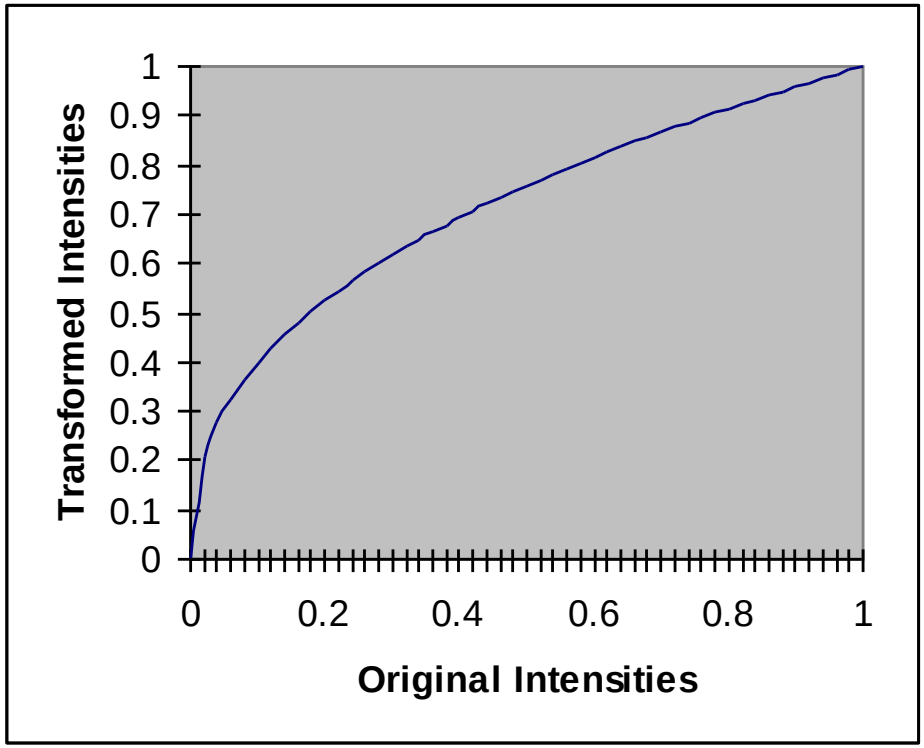
$$\gamma = 0.6$$



Credits: Brian Mac, Dublin Institute of Technology

Power Law Example (cont…)

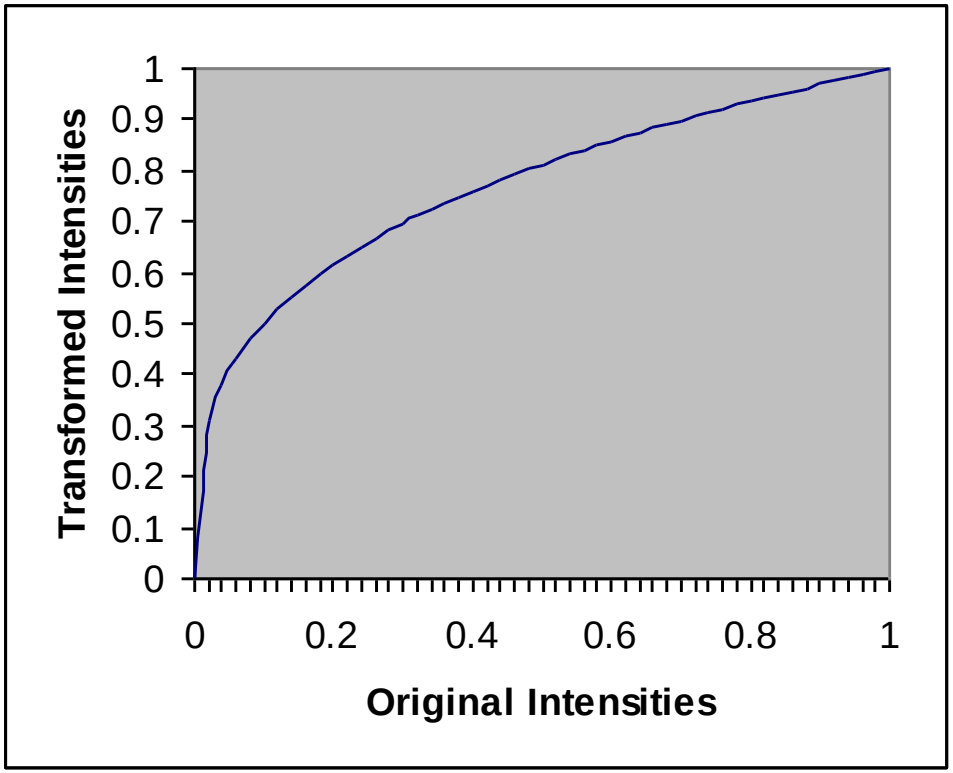
$$\gamma = 0.4$$



Credits: Brian Mac, Dublin Institute of Technology

Power Law Example (cont…)

$$\gamma = 0.3$$

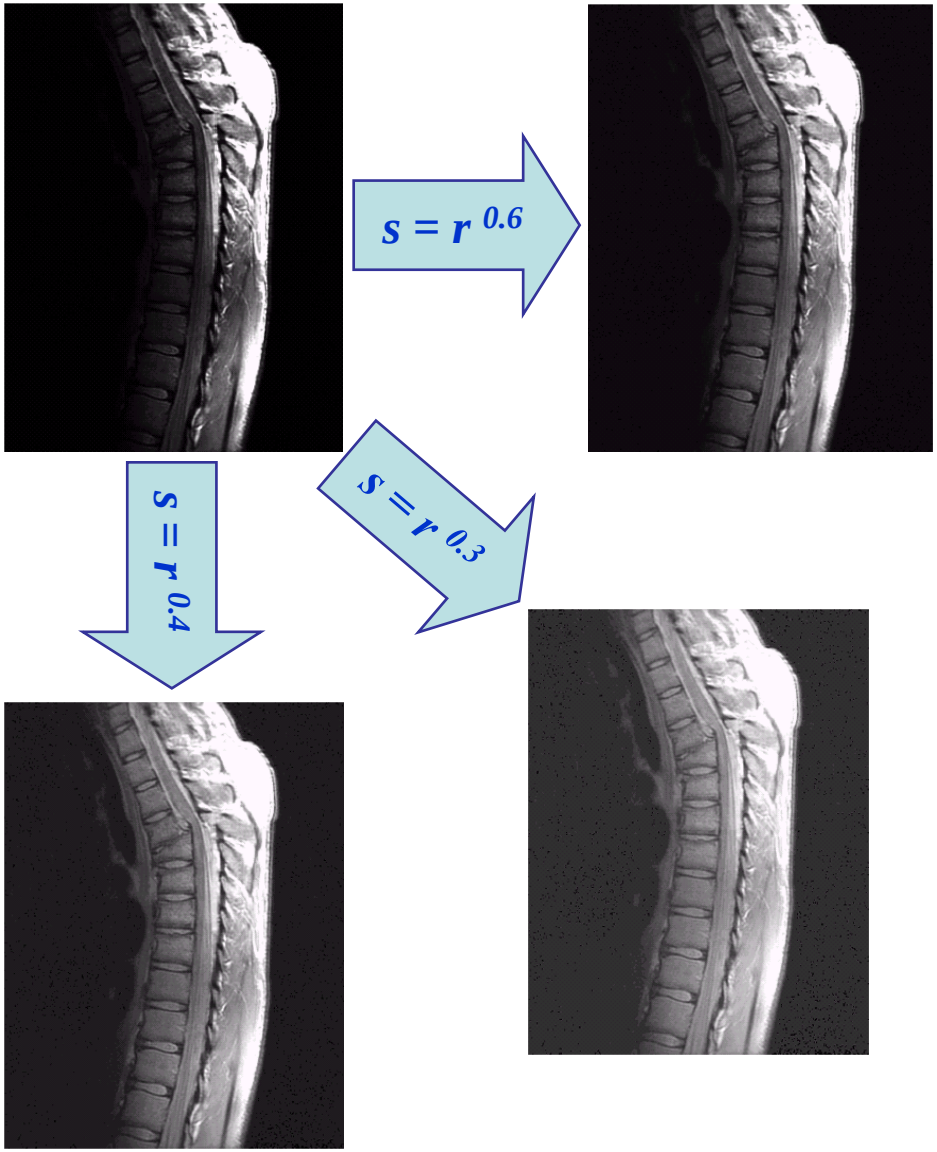


Credits: Brian Mac, Dublin Institute of Technology

Power Law Example (cont…)

The images to the right show a magnetic resonance (MR) image of a fractured human spine

Different curves highlight different detail



Credits: Brian Mac, Dublin Institute of Technology

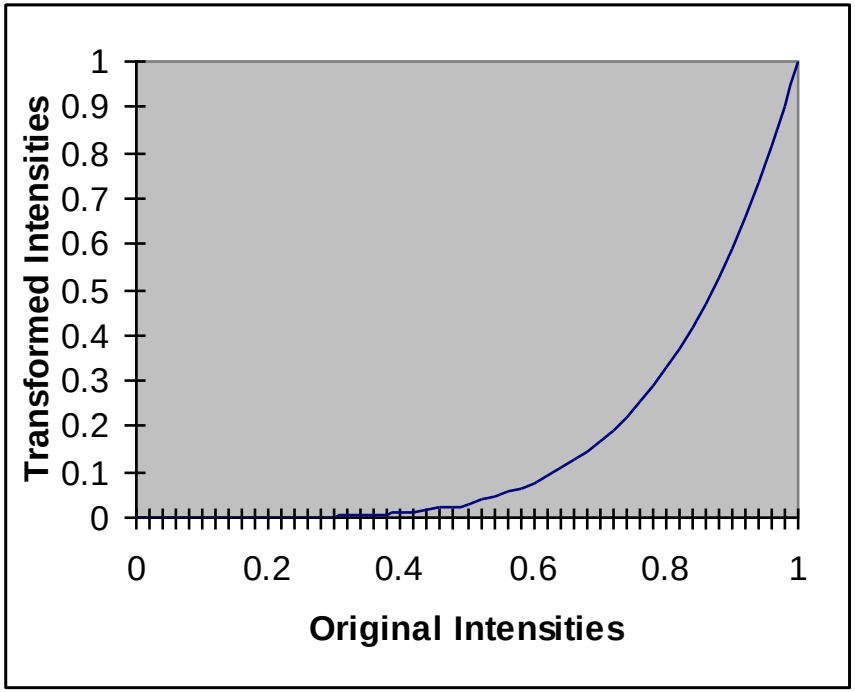
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Power Law Example

Credits: Brian Mac, Dublin Institute of Technology

Power Law Example (cont…)

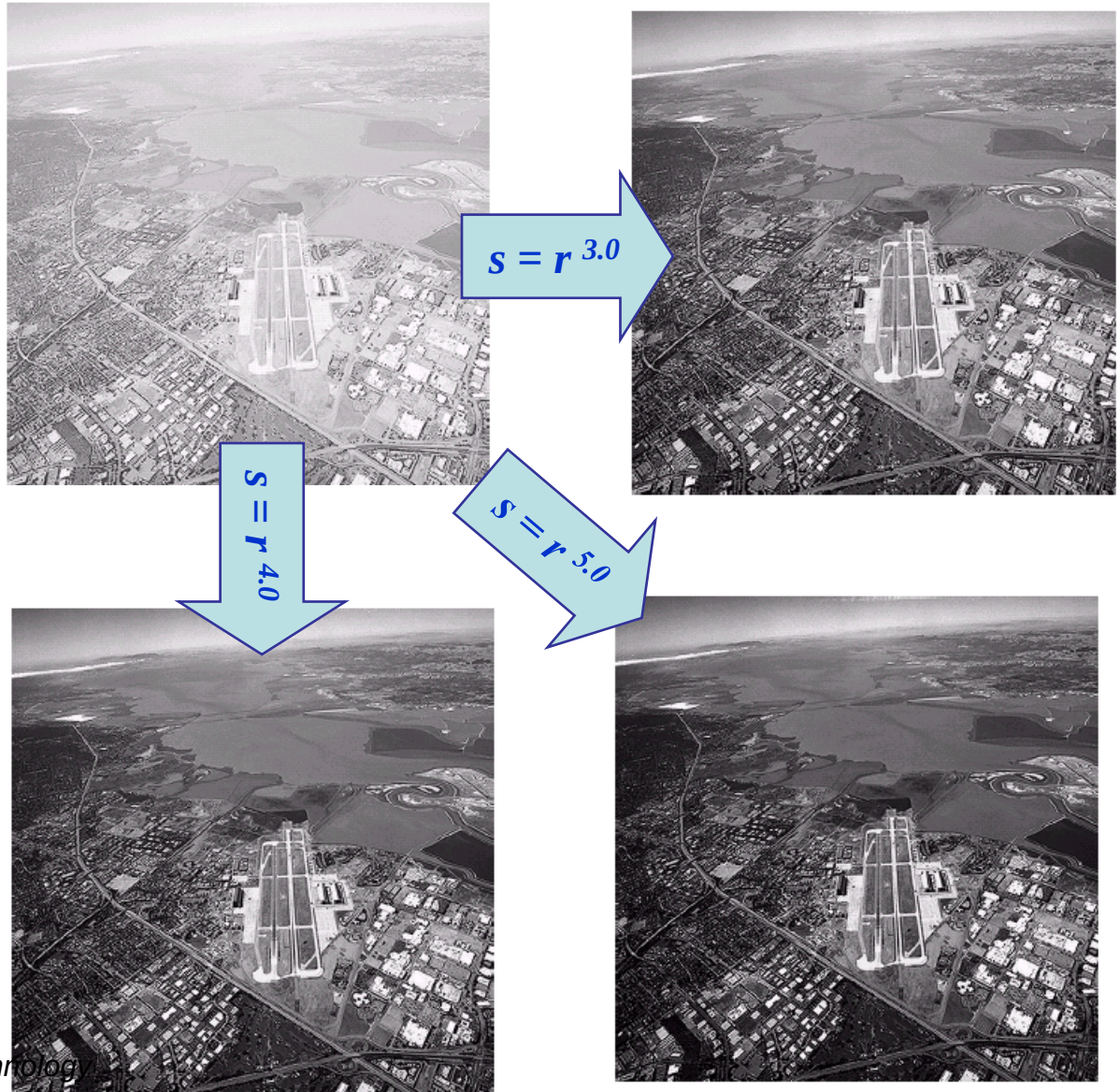
$$\gamma = 5.0$$



Credits: Brian Mac, Dublin Institute of Technology

Power Law Transformations (cont…)

- An aerial photo of a runway is shown
- This time power law transforms are used to darken the image
- Different curves highlight different detail



Credits: Brian Mac, Dublin Institute of Technology

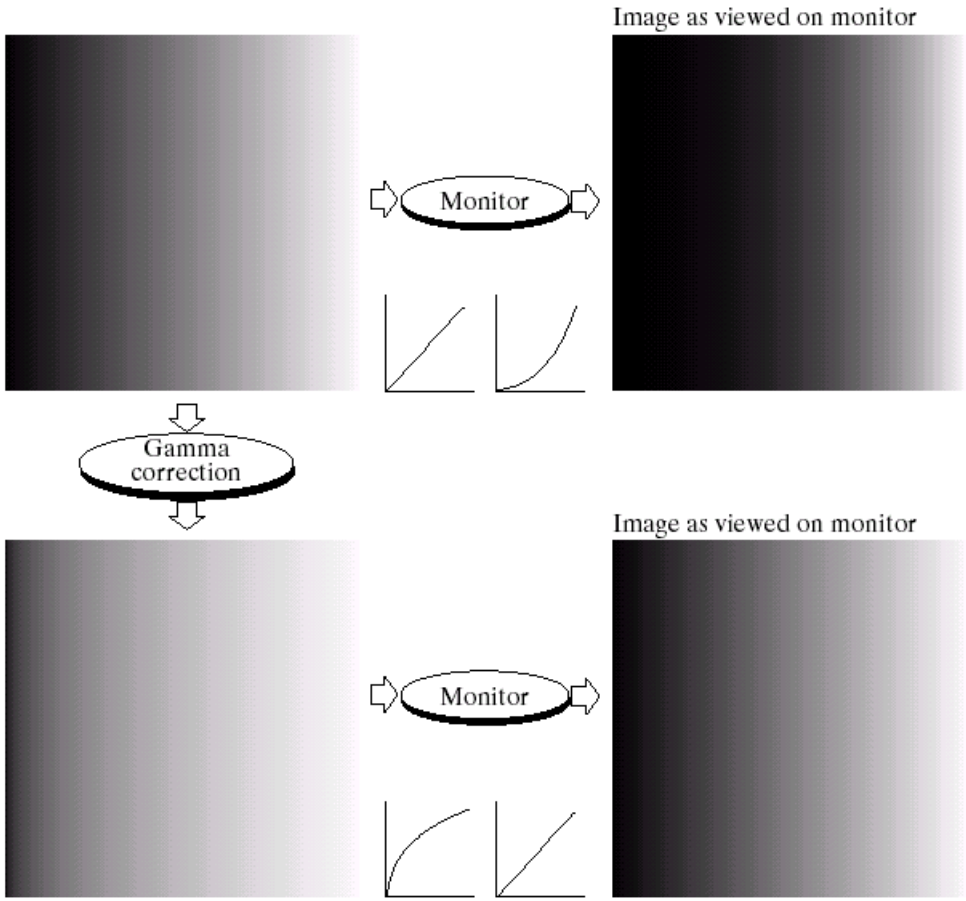
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Gamma Correction

Many of you might be familiar with gamma correction of computer monitors

Problem is that display devices do not respond linearly to different intensities

Can be corrected using a log transform



Credits: Brian Mac, Dublin Institute of Technology

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

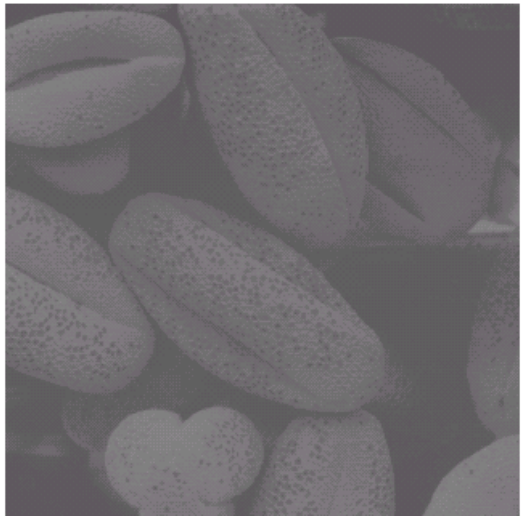
Gamma Correction

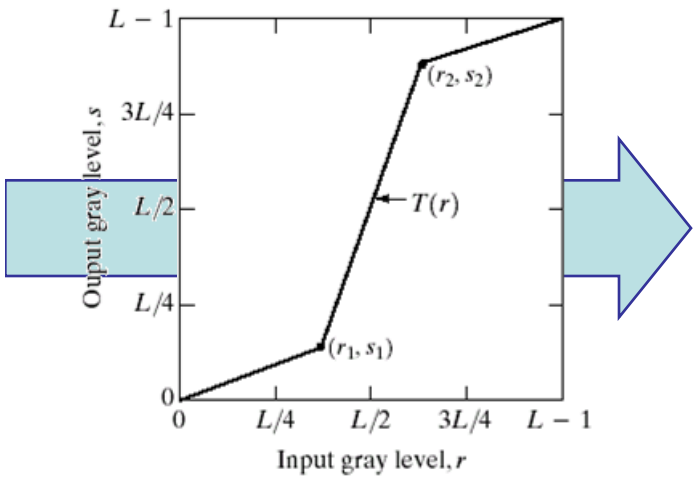
A variety of devices used for image capture, printing, and display respond according to a power law. By convention, the exponent in the power-law equation is referred to as *gamma* [hence our use of this symbol in Eq. (3.2-3)]. The process used to correct these power-law response phenomena is called *gamma correction*. For example, cathode ray tube (CRT) devices have an intensity-to-voltage response that is a power function, with exponents varying from approximately 1.8 to 2.5. With reference to the curve for $\gamma = 2.5$ in Fig. 3.6, we see that such display systems would tend to produce images that are darker than intended. This effect is illustrated in Fig. 3.7. Figure 3.7(a) shows a simple intensity-ramp image input into a monitor. As expected, the output of the monitor appears darker than the input, as Fig. 3.7(b) shows. Gamma correction in this case is straightforward. All we need to do is preprocess the input image before inputting it into the monitor by performing the transformation $s = r^{1/2.5} = r^{0.4}$.

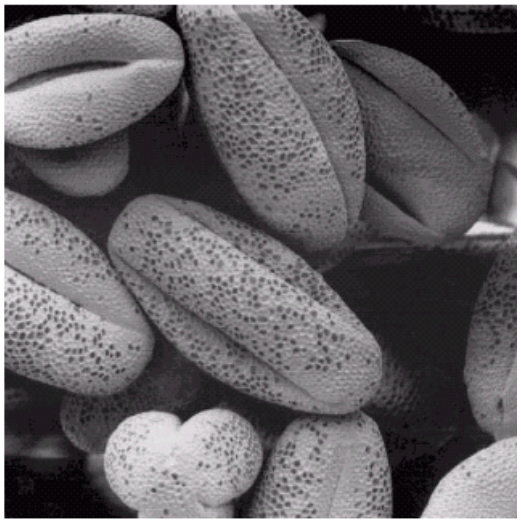
Piecewise Linear Transformation Functions

Rather than using a well defined mathematical function we can use arbitrary user-defined transforms

The images below show a contrast stretching linear transform to add contrast to a poor quality image



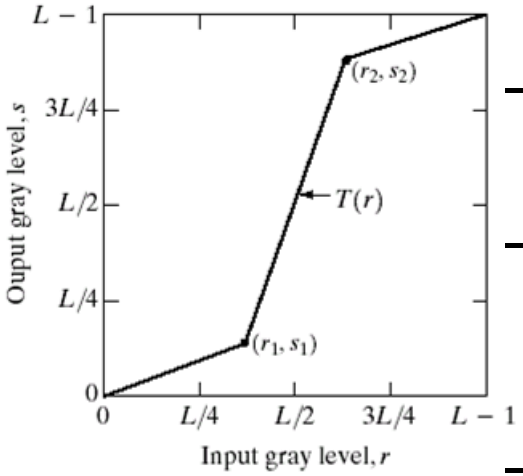




Images taken from Gonzalez & Woods, Digital Image Processing (2002)

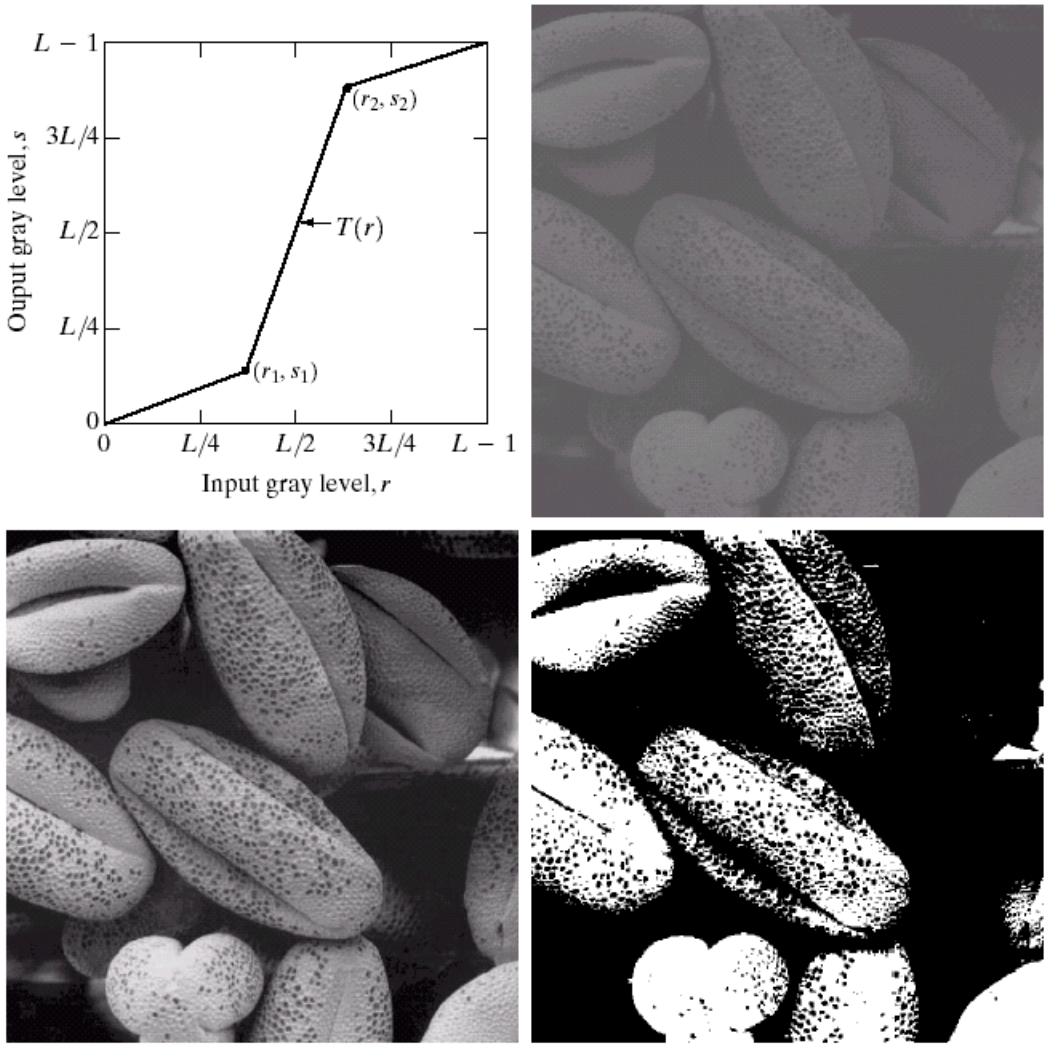
Contrast-Stretching

The locations of (r_1,s_1) and (r_2,s_2) control the shape of the transformation function.



- If $r_1 = s_1$ and $r_2 = s_2$ the transformation is a linear function and produces no changes.
- If $r_1=r_2$, $s_1=0$ and $s_2=L-1$, the transformation becomes a thresholding function that creates a binary image.
- Intermediate values of (r_1,s_1) and (r_2,s_2) produce various degrees of spread in the gray levels of the output image, thus affecting its contrast.
- Generally, $r_1 \leq r_2$ and $s_1 \leq s_2$ is assumed.

Piecewise-Linear Transformation Functions
Contrast Stretching



a b
c d

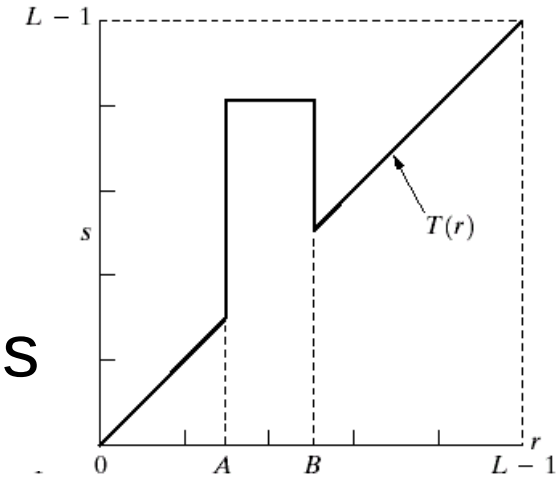
FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

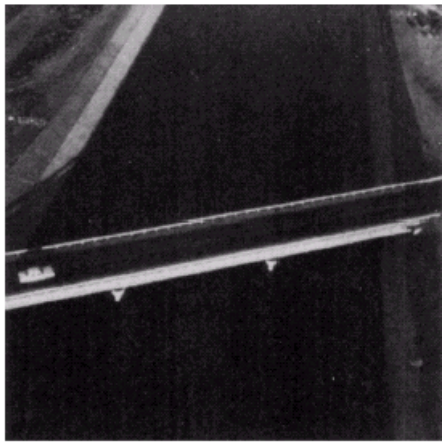
© 2002 R. C. Gonzalez & R. E. Woods

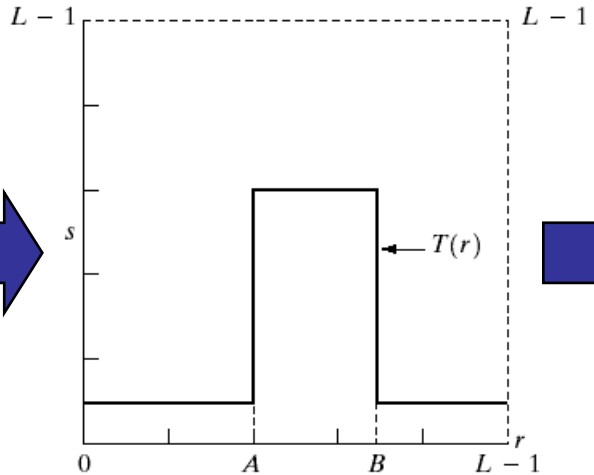
Intensity-level (Gray Level) Slicing

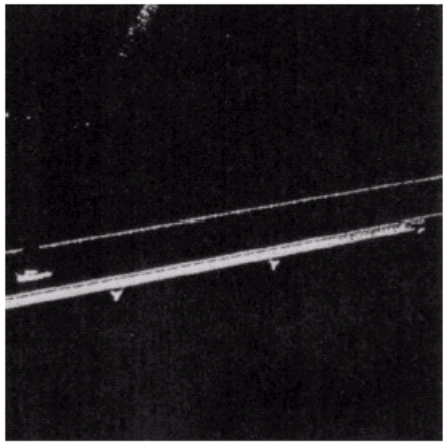
Highlights a specific range of grey levels

- Similar to thresholding
- Other levels can be suppressed or maintained
- Useful for highlighting features in an image







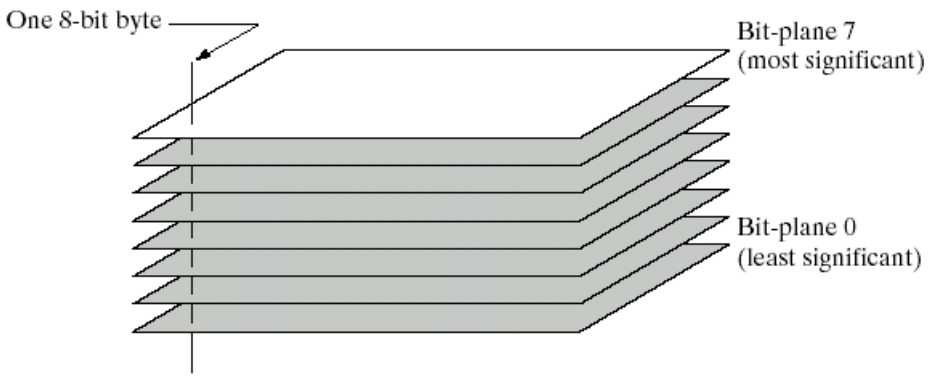


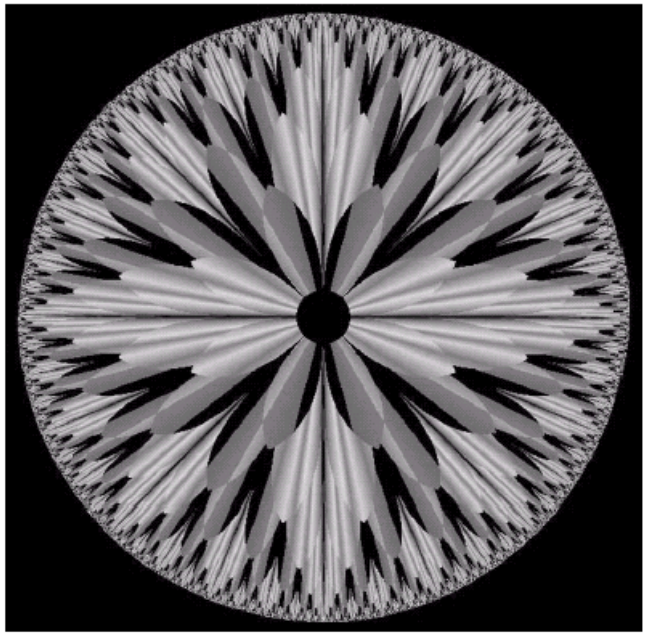
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing

Often by isolating particular bits of the pixel values in an image we can highlight interesting aspects of that image

- Higher-order bits usually contain most of the significant visual information
- Lower-order bits contain subtle details





Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)

[10000000]

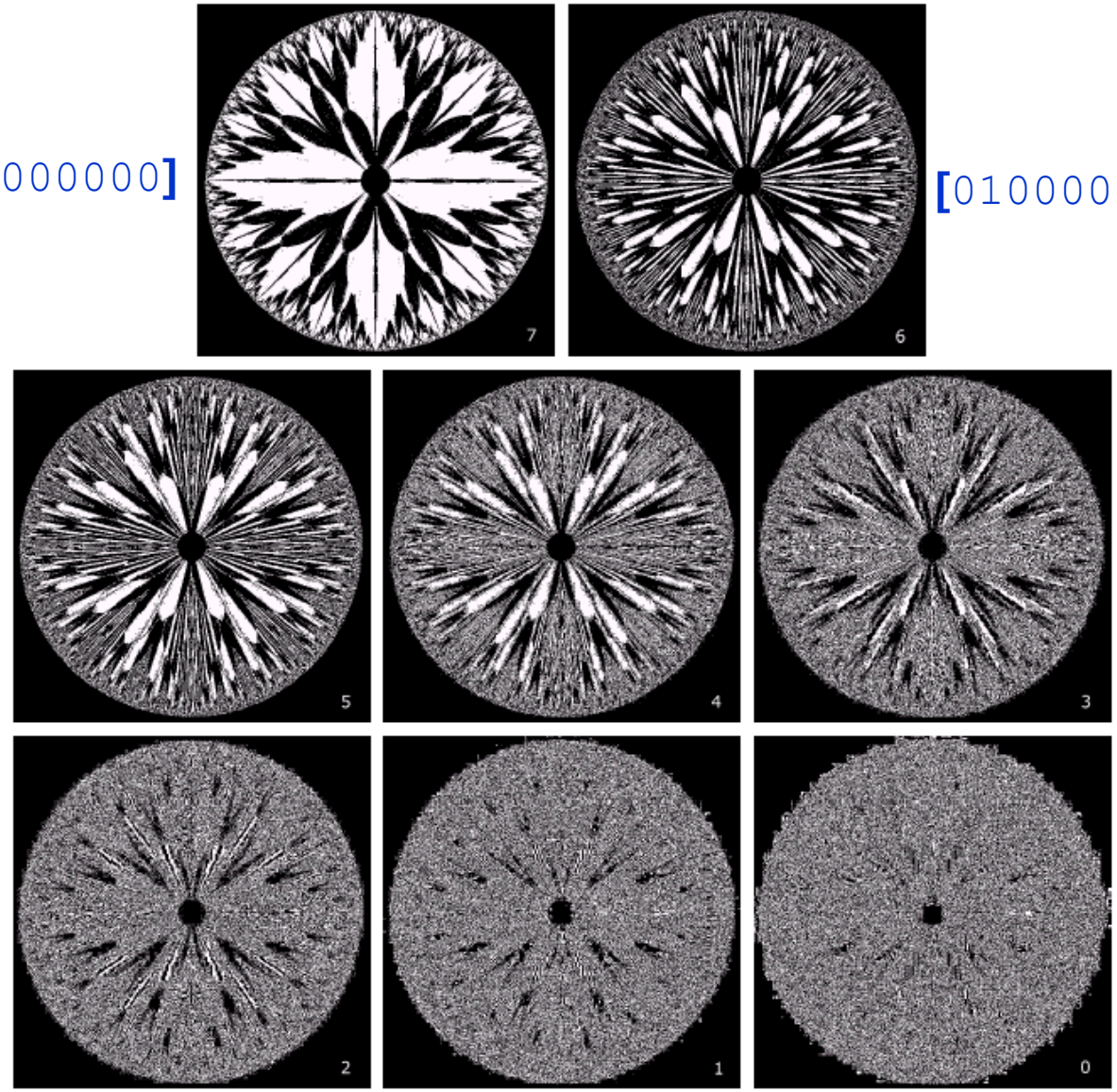
[01000000]

