Chap 3 Image Enhancement in Spatial Domain

Introduction

- The main objective of enhancement is to process an image so that the result is more suitable than the original image for a specific application.
- Two categories of image enhancement
  - **Spatial domain** (image plane, Chap. 3)
  - **Frequency domain** (Fourier transform, Chap. 4)
- The spatial domain techniques operate directly on the pixels of an image as opposed to the frequency domain in which operations are performed on the Fourier transform of an image.
- All the techniques discussed in this section are implemented in the **spatial domain**.

Introduction
-- Image Enhancement Approaches

- Enhancement techniques are as follows:
  - **Point operations** : each pixel is modified by an equation that is not dependent on other pixel values
  - **Mask operations** : each pixel is modified according to the values in a small neighborhood (or subimage)
  - **Global operations** : all pixel values in the image are taken into consideration

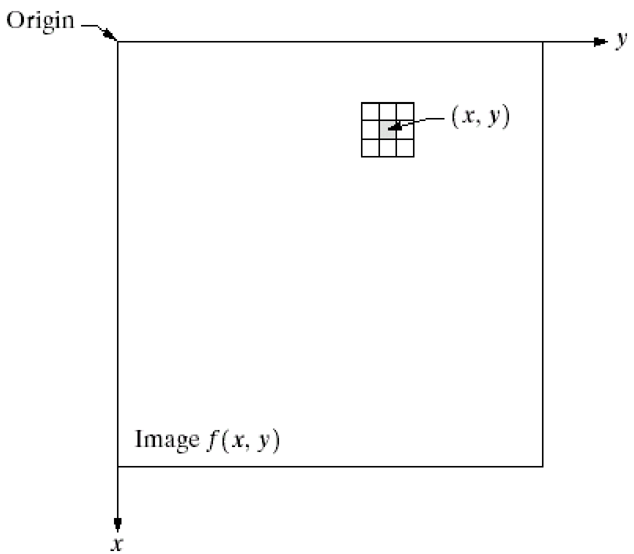
3.1 Background

- Spatial domain processes on images can be described as

$$g(x, y) = \mathbf{T}[f(x, y)]$$

  - where $f(x, y)$ is the input image, $g(x, y)$ is the output image, T is an operator
  - For mask operations, T operates on the neighbors of $f(x, y)$
    - Typically, the neighborhood is rectangular, centered on (x, y), and much smaller in size than the image.
    - A square or rectangular sub-image centered at (x, y) is placed to yield the output $g(x, y)$.

3.1 Background



$g(x, y) = \boldsymbol{T}[f(x, y)]$

$f(x, y)$: input image

$g(x, y)$: processed image

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

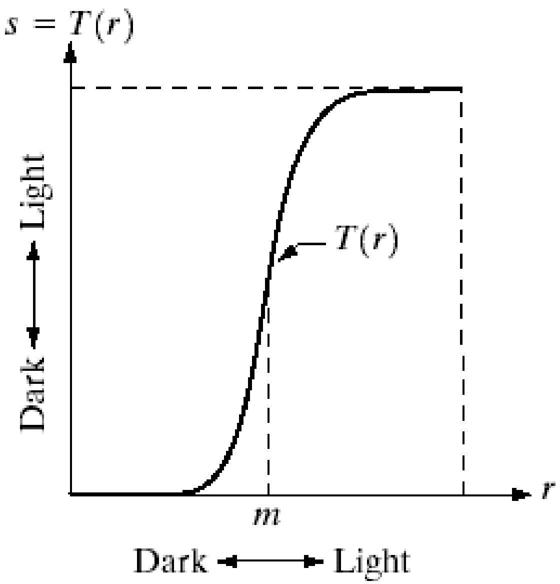
- The process consists of moving the origin of the neighborhood from pixel to pixel and applying the operator T to the pixels in the neighborhood to yield the output at that location.
- This procedure is called **spatial filtering**, in which the neighborhood along with a predefined operation is called a **spatial mask** (or **filter kernel**, **template**, or **window**)

3.1 Background

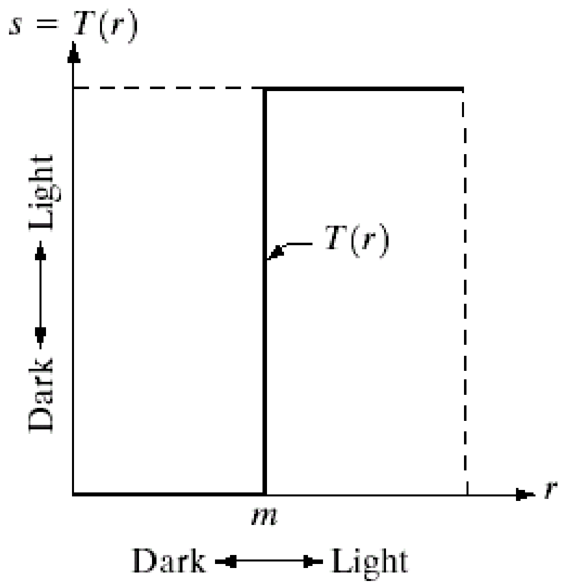
- The simplest form of T is the neighborhood of size 1×1.
- g depends on the value of f at (x, y), which is called **intensity** (or called **gray level** or **mapping) transformation** formulated as $\boldsymbol{s = T(r)}$
  - r is the **gray-level (digital number)** of $f(x, y)$ at any point (x, y).
  - s is the **gray-level (digital number)** of $g(x, y)$ at any point (x, y).
  - Enhancement of any point depends on that point only.
    - Point processing
- **Larger neighborhood provides more flexibility**.
  - Mask processing or filtering

Gray-level Transformation Function

- Point processing: $s = T(r)$



Contrast stretching



Thresholding

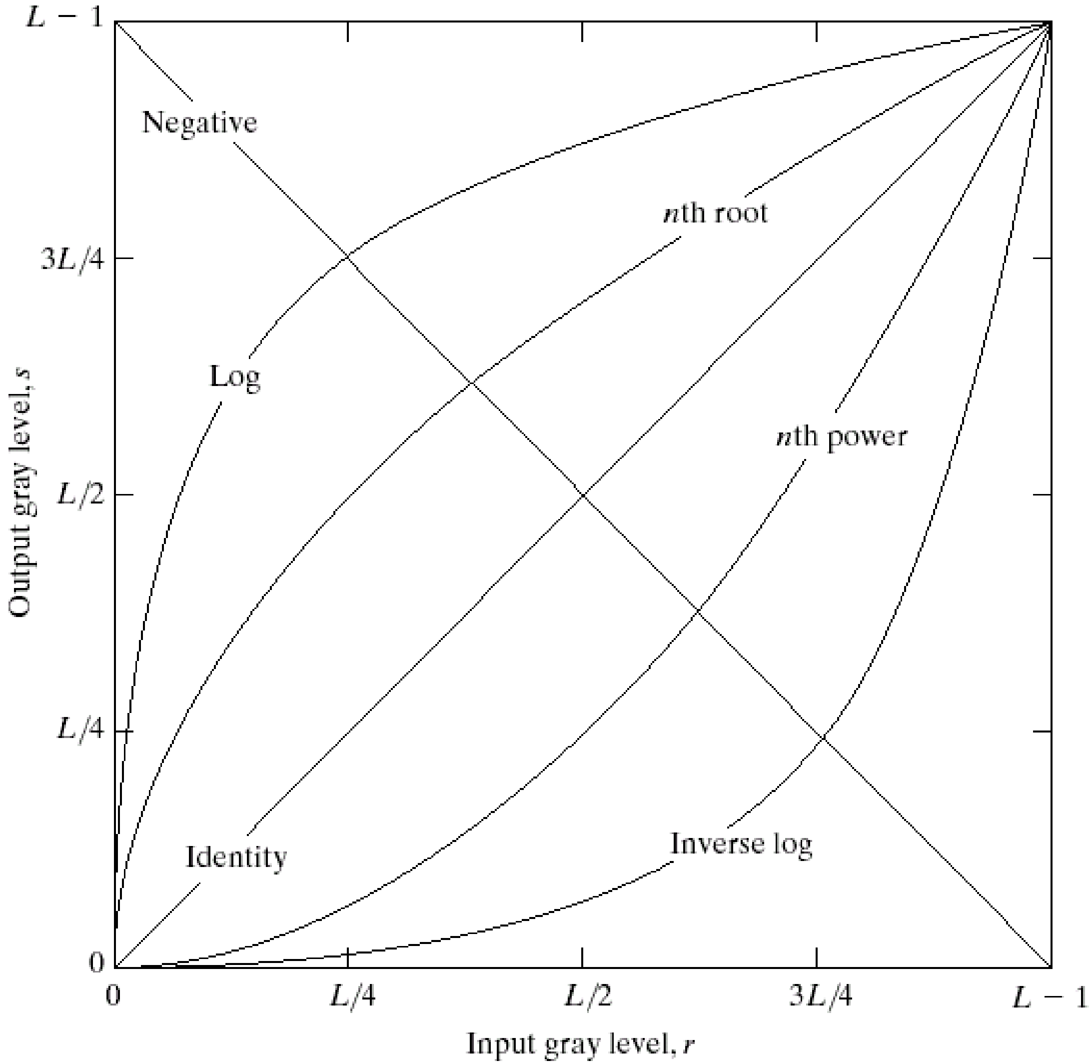
3.2 Basic Intensity Transformations

- Three basic functions used frequently for image enhancement
  - **Linear** (negative and identity transformation)

    ex: $s = (L - 1) - r$ (**negative transformation**)
  - **Logarithmic** (log and inverse log)

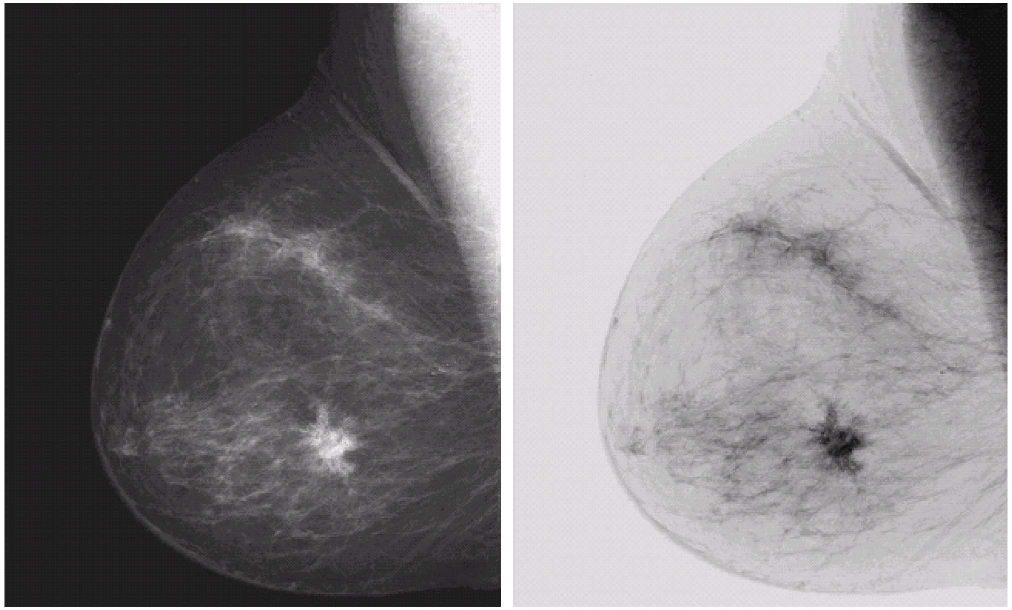
    ex: $s = clog(1 + r)$ (**log transformation**)
  - **Power law (Gamma)** (γ-th power and γ-th root transformation)

    ex: $s = cr^{\gamma}$ or $s = c(r + \varepsilon)^{\gamma}$(**$\boldsymbol{\gamma}$-th power trans.**)


L − 1
3L/4
L/2
L/4
0
0
L/4
L/2
3L/4
L − 1
Negative
nth root
Log
nth power
Identity
Inverse log
Output gray level, s
Input gray level, r

3.2.1 Image Negatives

- This type of processing is particularly suited for enhancing white or gray detail embedded in dark regions of an image, especially when the black areas are dominant in size.
- The image is a digital mammogram (X-ray imaging) showing a small lesion inside.



a b

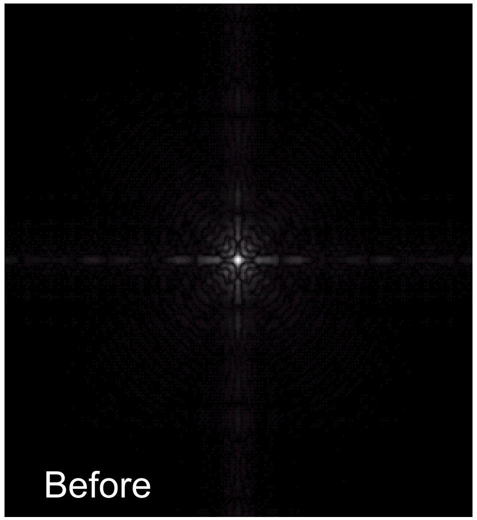
FIGURE 3.4 (a) Original digital mammogram. (b) Negative image obtained using the negative transformation in Eq. (3.2-1). (Courtesy of G.E. Medical Systems.)

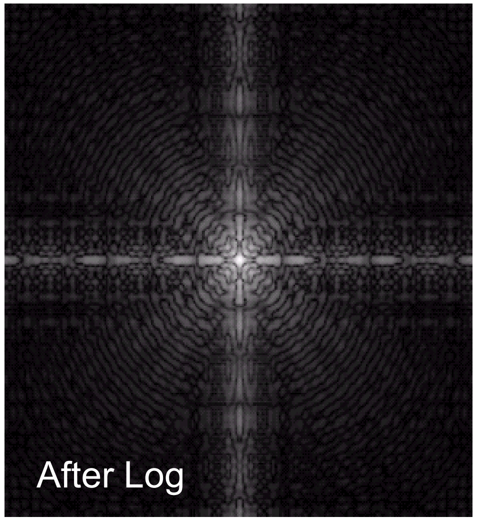
3.2.2 Log Transformations

- This transformation maps a narrow range of low intensity in the input into a wider range of output levels. The opposite is true of higher values of input levels.
- In other words, this transformation is used to expand the values of dark pixels while compressing the higher-level values. The opposite is true of the inverse log transformation.
- And, it can **compress the dynamic range of images** with large variations in pixel values.

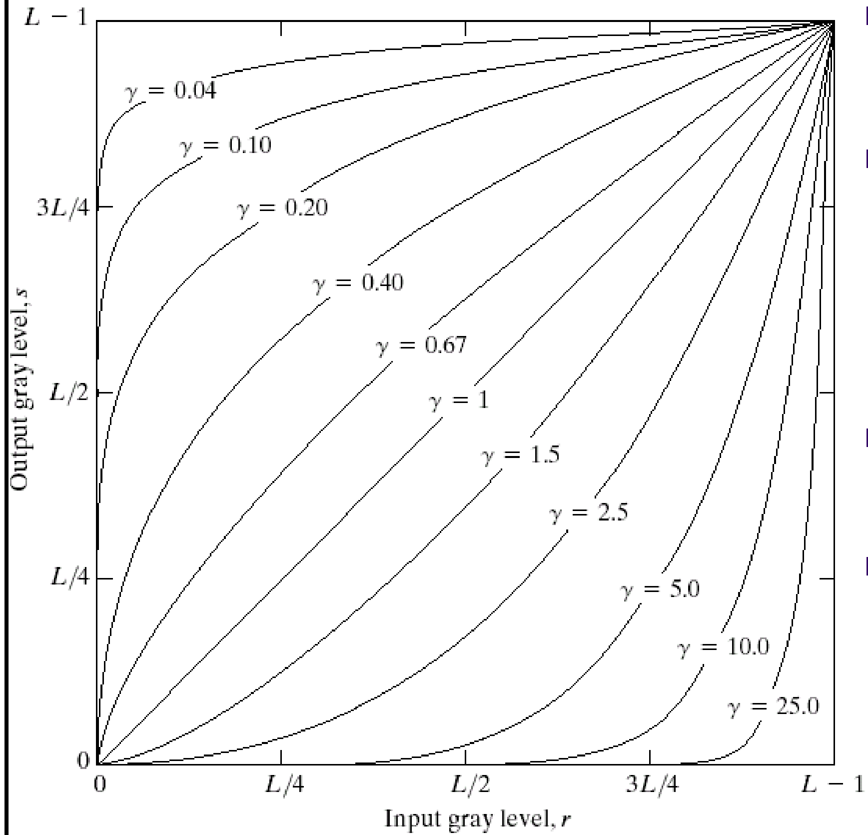
Fourier spectrum:
$0 \sim 1.5 * 10^6$

Compress the dynamic range of the Fourier spectrums with large variations.





3.2.3 Power-Law Transformations



- Equation:

  $s = cr^{\gamma}$ or $s = c(r + \varepsilon)^{\gamma}$

- Unlike the log function, a family of possible transformation curves obtained simply by varying r.
- $r > 1$: Enhance light areas.

  $r < 1$: Enhance dark areas.
- The curves generated with values of $r > 1$ have exactly the opposite effect as those generated with values of $r < 1$.

Examples of Power-Law Transformation

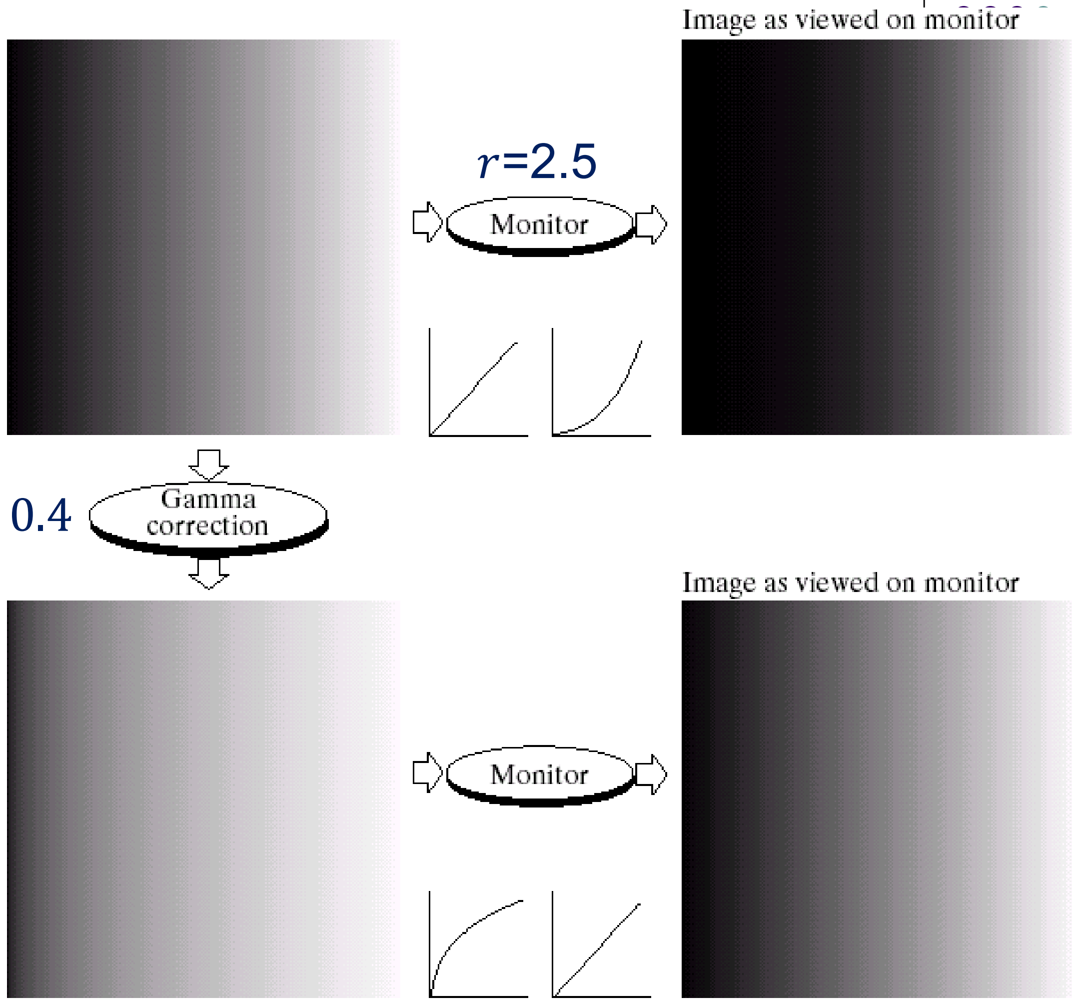
- By convention, the exponent in the power-law equation is referred to as gamma. The process used to correct these power-law response phenomena is called gamma correction.
- γ correction, for example
  - The cathode ray tube (CRT) devices have an intensity-to-voltage response which is a power function with γ generally ranges from 1.8 to 2.5.
  - Without γ correction, the monitor output will become darker than the original input.

a	b
c	d

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

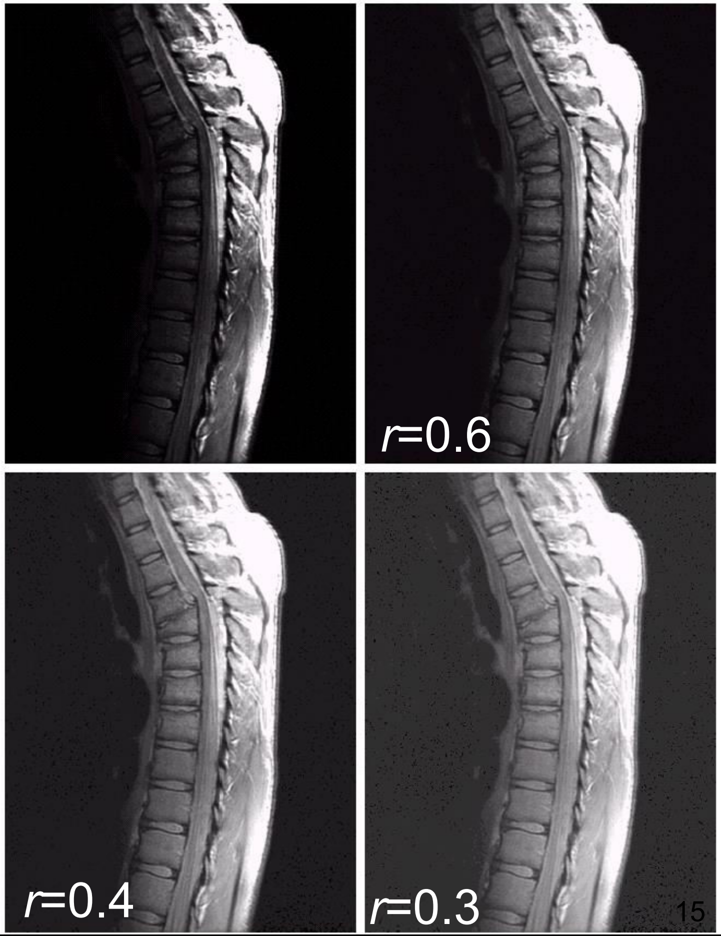
$r = 2.5$

$$r = \frac{1}{2.5} = 0.4$$

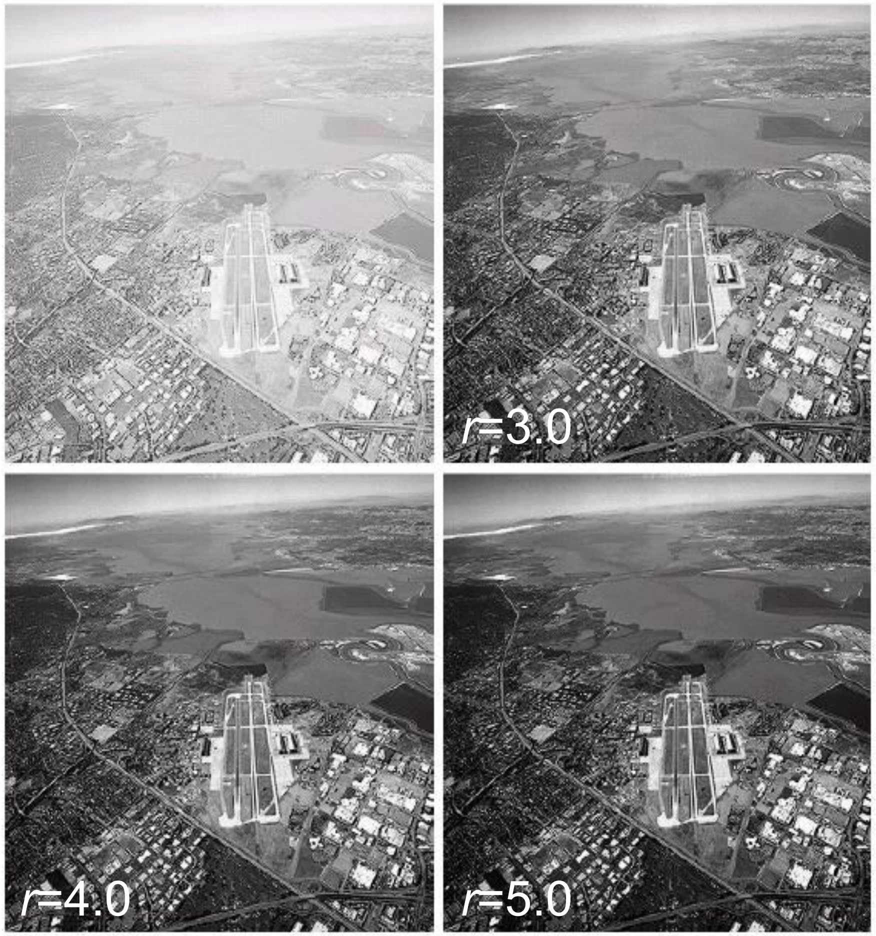


Examples of Gamma Correction

- In addition to gamma correction, power-law transformations are useful for general-purpose contrast manipulation.
- This example shows a magnetic resonance image (MRI) of a fractured human spine.
- Because the given image is predominantly dark, a power-law transformation with a fractional exponent is applied.



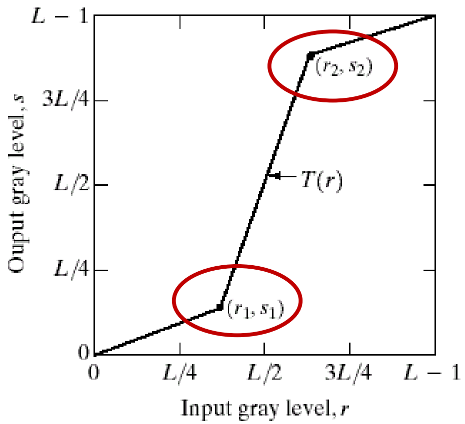
Examples of Gamma Correction

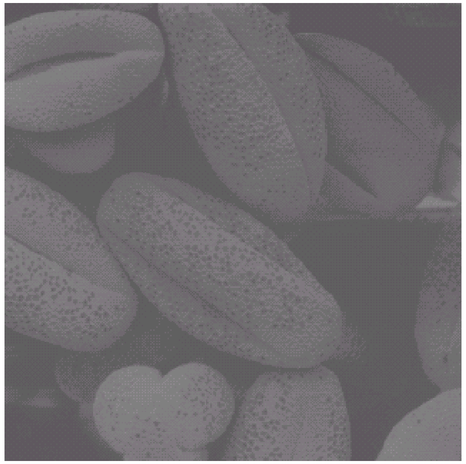


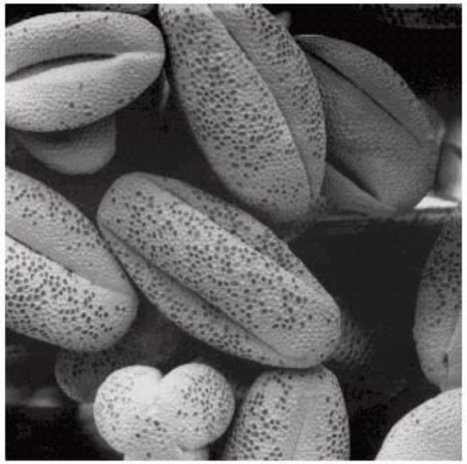
- The aerial image has a washed-out appearance, indicating that a enhancement of high gray levels is desirable.
- This can be accomplished with gamma transformation using $\boldsymbol{r > 1}$.
- Suitable results were obtained with $r = 3$ and $r = 4$.
- The result obtained with $r = 5$ has areas that are too dark (some detail is lost).

3.2.4 Piecewise Linear Transformation

- The principal advantage of **piecewise linear function** over the functions mentioned previously is that this piecewise form can be arbitrarily complex.
- The main disadvantage of piecewise functions is that they requires considerably more user input.
- Example1: **contrast stretching**







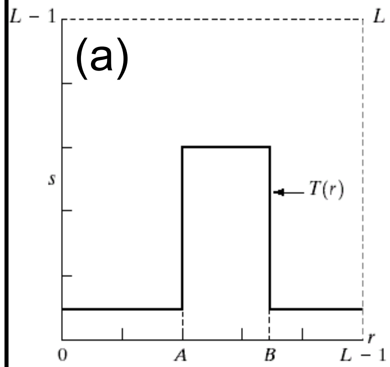
- Intermediate values of (r_1, s_1) and (r_2, s_2) produce various degrees of spread in the intensity levels of the output images, thus affecting its contrast.
- In general, the values are assigned so that the function is **single valued and monotonically increasing**. This condition preserves the order of intensity levels, thus preventing the creation of intensity artifacts.

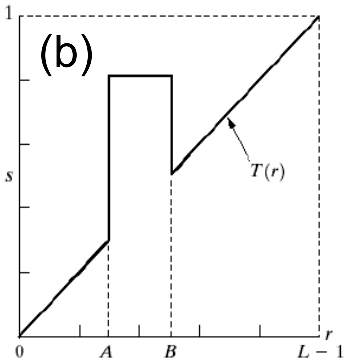
$(r_1,s_1)=(r_{min},0),(r_2,s_2)=(r_{max},L-1)$

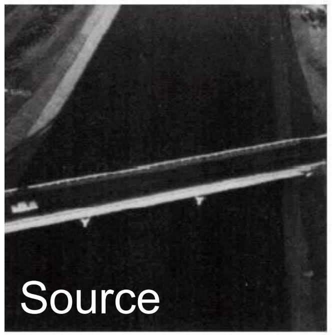
$r_1=r_2=m,\ s_1=0,\ s_2=L-1$

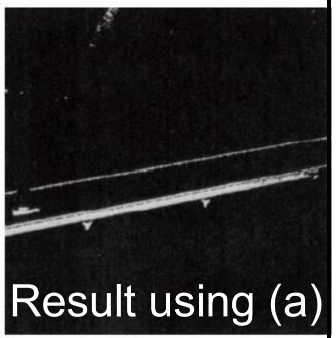
3.2.4 Piecewise Linear Transformation

- Example 2: **Gray-level slicing** -- highlight a specific range of gray levels
  - **The first approach (a)** is to display in one value (say, white) for all the values in the range of interest, and in another value (say, black) for all other intensities.
  - **In the second approach (b)**, brightens (or darkens) the desired range of intensities but leaves all other intensity levels in the image unchanged.



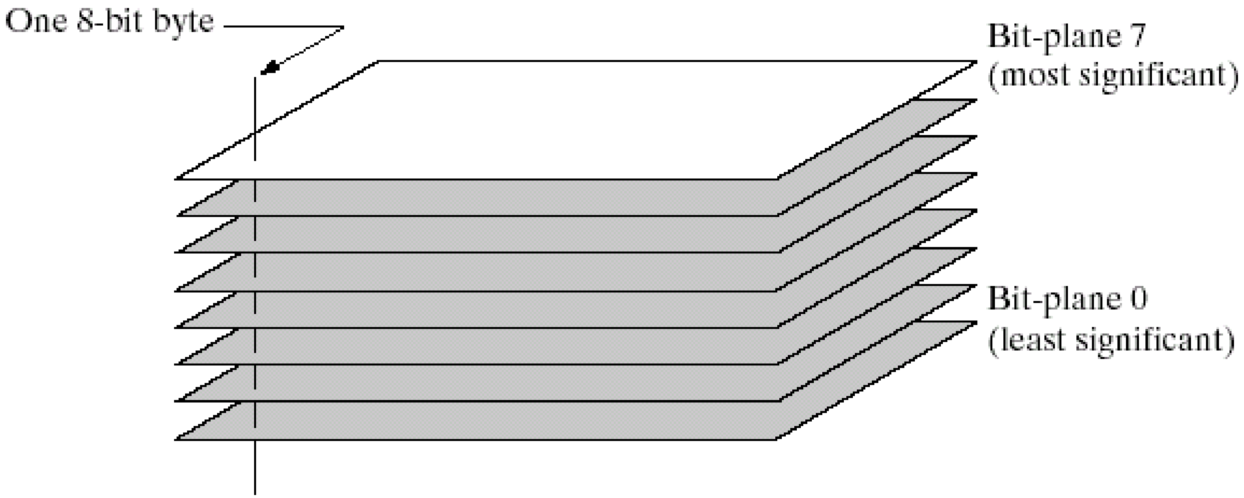






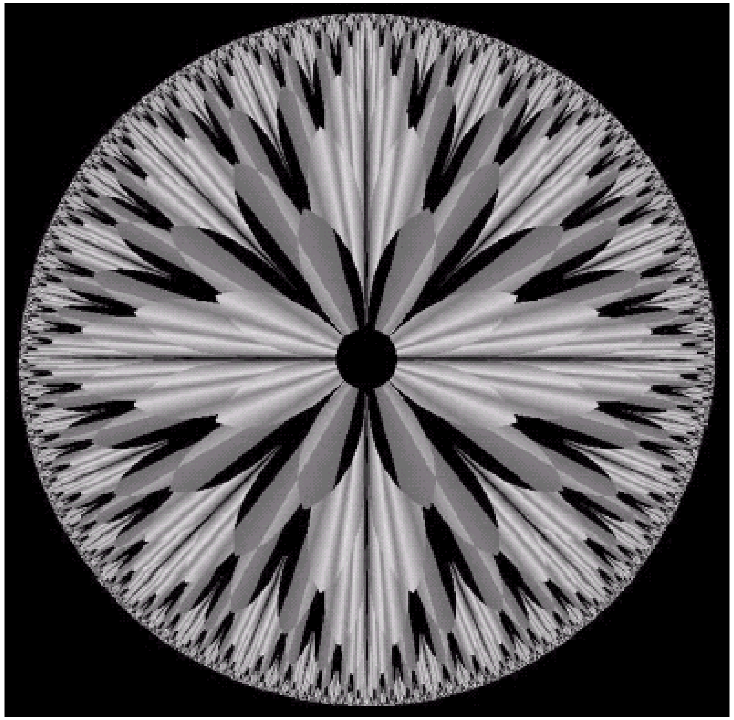
3.2.4 Piecewise Linear Transformation

- Example 3: **Bit-plane Slicing**
  - Pixels are digital numbers composed of bits. Instead of highlighting intensity-level ranges, we could highlight the contribution made to total image appearance by specific bits.
  - As illustrated, a 8-bit image may be considered as being composed of eight 1-bit planes.

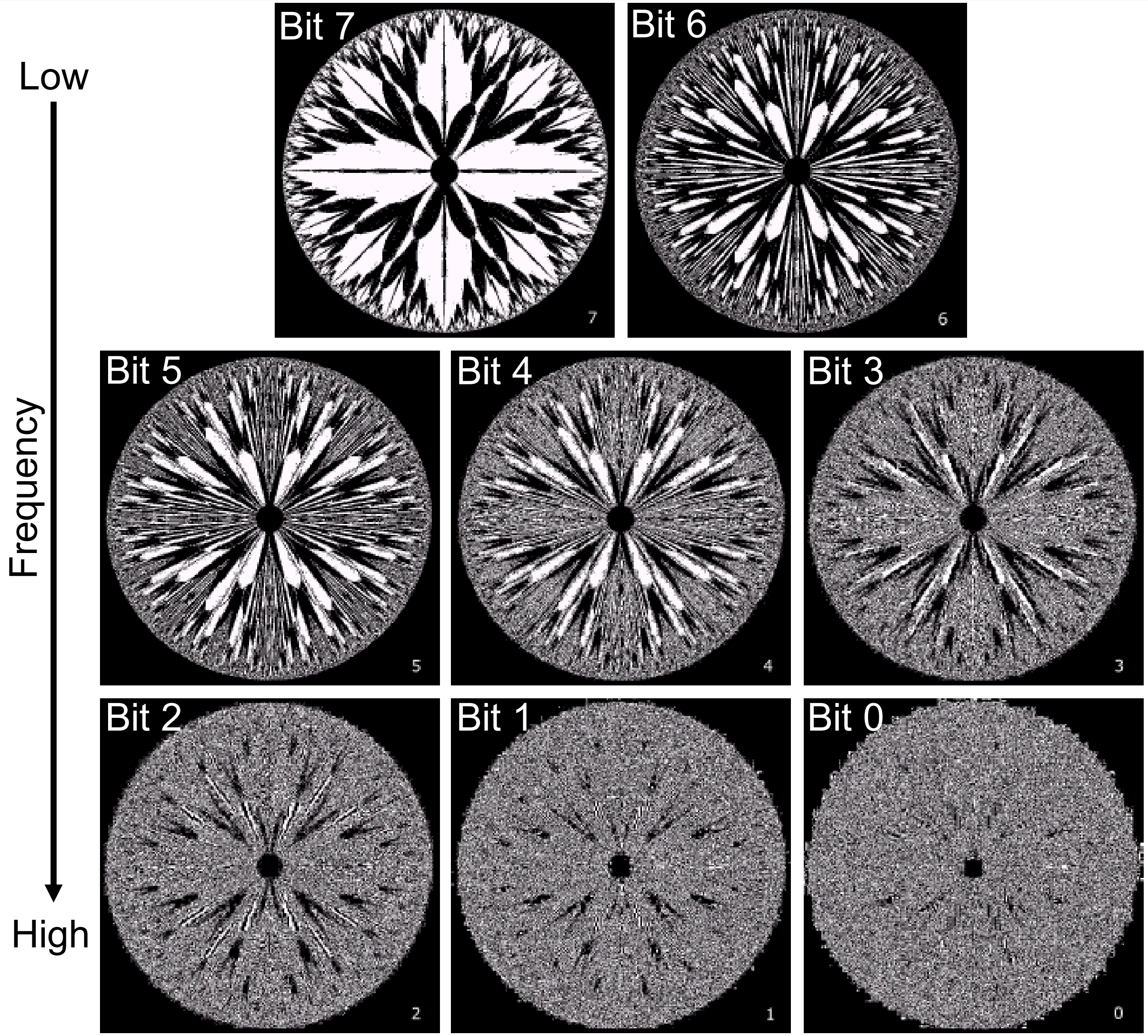


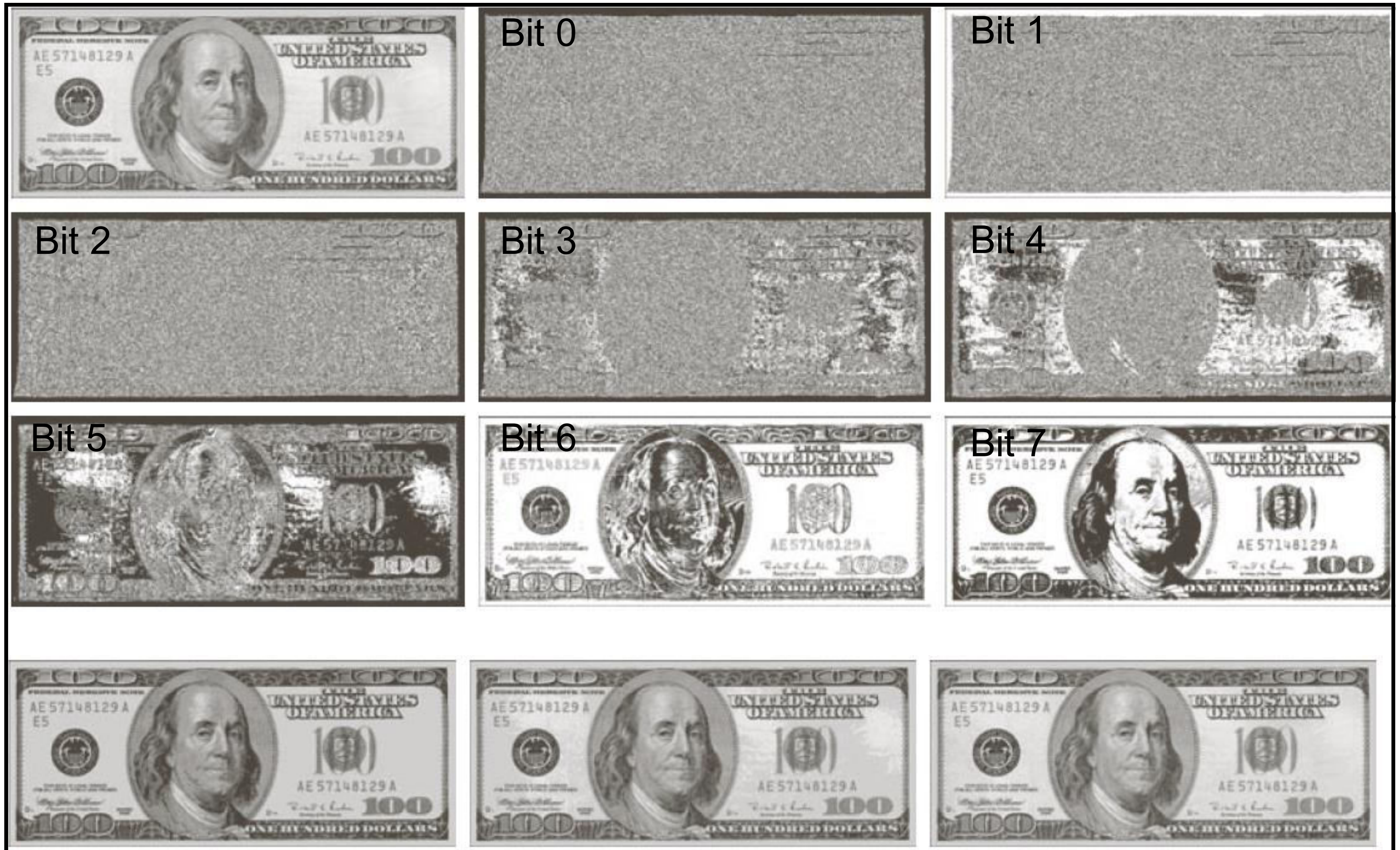
3.2.4 Piecewise-Linear Transformation

- **Bit-plane Slicing**



An 8-bit fractal image. (A fractal is an image generated from mathematical expressions). (Courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA.)