[00100000]

[00001000]

[00000100]

[00000001]



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)

[10000000]

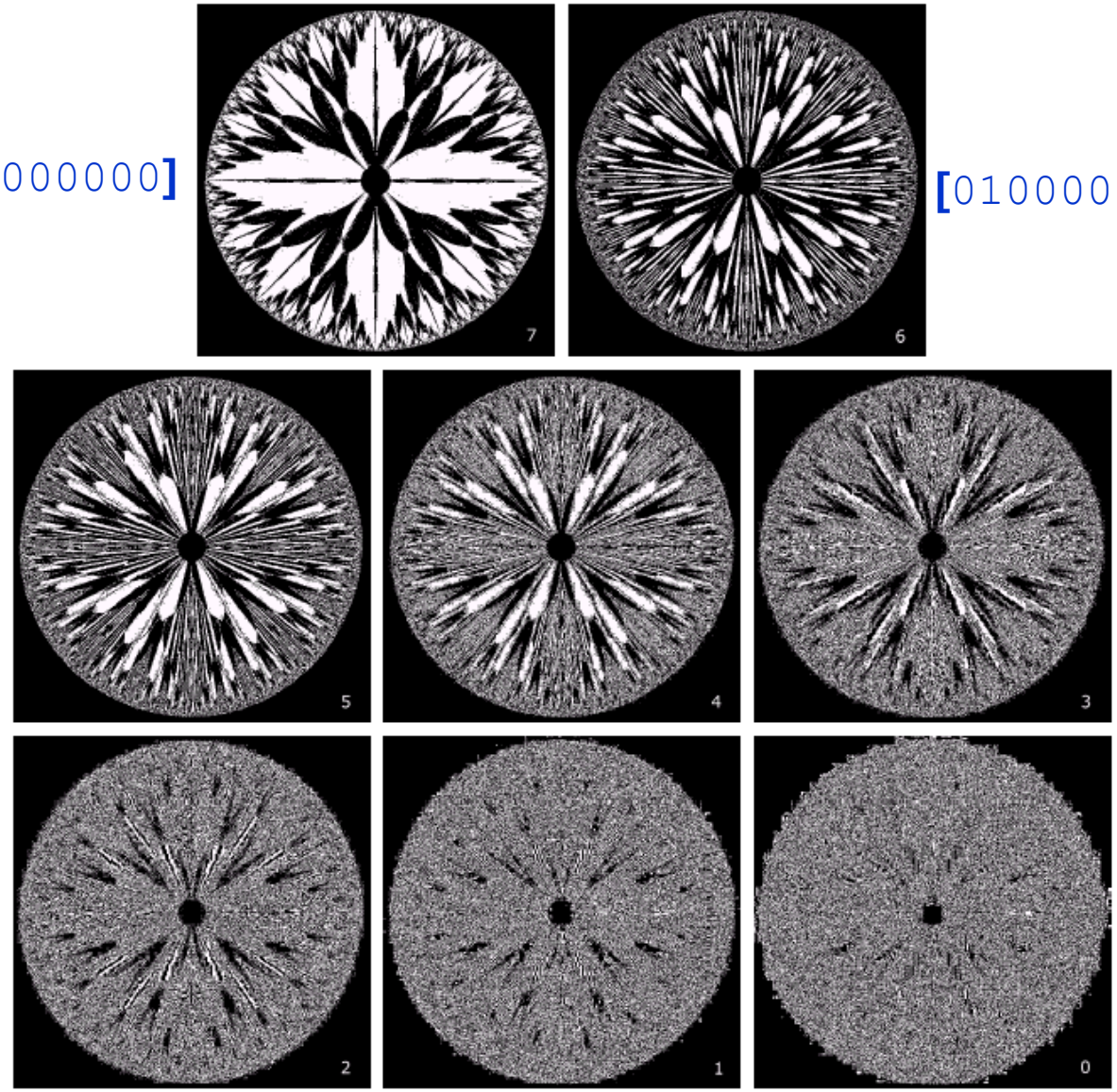
[01000000]

[00100000]

[00001000]

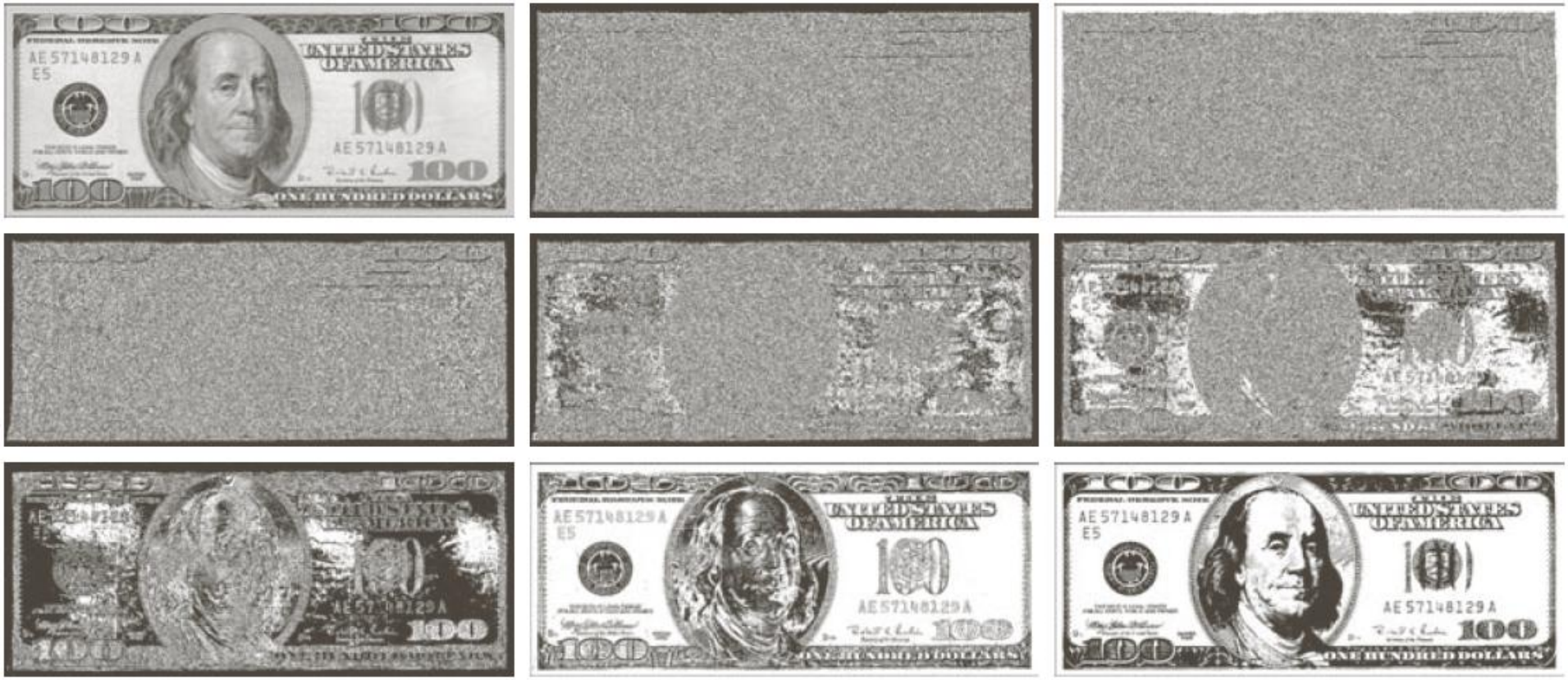
[00000100]

[00000001]



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)



a b c
d e f
g h i

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.

Bit Plane Slicing (cont…)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

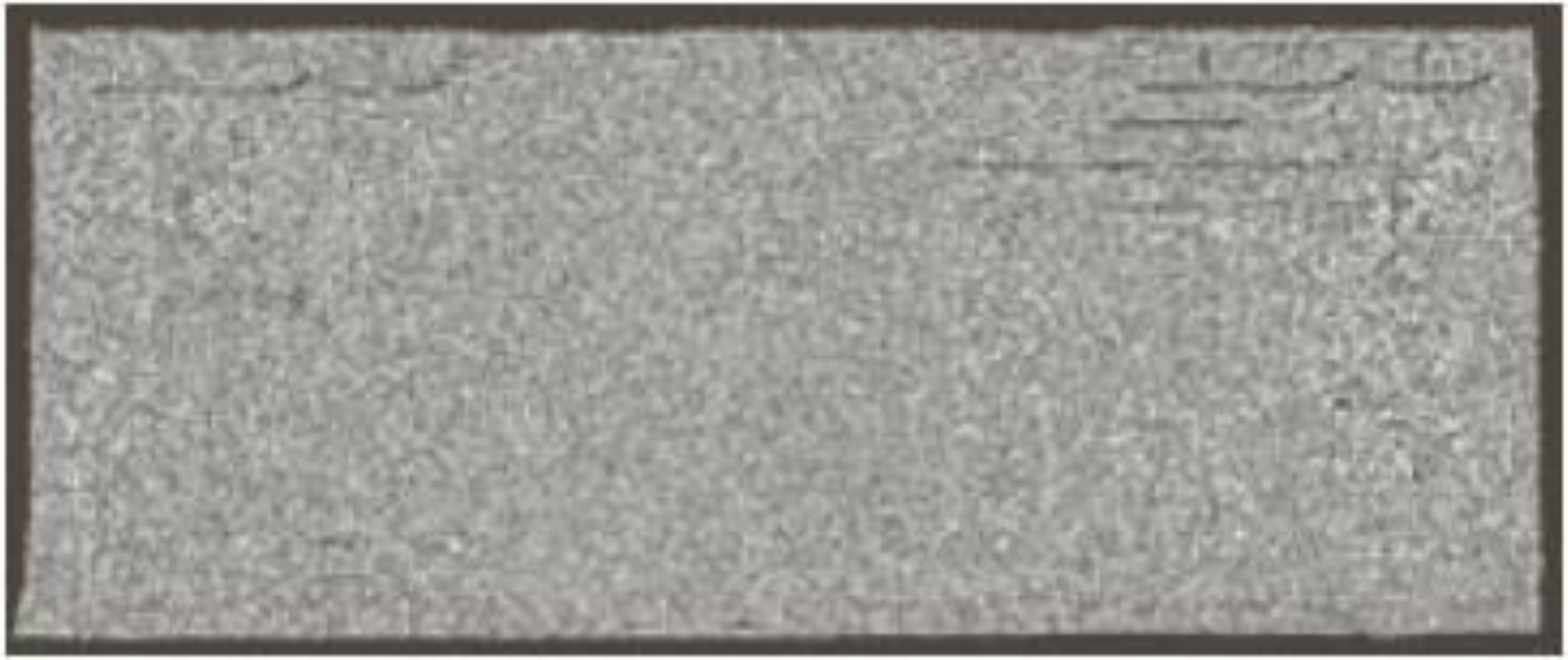
Bit Plane Slicing (cont…)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)

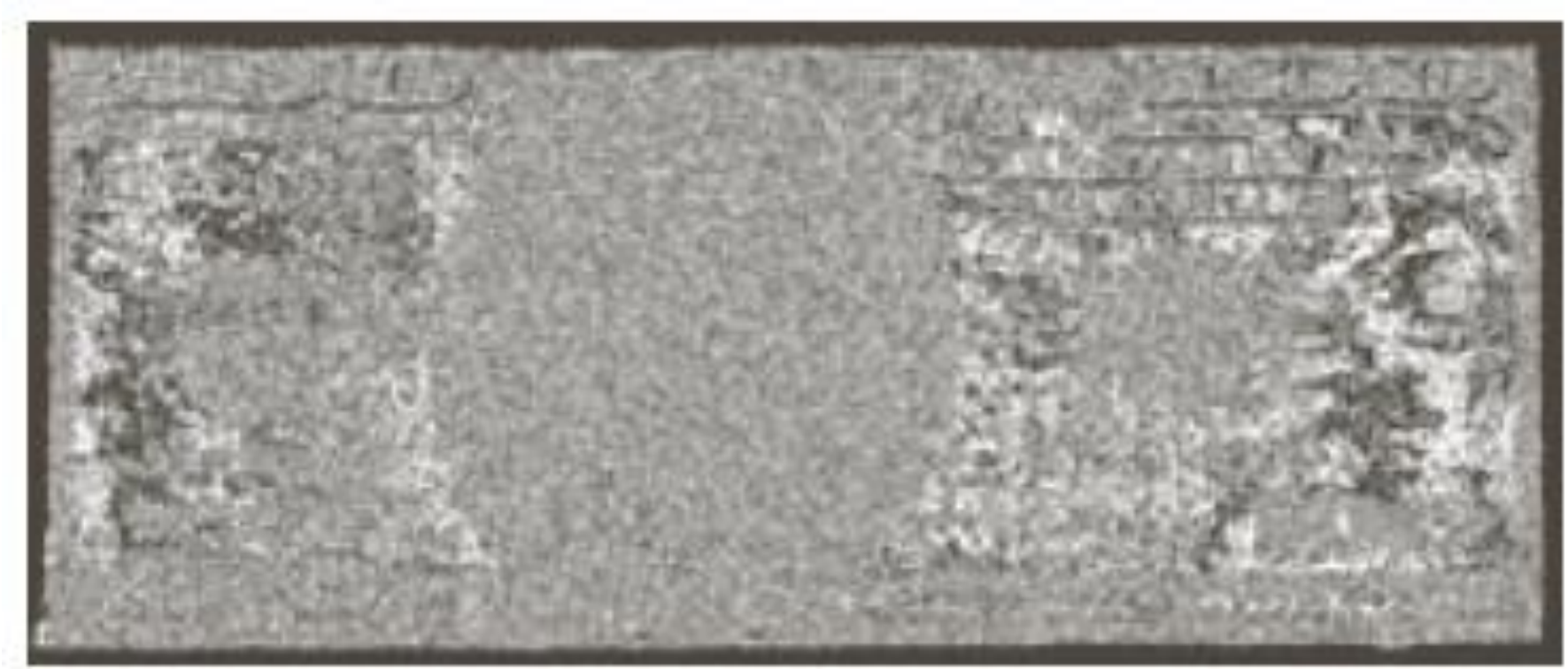
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)



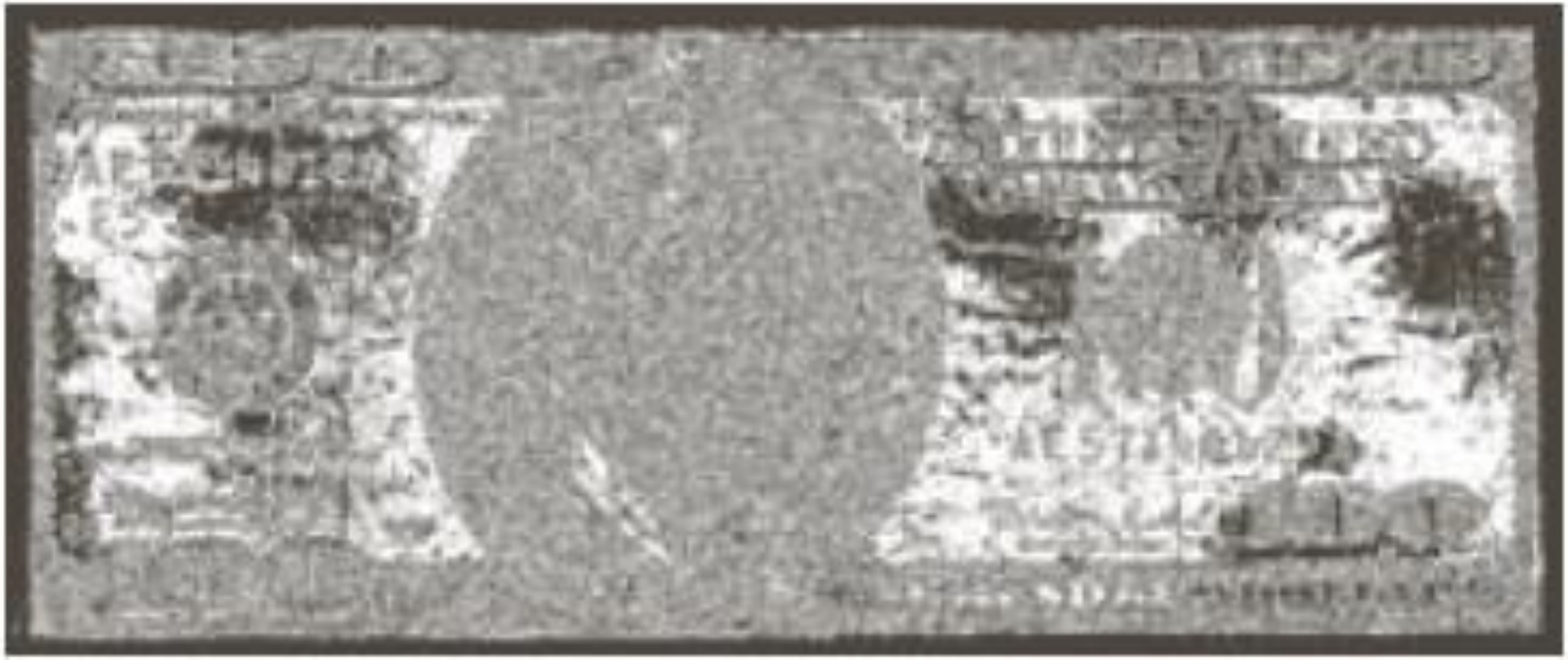
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Bit Plane Slicing (cont…)

Reconstructed image using only bit planes 8 and 7

Reconstructed image using only bit planes 8, 7 and 6

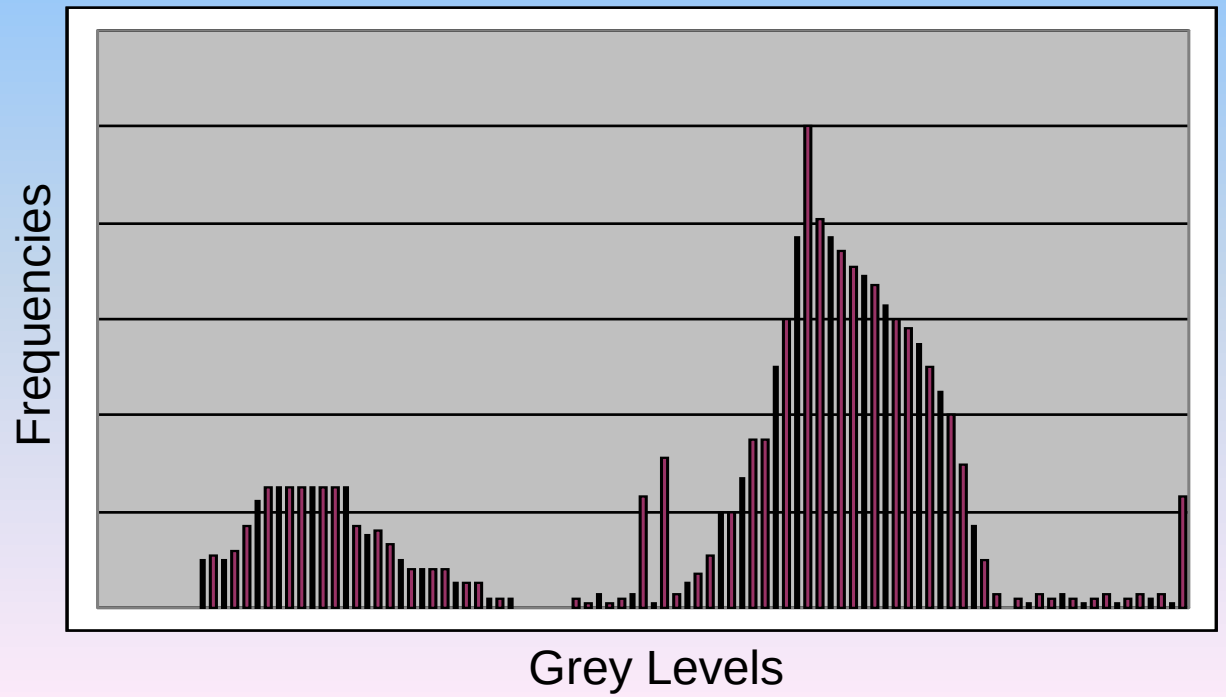
Reconstructed image using only bit planes 7, 6 and 5

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Image Histograms

The histogram of an image shows us the distribution of grey levels in the image

Massively useful in image processing, especially in segmentation



Histogram Processing

The *histogram* of a digital image with intensity levels in the range $[0, L-1]$ is a discrete function $h(r_k) = n_k$, where r_k is the kth intensity value and n_k is the number of pixels in the image with intensity r_k. It is common practice to normalize a histogram by dividing each of its components by the total number of pixels in the image, denoted by the product MN, where, as usual, M and N are the row and column dimensions of the image. Thus, a normalized histogram is given by $p(r_k) = n_k/MN$, for $k = 0, 1, 2, \ldots, L-1$. Loosely speaking, $p(r_k)$ is an estimate of the probability of occurrence of intensity level r_k in an image. The sum of all components of a normalized histogram is equal to 1.

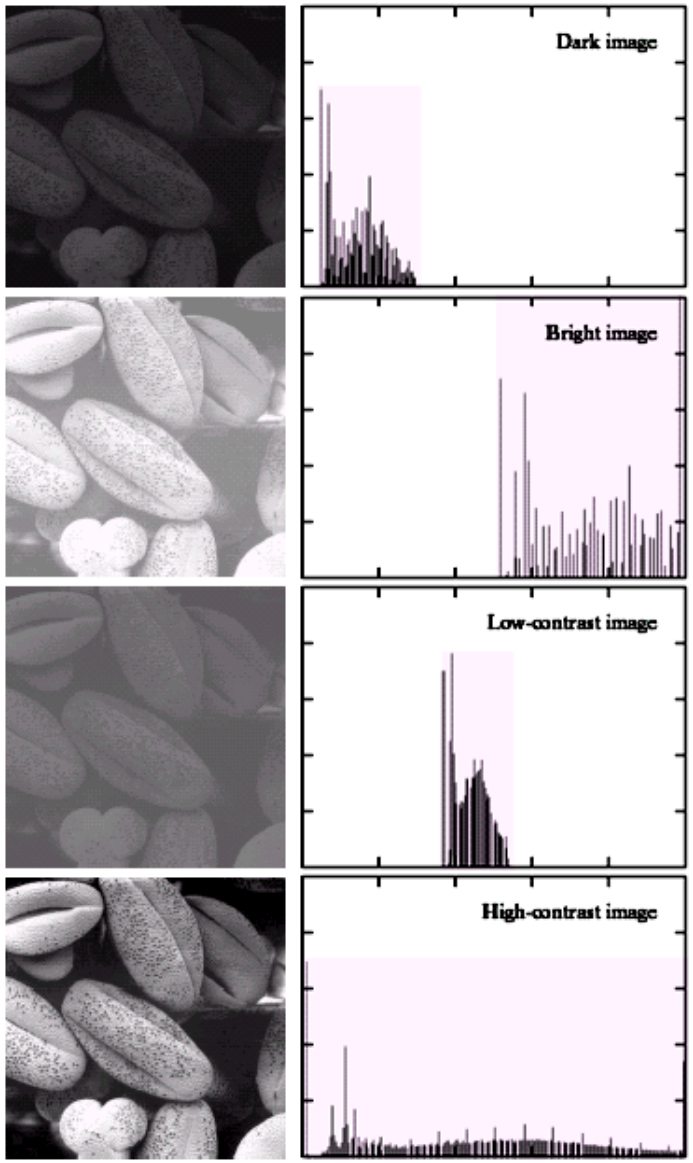
© 2002 R. C. Gonzalez & R. E. Woods

Histogram Examples (cont…)

A selection of images and their histograms

Notice the relationships between the images and their histograms

Note that the high contrast image has the most evenly spaced histogram

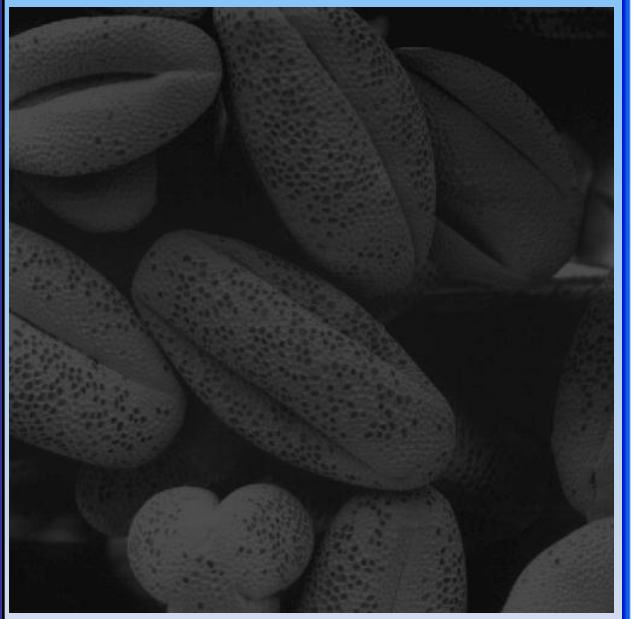


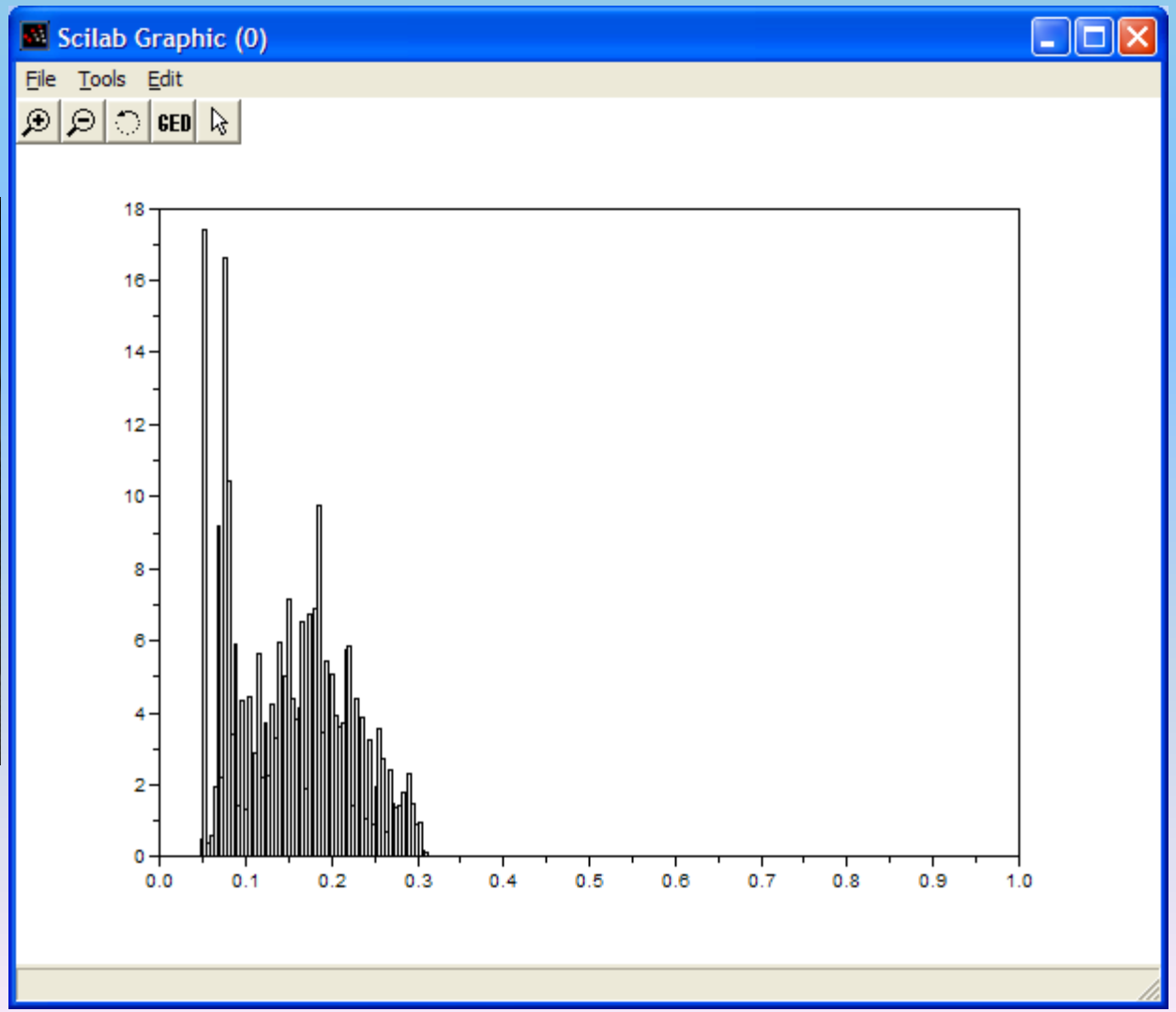
a b

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

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

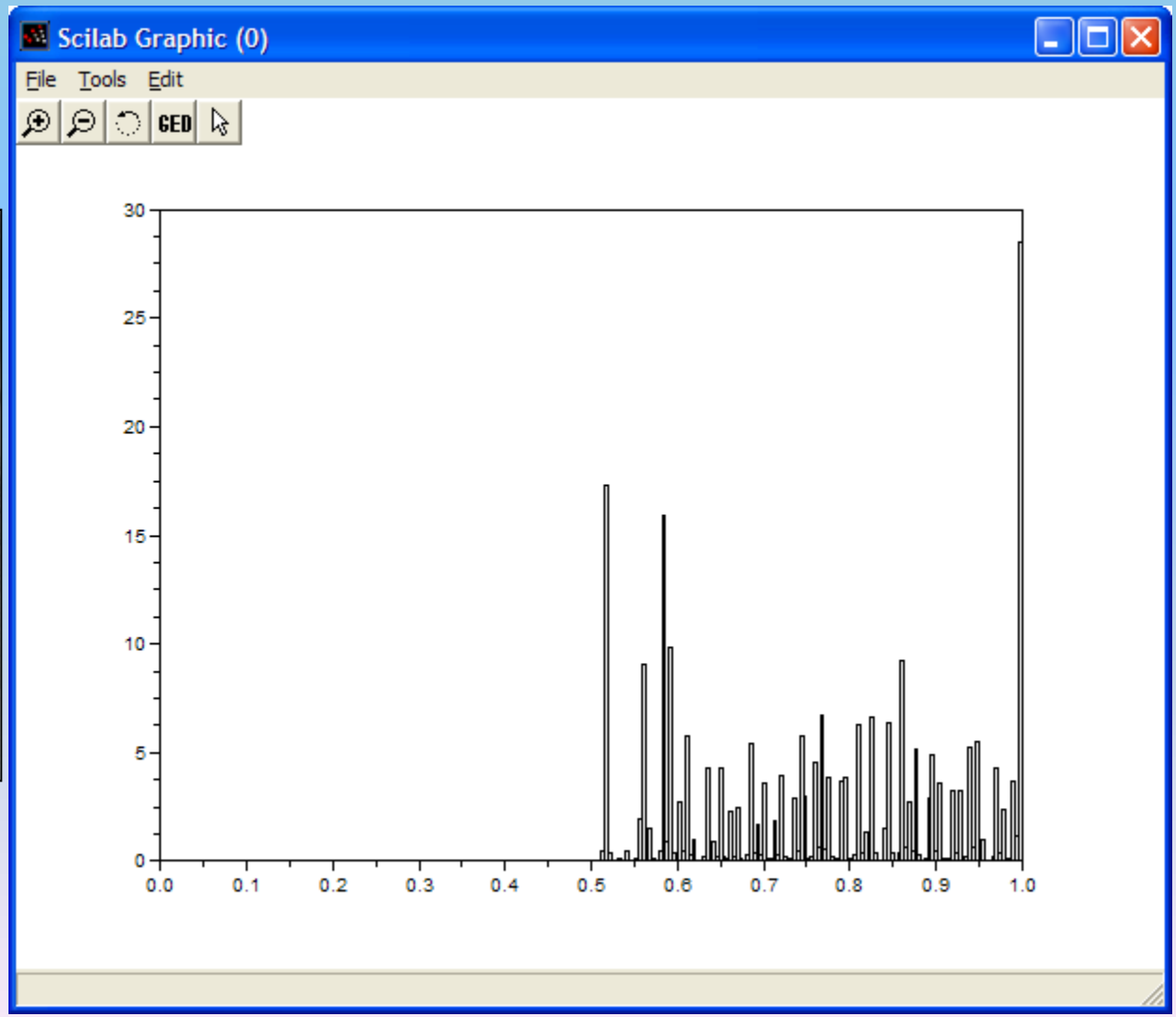
Histogram Examples (cont…)





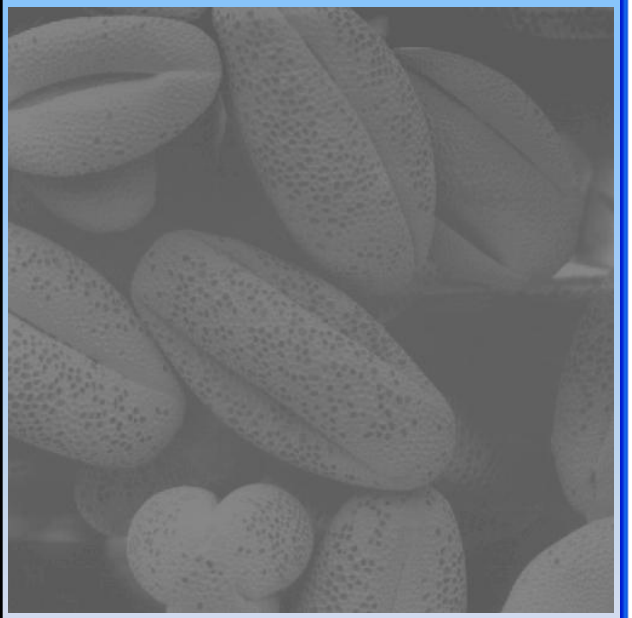
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

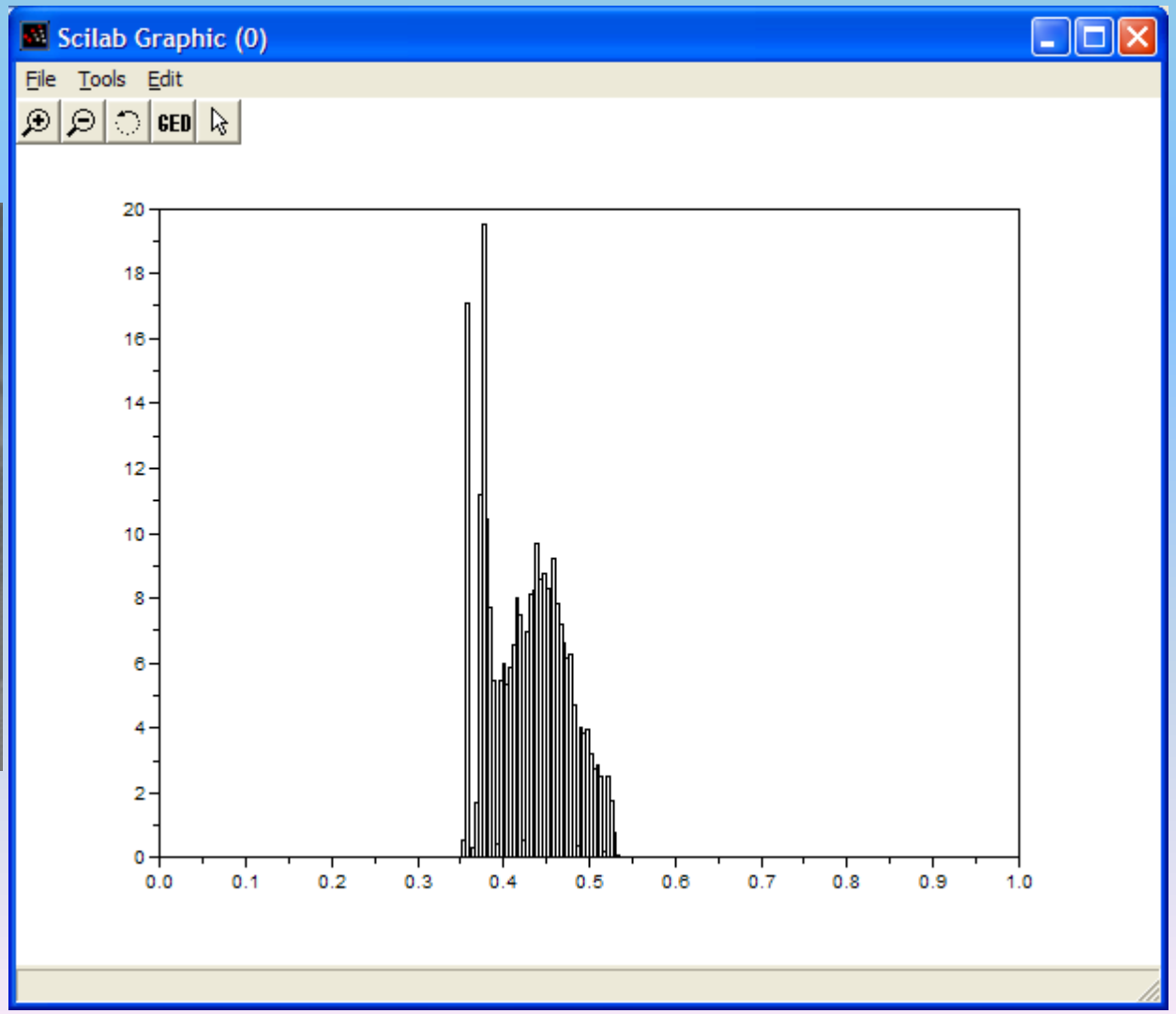
Histogram Examples (cont…)



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

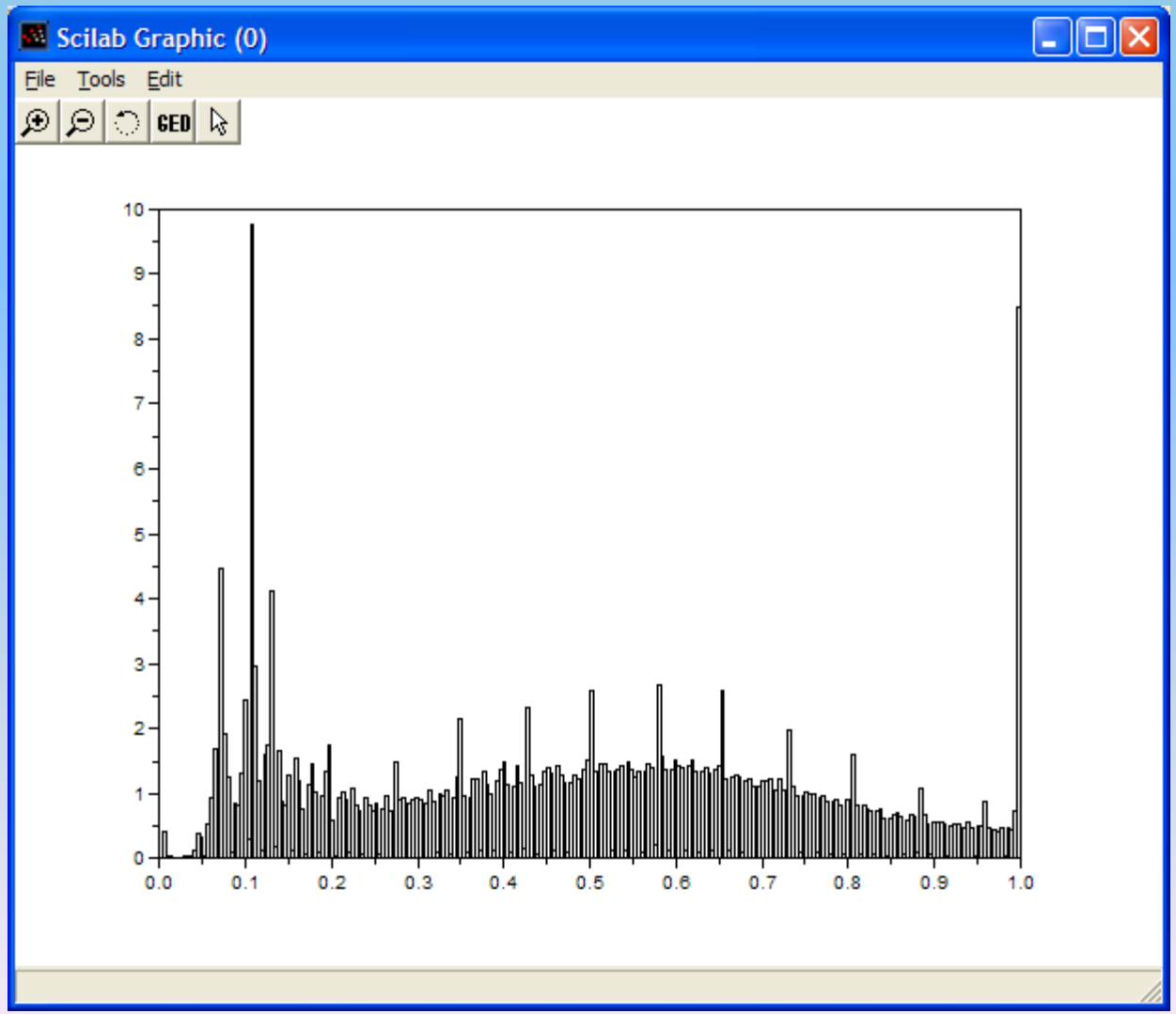
Histogram Examples (cont…)





Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Histogram Examples (cont…)



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Contrast Stretching

We can fix images that have poor contrast by applying a pretty simple contrast specification

The interesting part is how do we decide on this transformation function?

Histogram Stretching

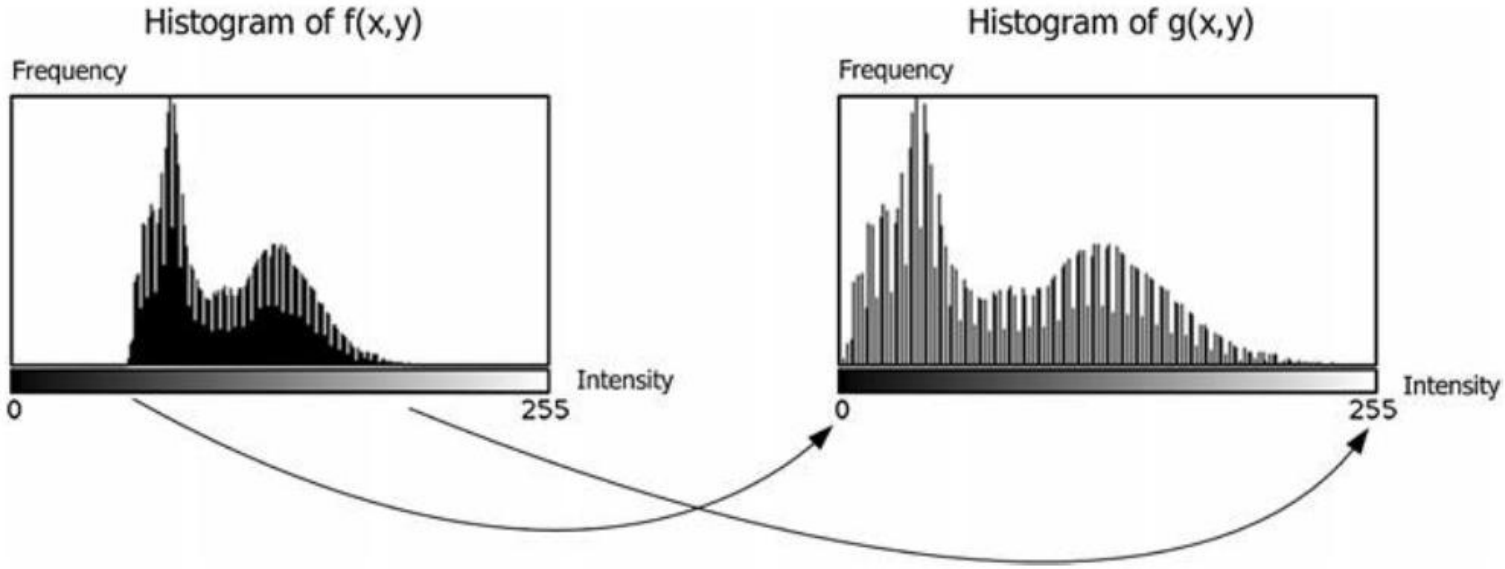
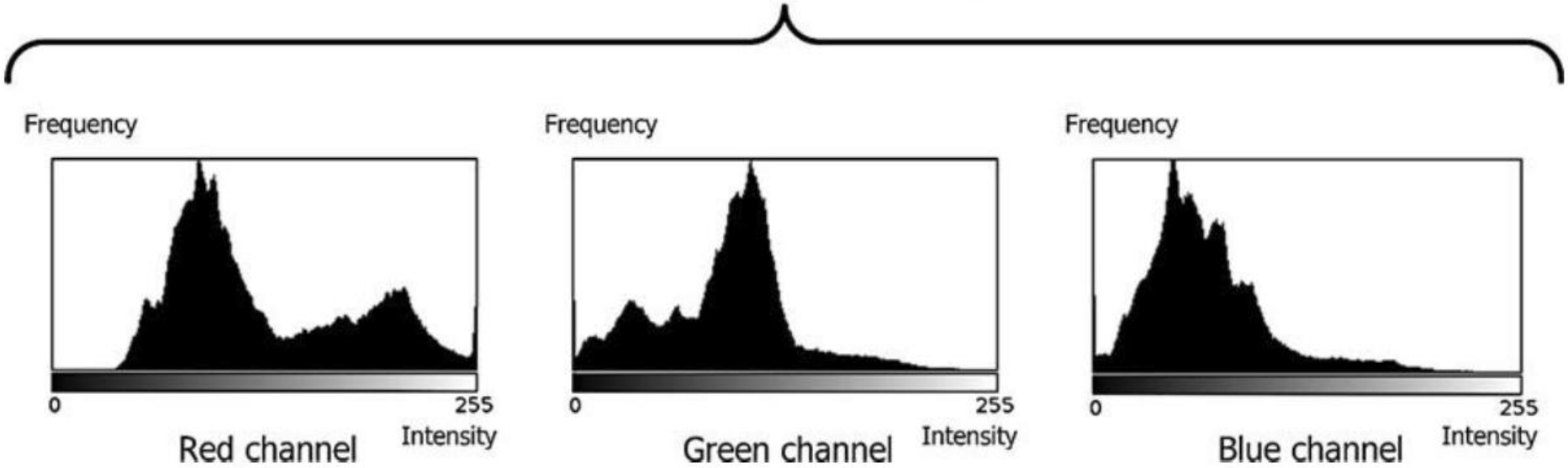


Fig. The concept of histogram stretching

$$g(x, y) = \frac{255}{f_2 - f_1} \cdot \left(f(x, y) - f_1\right)$$

Credits: S. Murala, EEL484 Lecture notes/slides, IIT Ropar



Credits: S. Murala, EEL484 Lecture notes/slides, IIT Ropar

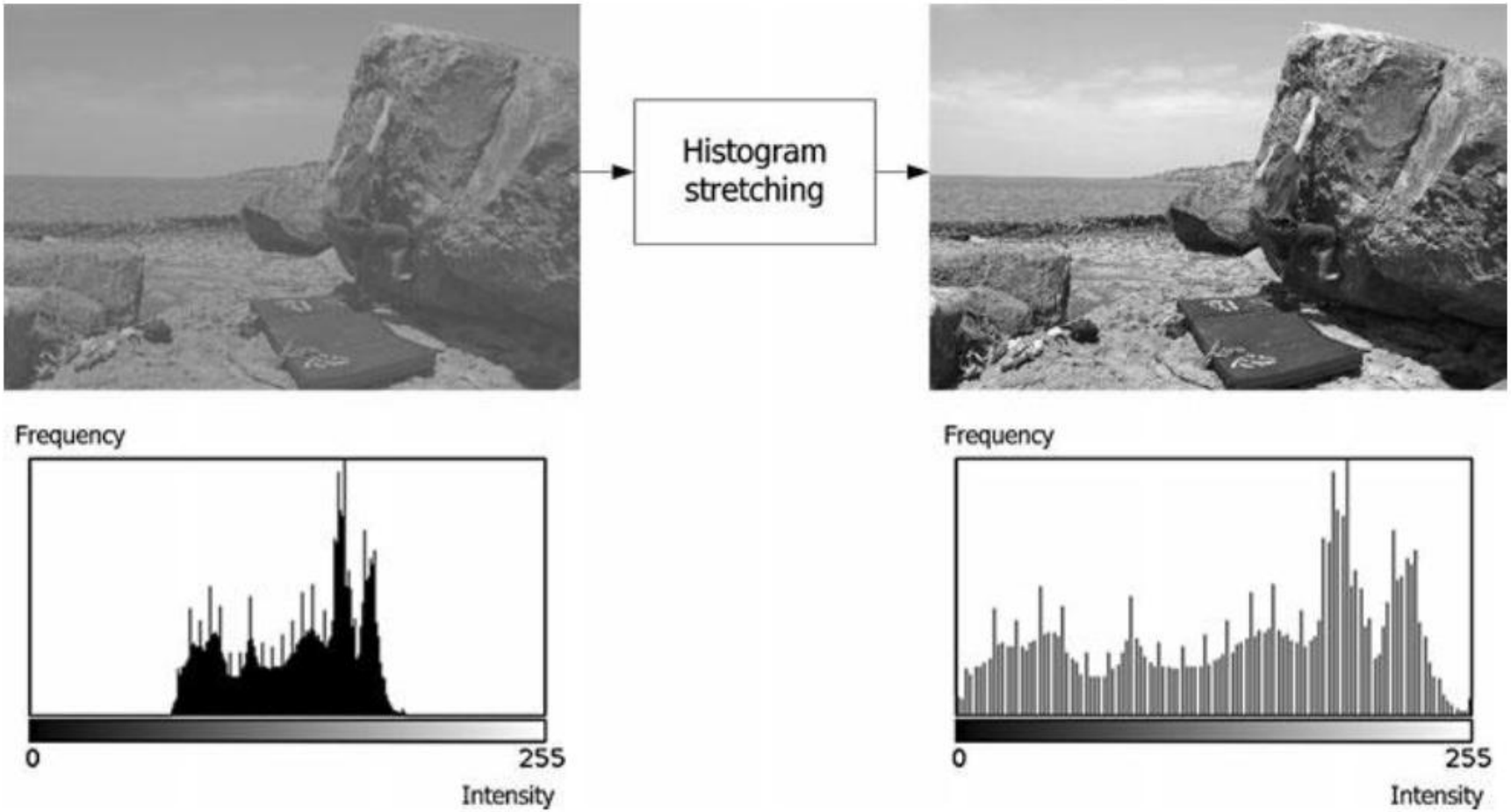


Fig. An example of histogram stretching

Credits: S. Murala, EEL484 Lecture notes/slides, IIT Ropar

- If just one pixel has the value 0 and another 255, histogram stretching will not work, since *f2 −f1 = 255.*

- *A solution is modified histogram stretching where small* bins in the histogram are removed by changing their values to those of larger bins.

- But if a significant number of pixels with very small and very high values exit, we still have *f2−f1 = 255, and hence the histogram (and image) remains the same*.

- A more robust method to improve the histogram (and image) is therefore to apply *histogram equalization.*

Histogram Equalization

- Histogram equalization:
  - To improve the contrast of an image
  - To transform an image in such a way that the transformed image has a nearly uniform distribution of pixel values
- Transformation:
  - Assume r has been normalized to the interval [0,1], with $r = 0$ representing black and $r = 1$ representing white

$$s = T(r) \qquad 0 \le r \le 1$$

  - The transformation function satisfies the following conditions:
    - $T(r)$ is single-valued and monotonically increasing in the interval $0 \le r \le 1$
    - $0 \le T(r) \le 1$ for $0 \le r \le 1$

© 2002 R. C. Gonzalez & R. E. Woods

Histogram Equalization

$$s = T(r) \quad 0 \le r \le L - 1$$

We assume that: **(a)** $T(r)$ is a monotonically† increasing function in the interval $0 \le r \le L - 1$; and

(b) $0 \le T(r) \le L - 1$ for $0 \le r \le L - 1$.

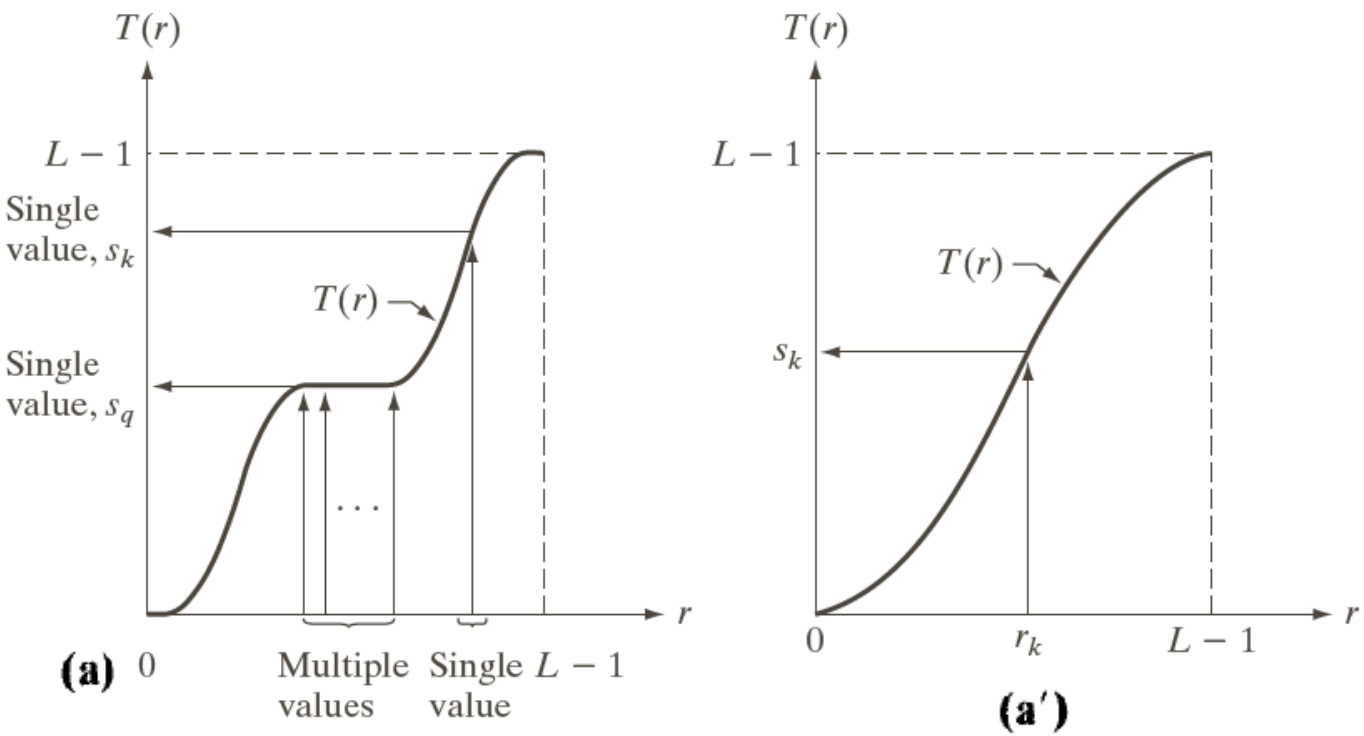
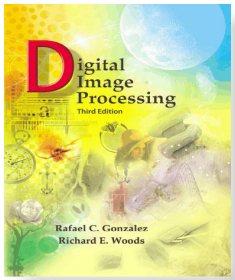


FIGURE 3.17 (a) Monotonically increasing function, showing how multiple values can map to a single value. **(a′)** Strictly monotonically increasing function. This is a one-to-one mapping, both ways.

© 1992–2008 R. C. Gonzalez & R. E. Woods



Histogram Equalization

The intensity levels in an image may be viewed as random variables in the interval $[0, L-1]$. A fundamental descriptor of a random variable is its probability density function (PDF). Let $p_r(r)$ and $p_s(s)$ denote the PDFs of r and s, respectively, where the subscripts on p are used to indicate that p_r and p_s are different functions in general. A fundamental result from basic probability theory is that if $p_r(r)$ and $T(r)$ are known, and $T(r)$ is continuous and differentiable over the range of values of interest, then the PDF of the transformed (mapped) variable s can be obtained using the simple formula

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| \tag{3.3-3}$$

Thus, we see that the PDF of the output intensity variable, s, is determined by the PDF of the input intensities and the transformation function used [recall that r and s are related by $T(r)$].

A transformation function of particular importance in image processing has the form

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw \tag{3.3-4}$$

© 1992–2008 R. C. Gonzalez & R. E. Woods

Histogram Equalization

- Histogram equalization is based on a transformation of the probability density function of a random variable.
- Let $p_r(r)$ and $p_s(s)$ denote the probability density function of random variable r and s, respectively.
- If $p_r(r)$ and $T(r)$ are known, then the probability density function $p_s(s)$ of the transformed variable s can be obtained

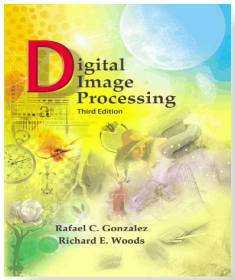
$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

- Define a transformation function $s = T(r) = \int_0^r p_r(w)dw$

  where w is a dummy variable of integration

  and the right side of this equation is the cumulative distribution function of random variable r.

© 1992–2008 R. C. Gonzalez & R. E. Woods



Histogram Equalization

- Given transformation function $T(r)$, $T(r) = \int_0^r p_r(w)dw$

$$\frac{dr}{ds} = \frac{dT(r)}{dr} = \frac{d}{dr}\left[\int_0^r p_r(w)dw\right] = p_r(r)$$

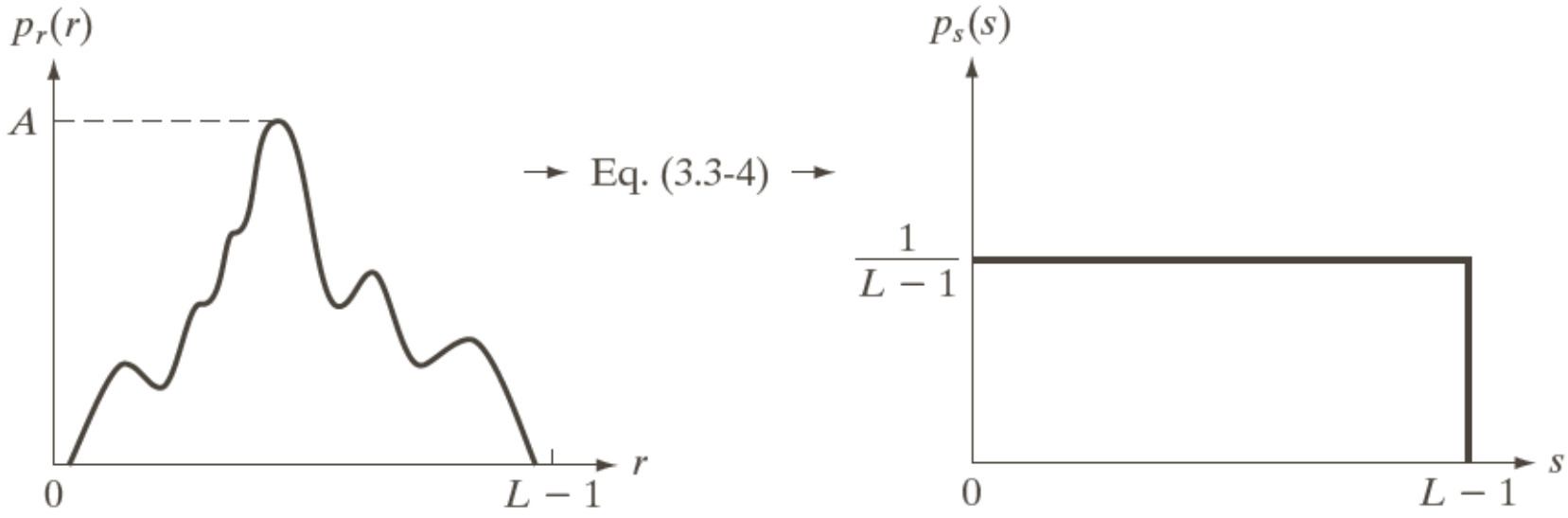
$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = p_r(r)\left|\frac{1}{p_r(r)}\right| = 1 \qquad 0 \le s \le 1$$

➡ $p_s(s)$ now is a uniform probability density function.

- $T(r)$ depends on $p_r(r)$, but the resulting $p_s(s)$ always is uniform.

© 1992–2008 R. C. Gonzalez & R. E. Woods

Histogram Equalization



a b

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.

© 1992–2008 R. C. Gonzalez & R. E. Woods

Histogram Equalization

- To fix ideas, consider the following simple example. Suppose that the (continuous) intensity values in an image have the PDF

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

From Eq. (3.3-4),

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw = \frac{2}{L-1}\int_0^r w\,dw = \frac{r^2}{L-1}$$

$$p_s(s) = \boxed{\begin{aligned} p_s(s) &= p_r(r)\left|\frac{dr}{ds}\right| = \frac{2r}{(L-1)^2}\left|\left[\frac{ds}{dr}\right]^{-1}\right| \\ &= \frac{2r}{(L-1)^2}\left|\left[\frac{d}{dr}\frac{r^2}{L-1}\right]^{-1}\right| \\ &= \frac{2r}{(L-1)^2}\left|\frac{(L-1)}{2r}\right| = \frac{1}{L-1} \end{aligned}}$$

© 1992–2008 R. C. Gonzalez & R. E. Woods

Histogram Equalization

- In discrete version:
  - The probability of occurrence of gray level r_k in an image is

$$p_r(r) = \frac{n_k}{n} \quad k = 0,1,2,\ldots, L-1$$

n : the total number of pixels in the image

n_k : the number of pixels that have gray level r_k

L : the total number of possible gray levels in the image

  - The transformation function is

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} \frac{n_j}{n} \quad k = 0,1,2,\ldots, L-1$$

  - Thus, an output image is obtained by mapping each pixel with level r_k in the input image into a corresponding pixel with level s_k.

© 1992–2008 R. C. Gonzalez & R. E. Woods

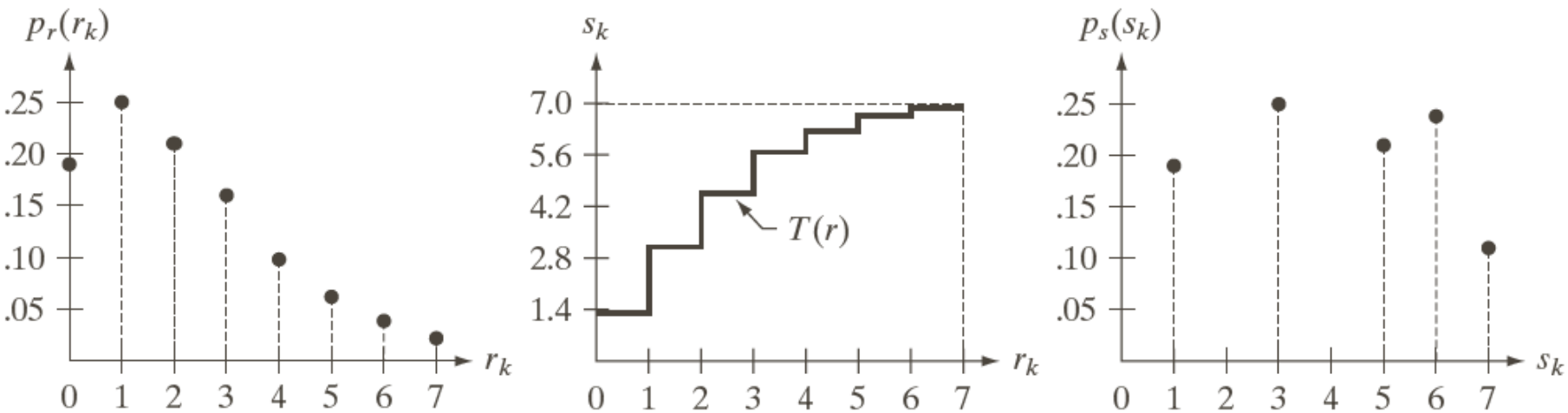
Histogram Equalization

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.

© 1992–2008 R. C. Gonzalez & R. E. Woods

Histogram Equalization



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.

$s_0 = 1.33 \rightarrow 1$	$s_4 = 6.23 \rightarrow 6$
$s_1 = 3.08 \rightarrow 3$	$s_5 = 6.65 \rightarrow 7$
$s_2 = 4.55 \rightarrow 5$	$s_6 = 6.86 \rightarrow 7$
$s_3 = 5.67 \rightarrow 6$	$s_7 = 7.00 \rightarrow 7$

© 1992–2008 R. C. Gonzalez & R. E. Woods

Histogram Equalization

Example: 3-bit image (L=8), 64x64 pixels:

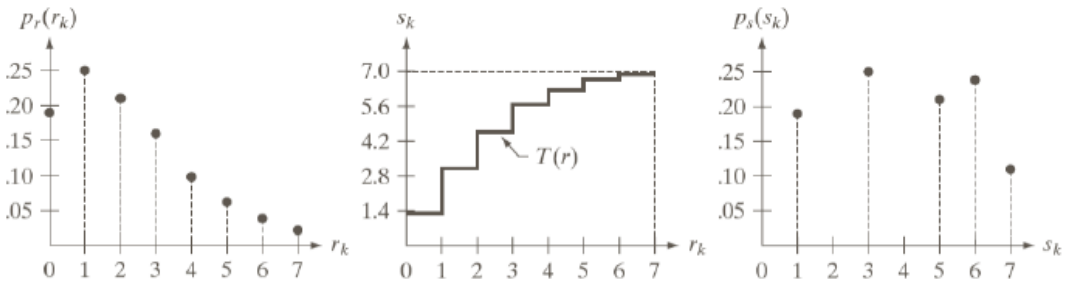
r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

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

Round intensity values to nearest integer:

$$s_0 = 1.33 \rightarrow 1 \quad s_4 = 6.23 \rightarrow 6$$
$$s_1 = 3.08 \rightarrow 3 \quad s_5 = 6.65 \rightarrow 7$$
$$s_2 = 4.55 \rightarrow 5 \quad s_6 = 6.86 \rightarrow 7$$
$$s_3 = 5.67 \rightarrow 6 \quad s_7 = 7.00 \rightarrow 7$$

=> 5 intensity levels



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 or *histogram linearization* transformation (discrete form):

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

$$= \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j$$

r_k: intensity levels in the *input* image

s_k: intensity levels in the *output* image

© 1992–2008 R. C. Gonzalez & R. E.