Low
Frequency
High
Bit 7
Bit 6
Bit 5
Bit 4
Bit 3
Bit 2
Bit 1
Bit 0



a b c

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

3.3 Histogram Processing

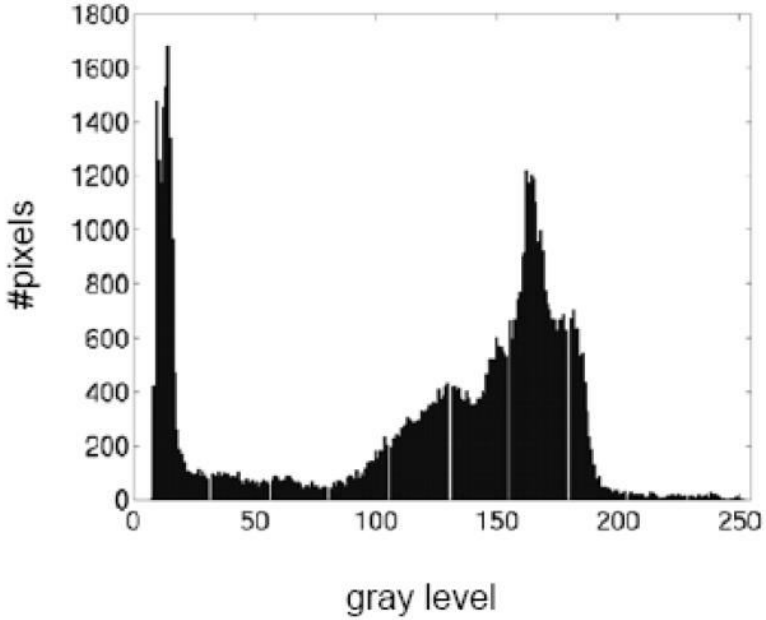
- Distribution of gray-levels can be judged by measuring a histogram. The processes are:
  - For a ***b*-bit** image, initialize 2^b counters with 0
  - Loop over all pixels (x, y)
  - When encountering gray level $f(x, y) = i$, increase counter $\#i$
- The histogram of a digital image with gray-levels in the range [0, L-1] is a discrete function $\boldsymbol{h(r_k) = n_k}$ where r_k is the k-th level and $\boldsymbol{n_k}$ **is the number of pixels having the gray-level** $\boldsymbol{r_k}$.

3.3 Histogram Processing

- Histograms are the basis for numerous spatial domain processing techniques. For example, histogram can provide useful image statistics, and histogram manipulation can be used for image enhancement.
- In addition to providing image statistics, the information inherent in the histogram is quite useful in other image processing applications, such as image segmentation and compression.
- Histograms are simple to calculate and implement in software, thus making them a popular tool for real-time image processing.

3.3 Histogram Processing

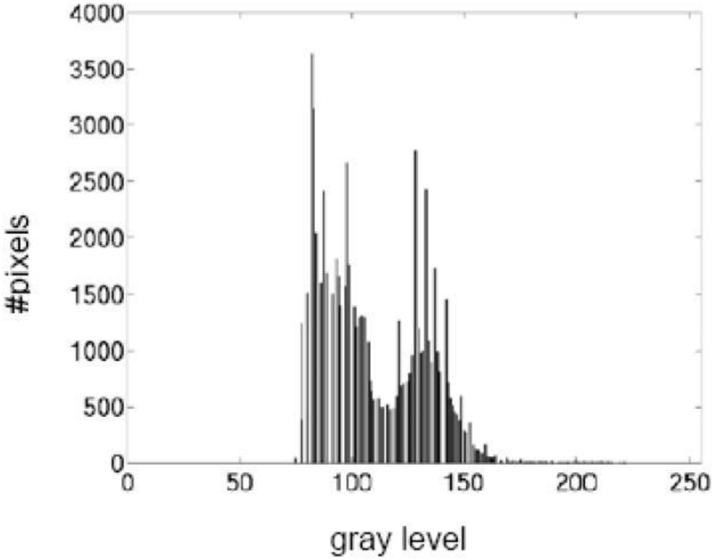
- A normalized histogram $p(r_k) = n_k/n$, n is the total number of pixels in the image.
- Histogram can be interpreted as probability density function (PDF).



Cameraman image

Histogram Example

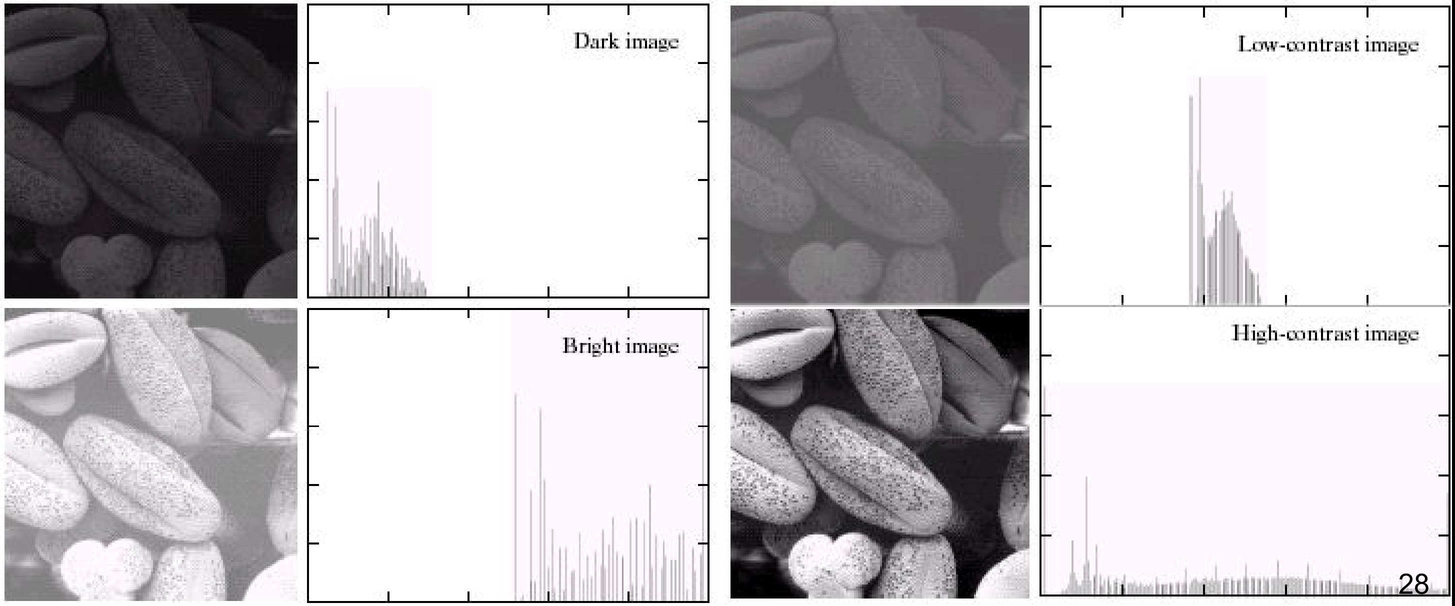
- An image with low contrast has a narrow histogram located typically (no always) toward the middle of the intensity scale.
- For a monochrome image, this implies a dull, washed-out gray look.



Pout image

Four Types of Images with Different Contrast

- The histogram of a high-contrast image covers a wide range of the intensity scale and **the distribution of pixel is not far from uniform** with very few vertical lines being much higher than the others.



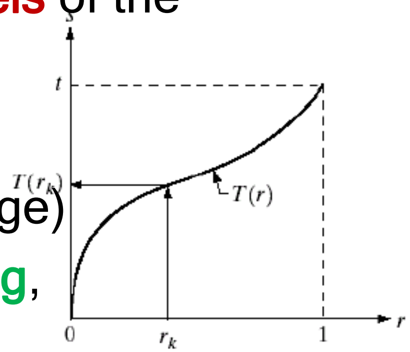
28

3.3.1 Histogram Equalization

- Idea: find a non-linear transformation

$$s = T(r), 0 \leq r \leq 1$$

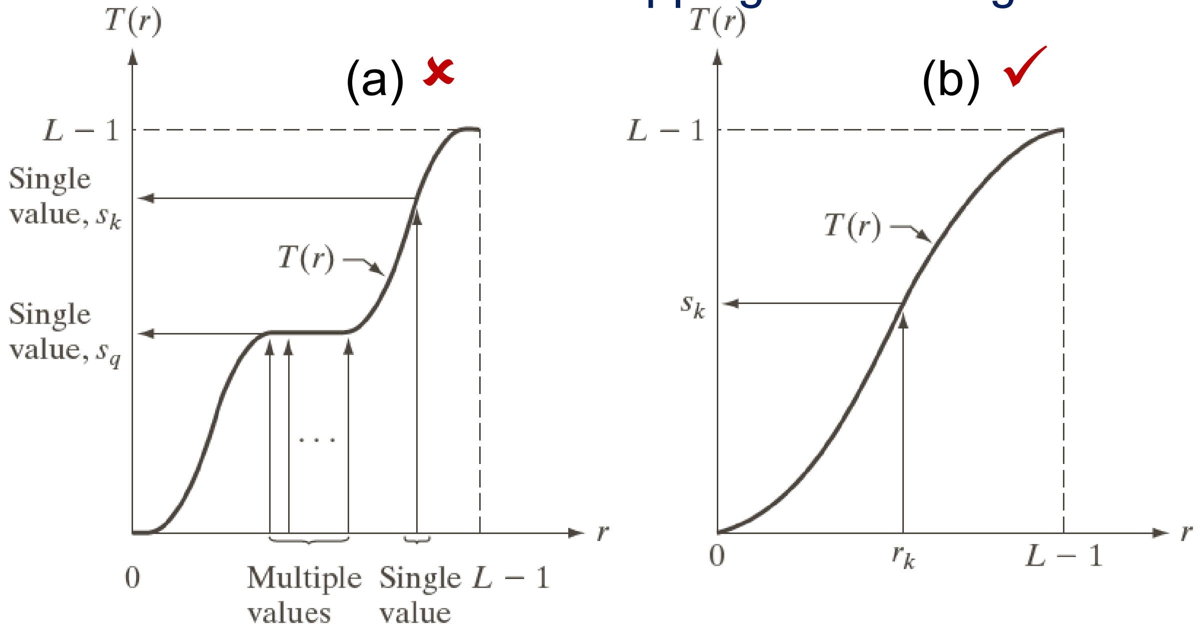
  to be applied to each pixel of the input image $r = f(x, y)$, such that an **uniform distribution of gray levels** of the output image $s = g(x, y)$ is obtained.



- Function T satisfies:
  - (a) $0 \leq r \leq 1, 0 \leq T(r) \leq 1$ (the same range)
  - (b) $T(r)$ is strictly **monotonically increasing**, i.e., there exists $r = T^{-1}(s)$
- The condition (b) guarantees that output intensity values will never be less than corresponding input values, thus preventing artifacts created by reversals of intensities.
- Goal: **PDF $p(s)$=const. over the range 0 ≤ s ≤1**

Monotonically Increasing Function

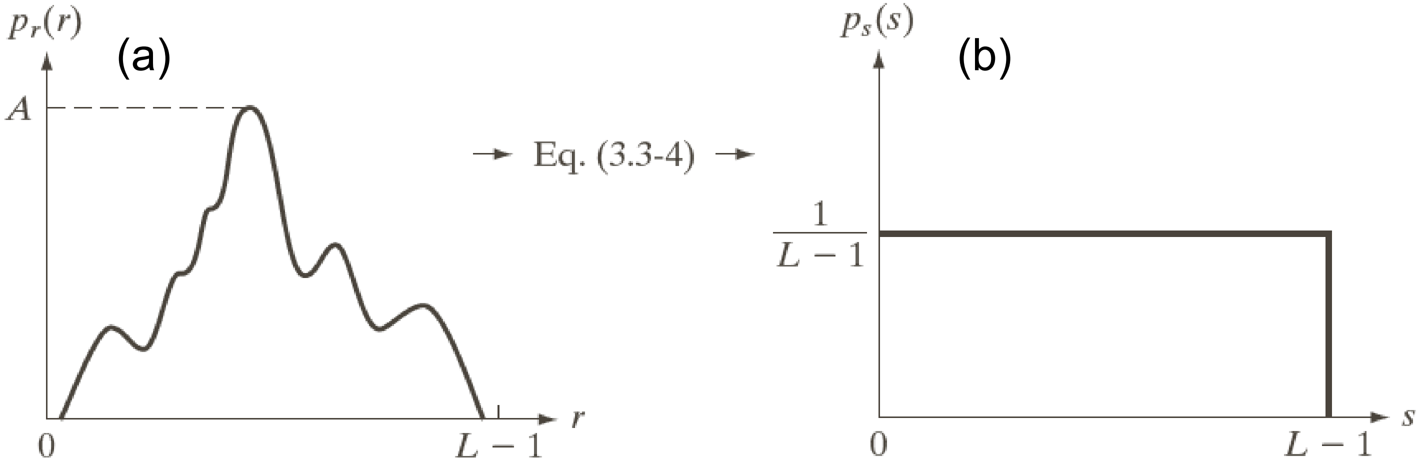
- The function in (a) contain many-to-one mapping that presents a problem if we wanted to recover the values of r uniquely from the mapped values.
- The function in (b) is a strictly monotonic function guarantees that the inverse mapping will be single valued.



30

3.3.1 Histogram Equalization

- (a) An arbitrary PDF.
- (b) The idea result of histogram equalization. The resulting intensities have a uniform PDF.



- The **gray-level** in an image may be viewed as a **random variable**, so we let $p_r(r)$ and $p_s(s)$ denote the probability density functions of random variables r and s.
- If $p_r(r)$ and $T(r)$ are known and $T(r)$ is single-valued and monotonically increased function, then the transformed variable s can be obtained by

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

- Assume the transformation function as

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

$T(r)$ is a **cumulative distribution function** (**CDF**) of the random variable r.

(http://en.wikipedia.org/wiki/Normal_distribution)

Proof: $$s = T(r) = (L-1)\int_0^r p_r(r)dr$$

$$\frac{ds}{dr} = \frac{dT(r)}{dr} = (L-1)\frac{d}{dr}\left[\int_0^r p_r(r)dr\right]$$
$$= (L-1)p_r(r)$$

From Leibniz's rule: The derivative of a definite integral with respect to its upper limit is the integrand evaluated at the limit.
(對一個定積分上限微分就是被積分式在該極限的值)

Substituting this result into Eq. (3.3-3)

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = p_r(r)\left|\frac{1}{(L-1)p_r(r)}\right| = \frac{1}{L-1} \quad , \quad 0 \le s \le 1 \quad (3.3\text{-}6)$$

We recognize the form of $p_s(s)$ in the last line of this equation as a uniform probability density function, which is independent of the form $p_r(r)$.

To fix ideas, consider the following simple example.

$$p_r(r) = \begin{cases} \frac{2r}{(L-1)^2}, for\ 0 \le r \le L-1 \\ 0, \quad otherwise \end{cases}$$

From Eq. (3.3-4),

$$s = T(r) = (L-1)\int_0^r p_r(\omega)d\omega = \frac{2}{L-1}\int_0^r \omega d\omega = \frac{r^2}{L-1}$$

Substitute $p_r(r)$ into Eq. (3.3-6) and using the fact that $s = r^2/L-1$; that is,

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

$$= \frac{2r}{(L-1)^2}\left|\frac{(L-1)}{2r}\right| = \frac{1}{L-1}$$

As expected, the result is a uniform PDF.

3.3.1 Histogram Equalization

- For discrete case, we deal with probabilities (histogram values) and summations instead of probability density functions and integrals.

$$p_r(r_k) = \frac{n_k}{MN}, \text{ for } k = 0,1,\cdots,L-1$$

MN: the total number of pixels in the image r

n_k: is the number of pixels that have intensity r_k

p_r: probability density function of image r

- The discrete version of the transformation function is

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

$$0 \le r_k \le 255,\ 0 \le s_k \le 255$$

- Advantage:
  - **Automatic, without the need of parameters (parameter-free).**

Example -- 3 bits 64 x 64 image

- Suppose that a 3-bit image (L=8) of size 64x64 pixel ($MN = 4096$) has the intensity distribution shown in the table, where the intensity levels are integers in the range [0, 7].

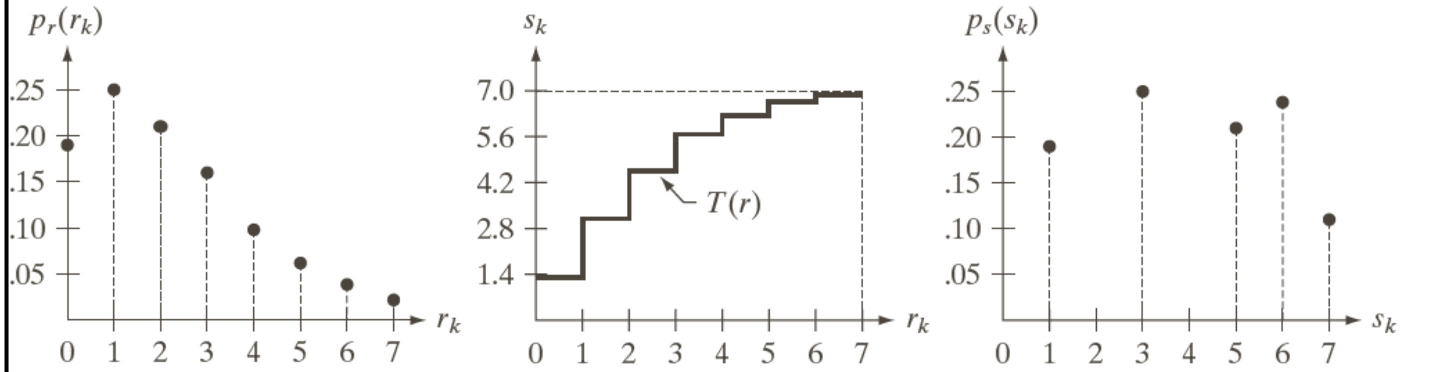
r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02
gray-level	# pixles	PDF

Values of the histogram equalization transformation function are obtained using Eq. (3.3-8). For instance,

$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7p_r(r_0) = 1.33$$

$$s_1 = T(r_1) = 7\sum_{j=0}^{1} p_r(r_j) = 7p_r(r_0)+7p_r(r_1) = 3.08$$

$$s_2 = 4.55,\ s_3 = 5.67,\ s_4 = 6.23,\ s_5 = 6.65, \cdots$$



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.

- These values are floats. Thus, we round them to the nearest integer:
  - $s_0 = 1.33 \rightarrow 1,\ s_1 = 3.08 \rightarrow 3,\ s_2 = 4.55 \rightarrow 5,$
  - $s_3 = 5.67 \rightarrow 6,\ s_4 = 6.23 \rightarrow 6, s_5 = 6.65 \rightarrow 7,$
  - $s_6 = 6.86 \rightarrow 7, s_5 = 7.00 \rightarrow 7$

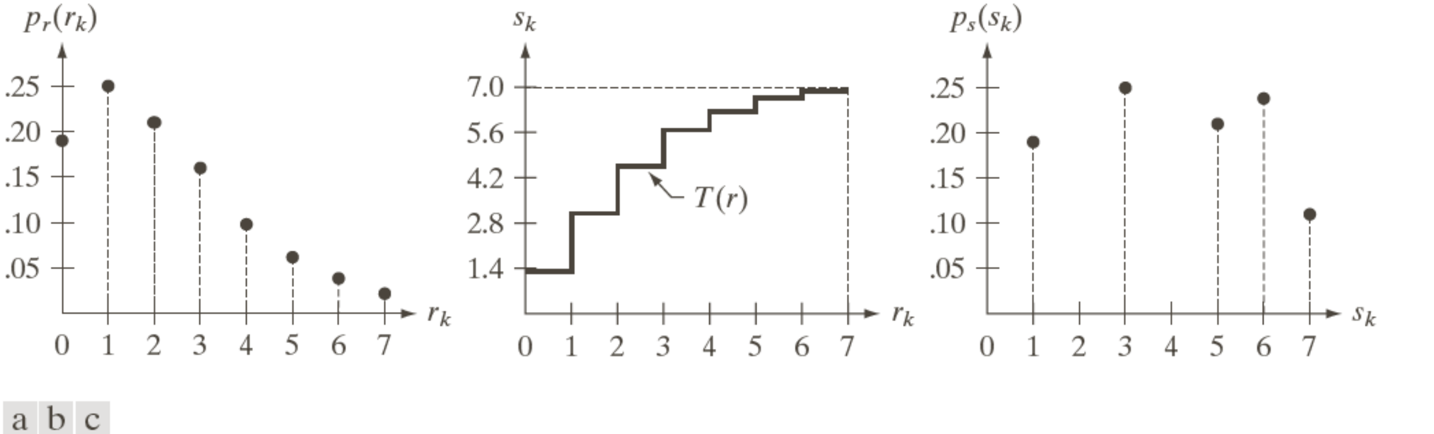


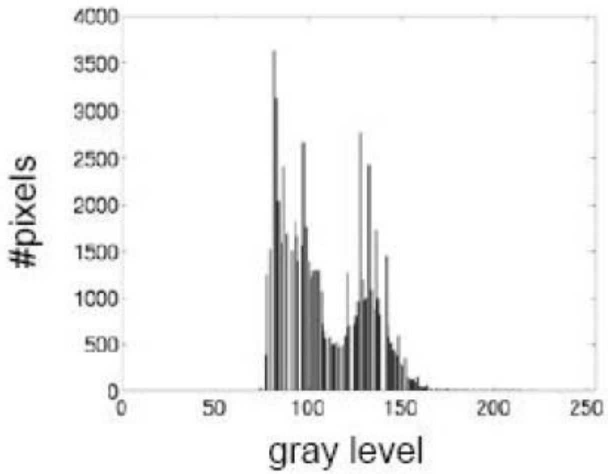
FIGURE 3.19 Illustration of histogram equalization of a 3-bit (8 intensity levels) image. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

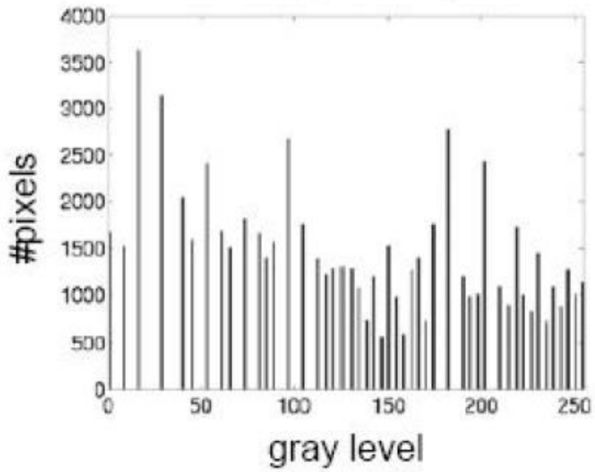
Histogram Equalization Example

Original Image

After histogram equalization

Histogram Equalization Example





Original Image

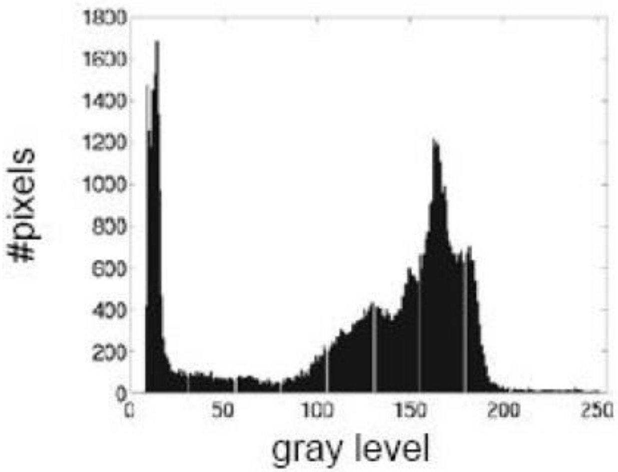
After histogram equalization

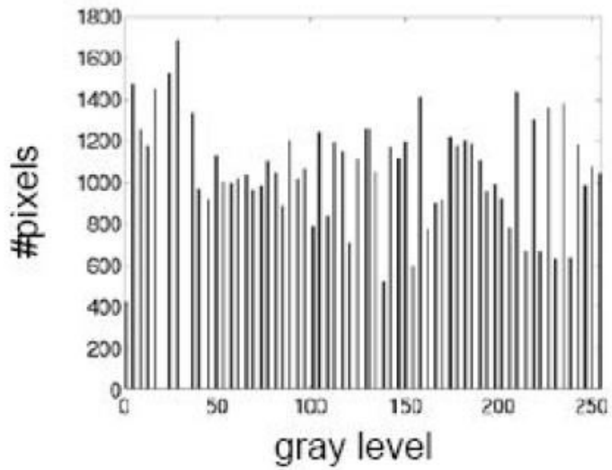
Histogram Equalization Example

Original Image

After histogram equalization

Histogram Equalization Example





Original Image

After histogram equalization

Histogram Equalization Example

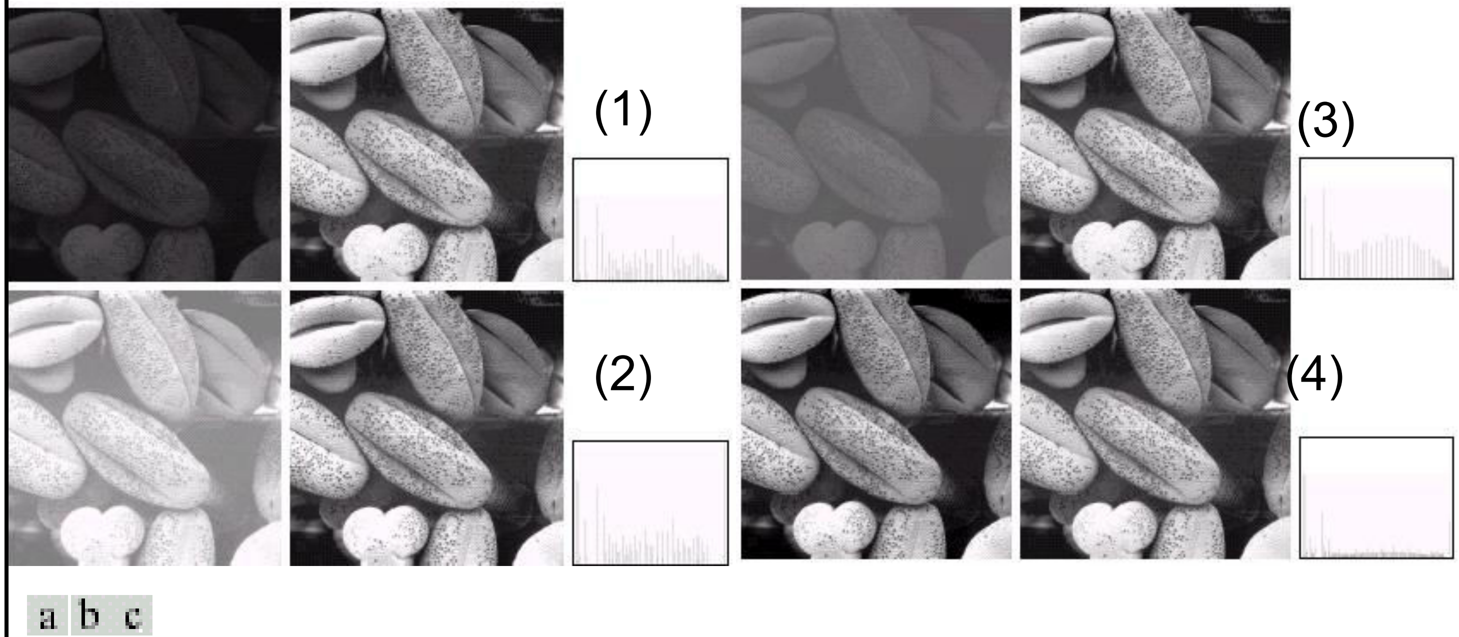
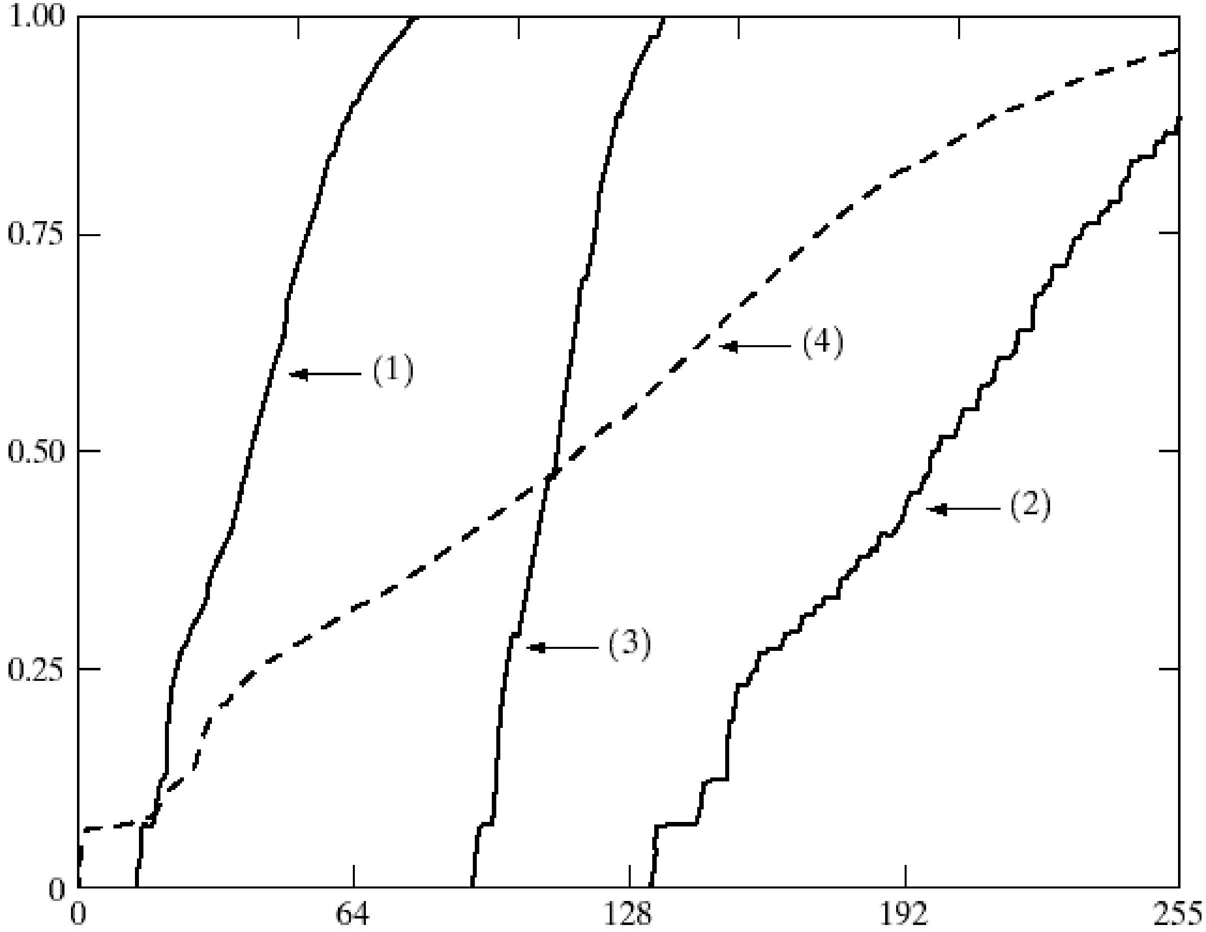


Figure 3.20 (a) Images from Fig. 3.15. (b) Results of histogram equalization. (c) Corresponding histograms.

The Intensity Transformation Functions

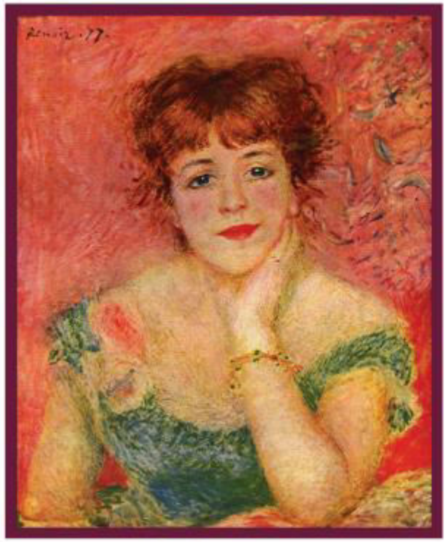


Histogram Equalizing for Color Images

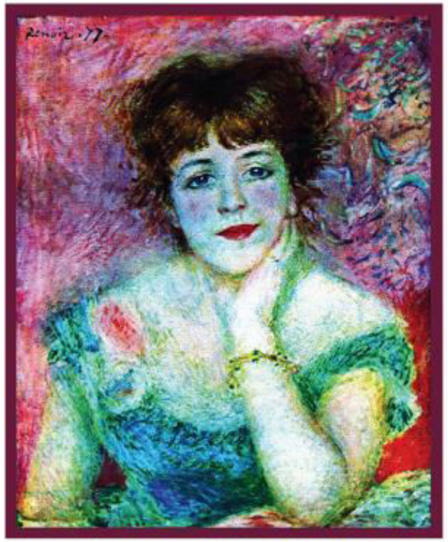
- Histogram equalization can also be done on color images by **performing the grayscale technique on each separate band of the image**.
- However, the colors will likely be dramatically altered. For example, the pixel values in each channel may be changed inconsistently and **the equalization will alter the color patterns present in the source**.
- **Histogram equalization of a color image is best performed on the intensity channel only**, which implies that the equalization should be done on the brightness band of an image using the HSV or other color spaces (will be introduced in Chap. 6).

Histogram Equalizing for Color Images

- Considering color image equalization,
  - (b) equalize each band independently
  - (c) equalize only the intensity



(a) Source.



(b) Equalized RGB.

(c) Equalized intensity.

3.3.1 Histogram Equalization

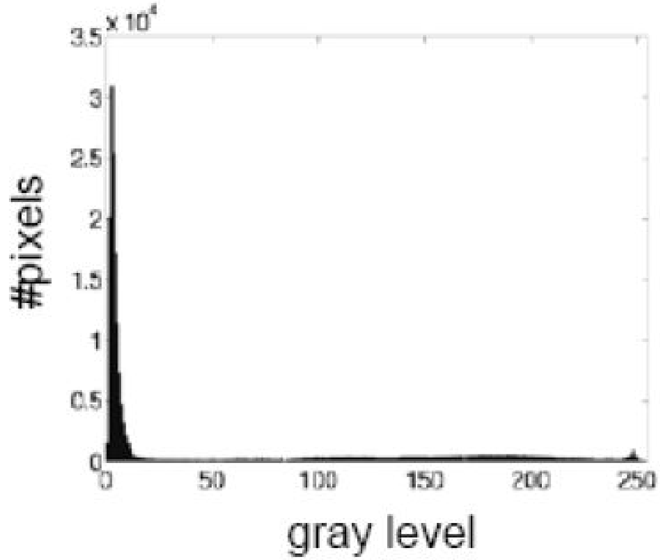
- Histogram equalization automatically determines a transformation function that seeks to produce an output image that has a uniform histogram.
- When automatic enhancement is desired, this is a good approach because the results from this technique are predictable and the method is easy to implement.
- But,……..sometime uniform histogram is not a good solution.

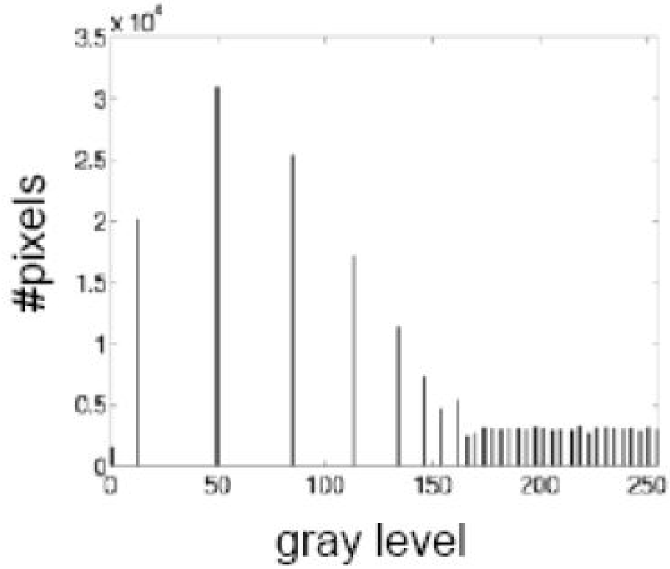
Histogram Equalization Example

Original Image

After histogram equalization

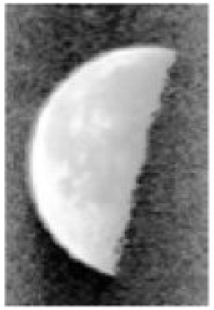
Original image *Moon* . . . after histogram equalization





Original Image

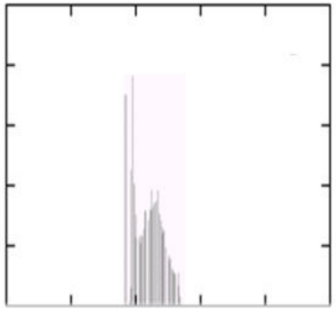
After histogram equalization



3.3.2 Histogram Matching

- It is useful sometime to be able to specify the shape of the histogram that we wish the processed image to have.
- The method used to generate a processed image that has a specified histogram is called **histogram matching** or **histogram specification**.

3.3.2 Histogram Matching

r

$T(r)$

T, G: histogram equalization functions

$z = \boldsymbol{G^{-1}}(\boldsymbol{T}(\boldsymbol{r})) = \boldsymbol{K}(\boldsymbol{r})$

K(*r*) ?

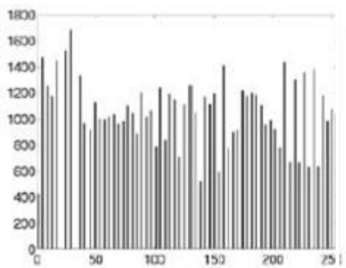
Unknown function

A common domain "s"

$G(z)$

$G^{-1}(s)$

z

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

$$s = \boldsymbol{G}(z) = \int_0^z p_z(t)\,dt$$

3.3.2 Histogram Matching

- **Question**: given an input image with $p_r(r)$, and a **specific output image with $\boldsymbol{p_z(z)}$**, find the transformation function between r and z. **In other words, $\boldsymbol{p_z(z)}$ is the specified probability density function that we wish the output image to have**.
- Let $s = \boldsymbol{T}(r) = \int_0^r p_r(w)\,dw$, where w is a dummy variable of integration.
- Support that a desired random variable z with the property is given

  $s = \boldsymbol{G}(z) = \int_0^z p_z(t)\,dt$, where t is a dummy variable
- From the above equation $G(z) = T(r)$, we have

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

3.3.2 Histogram Matching

For discrete cases:

- From an input histogram $p_r(r_j)$ (input image),

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

  where n_j is the number of pixels that have intensity value r_j, and n represents the total number of pixels.

- From a given histogram $p_z(z_j)$ (user-specified),

$$v_k = G(z_k) = \sum_{j=0}^{k} p_z(z_j) = s_k,\ \mathrm{i} = 1, \cdots, L-1$$

- Finally, we have $T(r_k) = G(z_k) = s_k$, and we find the desired z_k by using the inverse transformation.

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

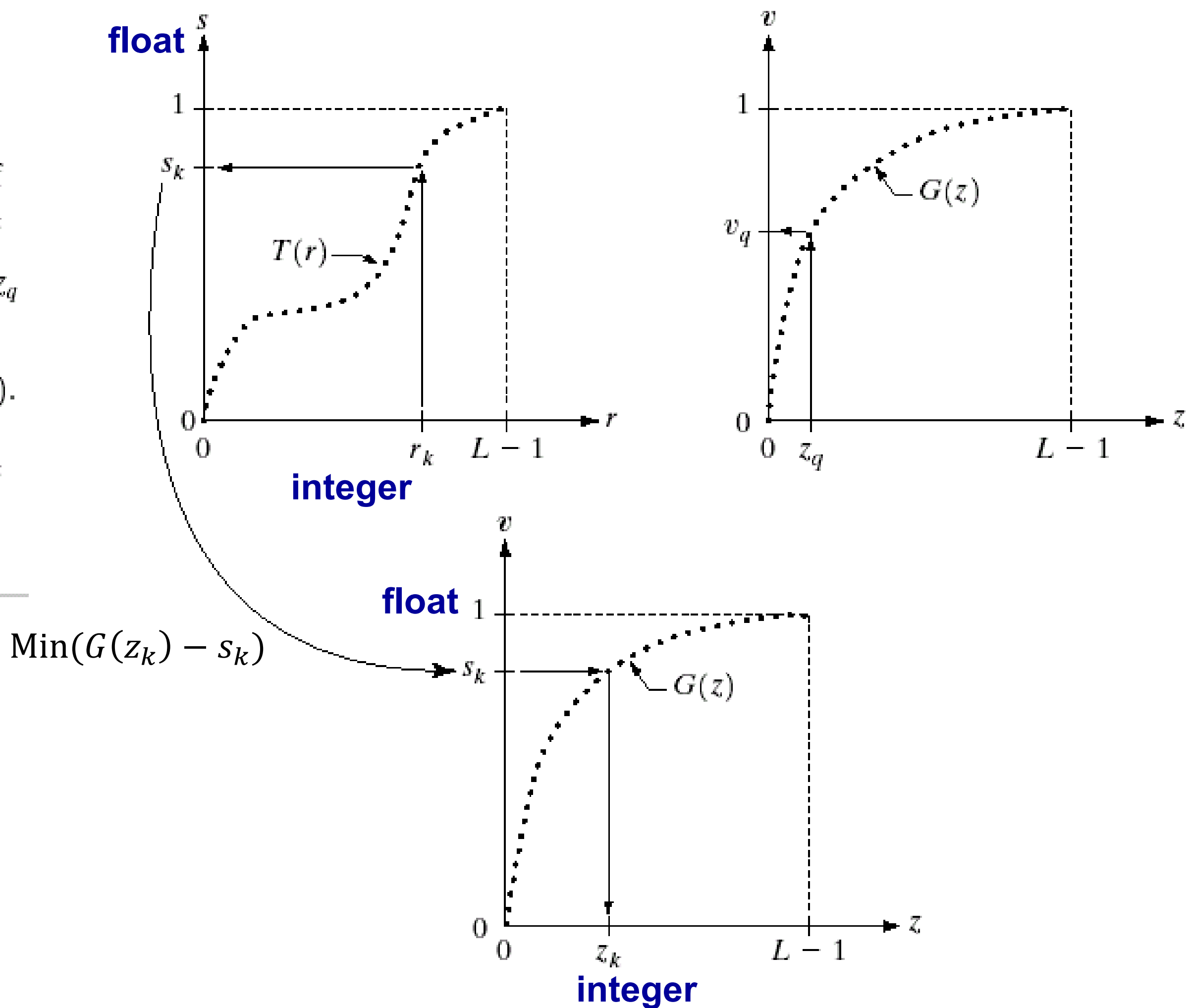
3.3.2 Histogram Matching

Summarize the histogram matching procedure as follows:

1. Obtain the histogram of the input image r.
2. Pre-compute a mapped level s_k for each level r_k. ($\boldsymbol{T}: \boldsymbol{r} \rightarrow \boldsymbol{s}$)
3. Obtain the transformation function G from the user-specified histogram $p(z)$ ($\boldsymbol{G}: \boldsymbol{z} \rightarrow \boldsymbol{s}$)
4. Precompute z_k for each value s_k using an **iterative scheme** as follows:
    1. Find $z_k = G^{-1}(s_k)$. However, **it may not exist such z_k.**
    2. Since we are **dealing with integer**, the closest we can get to satisfying $G(z_k) - s_k = 0$.

a b
c

(a) Graphical interpretation of mapping from r_k to s_k via $T(r)$. (b) Mapping of z_q to its corresponding value v_q via $G(z)$. (c) Inverse mapping from s_k to its corresponding value of z_k.