Histogram Equalisation

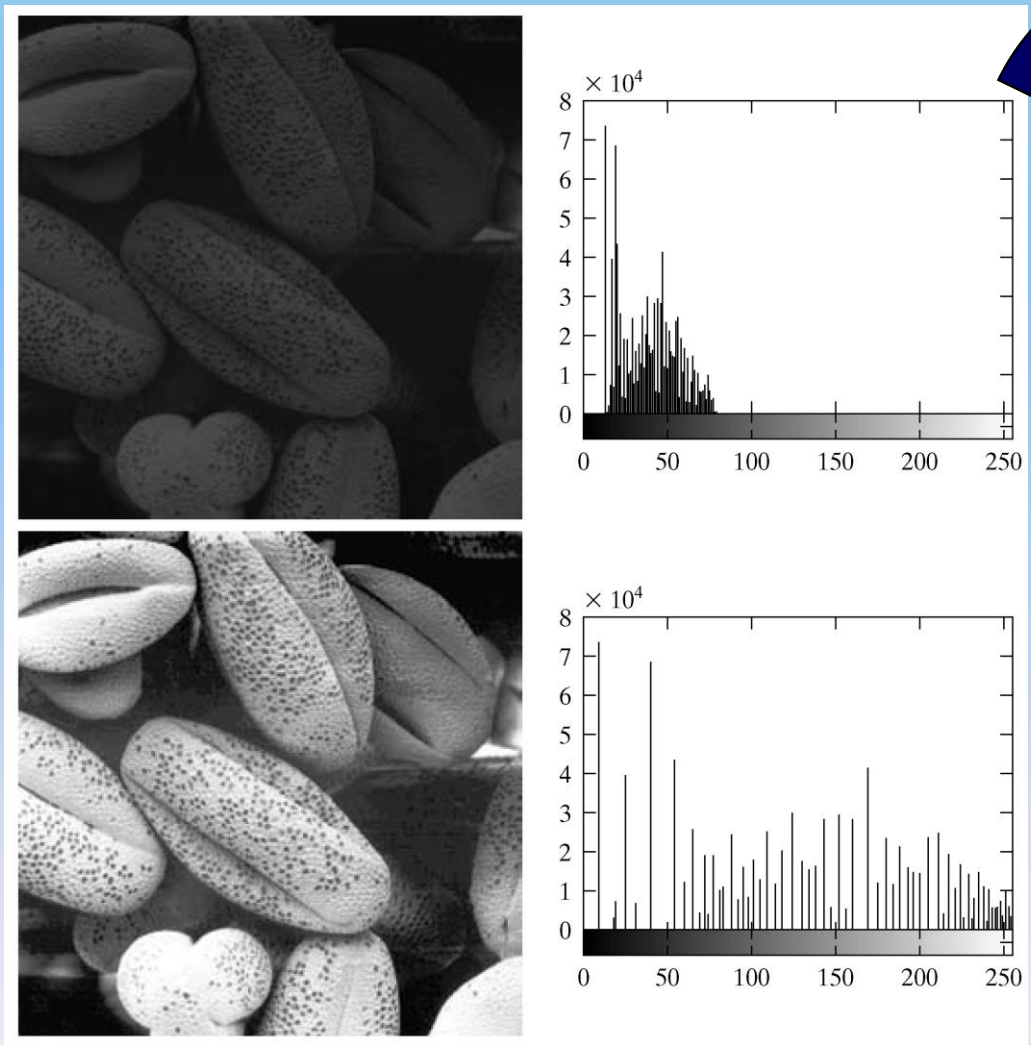
Spreading out the frequencies in an image (or equalising the image) is a simple way to improve dark or washed out images

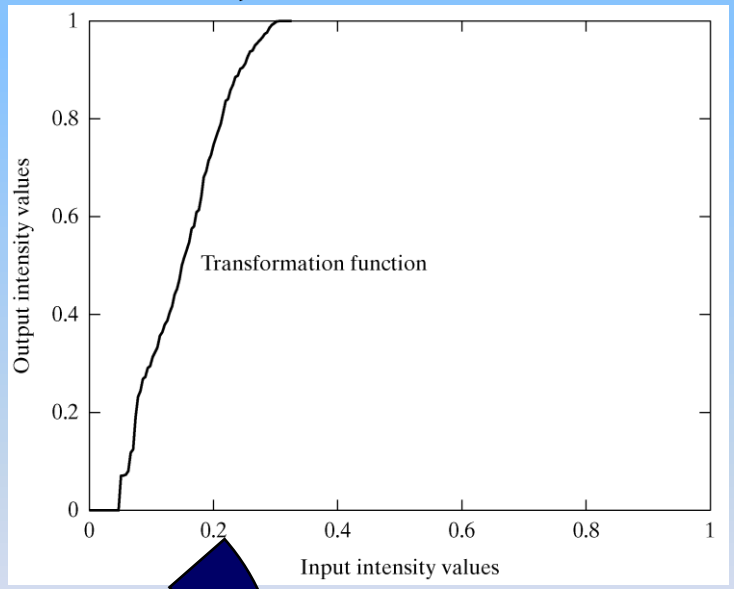
The formula for histogram equalisation is given where

- r_k: input intensity
- s_k: processed intensity
- k: the intensity range (e.g 0.0 – 1.0)
- n_j: the frequency of intensity j
- n: the sum of all frequencies

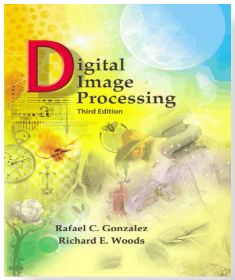
$$s_k = T(r_k) = \sum_{j=1}^{k} p_r(r_j) = \sum_{j=1}^{k} \frac{n_j}{n}$$

Equalisation Transformation Function



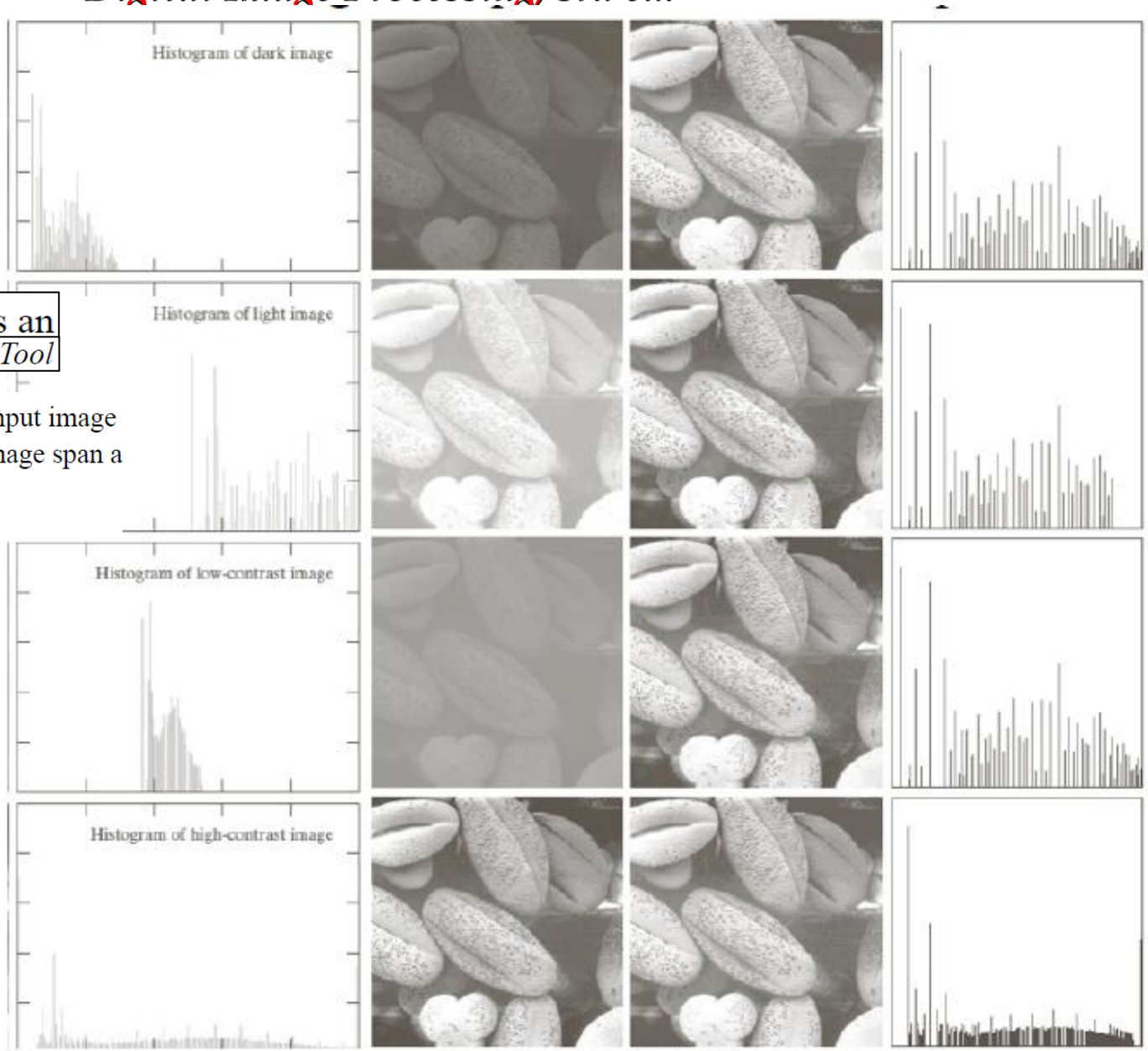


Images taken from Gonzalez & Woods, Digital Image Processing (2002)



Histogram equalization as an *Adaptive Contrast Enhancement Tool*

Tend to spread the histogram of the input image
⇒ intensity levels of the equalized image span a wider range
⇒ *contrast enhancement*



a) Original Histograms | *b) Original Images* | *c) Histogram-equalised im.* | *Histograms of c)*

© 1992–2008 R. C. Gonzalez & R. E. Woods

Histogram Equalization

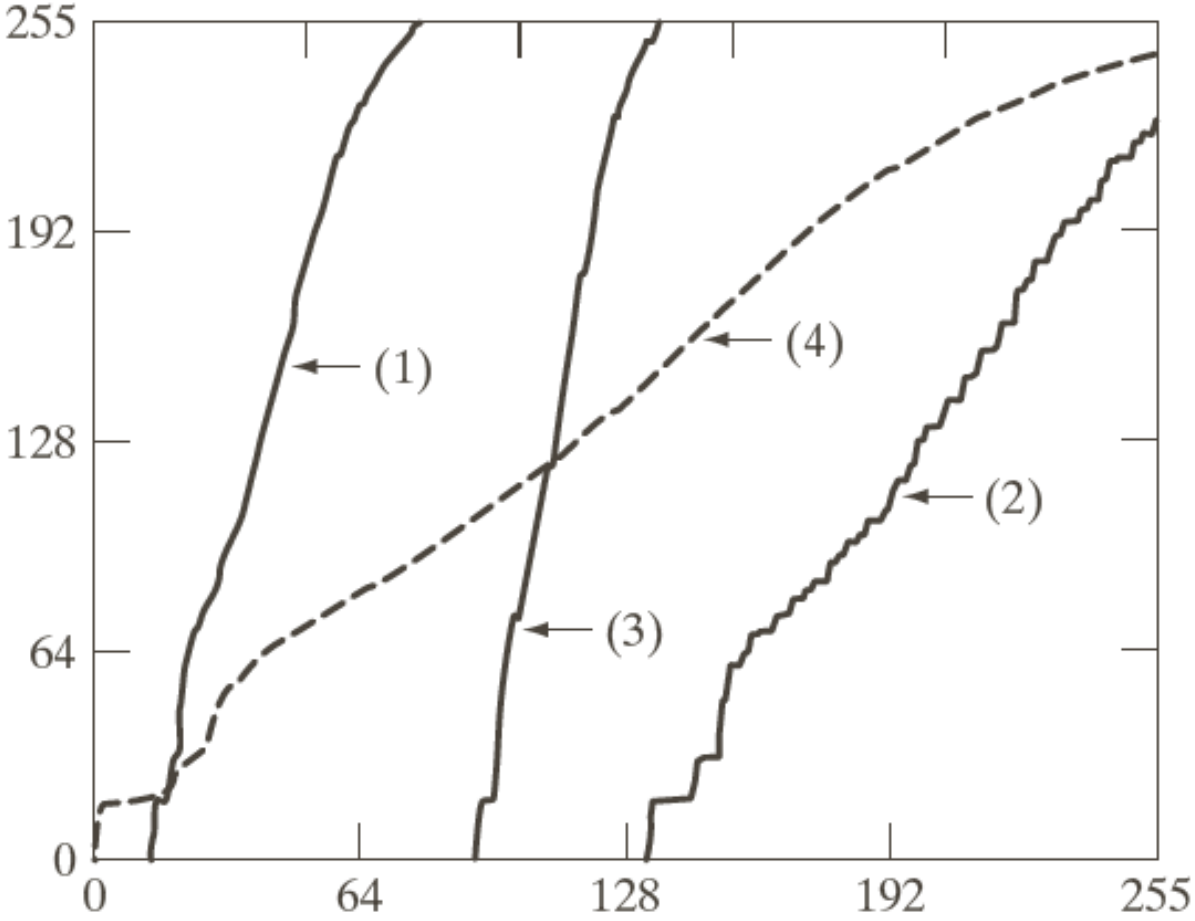


FIGURE 3.21 Transformation functions for histogram equalization. Transformations (1) through (4) were obtained from the histograms of the images (from top to bottom) in the left column of Fig. 3.20 using Eq. (3.3-8).

© 1992–2008 R. C. Gonzalez & R. E. Woods

Histogram Equalization

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

- Histogram equalization (HE) results are similar to contrast stretching but offer the advantage of full automation, since HE automatically determines a transformation function to produce a new image with a uniform histogram.

Histogram Matching

- Histogram equalization does not allow interactive image enhancement and generates only one result: an approximation to a uniform histogram.
- Sometimes though, we need to be able to specify particular histogram shapes capable of highlighting certain gray-level ranges.
- Histogram matching is similar to histogram equalization, except that instead of trying to make the output image have a flat histogram, we would like it to have a histogram of a specified shape, say $p_z(z)$.

© 2002 R. C. Gonzalez & R. E. Woods

Histogram Matching

We can estimate $p_r(r)$ from the given input image, while $p_z(z)$ is the *specified* probability density function that we wish the output image to have.

Let s be a random variable with the property

$$s = T(r) = (L - 1)\int_0^r p_r(w)\,dw \tag{3.3-10}$$

where, as before, w is a dummy variable of integration. We recognize this expression as the continuous version of histogram equalization given in Eq. (3.3-4).

Suppose next that we define a random variable z with the property

$$G(z) = (L - 1)\int_0^z p_z(t)\,dt = s \tag{3.3-11}$$

where t is a dummy variable of integration. It then follows from these two equations that $G(z) = T(r)$ and, therefore, that z must satisfy the condition

$$z = G^{-1}[T(r)] = G^{-1}(s) \tag{3.3-12}$$

© 2002 R. C. Gonzalez & R. E. Woods

Histogram Matching

Procedure to obtain an image whose intensity levels have a specified PDF:

1. Obtain $p_r(r)$ from the input image, and use Eq. (3-3.1) to obtain the values of s
2. Use the specified PDF in Eq. (3-3.2) to obtain the transformation function $G(z)$
3. Obtain the inverse transformation $z = G^{-1}(s)$ (i.e. *mapping* from s to z)
4. Obtain the output image by equalizing the input image using Eq. (3-3.1). For each pixel with value s in the equalized image, perform the inverse mapping $z = G^{-1}(s)$ to obtain pixels in the output image.

© 1992–2008 R. C. Gonzalez & R. E. Woods

Histogram Matching

Discrete Formulation

Input image → output image
$r_k \rightarrow z_k$ (intensity levels)

$p_z(z_k)$: *specified* PDF

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$
$$= \frac{(L-1)}{MN}\sum_{j=0}^{k} n_k \qquad (3\text{-}3.4)$$

Transformation Function: $$G(z_q) = (L-1)\sum_{i=0}^{q} p_z(z_i) \qquad (3\text{-}3.5)$$

Mapping: $$z_q = G^{-1}(s_k) \qquad (3\text{-}3.6)$$

© 1992–2008 R. C. Gonzalez & R. E. Woods

Histogram Matching

s_k being the values of the histogram-equalized image,

Discrete formulation of the histogram-specification procedure:

1. Compute $p_r(r)$ of input image, and use Eq. (3-3.4) to obtain the values of s_k, *rounded* to the integer range $[0,L-1]$
2. Compute all values of G using Eq. (3-3.5) for $q=0,1,\ldots,L-1$ and the specified histogram values. Round the values of G to integers in the range $[0,L-1]$ and store those values in a table.
3. For every value of s_k, use stored values of G to find the corresponding value of z_q so that $G(z_q)$ is closest to s_k and store these mapping from s to z. When mapping not unique, choose the smallest value by convention.
4. Form the histogram-specified image by "histogram-equalizing" the input image and then mapping every equalized pixel value s_k of this image to the corresponding value z_q using the mapping found in step 3.

© 1992–2008 R. C. Gonzalez & R. E. Woods

Histogram Matching

Example of Histogram Specification

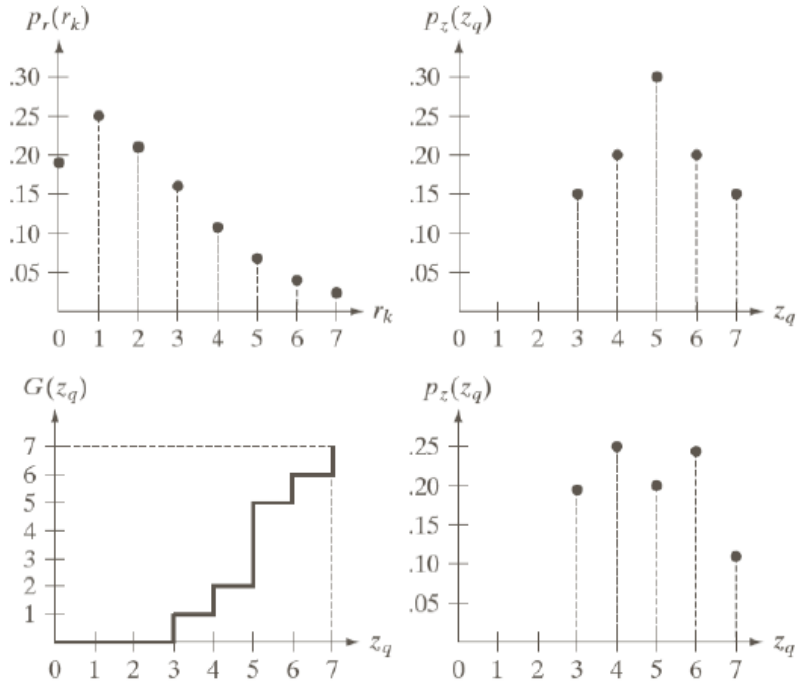
Original histogram

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 $p_z(z_q)$	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

Specified histogram

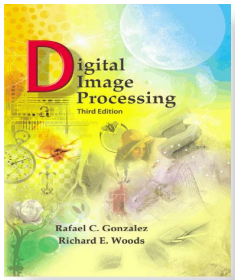
TABLE 3.2 Specified and actual histograms (the values in the third column are from the computations performed in the body of Example 3.8).



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).

© 1992–2008 R. C. Gonzalez & R. E. Woods



Histogram Matching

Original 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

Example of Histogram Specification

Step 1: obtain scaled histogram-equalized values:

$$s_0 = 1.33 \rightarrow 1 \quad s_4 = 6.23 \rightarrow 6$$
$$s_1 = 3.08 \rightarrow 3 \quad s_5 = 6.65 \rightarrow 7$$
$$s_2 = 4.55 \rightarrow 5 \quad s_6 = 6.86 \rightarrow 7$$
$$s_3 = 5.67 \rightarrow 6 \quad s_7 = 7.00 \rightarrow 7$$

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

TABLE 3.2 Specified and actual histograms (the values in the third column are from the computations performed in the body of Example 3.8).

Step 2: compute all values of G, rounded:

z_q	$G(z_q)$
$z_0 = 0$	0
$z_1 = 1$	0
$z_2 = 2$	0
$z_3 = 3$	1
$z_4 = 4$	2
$z_5 = 5$	5
$z_6 = 6$	6
$z_7 = 7$	7

TABLE 3.3 All possible values of the transformation function G scaled, rounded, and ordered with respect to z.

© 1992–2008 R. C. Gonzalez & R. E. Woods

Histogram Matching

$s_0 \rightarrow 1$
$s_1 \rightarrow 3$
$s_2 \rightarrow 5$
$s_3 \rightarrow 6$
$s_4 \rightarrow 6$
$s_5 \rightarrow 7$
$s_6 \rightarrow 7$
$s_7 \rightarrow 7$

z_q	$G(z_q)$
$z_0 = 0$	0
$z_1 = 1$	0
$z_2 = 2$	0
$z_3 = 3$	1
$z_4 = 4$	2
$z_5 = 5$	5
$z_6 = 6$	6
$z_7 = 7$	7

Step 3: for every s_k, find the smallest value of z_q so that $G(z_q)$ is the closest to s_k:

s_k	$\rightarrow$	z_q
1	$\rightarrow$	3
3	$\rightarrow$	4
5	$\rightarrow$	5
6	$\rightarrow$	6
7	$\rightarrow$	7

TABLE 3.4 Mappings of all the values of s_k into corresponding values of z_q.

Step 4: use Table 3.4 to map pixels in the histogram equalized image into a pixel in the new histogram-specified image

Values of the resulting histogram:

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

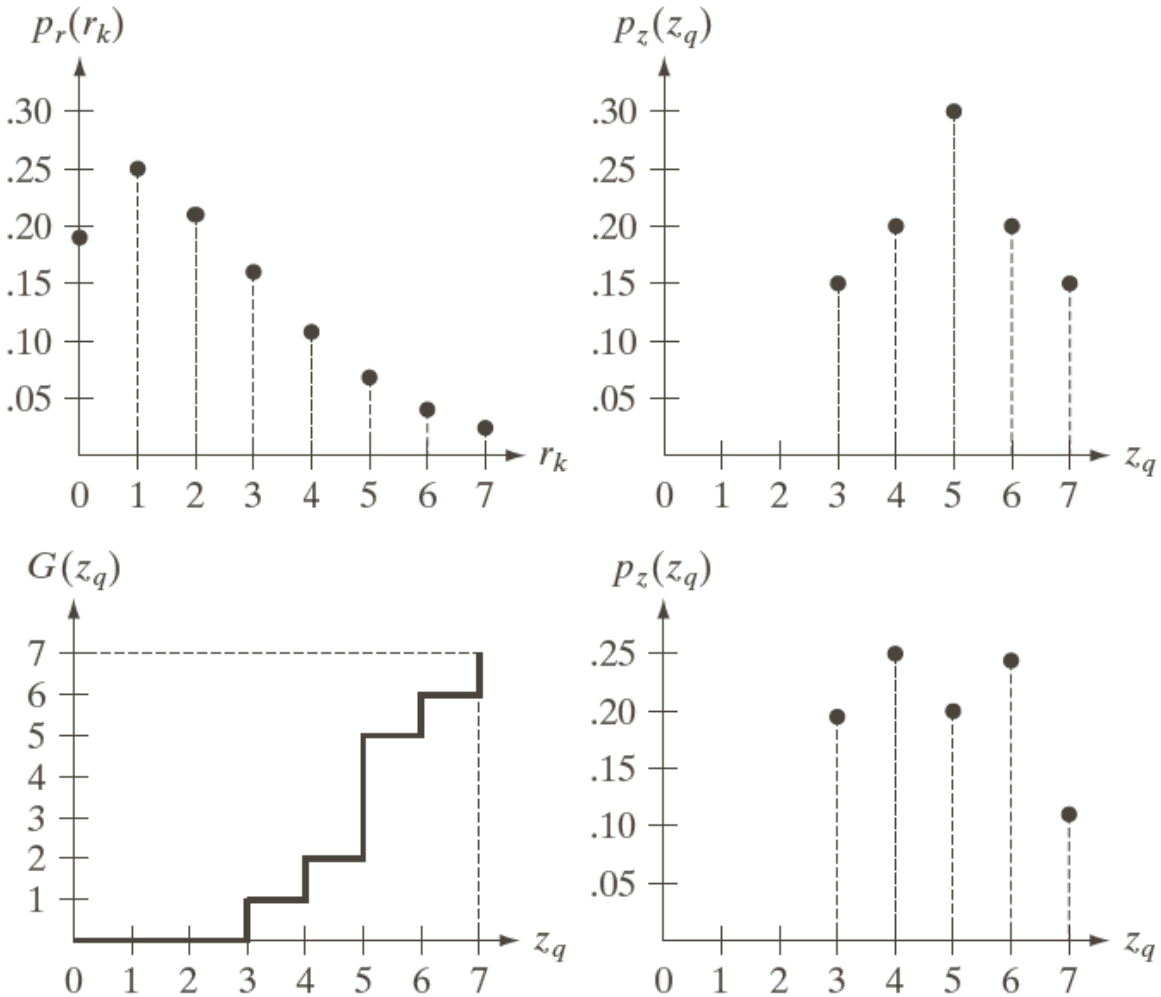
TABLE 3.2 Specified and actual histograms (the values in the third column are from the computations performed in the body of Example 3.8).

$s_0 = 1.33 \rightarrow 1$	$s_4 = 6.23 \rightarrow 6$
$s_1 = 3.08 \rightarrow 3$	$s_5 = 6.65 \rightarrow 7$
$s_2 = 4.55 \rightarrow 5$	$s_6 = 6.86 \rightarrow 7$
$s_3 = 5.67 \rightarrow 6$	$s_7 = 7.00 \rightarrow 7$

s_k	$\rightarrow$	z_q
1	$\rightarrow$	3
3	$\rightarrow$	4
5	$\rightarrow$	5
6	$\rightarrow$	6
7	$\rightarrow$	7

TABLE 3.4
Mappings of all the values of s_k into corresponding values of z_q.

© 1992–2008 R. C. Gonzalez & R. E. Woods



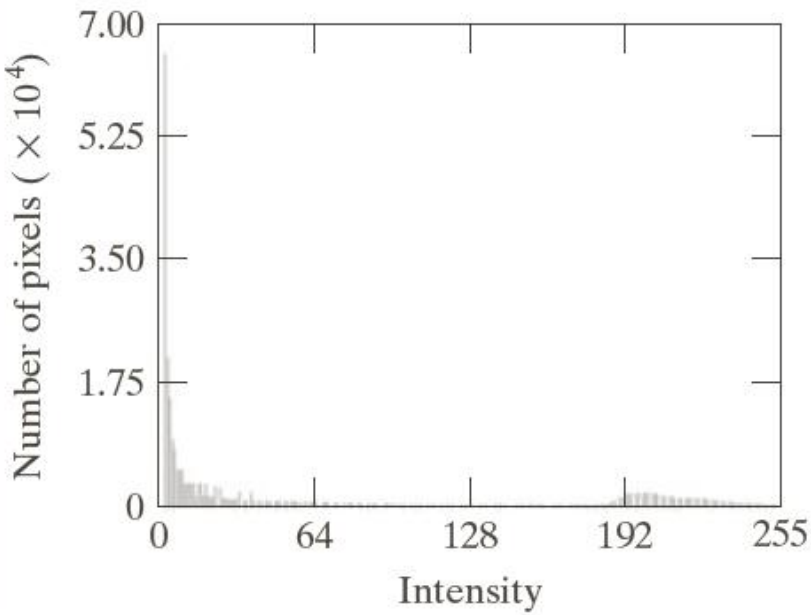
a b
c d

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).

© 1992–2008 R. C. Gonzalez & R. E. Woods

Chapter 3
Intensity Transformations & Spatial Filtering

Example: Comparison between histogram equalization and histogram matching

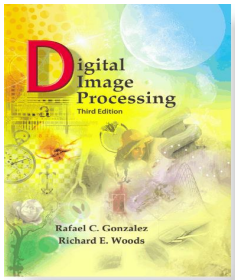


a b

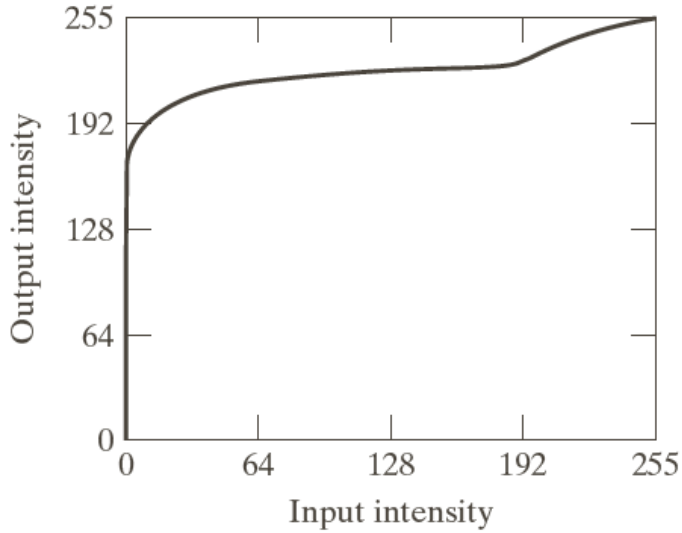
FIGURE 3.23 (a) Image of the Mars moon Phobos taken by NASA's *Mars Global Surveyor.* (b) Histogram. (Original image courtesy of NASA.)

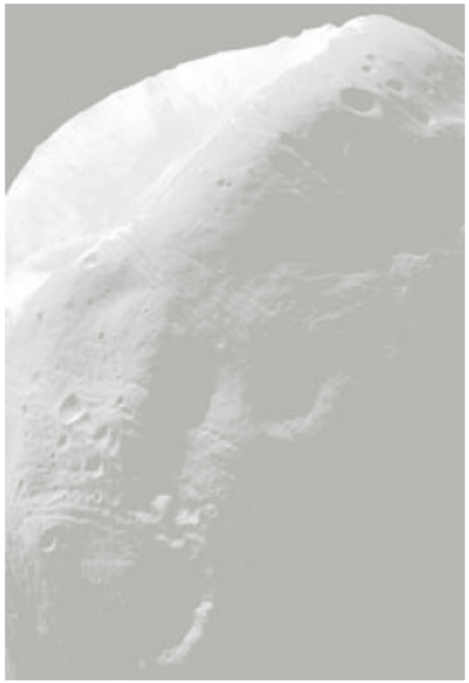
Image is dominated by large, dark areas

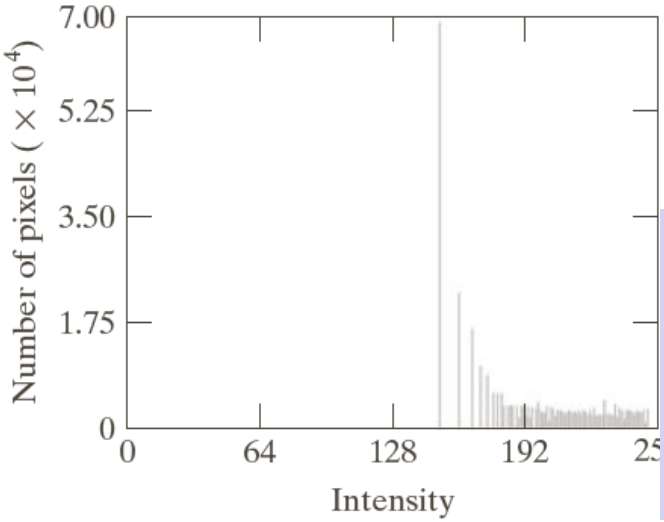
© 1992–2008 R. C. Gonzalez & R. E. Woods



Chapter 3
Intensity Transformations & Spatial Filtering







a b
c

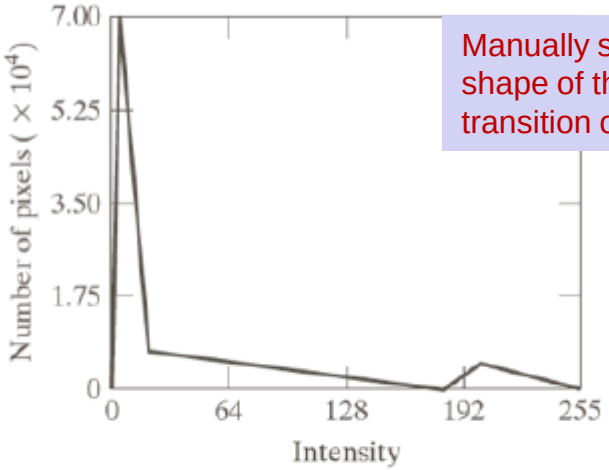
FIGURE 3.24 (a) Transformation function for histogram equalization. (b) Histogram-equalized image (note the washed-out appearance). (c) Histogram of (b).

Note that how fast Transformation function rises from Intensity level 0 to a level near 190.

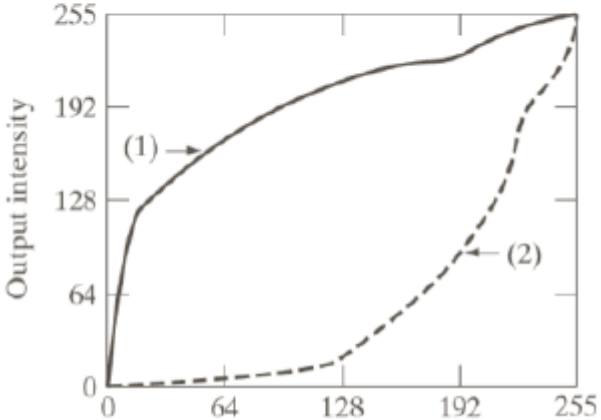
<u>Net effect:</u> *A very narrow interval of dark pixels mapped into the upper end of the gray scale of the output image (having light, washed-out appearance)*

© 1992 2008 R. C. Gonzalez & R. E. Woods

Chapter 3
Intensity Transformations & Spatial Filtering



Manually specified function that preserves the general shape of the original histogram, but has a smoother transition of levels in the dark region of the gray scale.



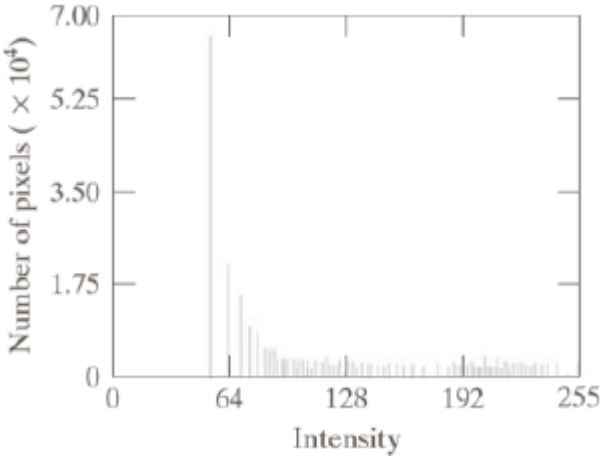


FIGURE 3.25 (a) Specified histogram. (b) Transformations. (c) Enhanced image using mappings from curve (2). (d) Histogram of (c).

Note: *Histogram specification is generally a trial-and-error process*

© 1992–2008 R. C. Gonzalez & R. E. Woods

Local Histogram Processing

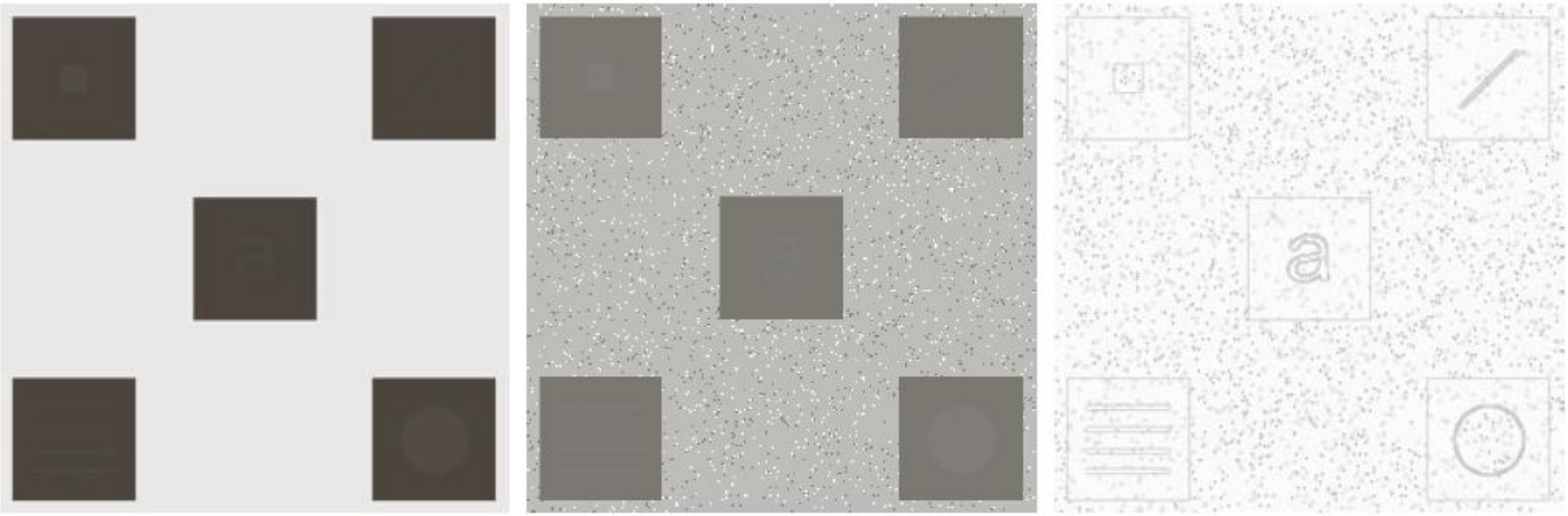
- At times, it is necessary to enhance details over small areas in an image.
  - The number of pixels in these areas may have negligible influence on the computation of a global transformation whose shape does not necessarily guarantee the desired local enhancement.
- Procedure:
  - Define a neighborhood and move its center from pixel to pixel.
  - At each location, the histogram of the points in the neighborhood is computed and accordingly, ,either a histogram equalization or histogram matching is performed.
  - The center of the neighborhood region is then moved to an adjacent pixel location and the procedure is repeated.