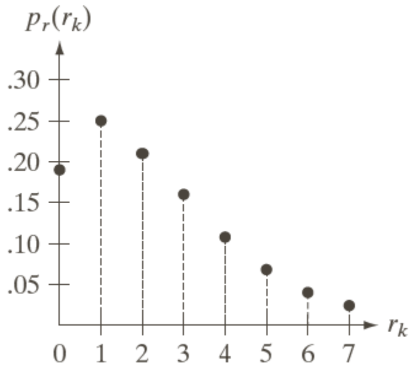


Histogram Matching Example

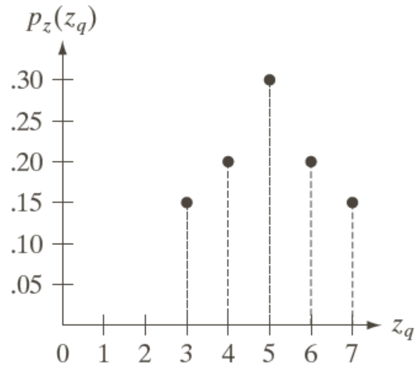
- Consider again the 3-bit 64x64 source image:

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

Source data



Source histogram



Target histogram

- And the desired (target) histogram is:

$$p_z(z_0) = 0.00 \quad p_z(z_1) = 0.00 \quad p_z(z_2) = 0.00 \quad p_z(z_3) = 0.15$$
$$p_z(z_4) = 0.20 \quad p_z(z_5) = 0.30 \quad p_z(z_6) = 0.20 \quad p_z(z_7) = 0.15$$

with $z_0 = 0, z_1 = 1, z_2 = 2, z_3 = 3, z_4 = 4, z_5 = 5, z_6 = 6, z_7 = 7.$

Histogram Matching Example

- The first step is to equalize the input (as before):

  $s_0 = 1, s_1 = 3, s_2 = 5, s_3 = 6, s_4 = 6, s_5 = 7, s_6 = 7, s_7 = 7$

- The next step is to equalize the output:

  $G(z_0) = 0, G(z_1) = 0, G(z_2) = 0, G(z_3) = 1,$

  $G(z_4) = 2, G(z_5) = 5, G(z_6) = 6, G(z_7) = 7$

- Notice that $G(z)$ is **not strictly monotonic**. We must resolve this ambiguity by choosing, e.g. **the smallest value**, for the inverse mapping.

Histogram Matching Example

- Perform inverse mapping: **find the smallest value of $G(z_q)$ that is closest to s_k**:

$s_k = T(r_i)$	$G(z_q)$
$s_0 = 1$	$G(z_0) = 0$
$s_1 = 3$	$G(z_1) = 0$
$s_2 = 5$	$G(z_2) = 0$
$s_3 = 6$	$G(z_3) = 1$
$s_4 = 6$	$G(z_4) = 2$
$s_5 = 7$	$G(z_5) = 5$
$s_6 = 7$	$G(z_6) = 6$
$s_7 = 7$	$G(z_7) = 7$

$$r_k \rightarrow s_k \rightarrow z_k$$

$$\boxed{0 \rightarrow 1 \rightarrow 3}$$

$$1 \rightarrow 3 \rightarrow 4$$

$$2 \rightarrow 5 \rightarrow 5$$

$$3,4 \rightarrow 6 \rightarrow 6$$

$$5,6,7 \rightarrow 7 \rightarrow 7$$

e.g. every pixel with value r_k=0 in the source image would have a value of 3 (z_k) in the histogram-specified image.

58

Histogram Matching Example

- Notice that due to **discretization**, the resulting histogram will rarely be exactly the same as the desired histogram.

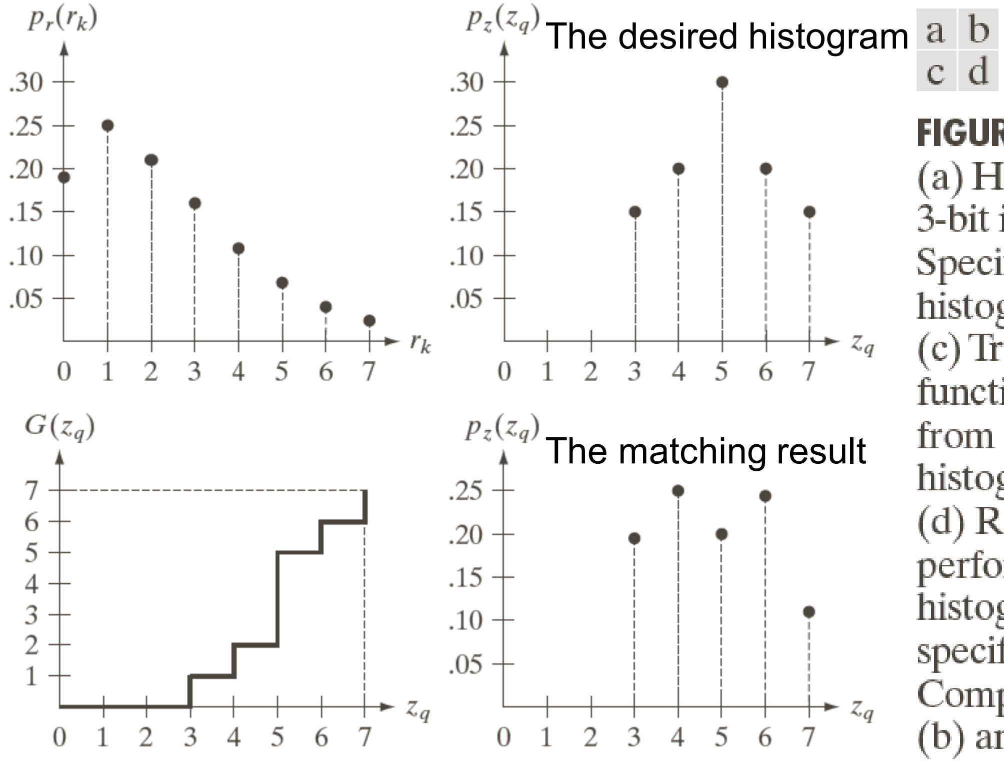
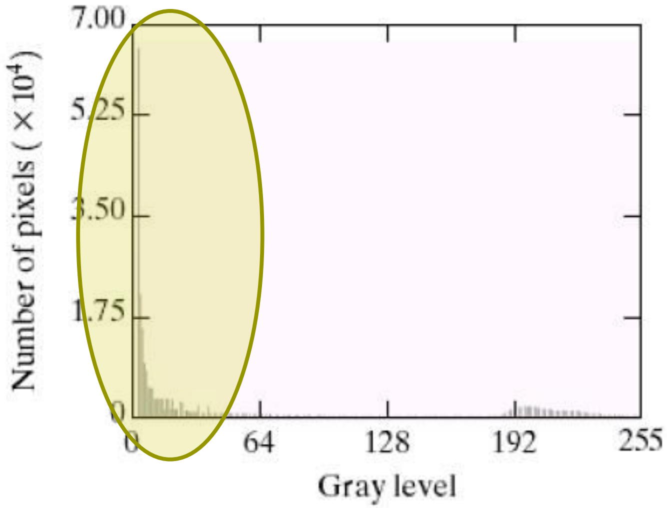


FIGURE 3.22 (a) Histogram of a 3-bit image. (b) Specified histogram. (c) Transformation function obtained from the specified histogram. (d) Result of performing histogram specification. Compare (b) and (d).

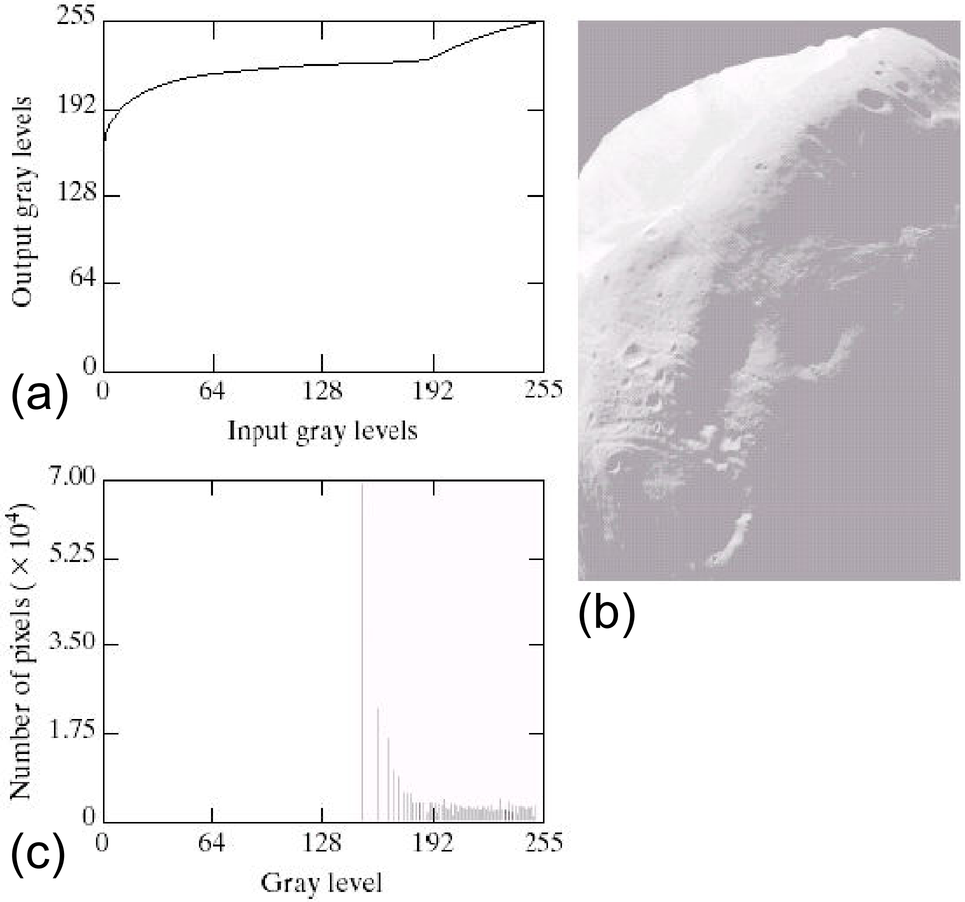
Comparison of Histogram Matching with Histogram Equalization

- This figure shows an image of the Mars moon, taken by NASA's *Mars Global Surveyor*. The image is dominated by large, dark areas, resulting in a histogram characterized by a large concentration of pixels in the dark end of the gray scale.

http://en.wikipedia.org/wiki/Mars_Global_Surveyor



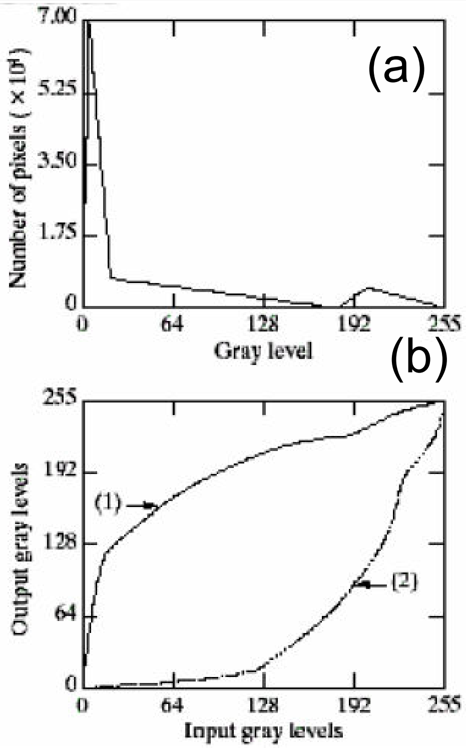
Histogram Equalization

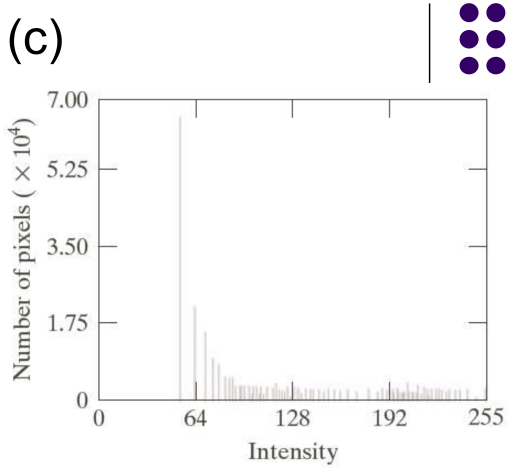


a b
c

Figure 3.24

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





(a) **User-specified histogram**: preserves the general shape of the original histogram, but has a smoother transition of levels in the dark region of the gray scale.

(b) curve 1: histogram equalization of (a) (function G(z)); curve 2: functionG^{-1}(z)

(c) The result using curve 2

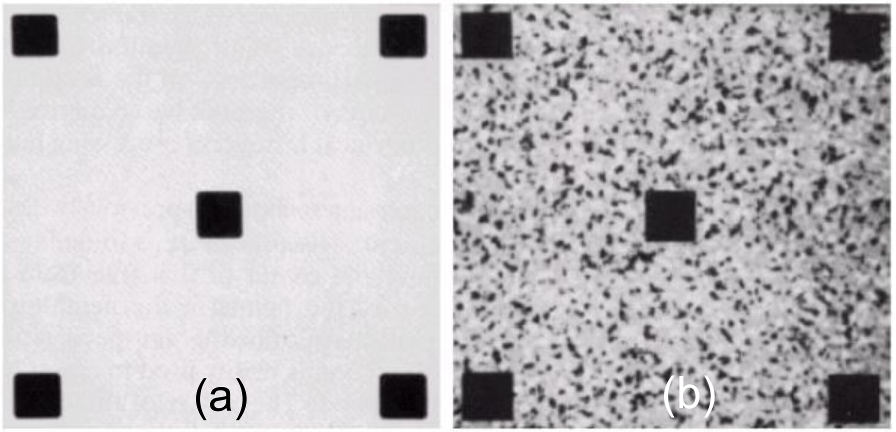
Conclusions

- In general, there are no rules for specifying histograms, and one mush resort to analysis on a **case-by-case basis** for any given enhancement task.
- The histogram matching is, for the most part, **a trial-and-error process**.

3.3.3 Global and Local Histogram Equalization

- The histogram processing methods discussed in the previous sections are global, in the sense that pixels are modified by a transformation function based on the intensity distribution of an entire image.
- Although this global approach is suitable for overall enhancement, there are cases in which 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.
- The solution is to devise transformation functions based on the intensity distribution in a neighborhood of every pixel in the image.

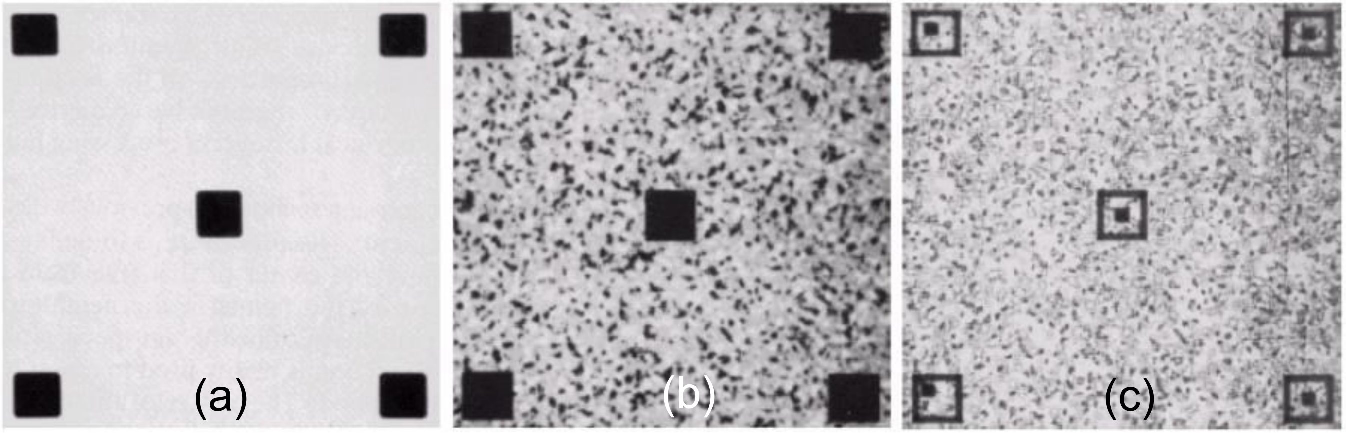
3.3.3 Global and Local Histogram Equalization



(a) Original Image: the image is slightly noisy, but the noise is imperceptible.

(b) **Global histogram equalization**: noise is enhanced. Aside from the noise, however, this case do not reveal any new significant details. There is a very faint hint that squares contain objects.

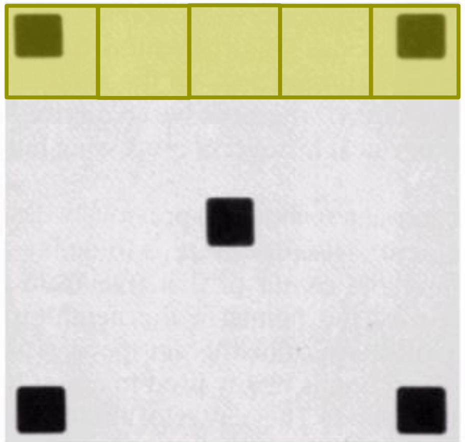
3.3.3 Global and Local Histogram Equalization



(c) **histogram equalization** (3*3mask): significant detail contained within the dark squares. The intensity value of these objects were too close to the intensity of the large squares, and their sizes were too small to influence global histogram equalization enough to show this detail.

3.3.3 Local Histogram Processing – Solution 1

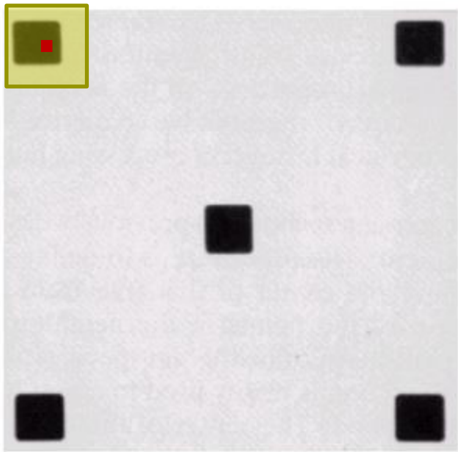
- **Partition the image into several disjoint blocks**. Then perform histogram equalization on each block to locally enhance the image.



- **Problem**: the enhanced image may have chessbox effect.

3.3.3 Local Histogram Processing – Solution 2

- The procedure is to **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 either a histogram equalization or histogram matching function is obtained.
- The function is then used to map the intensity of pixel centered in the neighborhood.



3.3.4 Using Histogram Statistics for Image Enhancement

- **Histogram Statistics**: Statistics obtained directly from an image histogram can be used for image enhancement.
  - r: a **discrete random variable** representing discrete gray-levels in the range [0, L-1]
  - $p(r_i)$: probability of occurrence of grey-level r_i
  - ***n*-th moment** (矩) of r about its mean m is defined as

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

    $$\mu_0(r) = 1,$$

    $$\mu_1(r) = 0,$$

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

  - The second moment is important. We recognize this expression as intensity variance, noted by σ^2.

- Before proceeding, it is useful to work through a simple numerical example to fix ideas. Consider the 2-bit image of size 5x5.

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

- The histogram has the components

$$p(r_0) = 6/25 = 0.24; \; p(r_1) = 7/25 = 0.28;$$
$$p(r_2) = 7/25 = 0.28; \; p(r_3) = 5/25 = 0.20;$$

- The average value of the intensities is

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

- Double check this result

$$m = \frac{1}{25} \sum_{x=0}^{4} \sum_{y=0}^{4} f(x, y)$$

3.3.4 Using Histogram Statistics for Image Enhancement

- We consider two uses of the mean and variance for enhancement purposes.
  - The global mean and variance are computed over an entire image and are useful for gross adjustments in overall intensity and contrast.
  - The local mean and variance are used as the basis for making changes that depend on image characteristics in a neighborhood about each pixel in an image.

3.3.4 Using Histogram Statistics for Image Enhancement

- Local Histogram Statistics
  - $S_{x,y}$: a neighborhood (subimage) centered at (x, y) of a specified size
  - **Local mean** $m_{x,y}$ is a measure of average gray level in neighborhood $S_{x,y}$

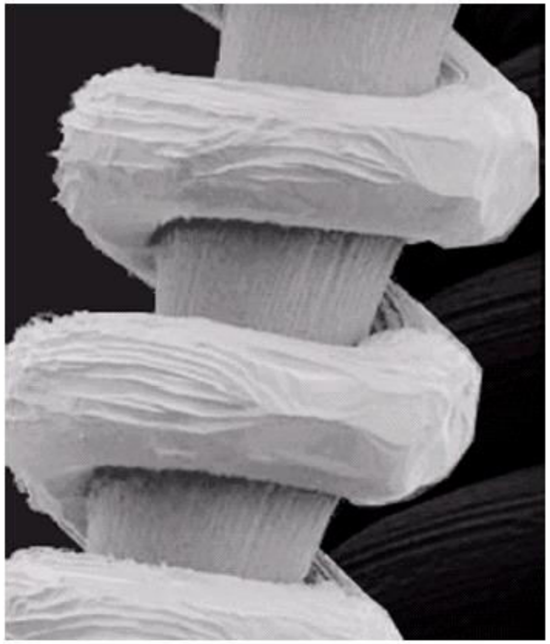
$$m_{S_{x,y}} = \sum_{i=0}^{L-1} r_i \, p_{S_{xy}}(r_i)$$

  - **Local variance** is a measure of gray level variance in neighborhood $S_{x,y}$

$$\sigma^2_{S_{x,y}} = \sum_{i=0}^{L-1} \left(r_i - m_{S_{x,y}}\right)^2 p_{S_{xy}}(r_i)$$

3.3.4 Using Histogram Statistics for Image Enhancement

- This tested image shows an SEM (scanning electron microscope) image of a tungsten filament (鎢燈絲).
- The filament in the center is quite clear. There is another filament structure on the dark side, but it is almost imperceptible.
- The problem is to **enhance dark areas** while **leaving the light area as unchanged as possible**.



3.3.4 Using Histogram Statistics for Image Enhancement

- $f(x, y)$: original image, $g(x, y)$: enhanced image

$$g(x,y) = \begin{cases} E \cdot f(x,y) & \text{if } m_{S_{x,y}} \le k_0 M_G \quad \text{and} \quad k_1 D_G \le \sigma_{S_{x,y}} \le k_2 D_G \\ f(x,y) & \text{otherwise} \end{cases}$$

M_G : global mean

D_G : global standard deviation

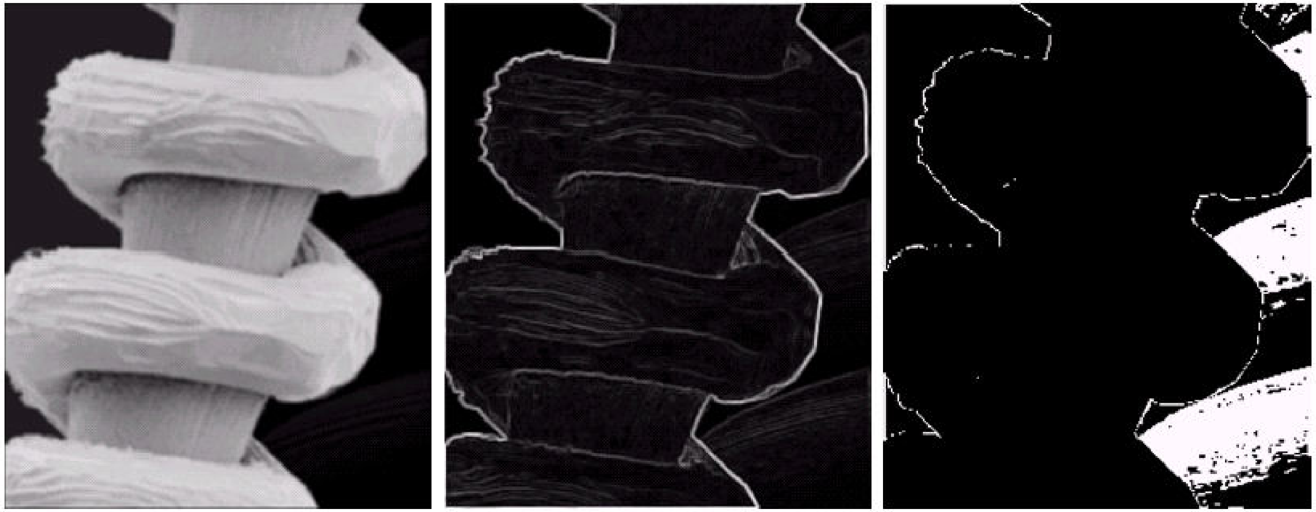
$m_{S_{x,y}}$: local mean

$\sigma_{S_{x,y}}$: local standard deviation

k_0, k_1, k_2: positive constant (<1.0);

E: positive constant

Example of Local Histogram Statistics



(a) Local mean image (3x3 block) (c)The detected areas

(b) Local standard deviation image (3x3 block)

$$m_{S_{x,y}} = \sum_{(s,t)\in S_{x,y}} r_{s,t}\, p(r_{s,t}) \qquad \sigma^2_{S_{x,y}} = \sum_{(s,t)\in S_{x,y}} (r_{s,t} - m_{S_{x,y}})^2\, p(r_{s,t})$$

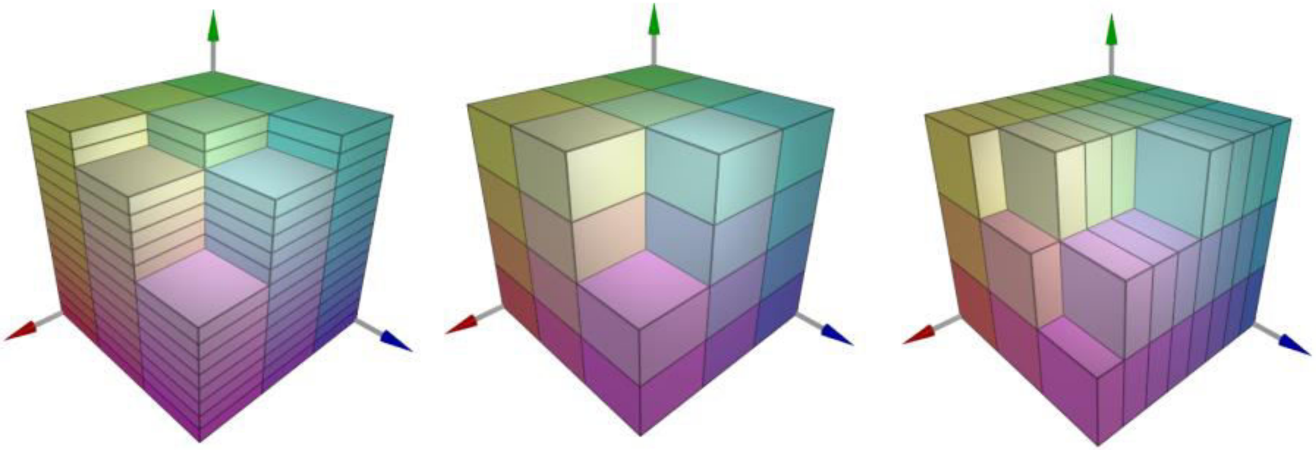
Result of Local Enhancement

76

Supplement
-- Color Histogram

- A color histogram is a 3D entity where each pixel of an image (rather than each sample) is placed into a bin.
- The color space is divided into **volumetric bins** and each of which represents a range of colors.
- Each axis of the color space may be divided independently of the others. This allows the axes to have different resolutions.

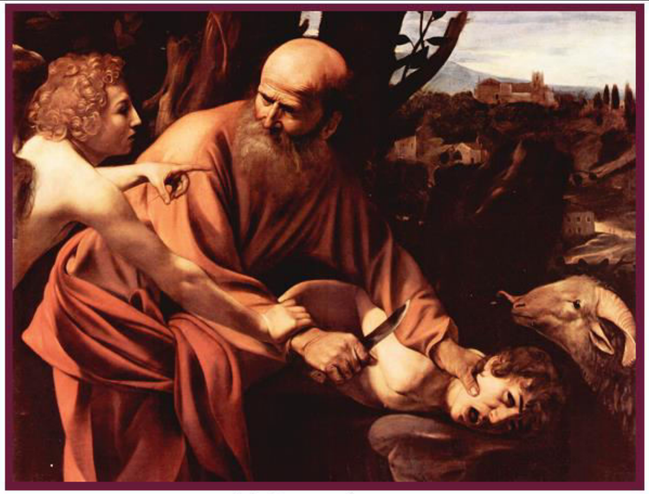
Supplement
-- Color Histogram



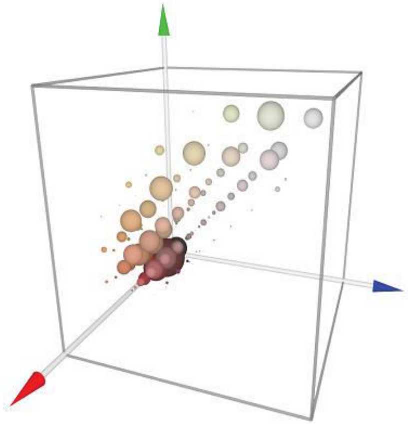
3x15x3 resolution 3x4x3 resolution 8x3x3 resolution

Consider the resolution of various color histogram binings in RGB space. The resolution of each axis may be set independently of the others.

Example

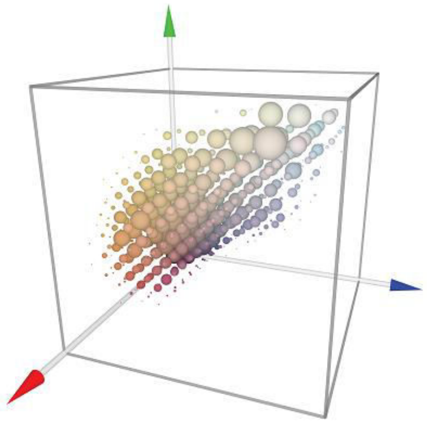


(a) Source image.



(b) RGB color histogram.

(c) Source image.



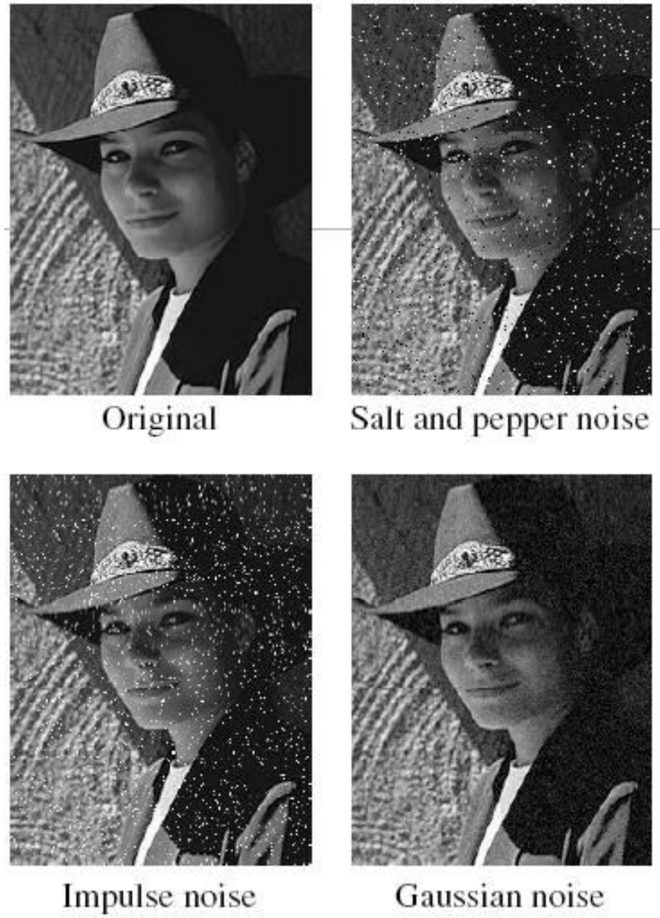
(d) RGB color histogram.

3.4 Fundamentals of Spatial Filtering

- The name **filter** is borrowed from frequency domain processing, where filtering refers to accepting (passing) or rejecting certain frequency components.
- In spatial domain, filters are operates on raw image data by using **a small neighborhood** in an image such as **3*3, 5*5, 7*7** and moving sequentially across and down the entire image
- Returns a result based on a **linear** or **non-linear** operation
- There are three types of filters:
  - **Smooth(Mean) filters**
  - **Median filters** (non-linear)
  - **Enhancement filters**

Noise Types

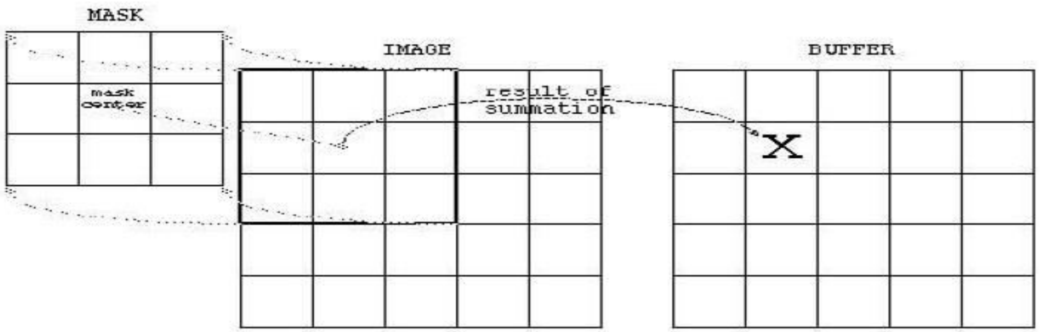
- Common types of noise:
  - **Salt and pepper noise**: random black and white pixels
  - **Impulse noise**: random white pixels
  - **Gaussian noise**: variations in intensity drawn from a Gaussian (normal) distribution



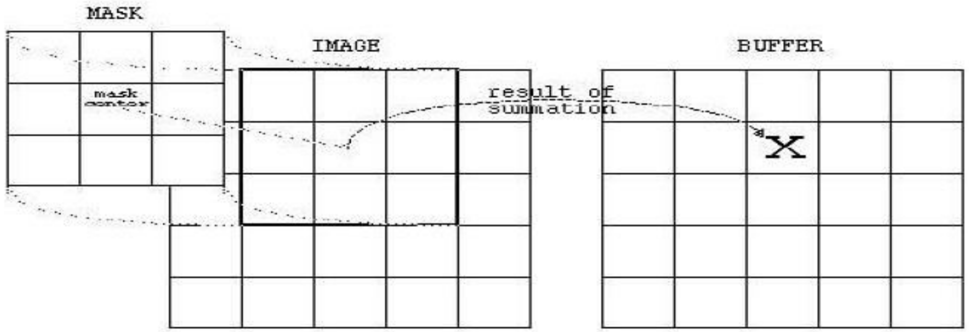
Linear Filter

- Many spatial filters are linear filters (filters implemented with a **convolution mask**; the result is a weighted sum of all values of pixels in a neighborhood)
- Mask coefficients: coefficients in the mask tend to effect the image in the following ways:
  - Coefficients are positive: **Blur the image**
  - Coefficients are alternating positive and negative: **Sharpen the image**
  - Coefficients sum to 1: **Brightness retained**
  - Coefficients sum to 0: **Dark image**

Image Convolution



a) Overlay the convolution mask in upper left corner of the image. Multiply coincident terms, sum, put result into the image buffer at the location that corresponds to the mask's current center, which is (r,c) = (1,1).



b) Move the mask one pixel to the right, multiply coincident terms, sum, and place the new result into the buffer at the location that corresponds to the new center location of the convolution mask, now at (r,c) = (1,2). Continue to the end of the row.

Basics of Spatial Filtering

- Spatial filtering: using a **filter kernel** (**convolution mask**, **template**, **window**) $w(x, y)$ which is a subimage to operate on the image $f(x, y)$.
- For **3x3 mask**, the response R of the pixel (x, y) after filtering is

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

The output image $g(x, y)$ is:

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

Convolution!

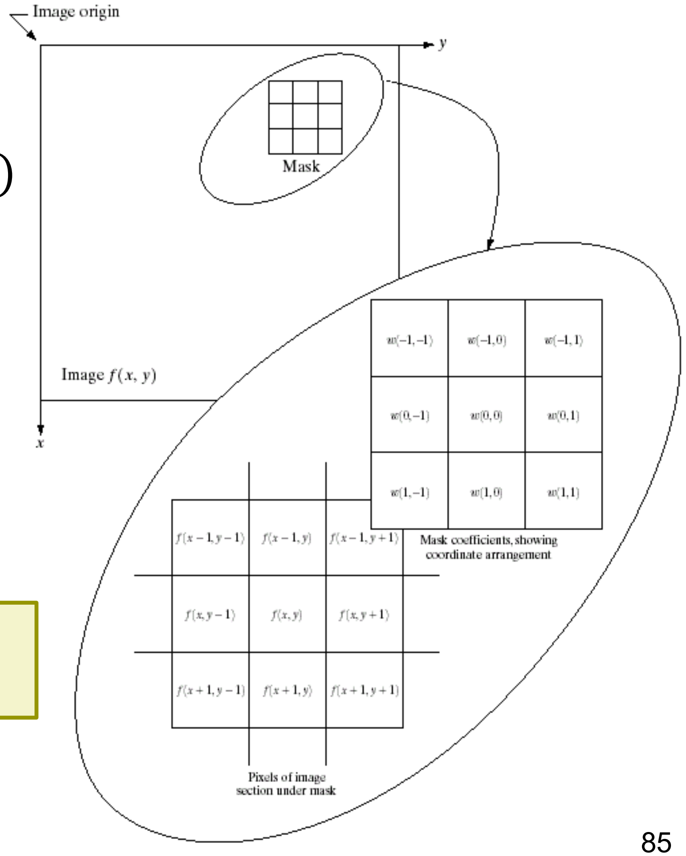
- Convolution demo (wiki): https://en.wikipedia.org/wiki/Convolution

Basics of Spatial Filtering

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

Ways of border handling:

- Limit the range of convolution
- Image padding
  - Zero padding
  - Replicating rows and columns



3.4.2 Spatial Correlation and Convolution

- There are two closely related concepts that must to be understood clearly: correlation and convolution.
- Correlation is the process of moving a filter mask over the image and computing the sum of products at each location.
- The mechanics of convolution are the same, except that the filter is first rotated by 180 degrees.

Correlation		Convolution	
(a)	Origin f `0 0 0 1 0 0 0 0` w `1 2 3 2 8`	Origin f `0 0 0 1 0 0 0 0` w rotated 180° `8 2 3 2 1`	(i)

(b)	`0 0 0 1 0 0 0 0`
	`1 2 3 2 8`
	Starting position alignment
		`0 0 0 1 0 0 0 0`
		`8 2 3 2 1`
			(j)

(c)	Zero padding
	`0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0`
	`1 2 3 2 8`
		`0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0`
		`8 2 3 2 1`
			(k)

(d)	`0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0`
	`1 2 3 2 8`
	Position after one shift
		`0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0`
		`8 2 3 2 1`
			(l)

(e)	`0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0`
	`1 2 3 2 8`
	Position after four shifts
		`0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0`
		`8 2 3 2 1`
			(m)

(f)	`0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0`
	`1 2 3 2 8`
	Final position
		`0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0`
		`8 2 3 2 1`
			(n)

(g)	Full correlation result `0 0 0 8 2 3 2 1 0 0 0 0`	Full convolution result `0 0 0 1 2 3 2 8 0 0 0 0`	(o)
(h)	Cropped correlation result `0 8 2 3 2 1 0 0`	Cropped convolution result `0 1 2 3 2 8 0 0`	(p)

3.4.2 Spatial Correlation and Convolution

- Summarizing the preceding discussion in equation form, we have that the correlation of a filter $w(x, y)$ of size $m \times n$ with an image $f(x, y)$ is given by

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

- The convolution of $w(x, y)$ and $f(x, y)$ is given by

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

- The minus signs on the right flip f, that is, rotate it by 180 degrees.