© 1992–2008 R. C. Gonzalez & R. E. Woods

Local Histogram Processing

As often is the case with histogram equalization of smooth, noisy regions, the image Fig. 3.26 (b) shows significant enhancement of the noise

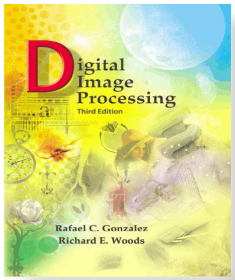
Here, we see the significant detail contained within the dark squares



a b c

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

As we discussed in Section 2.6.8, the nth moment of r about its mean is defined as

$$\mu_n(r) = \sum_{i=0}^{L-1} (r_i - m)^n p(r_i) \tag{3.3-17}$$

where m is the mean (average intensity) value of r (i.e., the average intensity of the pixels in the image):

$$m = \sum_{i=0}^{L-1} r_i p(r_i) \tag{3.3-18}$$

The second moment is particularly important:

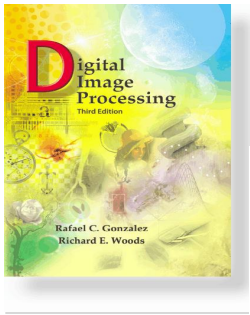
$$\mu_2(r) = \sum_{i=0}^{L-1} (r_i - m)^2 p(r_i) \tag{3.3-19}$$

Sample Mean and Sample Variance

$$m = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \qquad \sigma^2 = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[f(x, y) - m \right]^2$$

for $x = 0, 1, 2, \ldots, M - 1$ and $y = 0, 1, 2, \ldots, N - 1$.

© 1992–2008 R. C. Gonzalez & R. E. Woods



Using Histogram Statistics for Image Enhancement

Consider the following 2-bit image of size 5×5:

$$\begin{matrix} 0 & 0 & 1 & 1 & 2 \\ 1 & 2 & 3 & 0 & 1 \\ 3 & 3 & 2 & 2 & 0 \\ 2 & 3 & 1 & 0 & 0 \\ 1 & 1 & 3 & 2 & 2 \end{matrix}$$

The pixels are represented by 2 bits; therefore, $L = 4$ and the intensity levels are in the range $[0, 3]$. The total number of pixels is 25, so the histogram has the components

$$p(r_0) = \frac{6}{25} = 0.24; \quad p(r_1) = \frac{7}{25} = 0.28;$$

$$p(r_2) = \frac{7}{25} = 0.28; \quad p(r_3) = \frac{5}{25} = 0.20$$

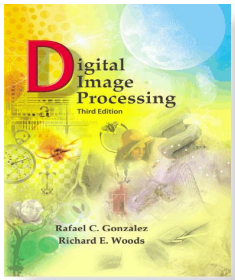
where the numerator in $p(r_i)$ is the number of pixels in the image with intensity level r_i. We can compute the average value of the intensities in the image using Eq. (3.3-18):

$$\begin{aligned} m &= \sum_{i=0}^{3} r_i p(r_i) \\ &= (0)(0.24) + (1)(0.28) + (2)(0.28) + (3)(0.20) \\ &= 1.44 \end{aligned}$$

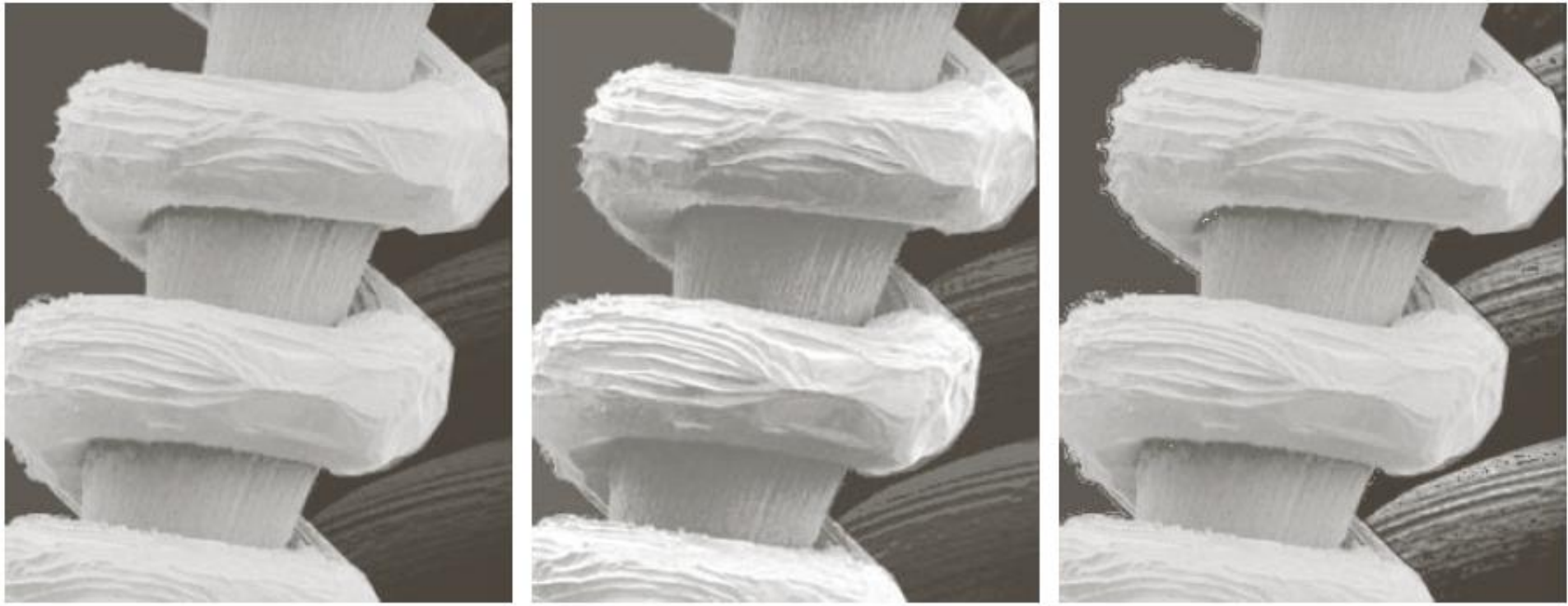
Sample Mean

$$\begin{aligned} m &= \frac{1}{25} \sum_{x=0}^{4} \sum_{y=0}^{4} f(x, y) \\ &= 1.44 \end{aligned}$$

© 1992–2008 R. C. Gonzalez & R. E. Woods

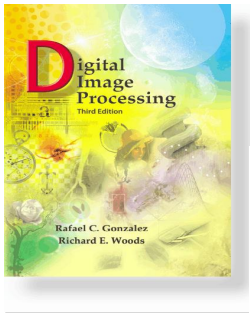


Using Histogram Statistics for Image Enhancement



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.)



Using Histogram Statistics for Image Enhancement

Let (x, y) denote the coordinates of any pixel in a given image, and let S_{xy} denote a neighborhood (subimage) of specified size, centered on (x, y). The mean value of the pixels in this neighborhood is given by the expression

$$m_{S_{xy}} = \sum_{i=0}^{L-1} r_i p_{S_{xy}}(r_i) \tag{3.3-22}$$

The variance of the pixels in the neighborhood similarly is given by

$$\sigma^2_{S_{xy}} = \sum_{i=0}^{L-1} (r_i - m_{S_{xy}})^2 p_{S_{xy}}(r_i) \tag{3.3-23}$$

Let $f(x, y)$ represen
value of an image at any image coordinates (x, y), and let $g(x, y)$ represent the corresponding enhanced value at those coordinates. Then,

$$g(x, y) = \begin{cases} E \cdot f(x, y) & \text{if } m_{S_{xy}} \le k_0 m_G \text{ AND } k_1\sigma_G \le \sigma_{S_{xy}} \le k_2\sigma_G \\ f(x, y) & \text{otherwise} \end{cases} \tag{3.3-24}$$

for $x = 0, 1, 2, \ldots, M - 1$ and $y = 0, 1, 2, \ldots, N - 1$, where, as indicated above, E, k_0, k_1, and k_2 are specified parameters, m_G is the global mean of the input image, and σ_G is its standard deviation. Parameters $m_{S_{xy}}$ and $\sigma_{S_{xy}}$ are the local mean and standard deviation, respectively. As usual, M and N are the row and column image dimensions.

© 1992–

Spatial Filtering: Fundamentals

Image $f(x,y) \rightarrow$ Image $g(x,y)$

Spatial filters (also called spatial *masks*, *kernels*, *templates* or *windows*)

3.4.1 The Mechanics of Spatial Filtering

Filtering creates a new pixel with coordinates equal to the coordinates of the center of the neighbourhood, with value = result of the filtering operation.

Example : linear spatial filter using a 3x3 neighbourhood:

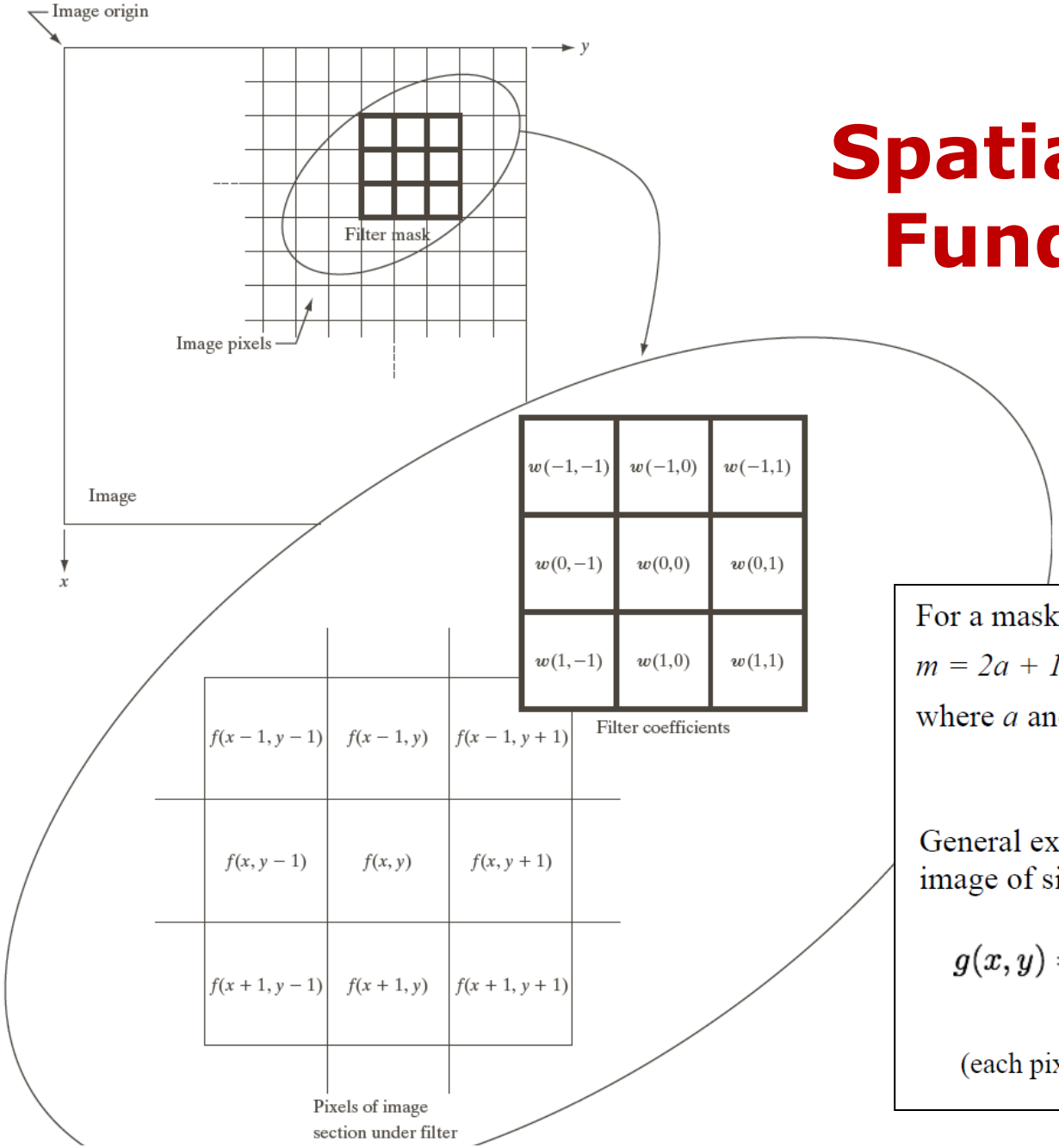
At any point (x,y) in the original image, response $g(x,y)$ of the filter:

$$g(x,y) = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + \ldots$$
$$+w(0,0)f(x,y) + \cdots + w(1,1)f(x+1,y+1)$$

Centre coefficient of filter

© 1992–2008 R. C. Gonzalez & R. E. Woods

Spatial Filtering: Fundamentals



For a mask of size *m*x*n*, we assume:

$m = 2a + 1$ and $n = 2b + 1$

where a and b are positive integers (*odd size*)

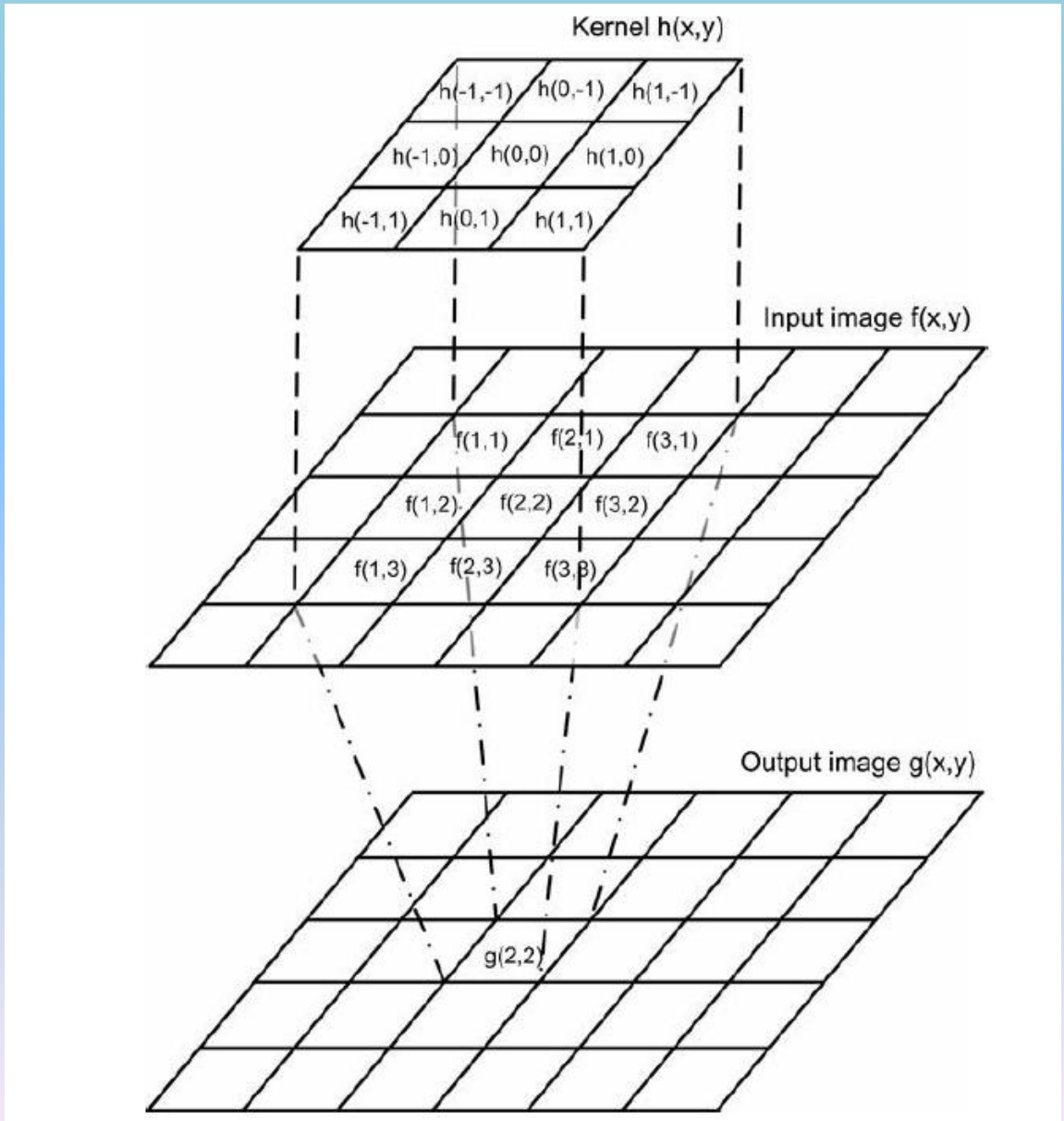
General expression of a linear spatial filtering of an image of size M x N with a filter of size m x n :

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

(each pixel in w visiting every pixel in 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.

Spatial Filtering: Fundamentals

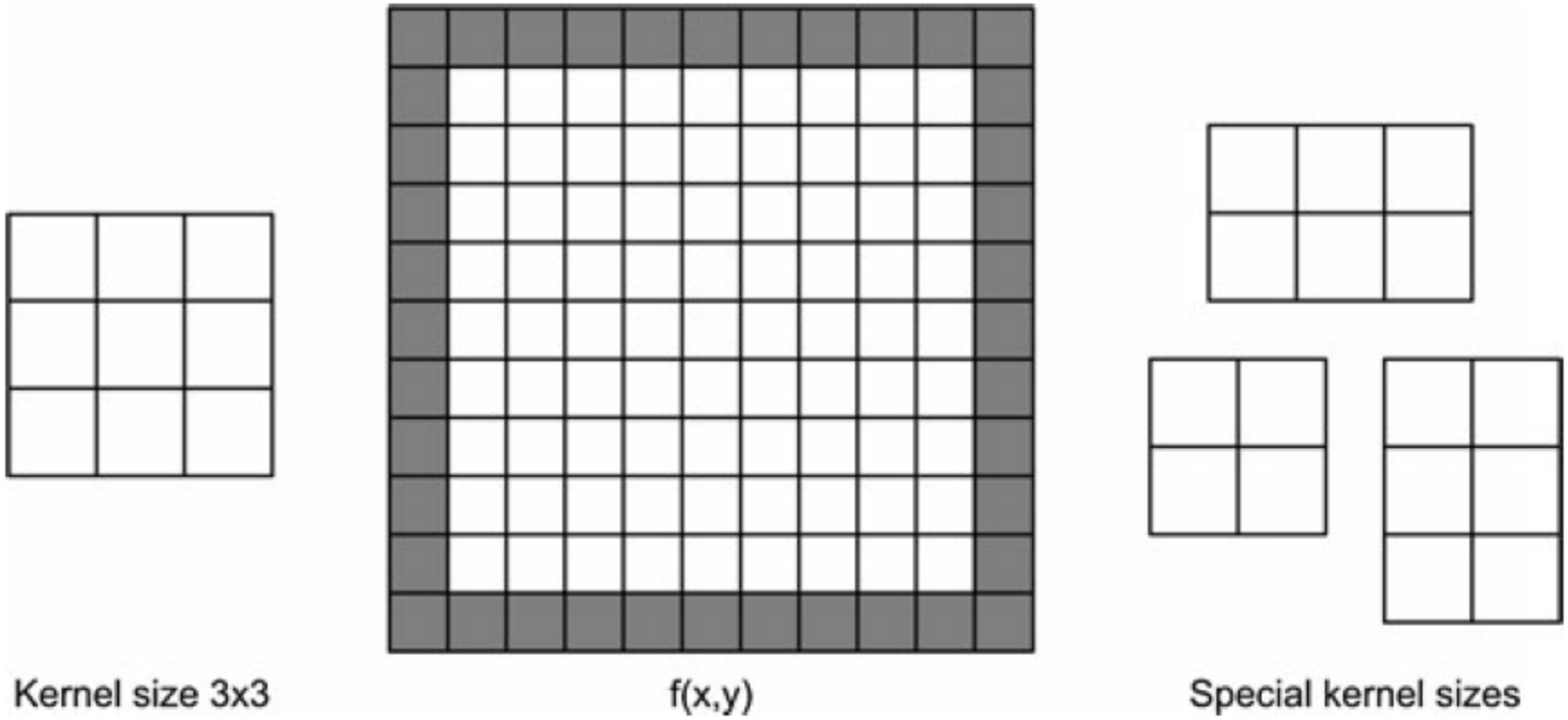


w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

FIGURE 3.31 Another representation of a general 3×3 filter mask.

Credits: S. Murala, EEL484 Lecture notes/slides, IIT Ropar

Spatial Filtering: Fundamentals



Credits: S. Murala, EEL484 Lecture notes/slides, IIT Ropar

Spatial Filtering: Fundamentals

1	1	1
1	1	1
1	1	1

3x3 Mean kernel

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

5x5 Gaussian kernel

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

3x3 Sobel kernel

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

1	1	1
1	1	1
1	1	1

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

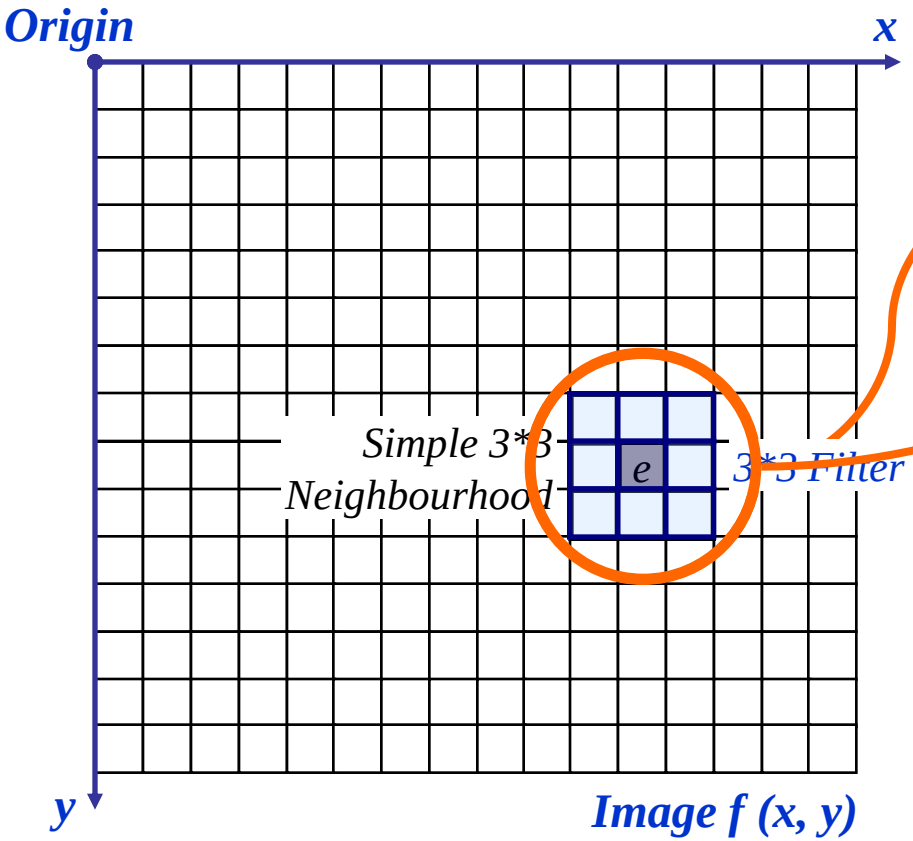
1	2	1
2	4	2
1	2	1

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.

Credits: S. Murala, EEL484 Lecture notes/slides, IIT Ropar

Spatial Filtering Refresher



a	b	c
d	e	f
g	h	i

Original Image Pixels

*

r	s	t
u	v	w
x	y	z

Filter

$$e_{processed} = v*e + r*a + s*b + t*c + u*d + w*f + x*g + y*h + z*i$$

The above is repeated for every pixel in the original image to generate the smoothed image

Credits: Brian Mac, Dublin Institute of Technology

Spatial Filtering: Fundamentals

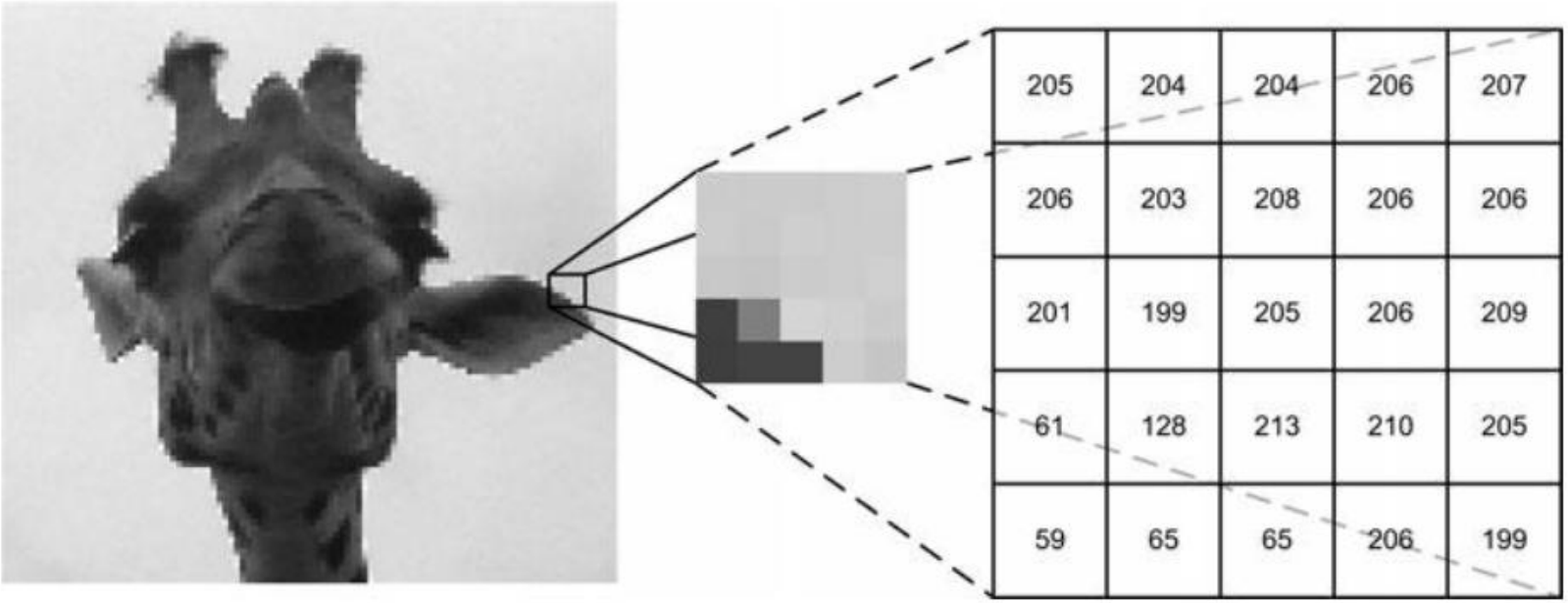
Fig. An example of how a mean filter can be used to hide the identity of a person.

Credits: S. Murala, EEL484 Lecture notes/slides, IIT Ropar

Spatial Filtering: Fundamentals

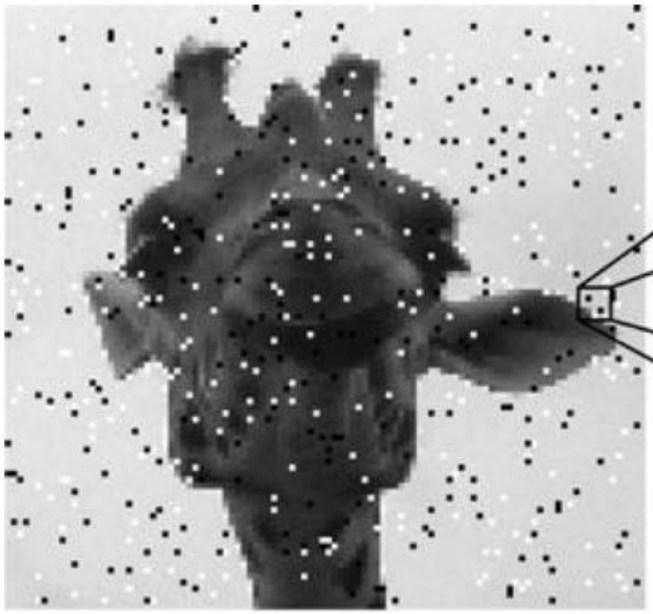
The Median Filter

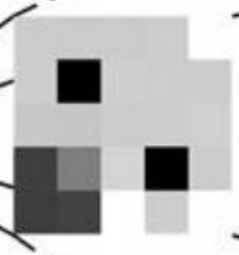
Application: Image Denoising



Credits: S. Murala, EEL484 Lecture notes/slides, IIT Ropar

Spatial Filtering: Fundamentals





205	204	204	206	255
206	0	208	206	206
201	199	205	206	209
61	128	213	0	205
59	65	255	206	255

$$\text{Ordering}: [0, 199, 201, 204, \underline{204}, 205, 205, 206, 208]$$

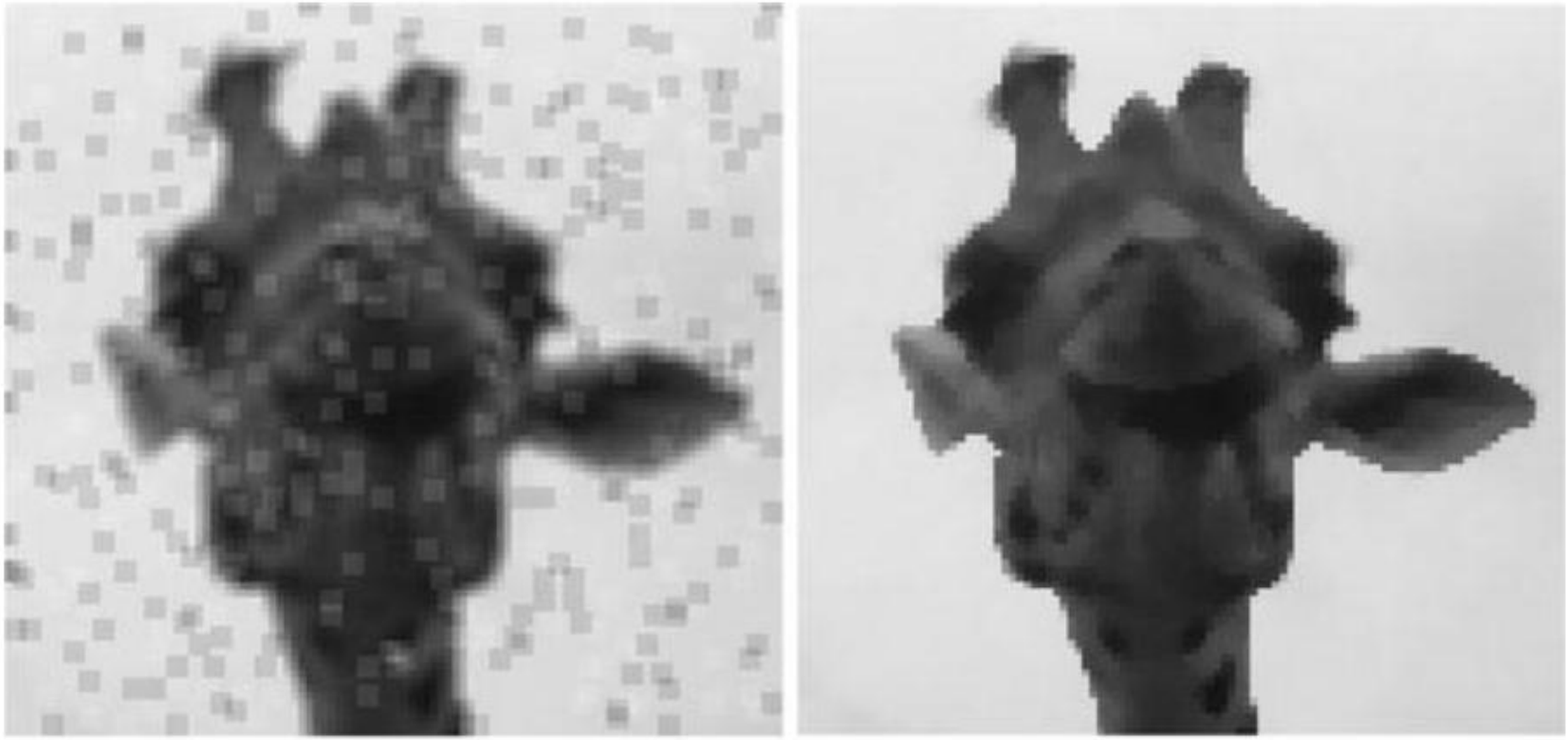
$$\text{Median} = 204$$

$$\text{Mean value} = \frac{205 + 204 + 204 + 206 + 0 + 208 + 201 + 199 + 205}{9}$$

$$= 181.3 \simeq 181$$

Credits: S. Murala, EEL484 Lecture notes/slides, IIT Ropar

Spatial Filtering: Fundamentals



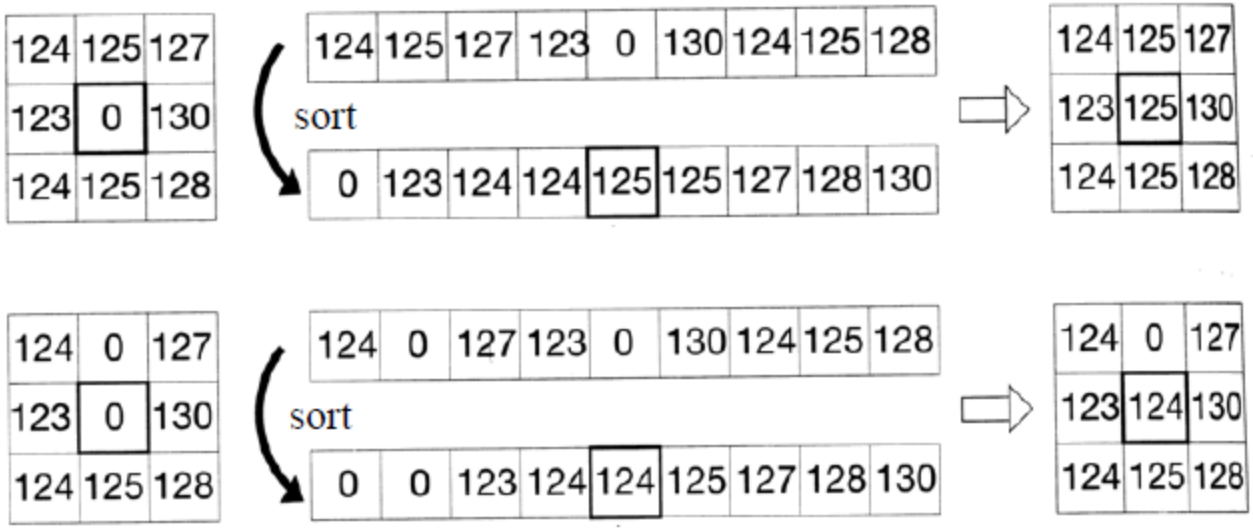
Mean filtered

Median filtered

Credits: S. Murala, EEL484 Lecture notes/slides, IIT Ropar

Chapter 3
Intensity Transformations & Spatial Filtering

Median filter :

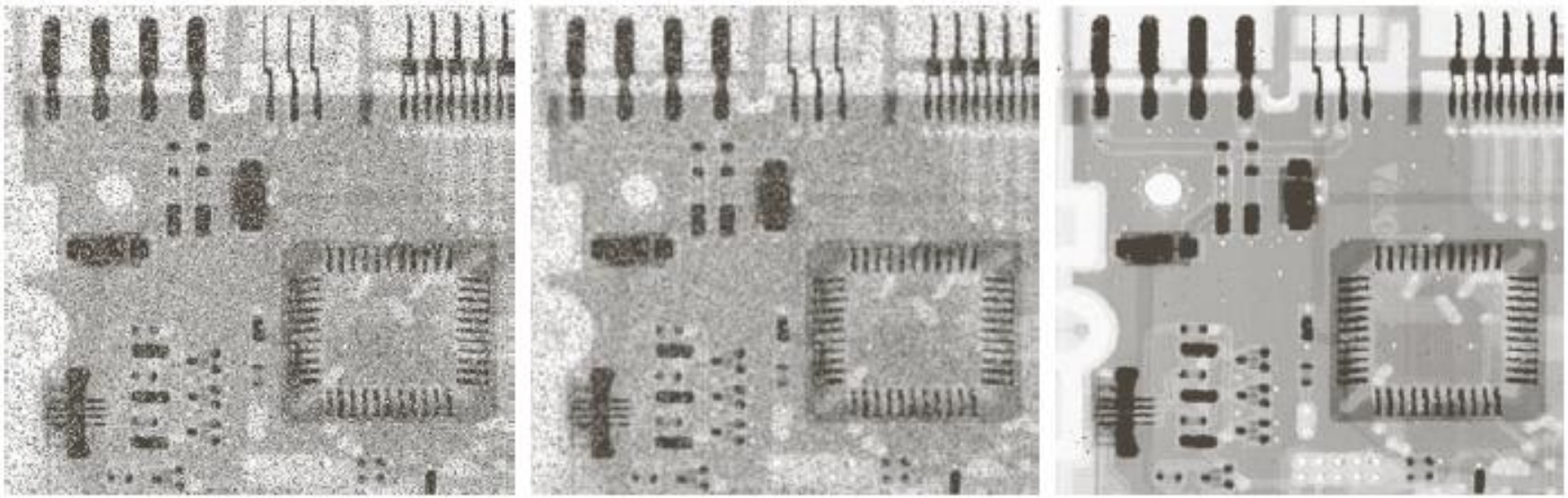


median value

[Bres, 2003]

© 1992–2008 R. C. Gonzalez & R. E. Woods

Chapter 3
Intensity Transformations & Spatial Filtering



a b c

FIGURE 3.35 (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.)

© 1992–2008 R. C. Gonzalez & R. E. Woods

Spatial Filtering: Fundamentals

Correlation

Difference between Correlation and Convolution

$$g(x, y) = \sum_{j=-R}^{R} \sum_{i=-R}^{R} h(i, j) \cdot f(x + i, y + j)$$

Correlation = process of moving a filter mask over the image and computing the sum of products at each location

Convolution = same mechanics but with a filter rotated by 180°

Credits: S. Murala, EEL484 Lecture notes/slides, IIT Ropar

Chapter 3
Intensity Transformations & Spatial Filtering

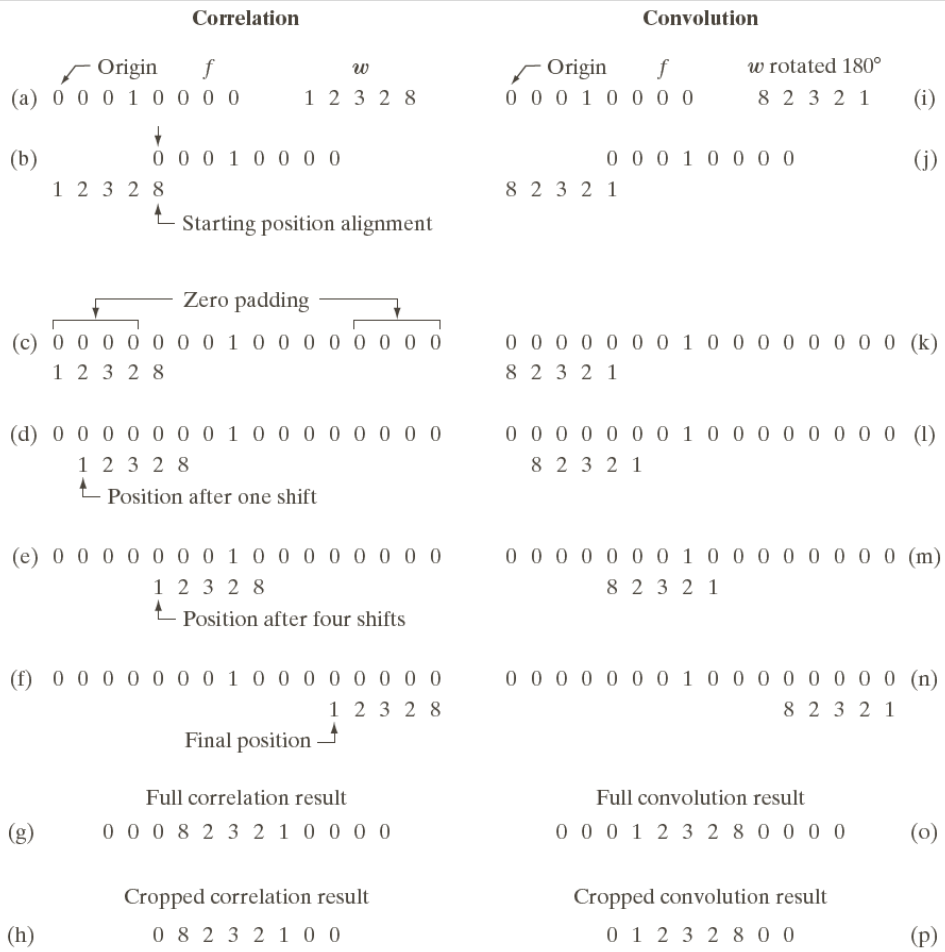


FIGURE 3.29 Illustration of 1-D correlation and convolution of a filter with a discrete unit impulse. Note that correlation and convolution are functions of *displacement*.

© 1992–2008 R. C. Gonzalez & R. E. Woods

3.4.2 Spatial Correlation and Convolution

- Correlation is a function of *displacement* of the filter
- If *f* contains all 0s and only one 1 (*discrete unit impulse*) => copy of *w* rotated by 180°
- i.e. correlation of a function with a discrete unit impulse => rotated version of the function at the location of the impulse
- Convolving a function with a unit impulse => copy of *w* at the location of the impulse
- Convolution: rotate one function by 180° and perform same operations as in correlation

© 1992–2008 R. C. Gonzalez & R. E. Woods

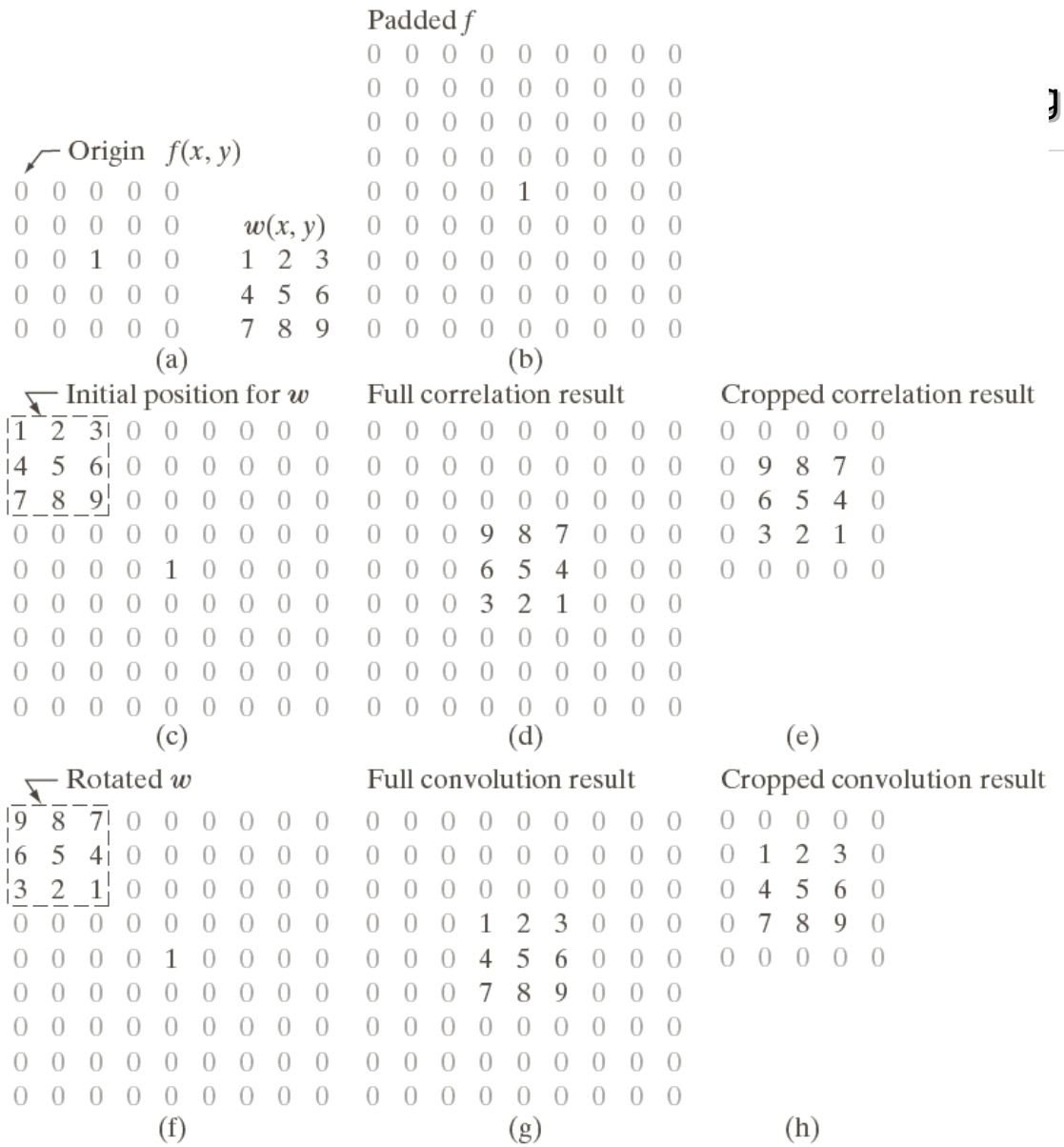


FIGURE 3.30 Correlation (middle row) and convolution (last row) of a 2-D filter with a 2-D discrete, unit impulse. The 0s are shown in gray to simplify visual analysis.

© 1992–2008 R. C. Gonzalez & R. E. Woods

3.4.2 Spatial Correlation and Convolution

- Correlation of a filter *w(x,y)* with an image *f(x,y)*:

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

For all values of x and y so that all elements of *w* visit every pixel in *f* (*f padded* appropriately)

- Convolution of *w(x,y)* and *f(x,y)*:

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

flipping

For all values of x and y so that all elements of *w* visit every pixel in *f* (*f* padded appropriately)

© 1992–2008 R. C. Gonzalez & R. E. Woods

3.4.3 Vector Representation of Linear Filtering

It is convenient sometimes to write the sum of product as:

$$R = w_1 z_1 + w_2 z_2 + \ldots w_{mn} z_{mn}$$

$$= \sum_{k=1}^{mn} w_k z_k \quad = \boldsymbol{w}^T \boldsymbol{z}$$

Where:

- w_k : coefficients of an mxn filter
- z_k : corresponding image intensities encompassed by the filter

Example:

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

FIGURE 3.31 Another representation of a general 3×3 filter mask.

© 1992–2008 R. C. Gonzalez & R. E. Woods

3.4.4 Generating Spatial Filter Masks

To generate a *mxn linear* filter, need to specify *mn* mask coefficients, selected based on what the filter is supposed to do

Example 1 : average filter with 3x3 neighbourhood: $R = \frac{1}{9}\sum_{i=1}^{9} z_i \quad \Rightarrow \quad w_i = \frac{1}{9}$

Example 2 : spatial filter mask based on a continuous function of two variables e.g. Gaussian Function:

$$h(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

3x3 mask:

$$\begin{aligned} w_1 &= h(-1,-1) \\ w_2 &= h(-1,0) \\ &\vdots \\ w_9 &= h(1,1) \end{aligned}$$

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

1	1	1
1	1	1
1	1	1

Averaging filter

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

1	2	1
2	4	2
1	2	1

Weighted average

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.

© 1992–2008 R. C. Gonzalez & R. E. Woods

3.5 Smoothing Spatial Filters

3.5.1 Smoothing Linear Filters

Output = average of pixels contained in the neighbourhood of the filter mask

Averaging filters or *lowpass filters*

⇒ Reduce "sharp" transitions in intensities

⇒ Application 1: noise reduction

⇒ Application 2: smoothing of false contours

FC caused generally by use of an insufficient number of intensity levels)

⇒ Blurring effect

Edges are characterized by sharp intensity transitions, so averaging filters have this undesirable side effect that they blur images

© 1992–2008 R. C. Gonzalez & R. E. Woods

Smoothing Spatial Filters

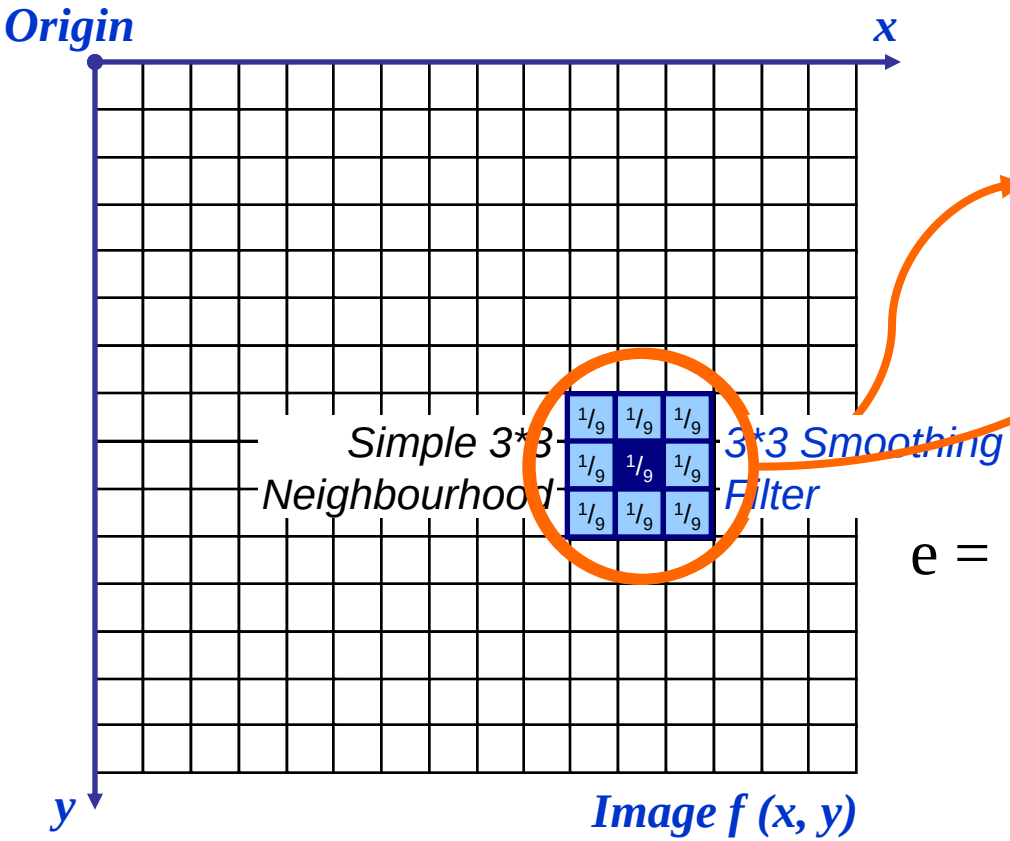
One of the simplest spatial filtering operations we can perform is a smoothing operation

- Simply average all of the pixels in a neighbourhood around a central value
- Especially useful in removing noise from images
- Also useful for highlighting gross detail

$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$

Simple averaging filter

Smoothing Spatial Filtering



104	100	108
99	106	98
95	90	85

Original Image Pixels

*

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

Filter

$$
\begin{aligned}
e = \; & \tfrac{1}{9}*106 + \\
& \tfrac{1}{9}*104 + \tfrac{1}{9}*100 + \tfrac{1}{9}*108 + \\
& \tfrac{1}{9}*99 + \tfrac{1}{9}*98 + \\
& \tfrac{1}{9}*95 + \tfrac{1}{9}*90 + \tfrac{1}{9}*85 \\
= \; & 98.3333
\end{aligned}
$$

The above is repeated for every pixel in the original image to generate the smoothed image

Credits: Brian Mac, Dublin Institute of Technology

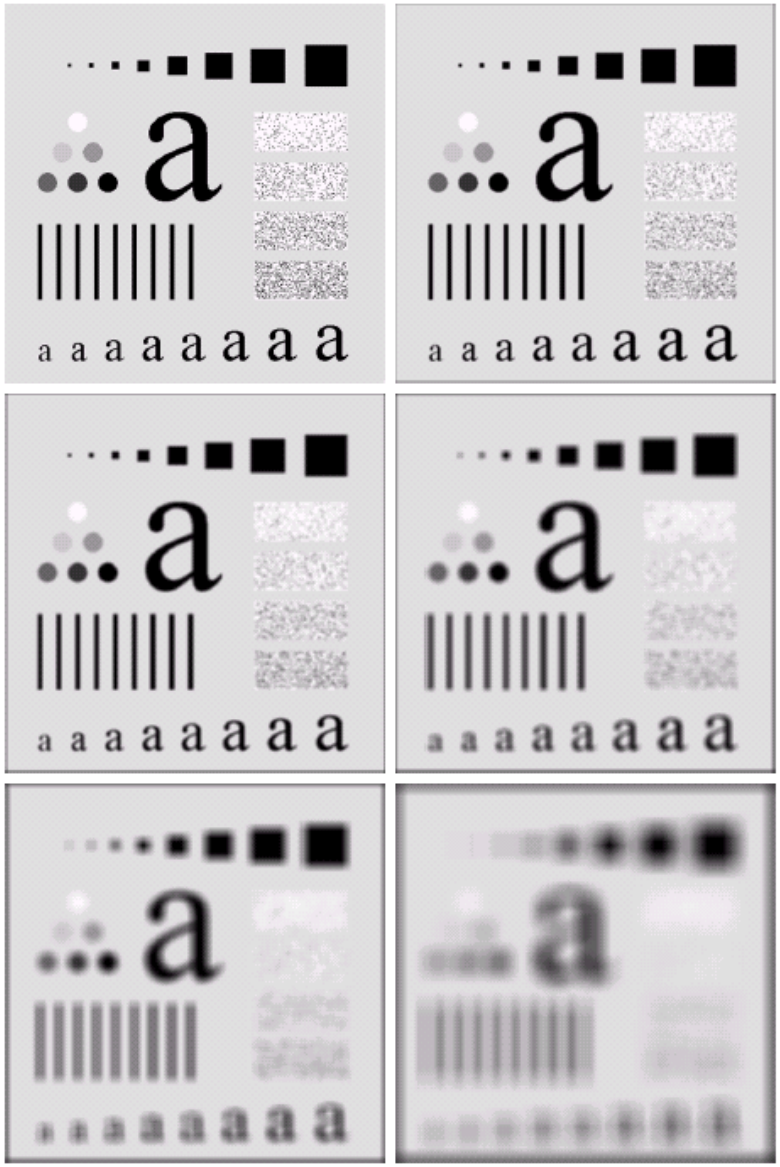
Image Smoothing Example

The image at the top left is an original image of size 500*500 pixels

The subsequent images show the image after filtering with an averaging filter of increasing sizes

- 3, 5, 9, 15 and 35

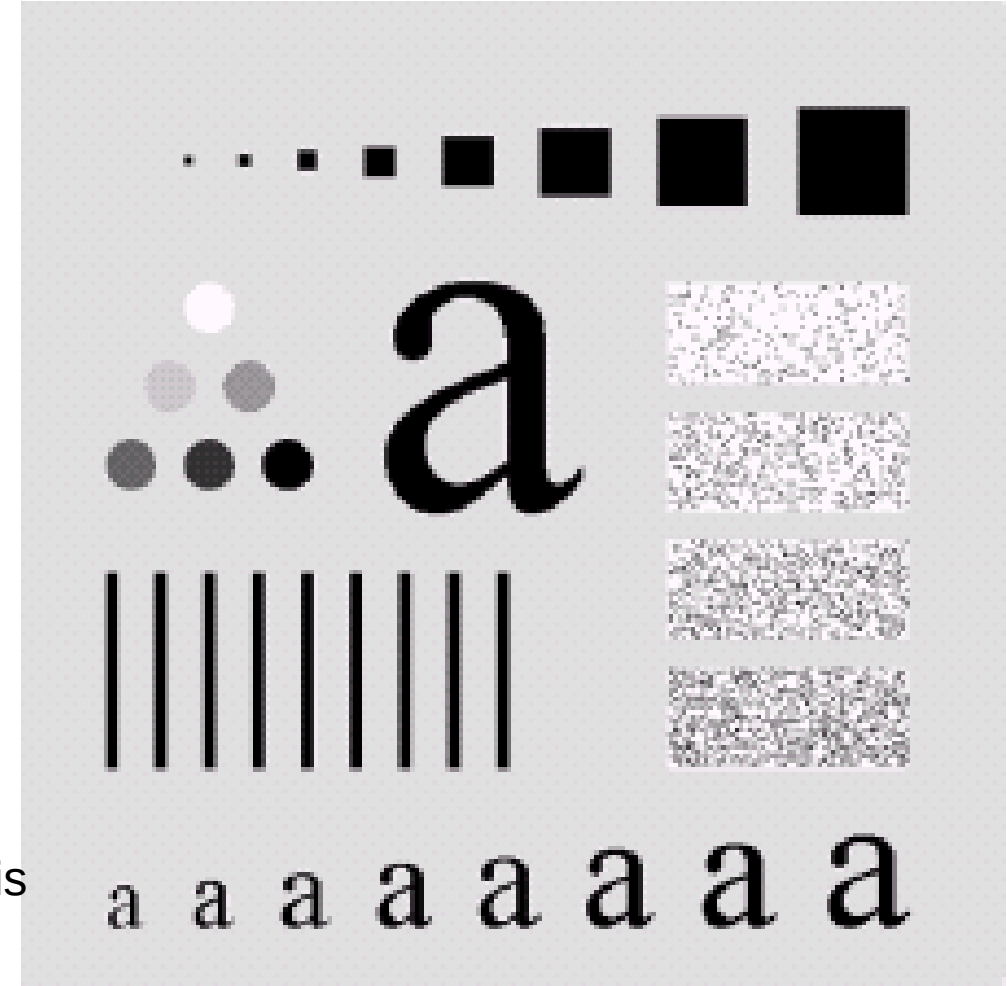
Notice how detail begins to disappear



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Image Smoothing Example

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

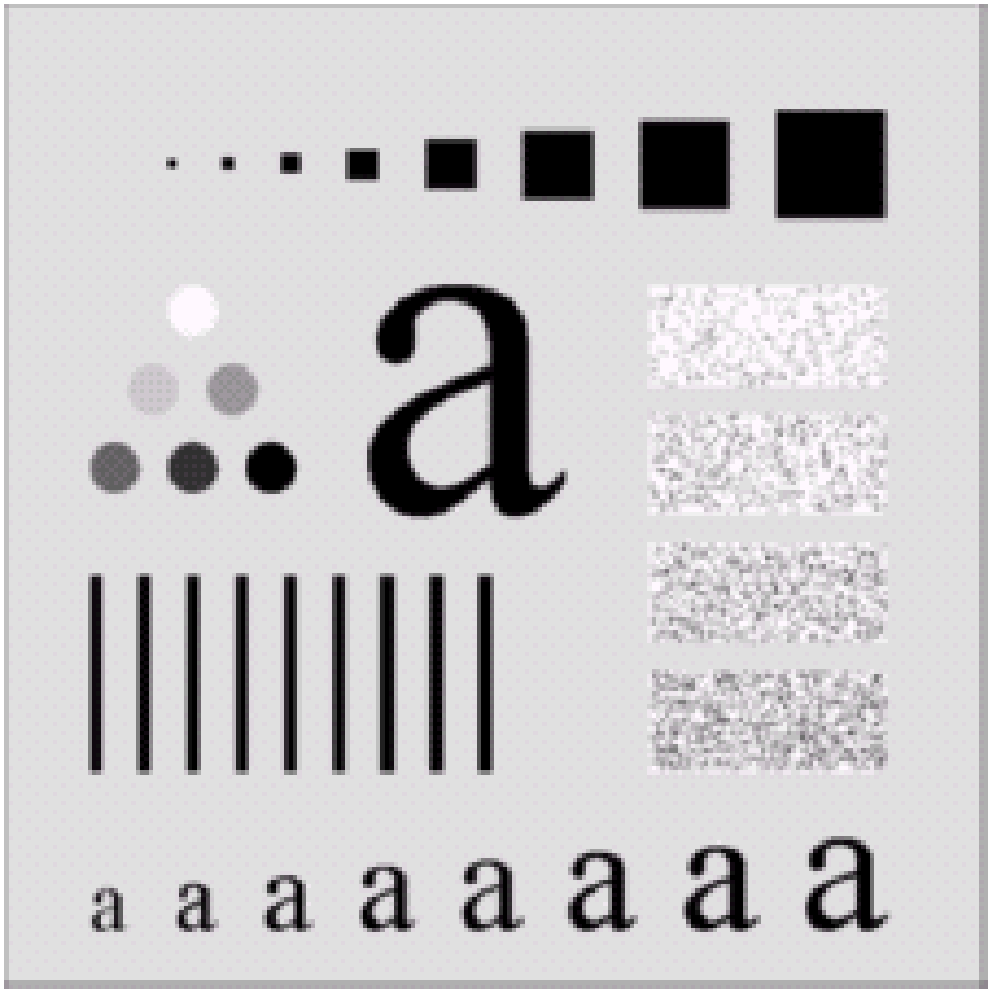


Original Image

Note that the vertical bars are 5 pixels wide, 100 pixels high, and their separation is 20 pixels.

Image Smoothing Example

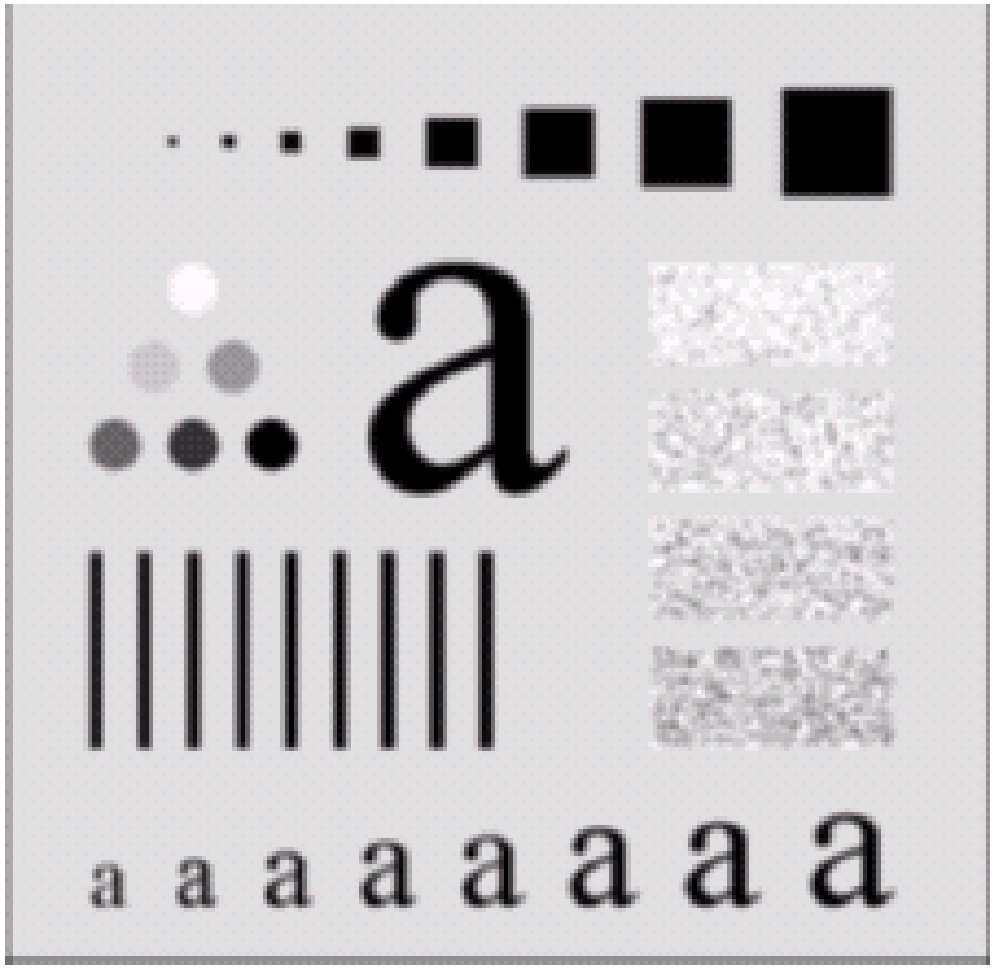
Note:
•Significant blurring for small objects.
•Noise is less pronounced and the jagged borders of the characters are pleasingly smoothed.



Smoothened using averaging mask of size 3x3

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Image Smoothing Example

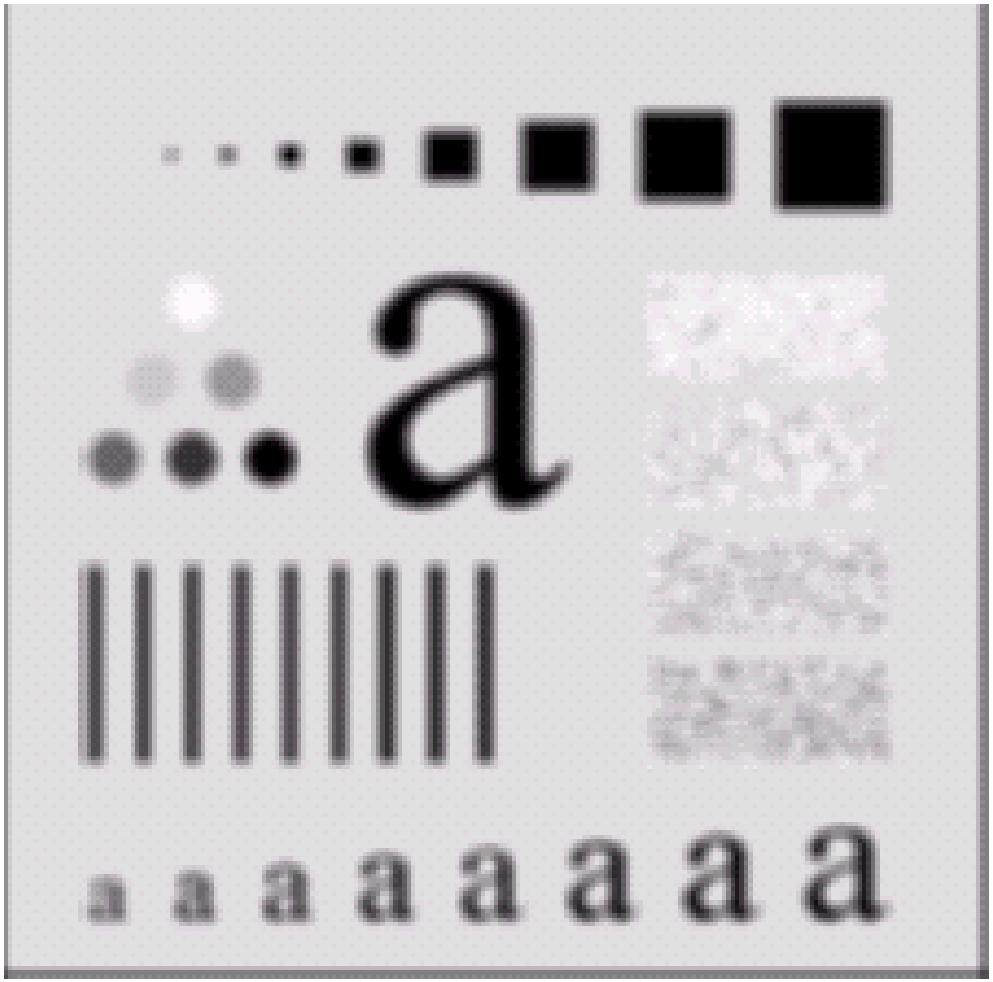


Smoothened using averaging mask of size 5x5

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Image Smoothing Example

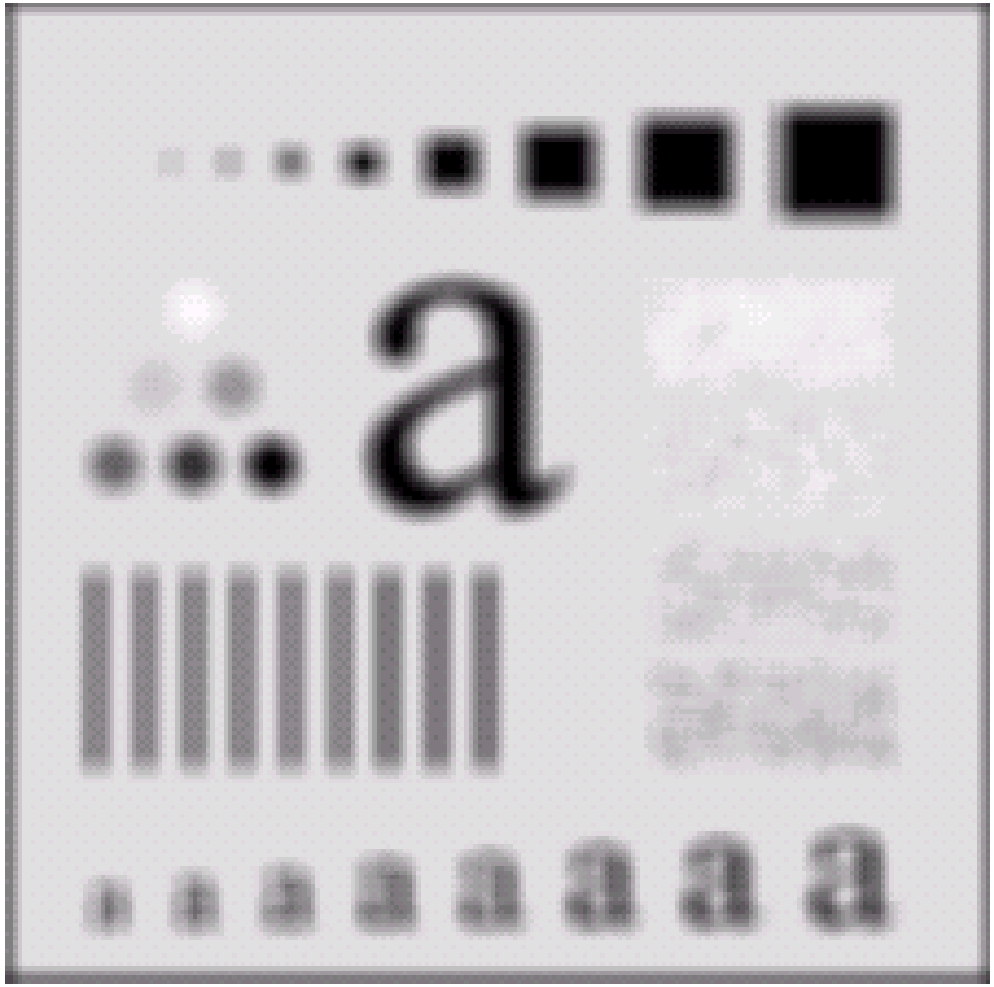
Note:20% black circle is not nearly as distinct from the background as in the previous 3 images



Smoothened using averaging mask of size 9x9

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Image Smoothing Example



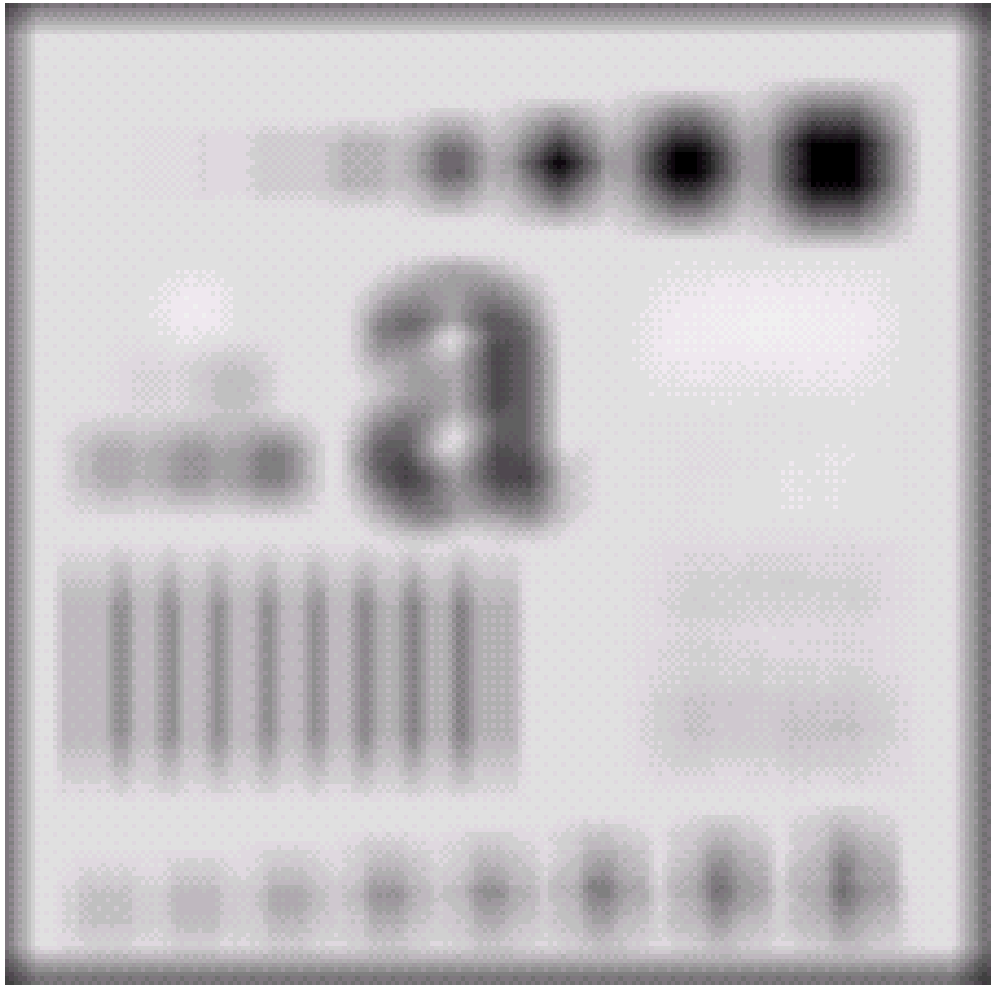
Smoothened using averaging mask of size 15x15

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Image Smoothing Example

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Note:
- Small objects almost eliminated using this aggressive blurring.
- Larger objects become bloblike and easy to detect.