Origin $f(x, y)$

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

$w(x, y)$

1	2	3
4	5	6
7	8	9

(a)

Padded f

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

(b)

Initial position for w

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

(c)

Full correlation result

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

(d)

Cropped correlation result

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

(e)

Rotated w

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

(f)

Full convolution result

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

(g)

Cropped convolution result

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

(h)

3.4.2 Spatial Correlation and Convolution

- Convolution is a cornerstone of linear system theory. The property that the convolution of a function with a unit impulse copies the function at the location of the impulse plays a central role in a number of important applications.
- You are likely to encounter the terms, *convolution filter*, *convolution mask* or *convolution kernel* in the image processing literature. As a rule, these terms are used to denote a spatial filter and **not necessarily that the filter will be used for true convolution**.

3.5 Mean (Average) Filter

- **Tend to blur the image**
- **Adds a softer look to the image**
- Convolution mask used is 3x3:

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

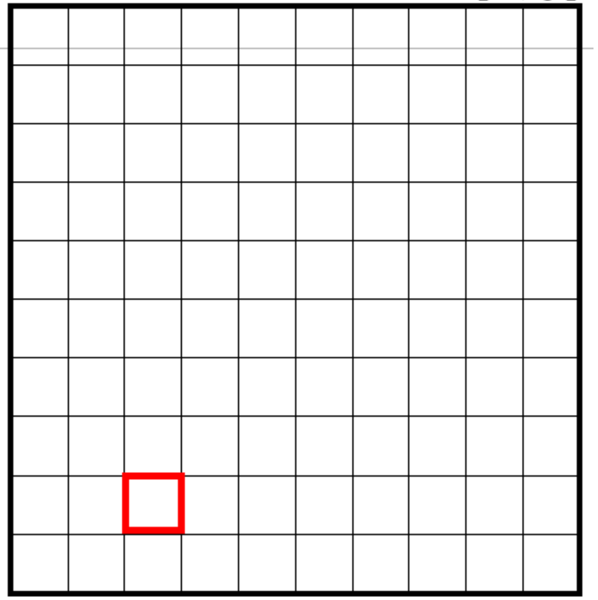
$$g(x, y) = \frac{\sum_{s=-a}^{a} \sum_{s=-b}^{b} w(s,t) f(x+s, y+t)}{\sum_{s=-a}^{a} \sum_{s=-b}^{b} w(s,t)}$$

Mean Filter Example

$I(x, y)$

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

$\tilde{I}(x, y)$



- Replace each pixel with an average of the pixels in the $k \times k$ mask (the red box) around it.
- In 3 × 3 case: $\tilde{I}(x, y) = \frac{1}{9}\sum_{u=0}^{2}\sum_{v=0}^{2} I(x+u-1, y+v-1)$

92

Mean Filter Example

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

$I(x, y)$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

$\tilde{I}(x, y)$

- What about border pixels?

Some options

- Skip them - image gets smaller each time a filter is applied
- Pad the image with more rows and columns on the top, bottom, left, and right.
  - Option1: copy the border pixels. Add $I(2,3) = 0$ to $\tilde{I}(10,3)$ in the case below.
  - Option2: reflect the image about the border. Add $I(9,3) = 90$ to $\tilde{I}(10,3)$.

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

$F[x, y]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

$G[x, y]$

94

Mean Filter Example

Original image

Mean filtered image

Effect of filter size

- What happens if we use a larger mean filter?

  such as 5×5? 7×7?

Gaussian noise | Salt and pepper noise

3x3

5x5

7x7

Gaussian Filter (Weighted Average Filtering)

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

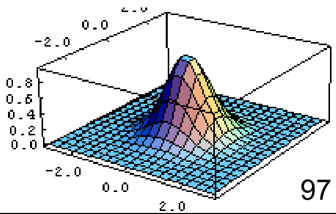
$F[x, y]$

$\frac{1}{16}$

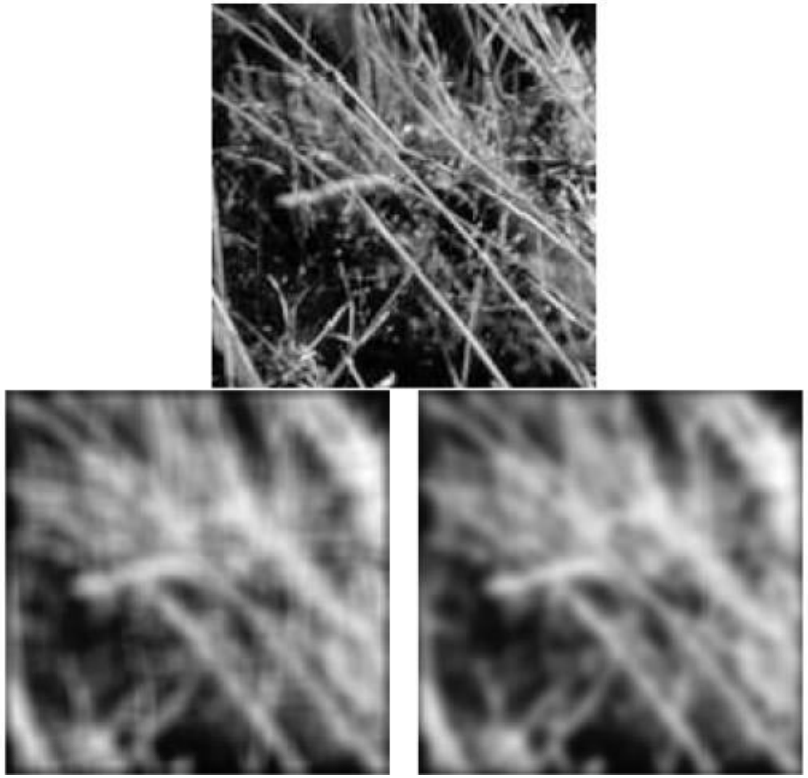
1	2	1
2	4	2
1	2	1

$H[u, v]$

- This filter H is a good approximation to $h(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{\sigma^2}}$
- Properties of Gaussian
  - more weight to the center
  - good model of blurring in optical systems
  - σ corresponds to width of the Gaussian



Comparison of mean and Gaussian filters



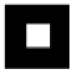
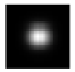
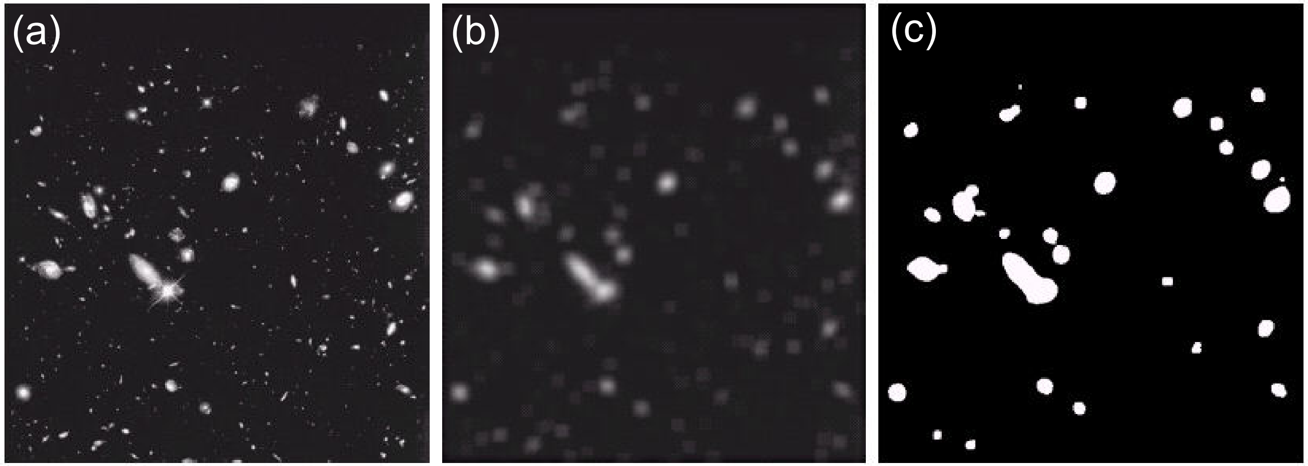
Filter kernel function  

Image Smoothing Example

- An image from the Hubble space telescope in orbit around the Earth
- Application: spatial averaging is to blur an image for the purpose of getting a gross representation of objects of interest, such that the intensity of **smaller objects blends with the background** and **larger objects become "blob-like"** and easy to detect.

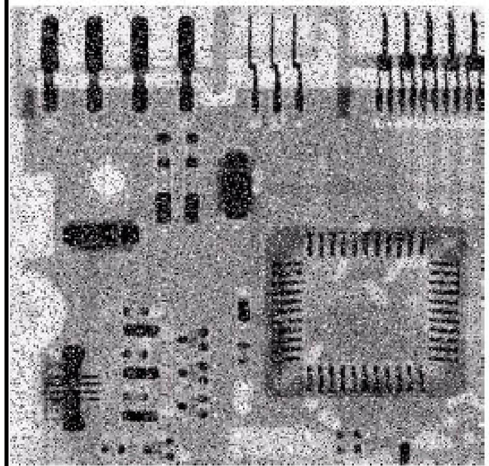


(a) Image from the Hubble Space Telescope. (b) Image processed by a 15×15 averaging mask. (c) Result of thresholding (b).

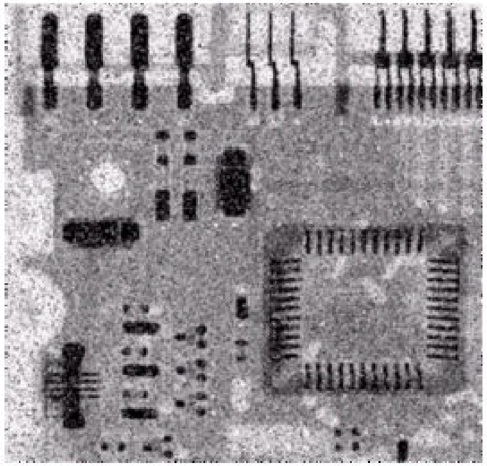
3.5.2 Non-linear (Order-statistic) Spatial Smoothing Filters

- **Median/Max/Min** Filter
  - The response is based on **ordering** (**ranking**) the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result (median, max or min value)
- For certain noise, such as salt-and-pepper noise, median filter is effective.
- For instance, suppose a 3x3 neighborhood has values (100, 20, 20, 20, 15, 20, 20, 25, 10). **These values are sorted as** (10, 15, 20, 20, 20, 20, 20, 25, 100), which results in **a min. of 10**, **a max. of 100**, and **a median of 20**.

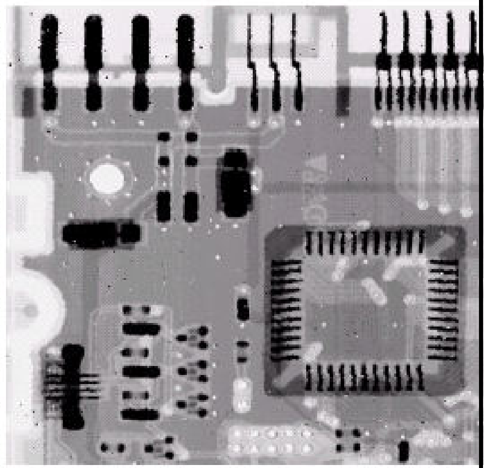
Comparison of average filter and median filter



(a)



(b)



(c)

(a)X-ray image of circuit board corrupted by salt-and-pepper noise

(b)Result of 3x3 averaging filter

(c)Result of 3x3 median filter

Smooth Filter Comparison

3.5.2 Non-linear (Order-statistic) Spatial Smoothing Filters

- Although the median filter is by far the most useful order-statistic filter in image processing, it is by no means the only one.
- The median represents the 50th percentile of a ranked set of numbers, but recall from basic statistics that ranking lends itself to many other possibilities.
- For instance,
  - using the 100th percentile results in the so-called **max filter**, which is useful for finding the brightest points.
  - The 0th percentile filter is the **min filter**, used for the opposite purpose.

3.6 Spatial Sharpening Filters

- The principal objective of sharpening is to highlight transitions in intensity.
- **Image averaging** – **low-pass filtering** –**spatial integration**
- **Image sharpening** – **high-pass filtering** –**spatial differentiation**
  - It enhances the **edges** and the other **discontinuities**
  - **First order difference**

$$\partial f/\partial x = f(x+1, y) - f(x, y)$$
$$\partial f/\partial y = f(x, y+1) - f(x, y)$$

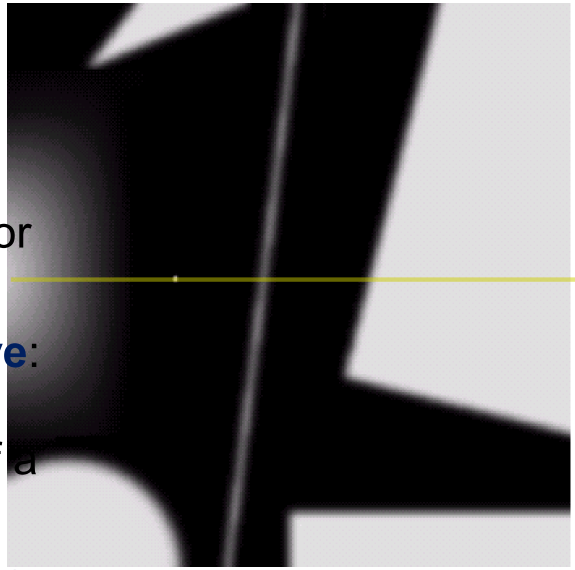
  - **Second order difference**

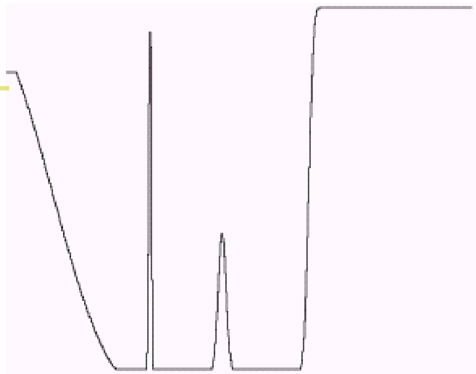
$$\begin{aligned}\partial^2 f/\partial x^2 &= \partial f(x,y)/\partial x - \partial f(x-1,y)/\partial x \\ &= f(x+1,y) + f(x-1,y) - 2f(x,y)\end{aligned}$$
$$\begin{aligned}\partial^2 f/\partial y^2 &= \partial f(x,y)/\partial y - \partial f(x-1,y)/\partial y \\ &= f(x,y+1) + f(x,y-1) - 2f(x,y)\end{aligned}$$

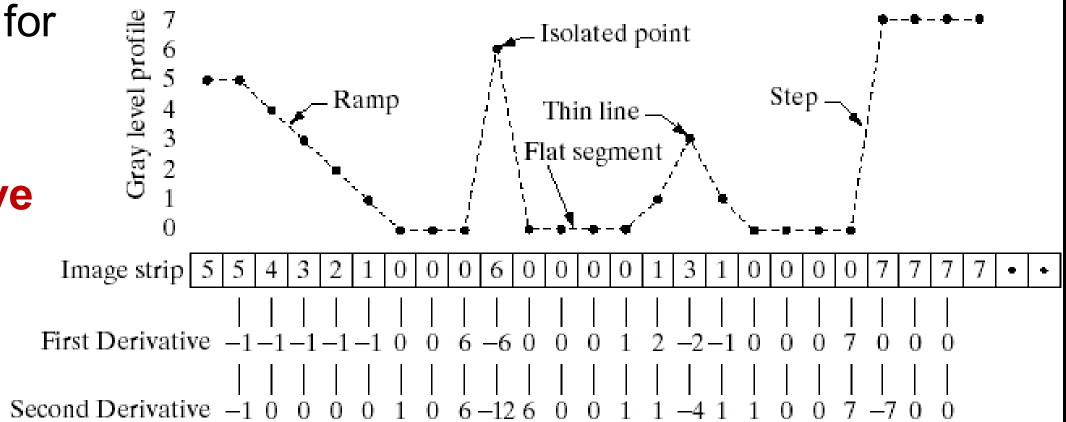
104

Note:

- **First derivative**: strong response for a step.
- **Second derivative**: double responses on the changes of a step.
- **Second derivative**: strong responses for thin lines and isolated points
- **Both are sensitive to noises**







3.6 Spatial Sharpening Filters

- Many image sharpening algorithms consist of three general steps:
  1. Extract high frequency information.
  2. Combine the high frequency image with the original image to emphasize image detail.
  3. Maximize image contrast via histogram manipulation.

Second-order derivative for enhancement - Laplacian

- **Isotropic filter**: response is independent of the direction of the discontinuities (**rotation invariant**, in the sense that rotating the image and then applying the filter gives the same result as applying the filter to the image then rotating the result).
- **Laplacian** of a function f (image):

$$\nabla^2 f = \partial^2 f / \partial x^2 + \partial^2 f / \partial y^2$$

$$\partial^2 f / \partial x^2 = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\partial^2 f / \partial y^2 = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

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

Laplacian Mask

- Mask (a) and (b) are symmetric along the x-axis and y-axis, meaning that these masks yields isotropic results in increments of 90^0.
- In the mask (c) and (d), because each diagonal term also contains a $-2f(x,y)$ term, the total subtracted from the difference terms now would be $-8f(x,y)$.
- These two masks are rotation invariant in increments of 45^0

(a)

0	1	0
1	−4	1
0	1	0

(c)

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

(b)

−1	−1	−1
−1	8	−1
−1	−1	−1

(d)

Image Enhancement Using Laplacian Filter

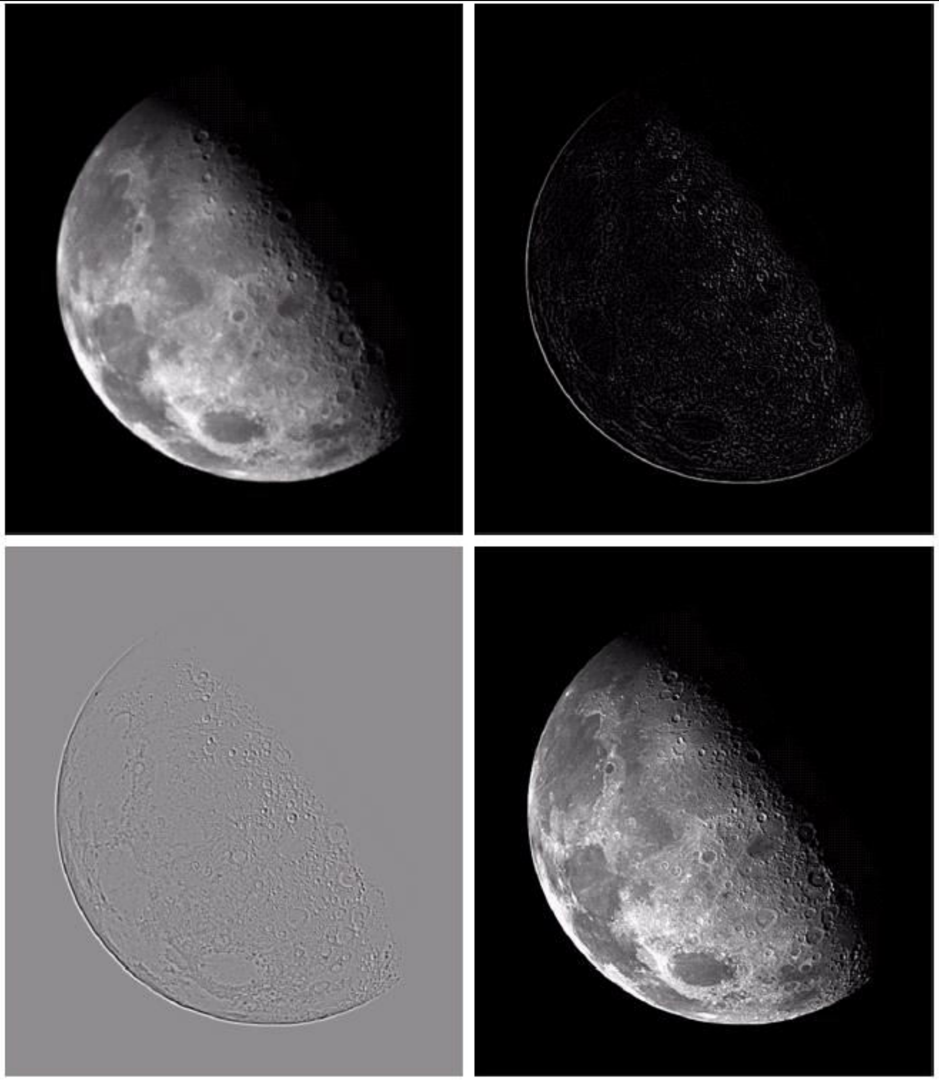
- Because the Laplacian is a derivative operator, its use highlights intensity discontinuities in an image and deemphasizes regions with slowly varying intensity levels. This will tend to produce images that have grayish edge lines and other discontinuities, all superimposed on a **dark and featureless background**.
- Background features can be recovered while still preserving the sharpening effect simply by adding the Laplacian image to the original.
- If the center coefficient of the Laplacian mask is **negative**