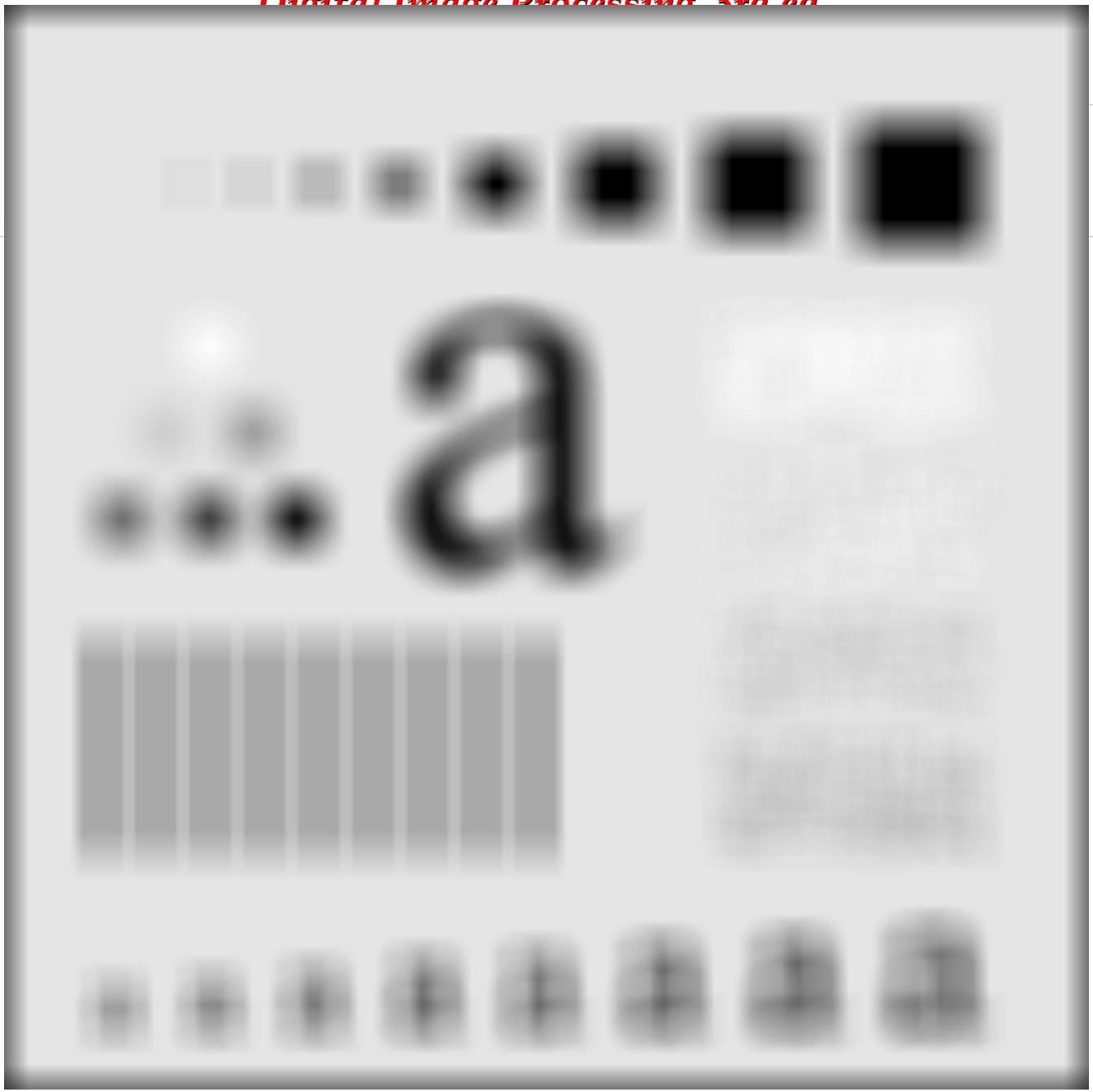
Smoothened using averaging mask of size 35x35

★3.21 The three images shown were blurred using square averaging masks of sizes $n = 23$, 25, and 45, respectively. The vertical bars on the left lower part of (a) and (c) are blurred, but a clear separation exists between them. However, the bars have merged in image (b), in spite of the fact that the mask that produced this image is significantly smaller than the mask that produced image (c). Explain the reason for this.

Remember that the vertical bars are 5 pixels wide, 100 pixels high, and their separation is 20 pixels.

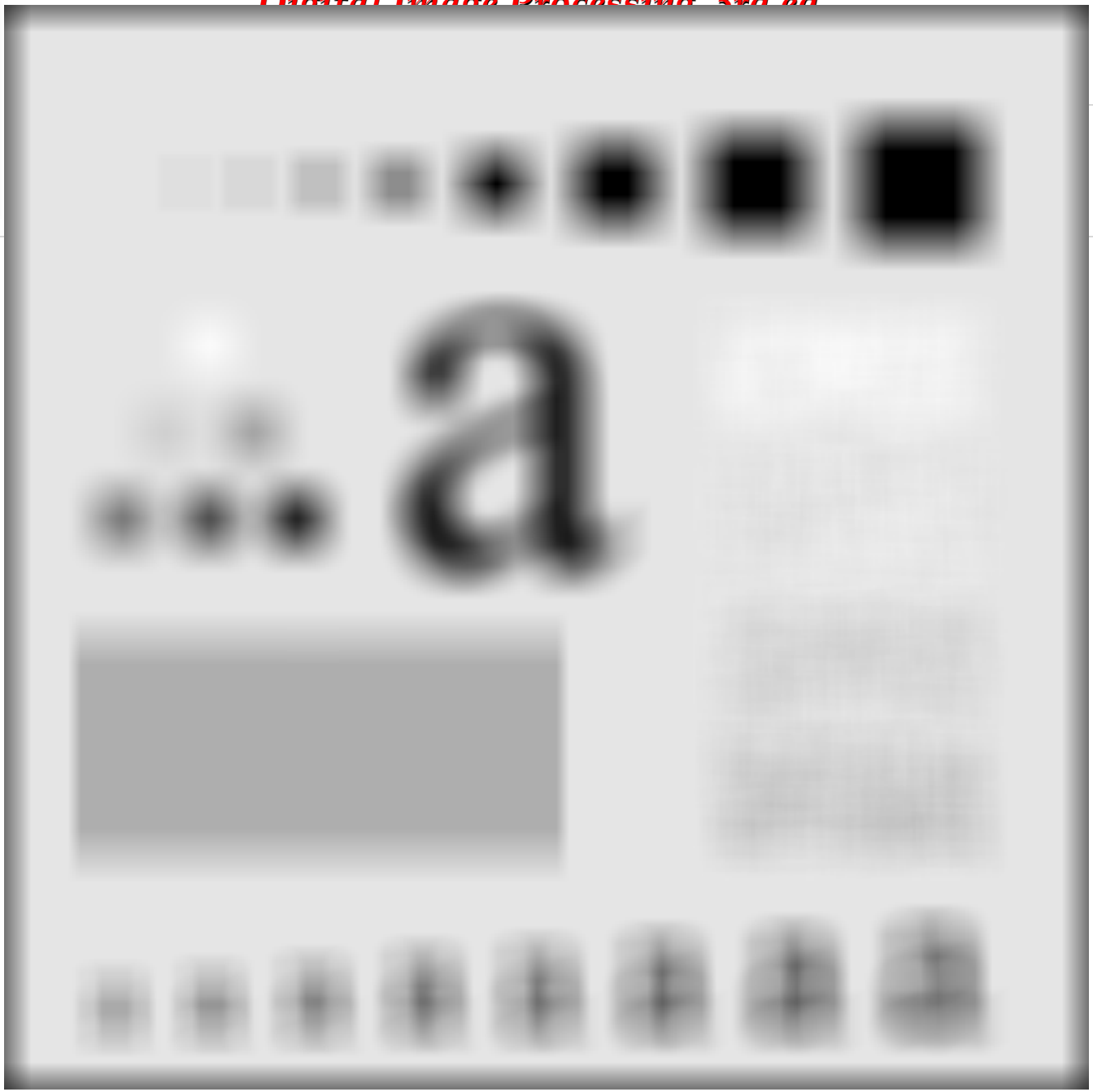
Original Image

© 1992–2008 R. C. Gonzalez & R. E. Woods



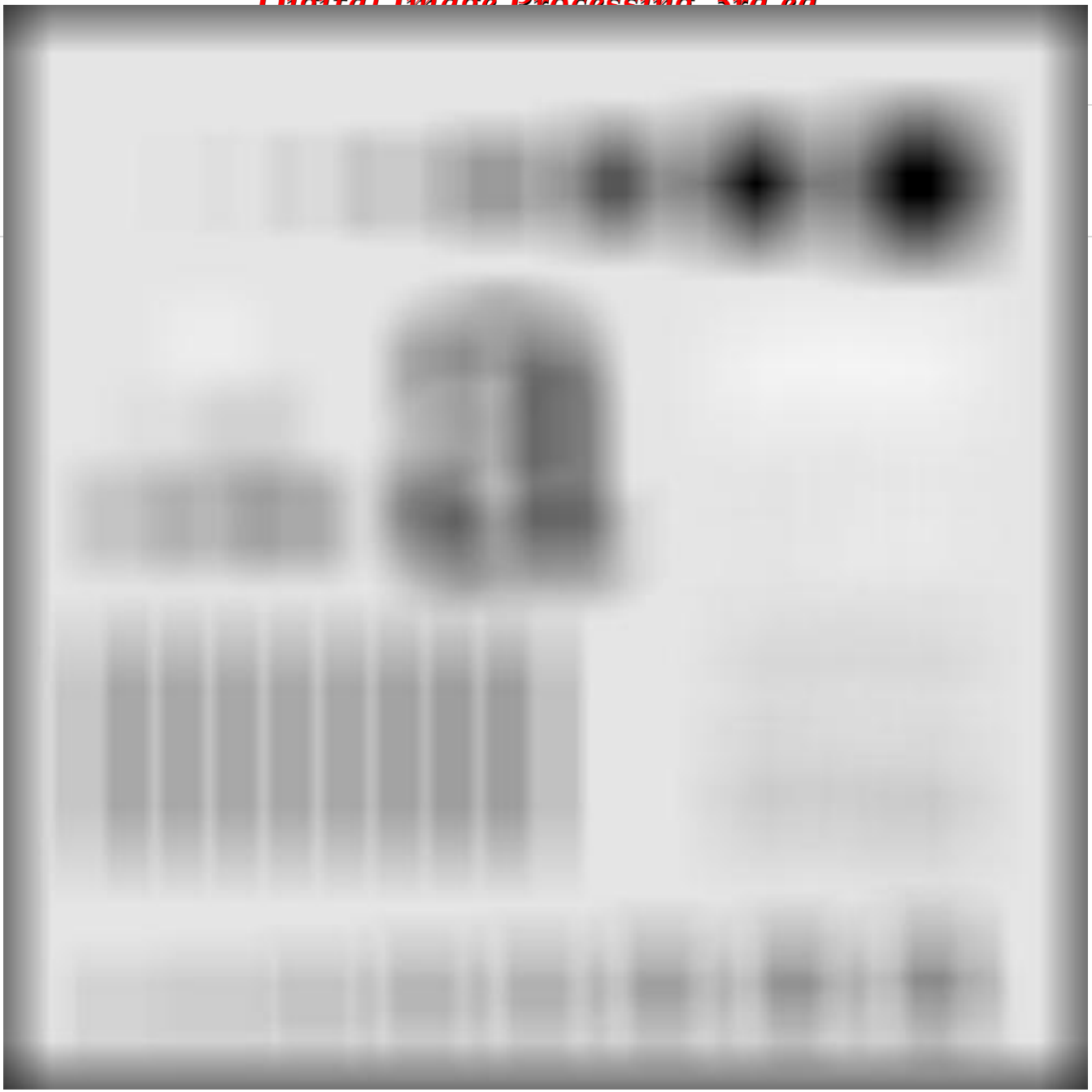
Smoothened using averaging mask of size 23x23

© 1992–2008 R. C. Gonzalez & R. E. Woods



Smoothened using averaging mask of size 25x25

© 1992–2008 R. C. Gonzalez & R. E. Woods



Smoothened using averaging mask of size 45x45

© 1992–2008 R. C. Gonzalez & R. E. Woods

Weighted Smoothing Filters

More effective smoothing filters can be generated by allowing different pixels in the neighbourhood different weights in the averaging function

- Pixels closer to the central pixel are more important
- Often referred to as a *weighted averaging*

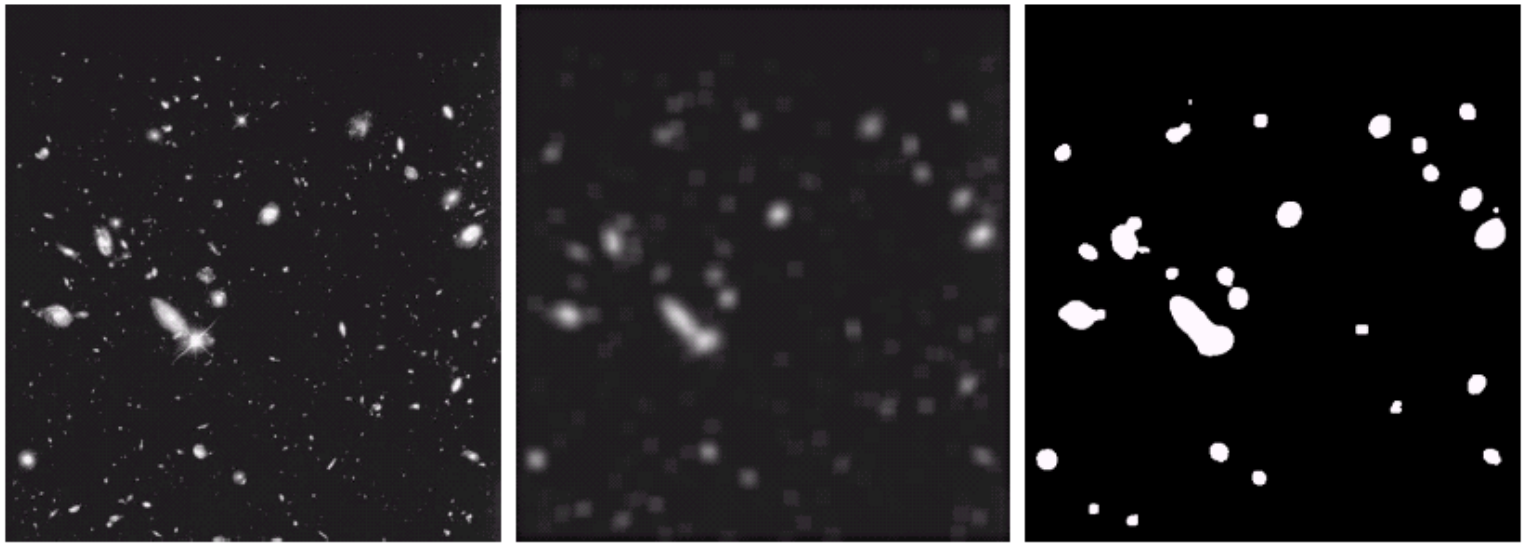
$^1/_{16}$	$^2/_{16}$	$^1/_{16}$
$^2/_{16}$	$^4/_{16}$	$^2/_{16}$
$^1/_{16}$	$^2/_{16}$	$^1/_{16}$

Weighted Averaging filter

Credits: Brian Mac, Dublin Institute of Technology

Another Smoothing Example

By smoothing the original image, we get rid of lots of the finer detail which leaves only the gross features for thresholding

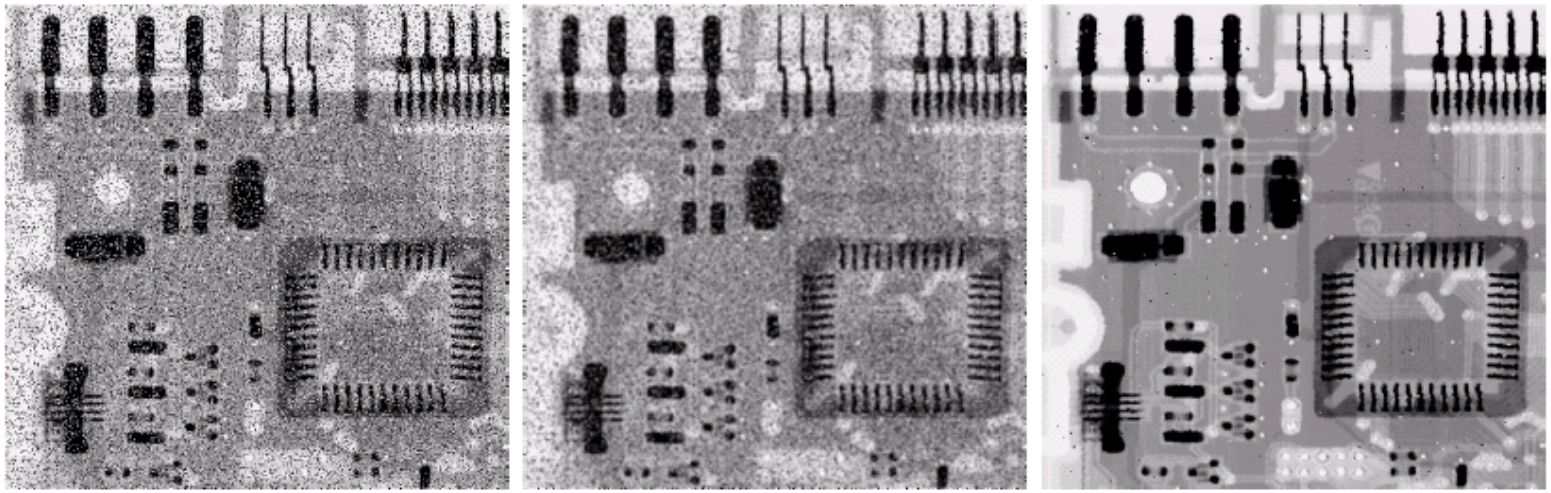


Original Image **Smoothed Image** **Thresholded Image**

Seems a reasonable representation of what we would consider to be the largest, brightest objects in that image.

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Averaging Filter Vs. Median Filter Example



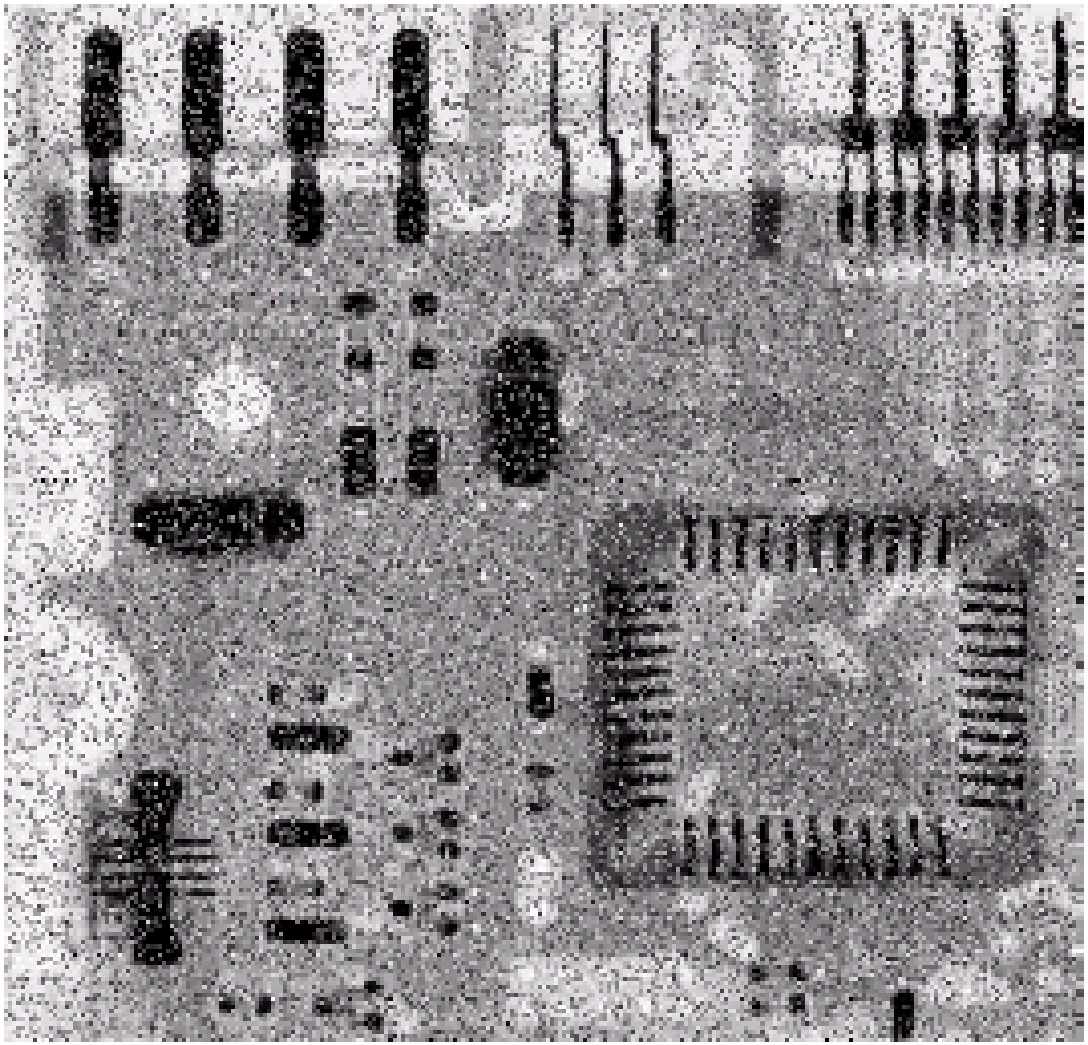
Original Image With Noise | **Image After Averaging Filter** | **Image After Median Filter**

Filtering is often used to remove noise from images

Sometimes a median filter works better than an averaging filter

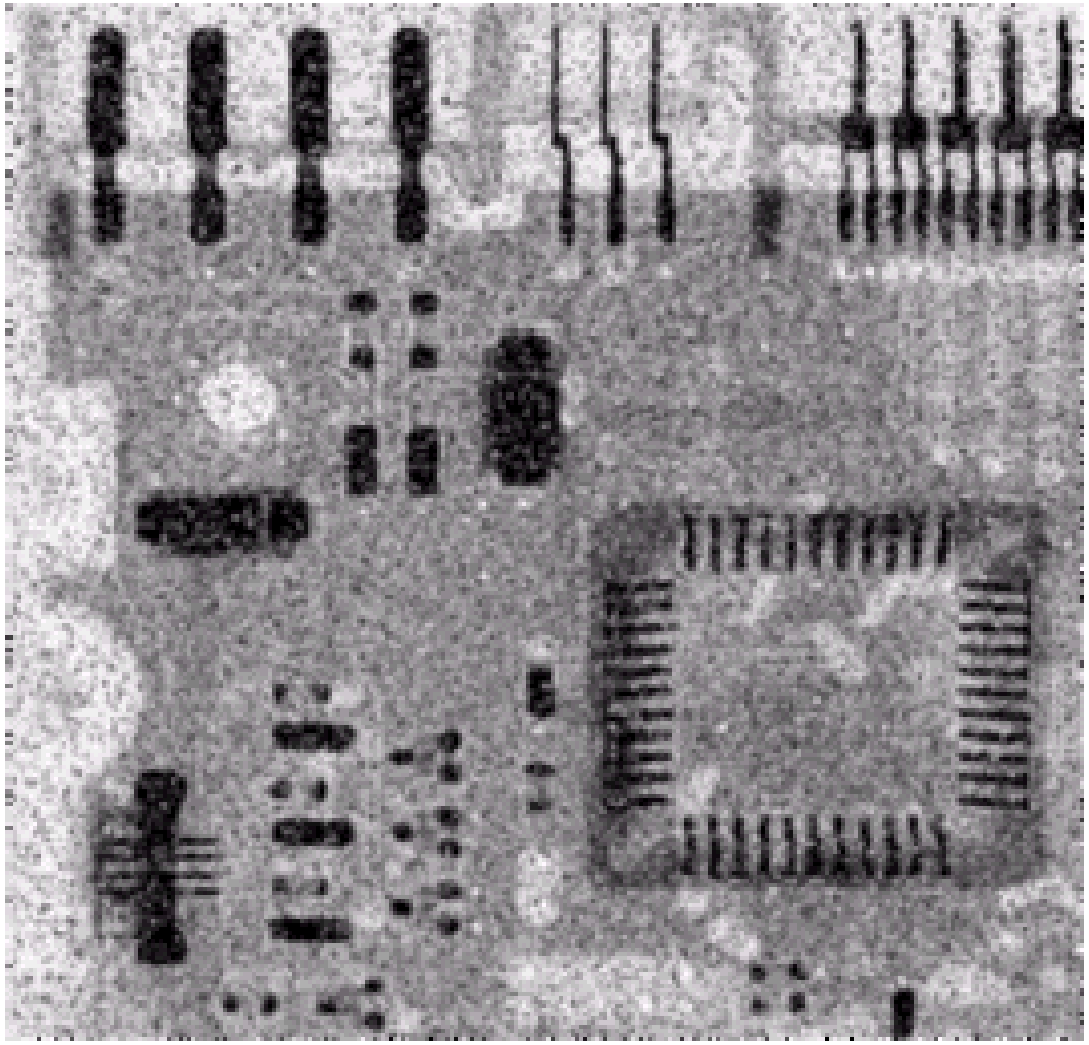
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Averaging Filter Vs. Median Filter Example



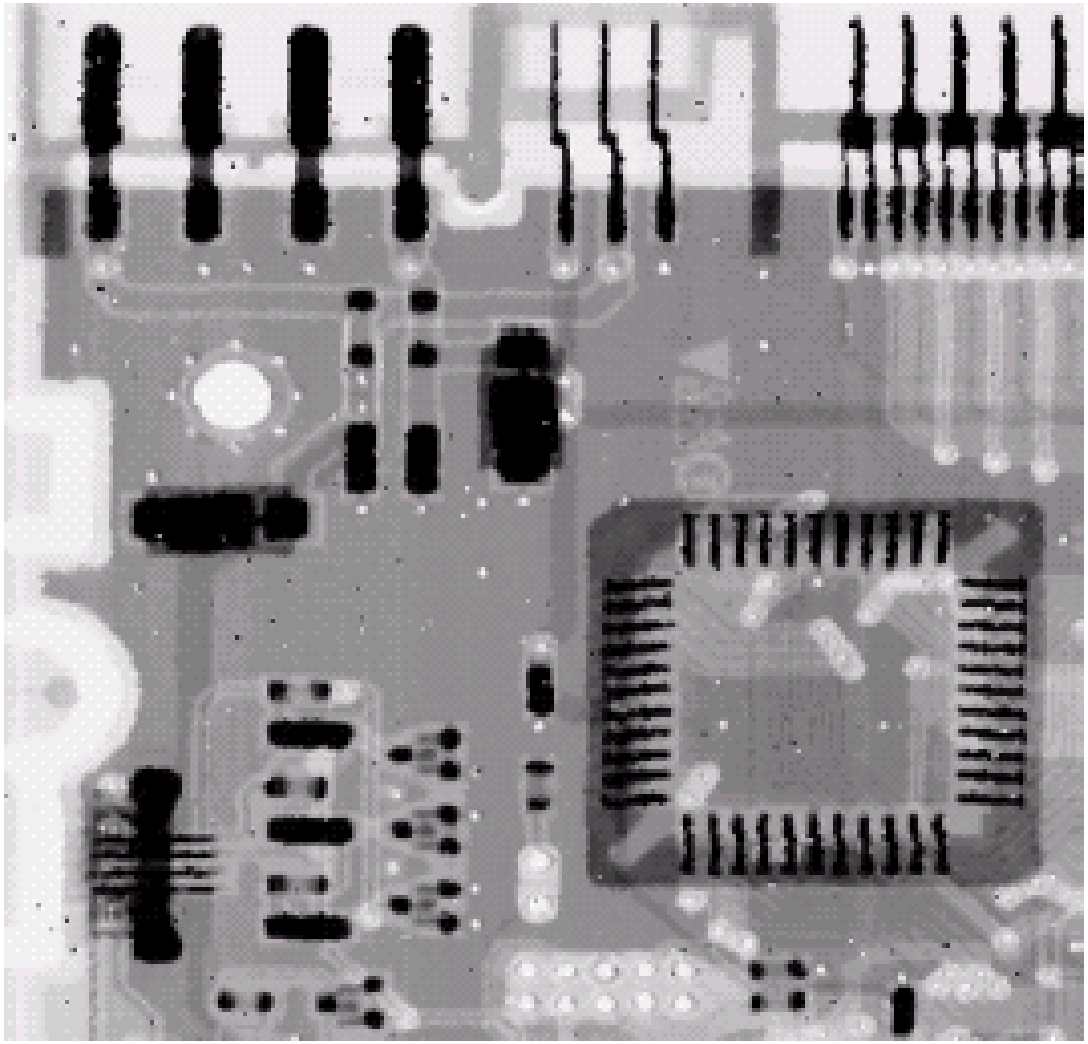
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Averaging Filter Vs. Median Filter Example



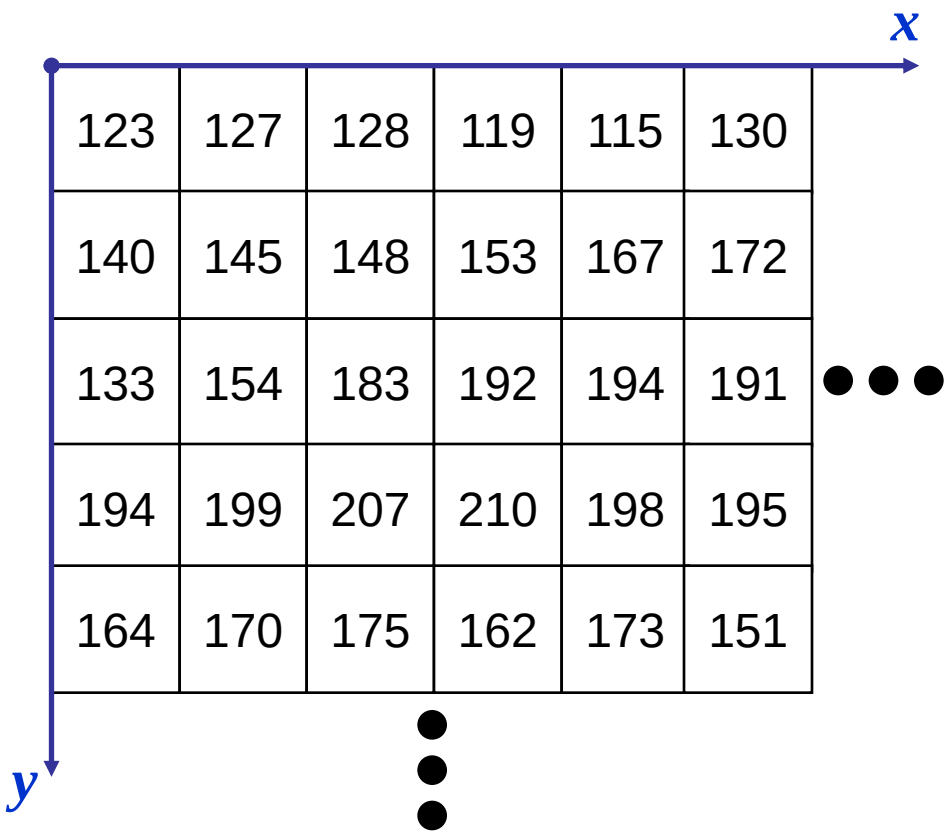
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Averaging Filter Vs. Median Filter Example



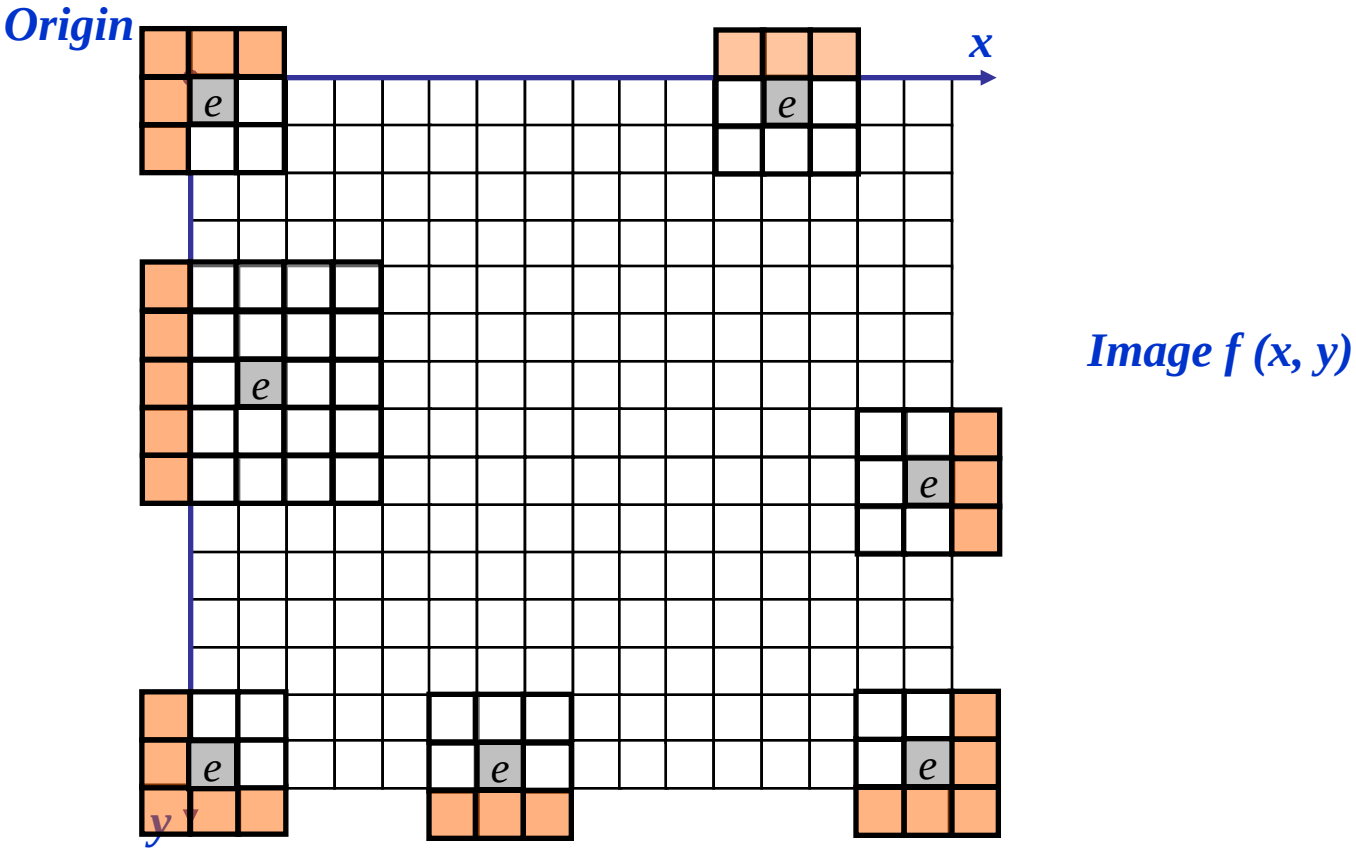
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Simple Neighbourhood Operations Example



Strange Things Happen At The Edges!