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

- If the center coefficient of the Laplacian mask is **positive**

$$g(x,y) = \boldsymbol{f(x,y)} + \nabla^2 f(x,y)$$

(a) Original Image (North Pole of the moon)



(b) Laplacian filter

(c) Enhancement of (b)

(d) (a)+(b)

110

Image Enhancement (Laplacian Filter)

- If the center coefficient of the Laplacian mask is **negative**

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

- If the center coefficient of the Laplacian mask is **positive**

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

- **Simplification**:

$$g(x,y) = f(x,y) - [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)] + 4f(x,y)$$

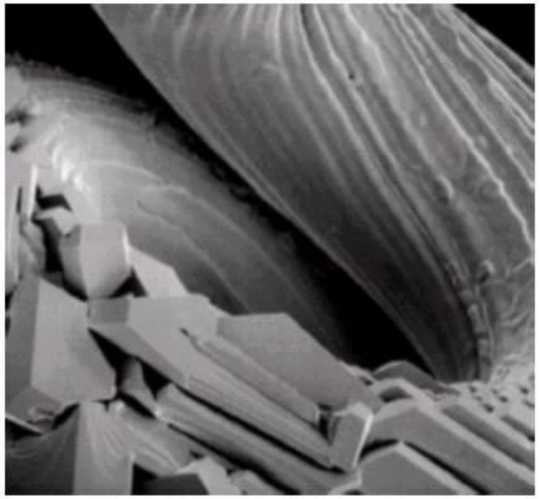
$$g(x,y) = 5f(x,y) - [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)]$$

0	−1	0
−1	5	−1
0	−1	0

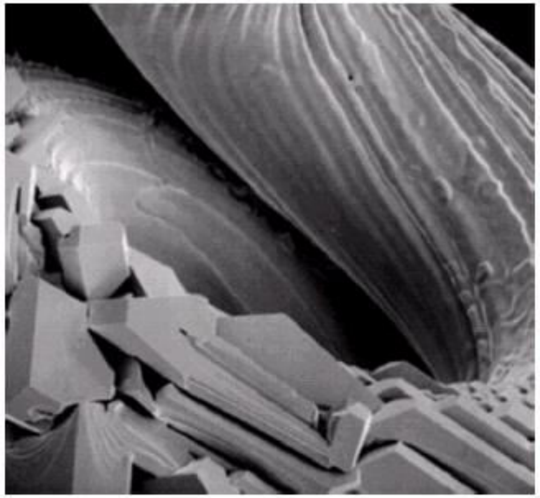
Mask (a)

−1	−1	−1
−1	9	−1
−1	−1	−1

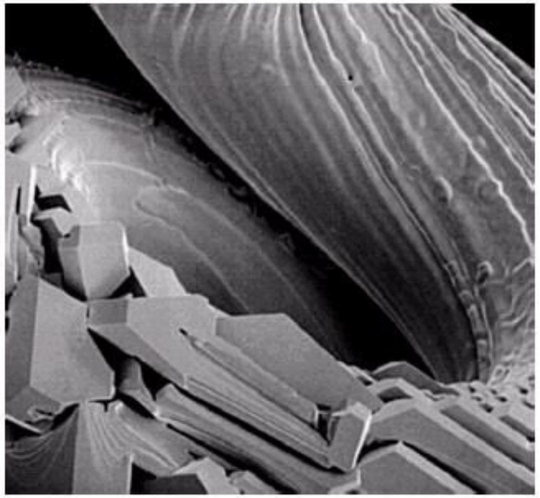
Mask (b)



Original Image



Mask (a)



Mask (b)

Dynamic Range Rescaling

- The intensity values of a sharpening result range may >255 or <0.
- A typical way to scale a Laplacian image is to add to its minimal value to bring the new minimum to zero and then scale the result to the full intensity range [0, L-1].
- The procedure is
  - Find the minimal intensity level t，search for the maximal intensity level M
  - If $M > 255$ or $t < 0$ ，recalculate the pixel intensity as
    $$p' = ((p - t)/(M - t)) \times 255$$
  - p' is the new intensity value of p 。

Use of First Derivative for Image Enhancement - Gradient

- First derivatives in image processing are implemented using the **magnitude of gradient**.
- Image gradient: $\nabla f = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix}$, This vector has the important geometrical property that it points in the direction of the greatest rate of change of f at location (x, y).
- Gradient magnitude (length):

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \sim \left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right| \text{ (approximation)}$$

- It is common practice to refer to this image as the **gradient image**.

114

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

- **Robert cross-gradient operator** (2x2 mark)

$$\|\nabla f(x,y)\| = \sqrt{(z_9 - z_5)^2 + (z_8 - z_6)^2}$$

$$= |z_9 - z_5| + |z_8 - z_6| \text{ (approximation)}$$

- **Sobel operator** (3x3 mask)

$$\|\nabla f(x,y)\| = |G_x| + |G_y| =$$

$$|(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)|$$

$$+|(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

First-Derivative – the Gradient

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

0	−1
1	0

Roberts operator

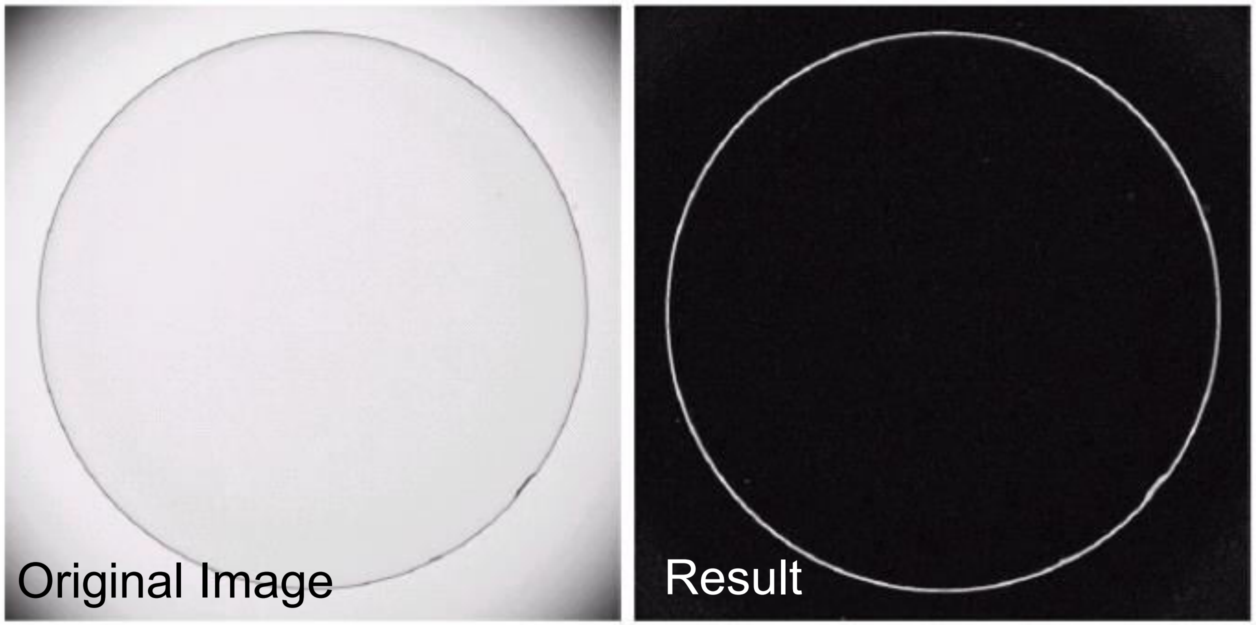
−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel operator

Example of Sobel Filter

- The gradient is used frequently in industrial inspection.
- This figure shows an optical image of a contact lens, illuminated by a lighting arrangement designed to highlight imperfections.



3.6.3 Unsharping mask and High-boost filtering

- A process that has been used for many years by the printing and publishing industry to sharpen images consists of subtracting a blurred version of an image from the original image. This process is called **unsharp masking**.

  $$f_s(x,y) = f(x,y) - f^*(x,y),$$

  where $f^*(x,y)$ is the **blurred image**.

- A slight further generalization of the unsharp masking, called **high boost filtering**

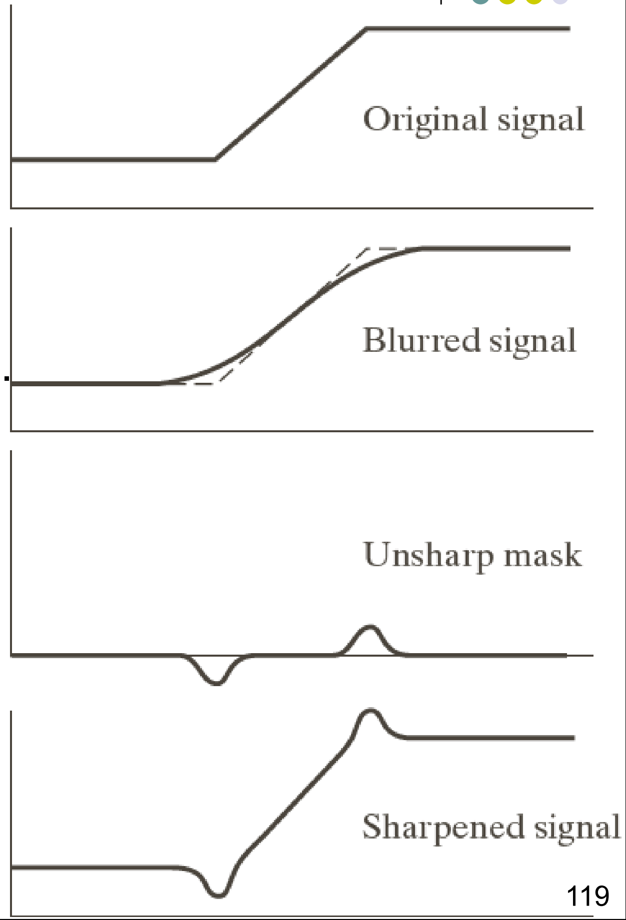
  $$\begin{aligned} f_{hb}(x,y) &= Af(x,y) - f^*(x,y) \\ &= (A-1)f(x,y) + f(x,y) - f^*(x,y) \\ &= (A-1)f(x,y) + f_s(x,y) \end{aligned}$$

- Using Laplacian

  $$f_{hb}(x,y) = Af(x,y) - \nabla^2 f(x,y)$$
  $$f_{hb}(x,y) = Af(x,y) + \nabla^2 f(x,y)$$

Unsharping mask and high-boost filtering

- This figure explains how unsharp masking works.
- **(a)** can be interpreted as a horizontal scan line through a vertical edge that transitions from a dark to a light region.
- **(b)** shows the result of smoothing.
- **(c)** is the unsharp mask, obtained by subtracting the blurred signal from the original. (c) is very similar to what we would obtain using a second-order derivative.
- **(d)** is the final sharpened result, obtained by adding the mask to the original signal.

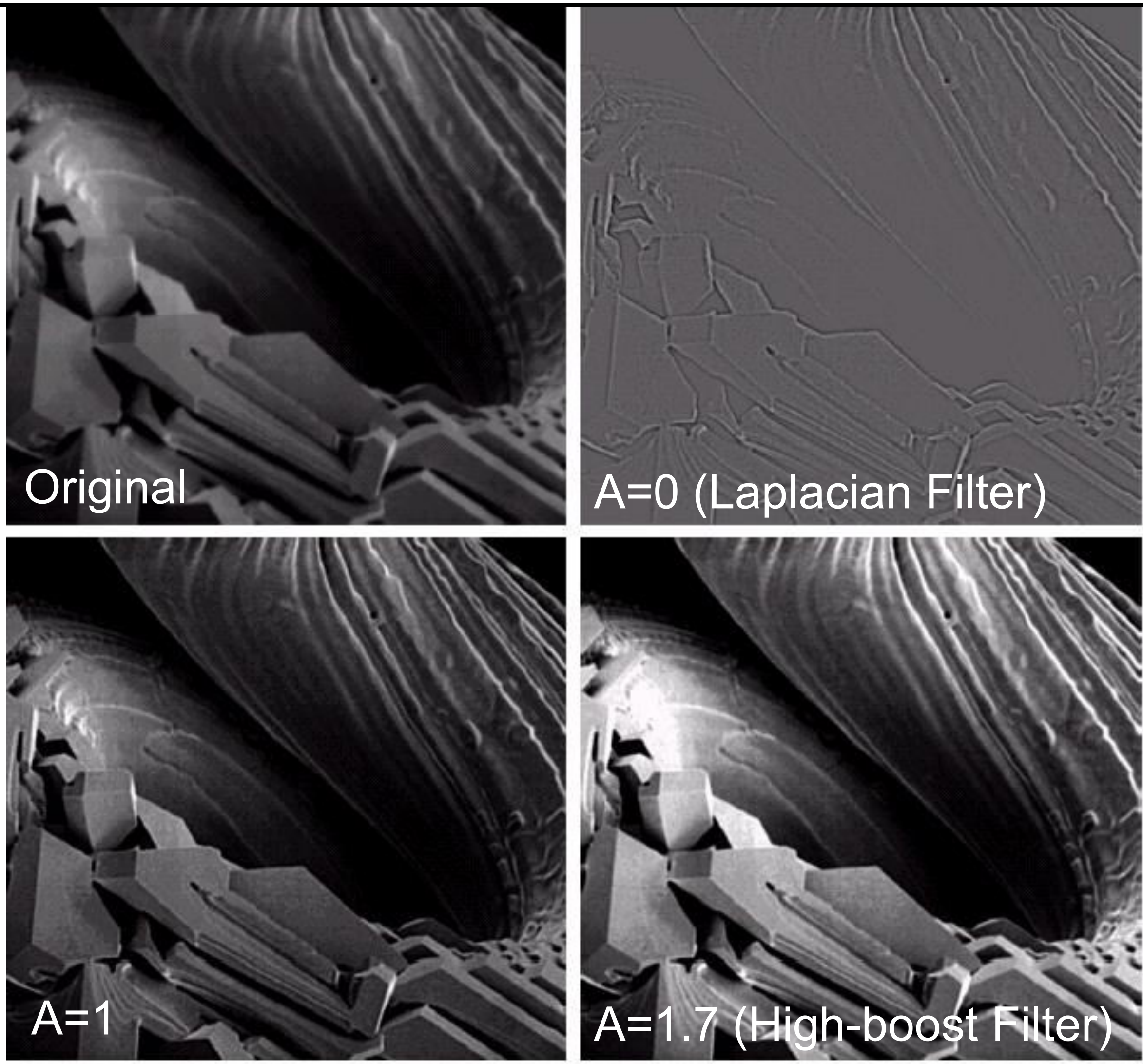


High-boost Filtering

0	-1	0
-1	$A + 4$	-1
0	-1	0

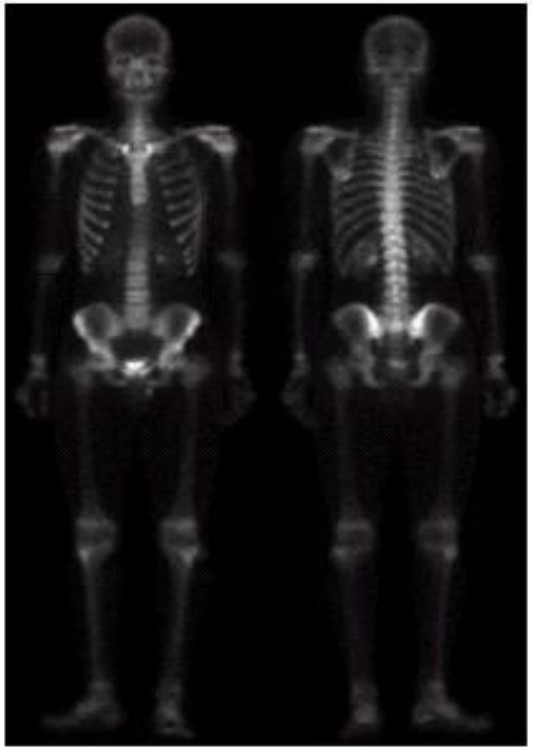
-1	-1	-1
-1	$A + 8$	-1
-1	-1	-1

- A=0: Laplacian Filter
- A=1: Laplacian enhancement
- A>1: High-boost Filter


Original
A=0 (Laplacian Filter)
A=1
A=1.7 (High-boost Filter)

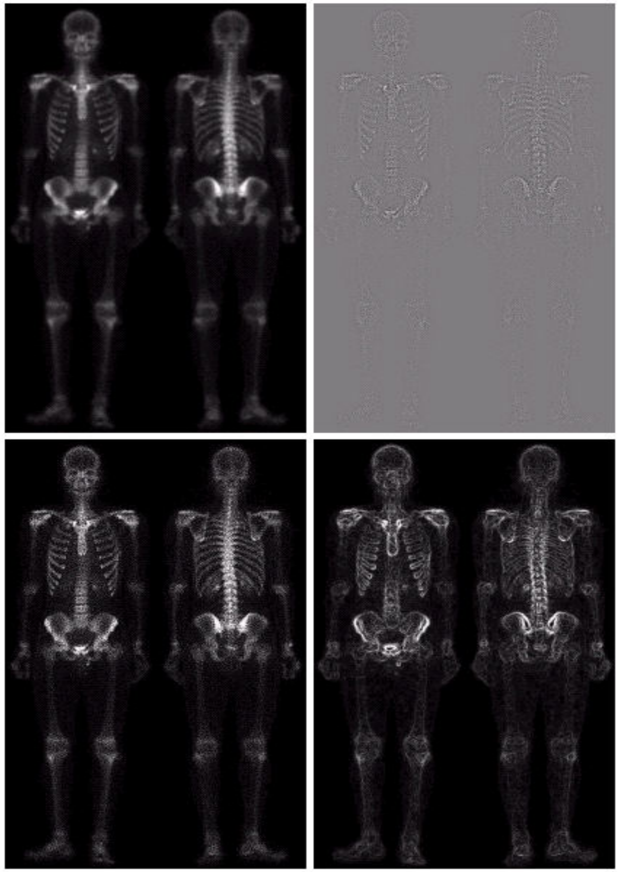
3.7 Combining Spatial Enhancement Methods

- Frequently, a given task will require application of several complementary techniques in order to achieve an acceptable result.
- The image show here is a unclear whole body bone scan, used to detect diseases such as bone infection and tumors.
- The objective is to enhance this image by sharpening it and by bringing out more of the skeletal detail.
- The narrow dynamic range of the intensity levels and high noise content make this image difficult to enhance.



122

3.7 Combining Spatial Enhancement Methods



a
+
b

a b
c d

Figure 3.43
(a) Image of whole body bone scan.
(b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b). (d) Sobel of (a).

cxe

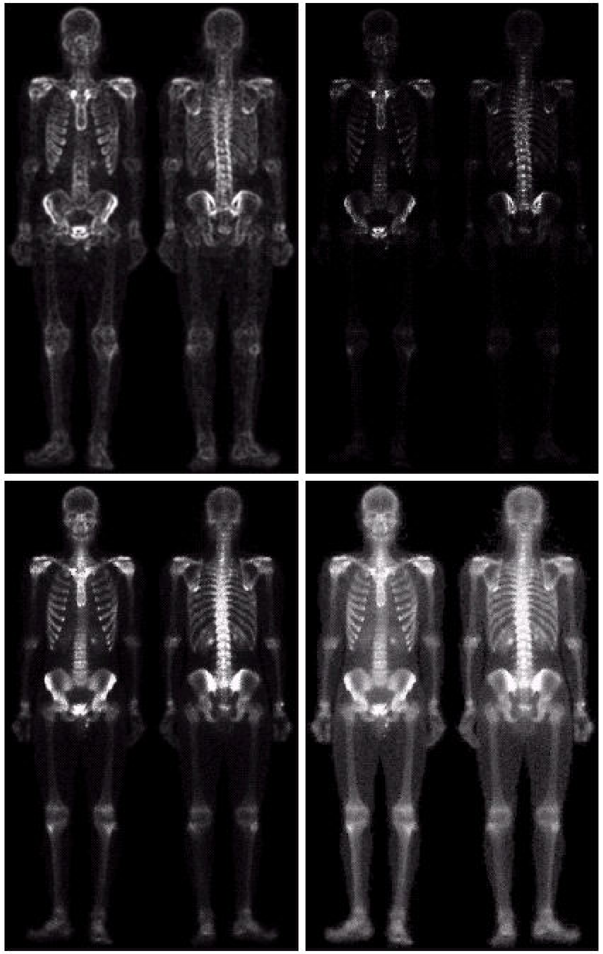


Figure 3.43

(Continued)
(e) Sobel image smoothed with a 5×5 averaging filter. (f) Mask image formed by the product of (c) and (e). (g) Sharpened image obtained by the sum of (a) and (f). (h) Final result obtained by applying a power-law transformation to (g). Compare (g) and (h) with (a). (Original image courtesy of G.E. Medical Systems.)

3.8 Using Fuzzy Techniques for Intensity Transformations and Spatial Filtering

- Final Project

a b c

FIGURE 3.54 (a) Low-contrast image. (b) Result of histogram equalization. (c) Result of using fuzzy, rule-based contrast enhancement.