At the edges of an image we are missing pixels to form a neighbourhood



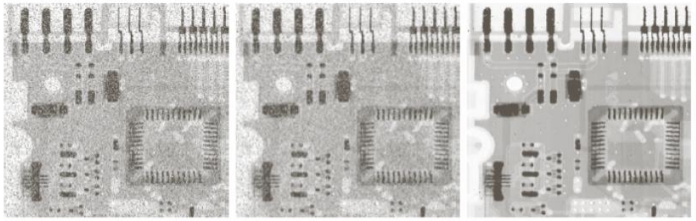
Credits: Brian Mac, Dublin Institute of Technology

Strange Things Happen At The Edges! (cont…)

There are a few approaches to dealing with missing edge pixels:

- Omit missing pixels
  - Only works with some filters
  - Can add extra code and slow down processing
- Pad the image
  - Typically with either all white or all black pixels
- Replicate border pixels
- Truncate the image
- Allow pixels *wrap around* the image
  - Can cause some strange image artefacts

Credits: Brian Mac, Dublin Institute of Technology



a b c

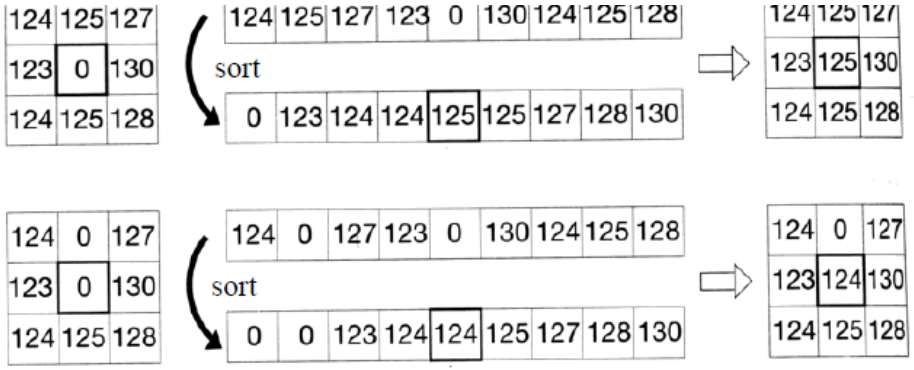
FIGURE 3.35 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

3.5.2 Order-Statistic (Nonlinear) Filters

Ordering (ranking) the pixels contained in the image area encompasses by the filter, then replacing the value of the centre pixel with the value of ranking result

Median filter: replaces the value of a pixel by the median of the intensity values in the neighbourhood. Good noise-reduction capabilities with less smoothing (e.g. *impulse noise*, or *salt-and-pepper noise*)

Median filter :



[Bres, 2003]

Max filter, min filter

© 1992–2008 R. C. Gonzalez & R. E. Woods

Sharpening Spatial Filters

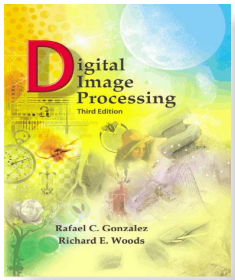
Previously we have looked at smoothing filters which remove fine detail

Sharpening spatial filters seek to highlight fine detail

- Remove blurring from images
- Highlight edges

Sharpening filters are based on *spatial differentiation*

The principal objective of sharpening is to highlight transitions in intensity. Uses of image sharpening vary and include applications ranging from electronic printing and medical imaging to industrial inspection and autonomous guidance in military systems. In the last section, we saw that image blurring could be accomplished in the spatial domain by pixel averaging in a neighborhood. Because averaging is analogous to integration, it is logical to conclude that sharpening can be accomplished by spatial differentiation. This, in fact, is the case.



Chapter 3
Intensity Transformations & Spatial Filtering

3.6 Sharpening Spatial Filters

Objective: highlight transitions in intensity

Spatial (digital) differentiation => enhances edges and other discontinuities

3.6.1 Foundation

Let's study the behavior of first- and second-order derivatives in areas of constant intensities, at the onset and end of discontinuities, and along intensity ramps

Basic definition of first-order derivative of a one-dimensional function f(x):

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

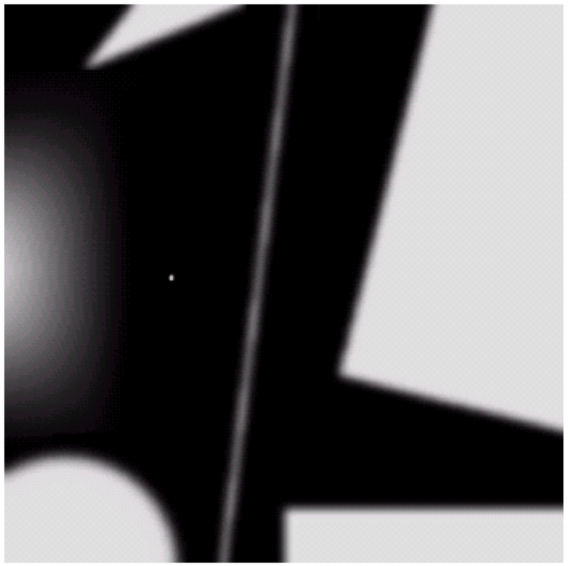
Second derivative: $$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

© 1992–2008 R. C. Gonzalez & R. E. Woods

Spatial Differentiation

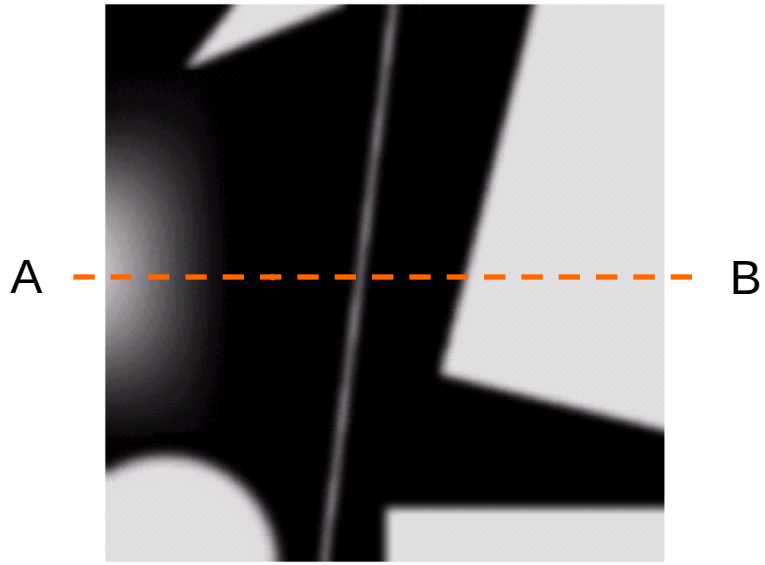
Differentiation measures the *rate of change* of a function

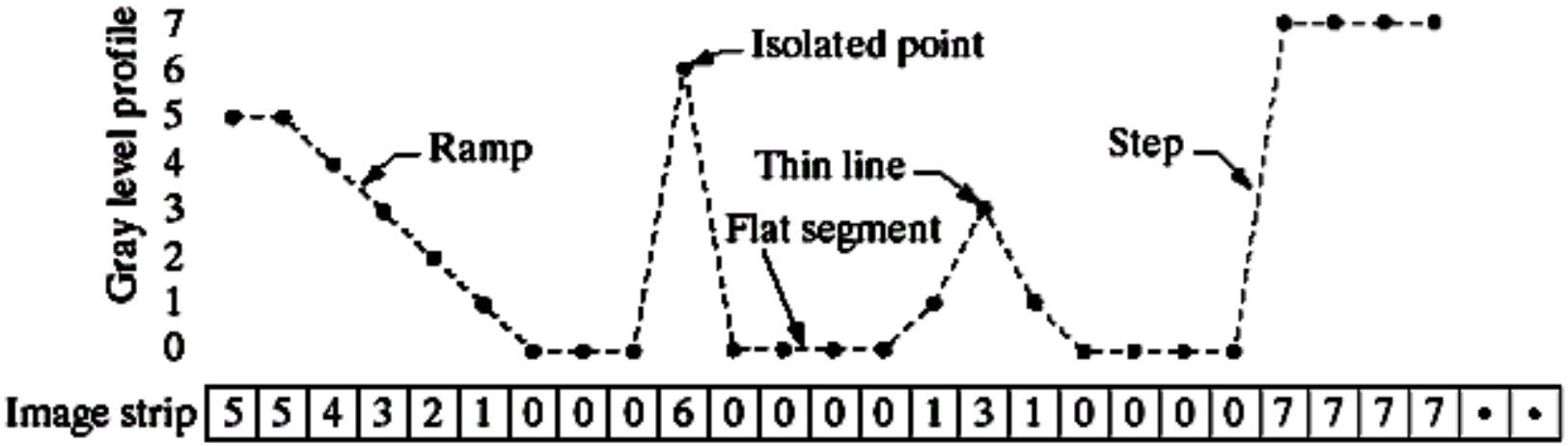
Let's consider a simple 1 dimensional example



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Spatial Differentiation





Images taken from Gonzalez & Woods, Digital Image Processing (2002)

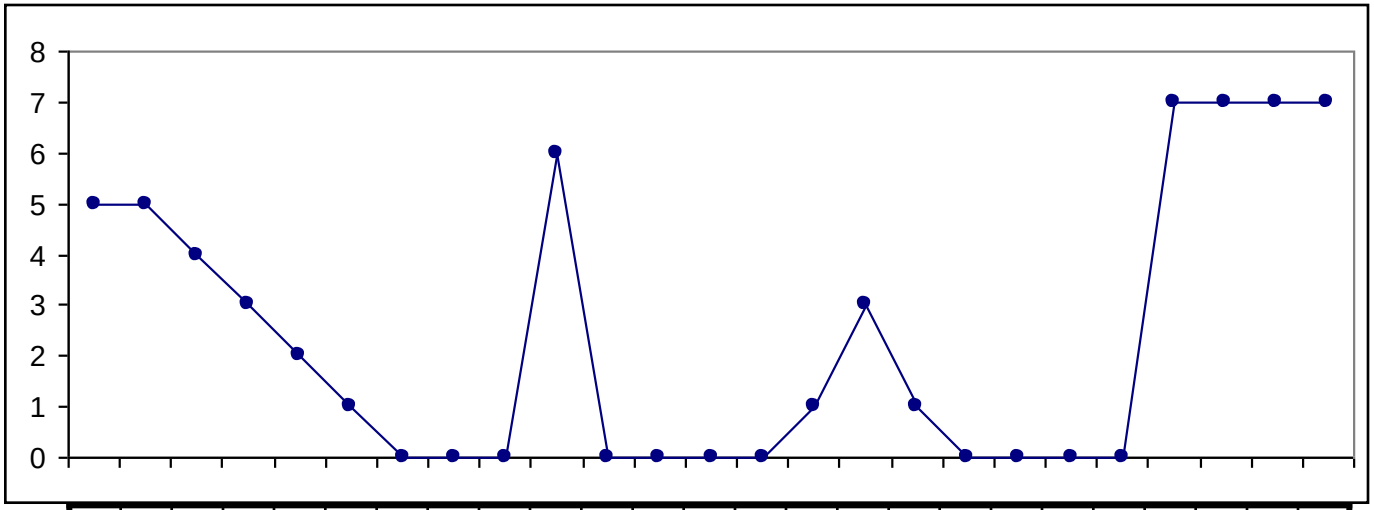
1st Derivative

The formula for the 1st derivative of a function is as follows:

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

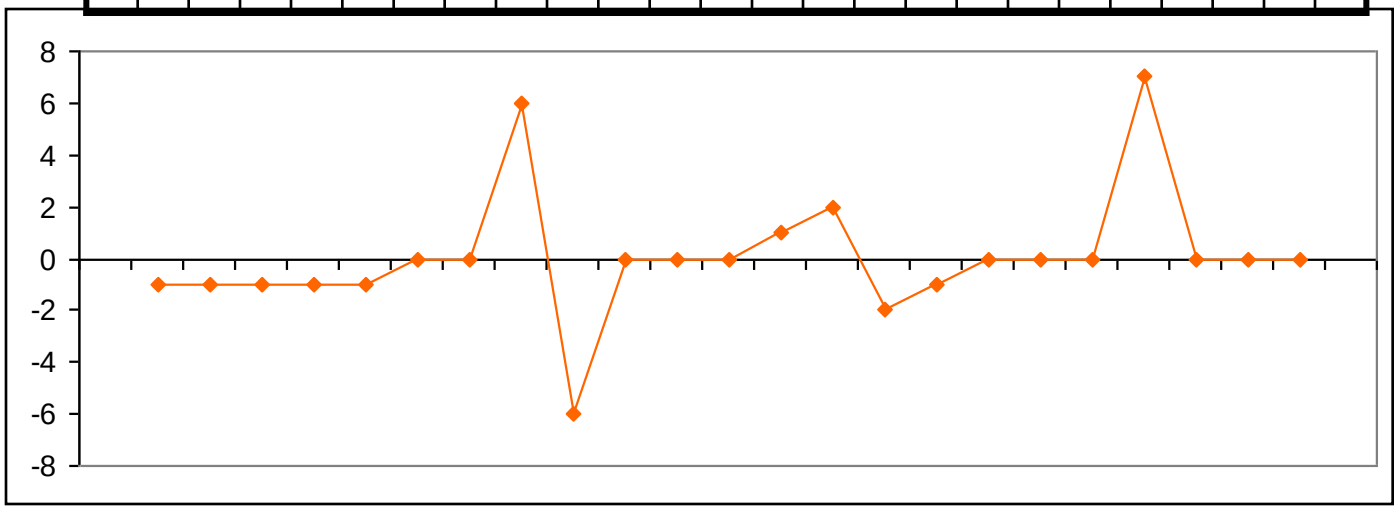
It's just the difference between subsequent values and measures the rate of change of the function

1st Derivative (cont…)



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

	0	-1	-1	-1	-1	0	0	6	-6	0	0	0	1	2	-2	-1	0	0	0	7	0	0	0	



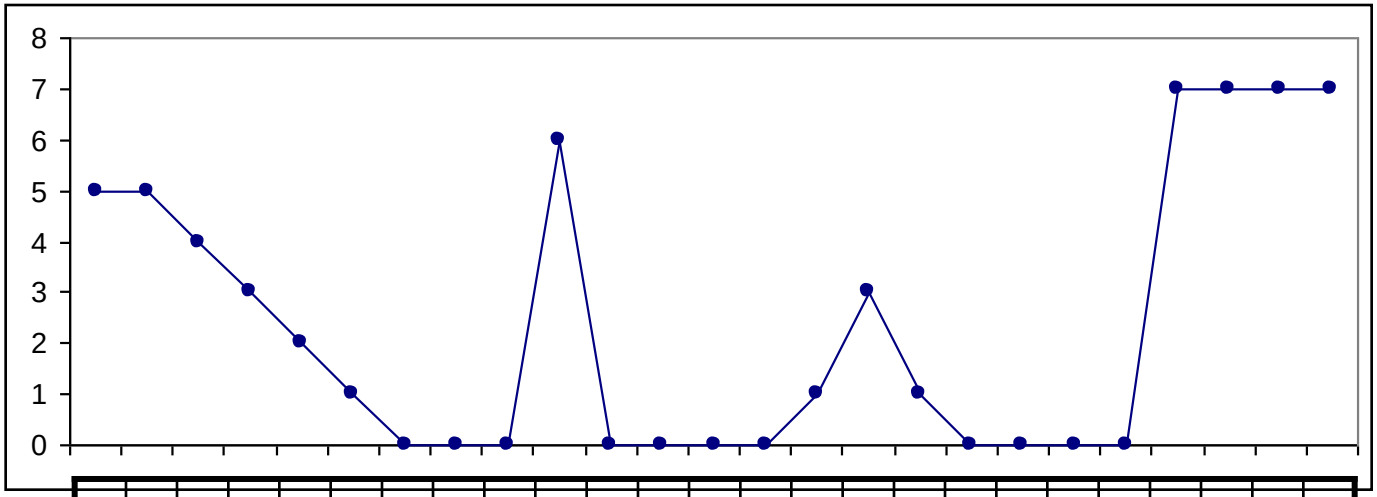
2nd Derivative

The formula for the 2nd derivative of a function is as follows:

$$\frac{\partial^2 f}{\partial^2 x} = f(x+1) + f(x-1) - 2f(x)$$

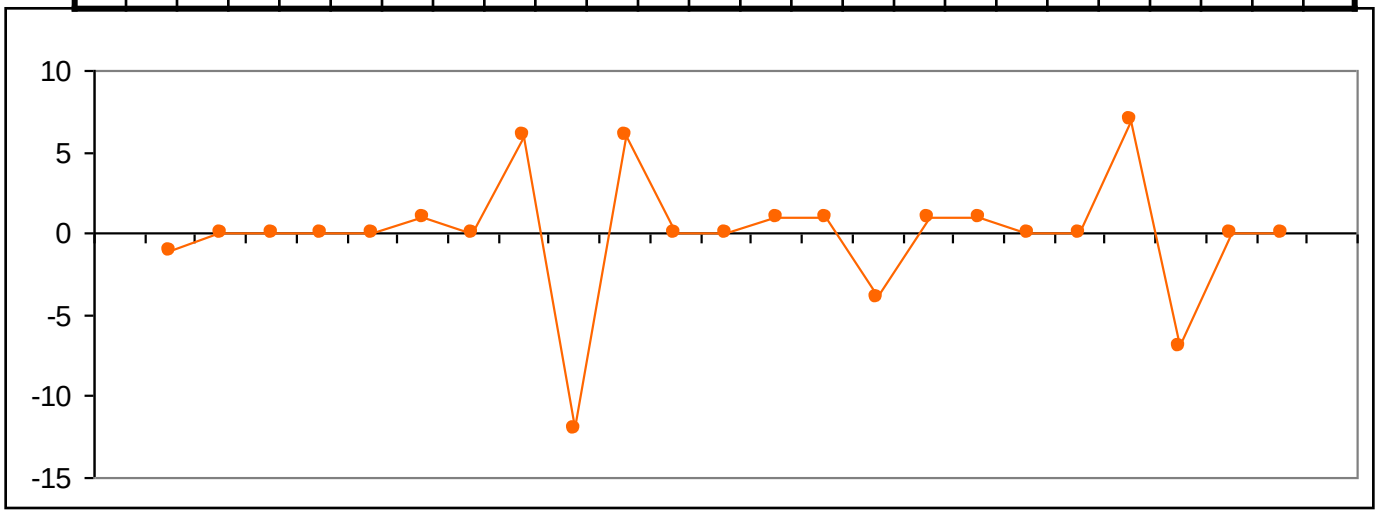
Simply takes into account the values both before and after the current value

2nd Derivative (cont…)



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

	-1	0	0	0	0	1	0	6	-12	6	0	0	1	1	-4	1	1	0	0	7	-7	0	0	



Using Second Derivatives For Image Enhancement

The 2nd derivative is more useful for image enhancement than the 1st derivative

- Stronger response to fine detail
- Simpler implementation
- We will come back to the 1st order derivative later on

The first sharpening filter we will look at is the *Laplacian*

- Isotropic
- One of the simplest sharpening filters
- We will look at a digital implementation

The Laplacian

The Laplacian is defined as follows:

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

where the partial 1st order derivative in the x direction is defined as follows:

$$\frac{\partial^2 f}{\partial^2 x} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

and in the y direction as follows:

$$\frac{\partial^2 f}{\partial^2 y} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

The Laplacian (cont…)

So, the Laplacian can be given as follows:

$$\nabla^2 f = [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)] - 4f(x,y)$$

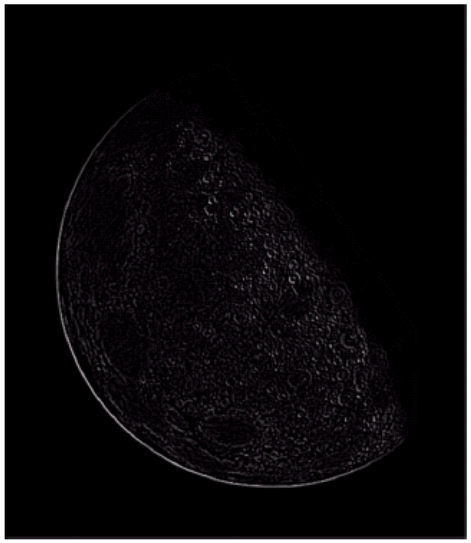
We can easily build a filter based on this

0	1	0
1	-4	1
0	1	0

The Laplacian (cont…)

Applying the Laplacian to an image we get a new image that highlights edges and other discontinuities

Original Image



Laplacian Filtered Image

Laplacian Filtered Image Scaled for Display

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

But That Is Not Very Enhanced!

The result of a Laplacian filtering is not an enhanced image

We have to do more work in order to get our final image

Subtract the Laplacian result from the original image to generate our final sharpened enhanced image

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

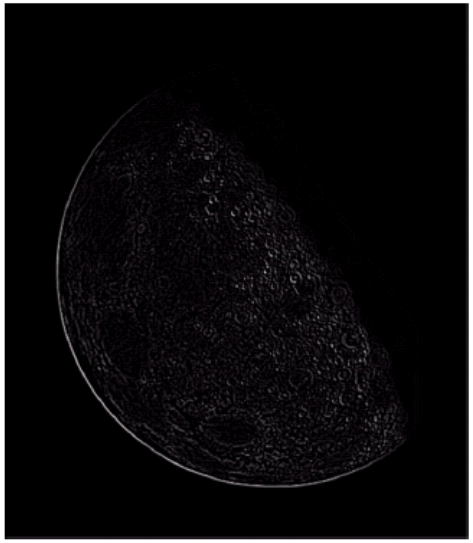
Laplacian
Filtered Image
Scaled for Display

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Laplacian Image Enhancement

Original Image

-



Laplacian Filtered Image

=

Sharpened Image

In the final sharpened image edges and fine detail are much more obvious

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Laplacian Image Enhancement

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

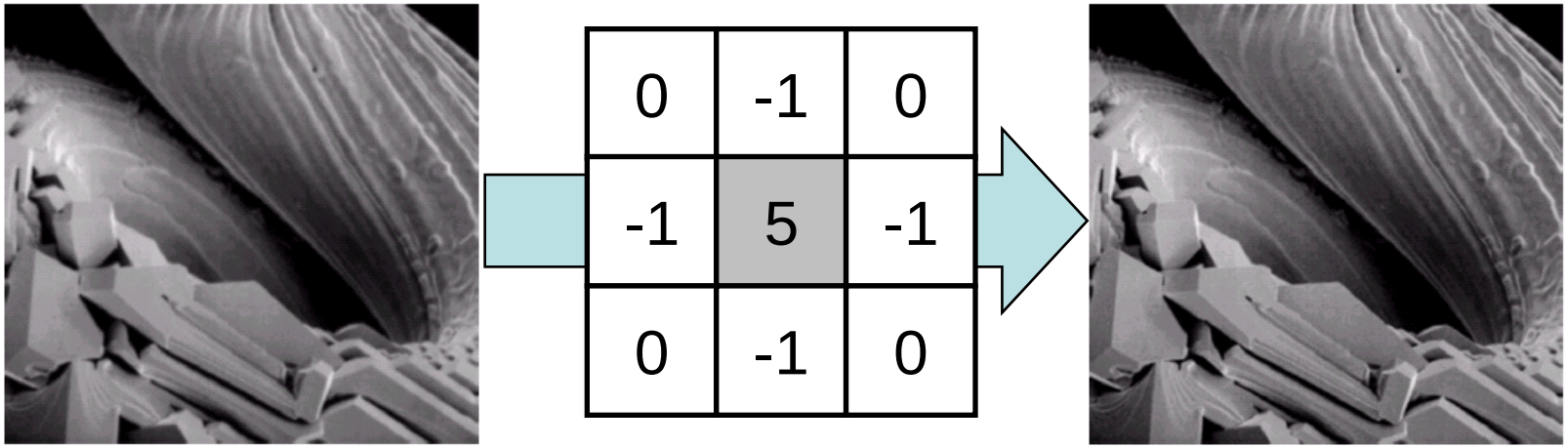
Simplified Image Enhancement

The entire enhancement can be combined into a single filtering operation

$$
\begin{aligned}
g(x,y) &= f(x,y) - \nabla^2 f \\
&= f(x,y) - [f(x+1,y) + f(x-1,y) \\
&\qquad + f(x,y+1) + f(x,y-1) \\
&\qquad - 4f(x,y)] \\
&= 5f(x,y) - f(x+1,y) - f(x-1,y) \\
&\qquad - f(x,y+1) - f(x,y-1)
\end{aligned}
$$

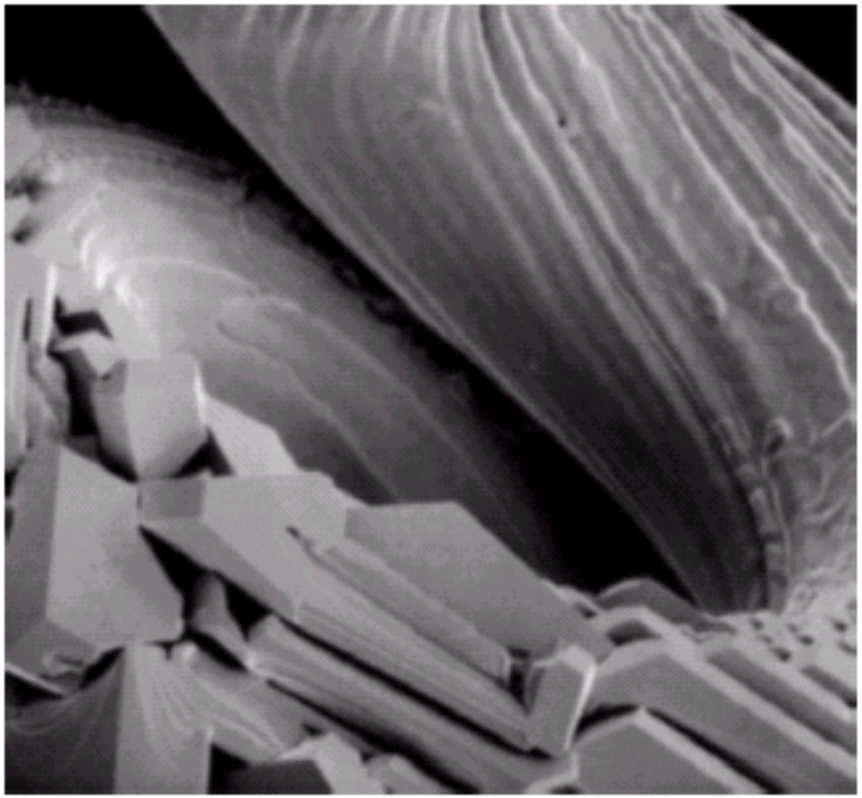
Simplified Image Enhancement (cont…)

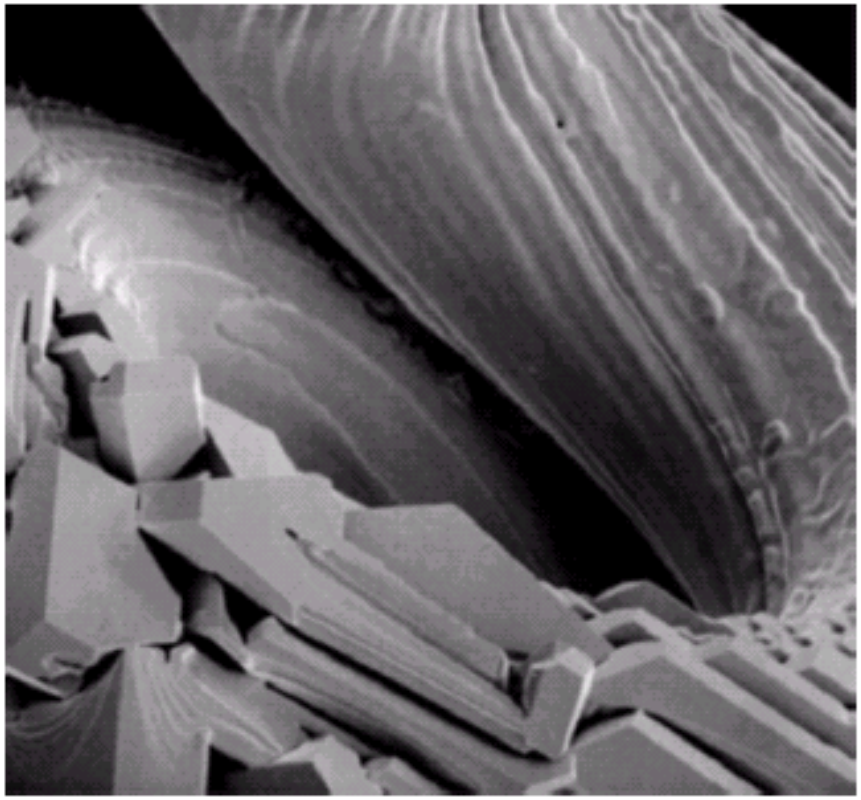
This gives us a new filter which does the whole job for us in one step



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Simplified Image Enhancement (cont…)





Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Variants On The Simple Laplacian

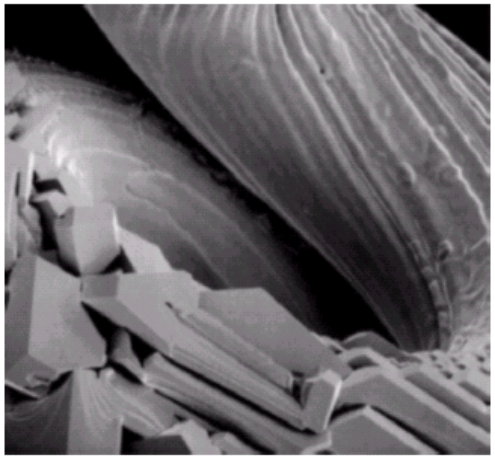
There are lots of slightly different versions of the Laplacian that can be used:

0	1	0
1	-4	1
0	1	0

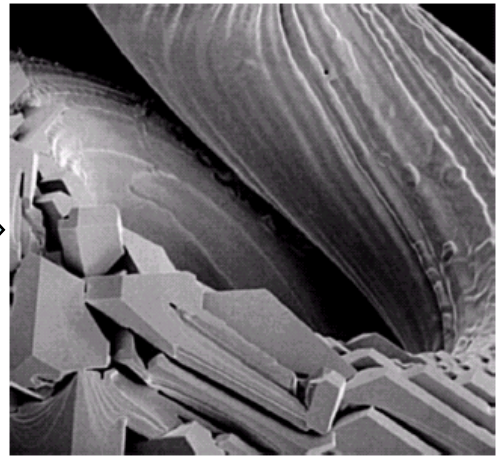
Simple Laplacian

1	1	1
1	-8	1
1	1	1

Variant of Laplacian



-1	-1	-1
-1	9	-1
-1	-1	-1



Images taken from Gonzalez & Woods, Digital Image Processing (2002)