Chapter 2 Image Enhancement

Fundamental image enhancement techniques:

1. Gray-Level Transformations

   $s = T(r)$,

   where r : input gray levels, s : output gray levels

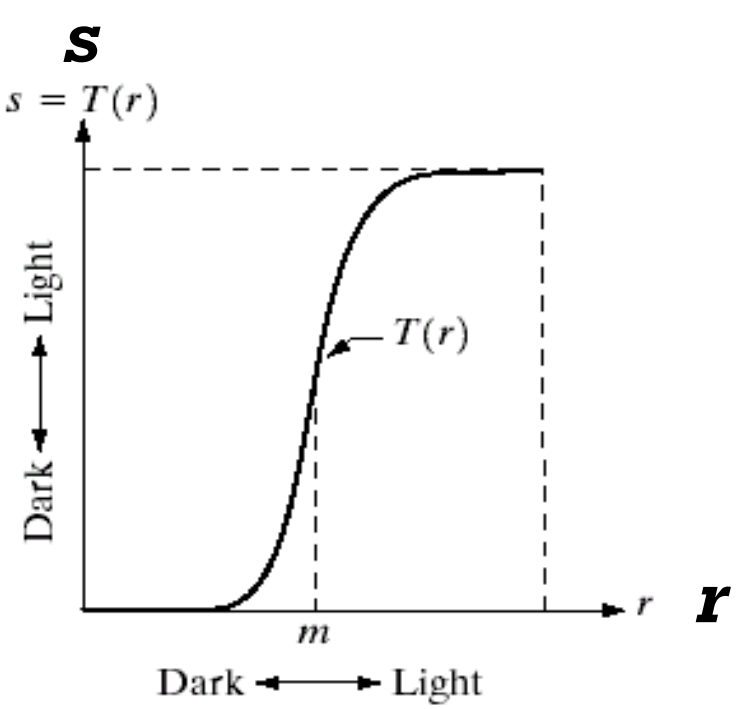
2. Histogram Processing

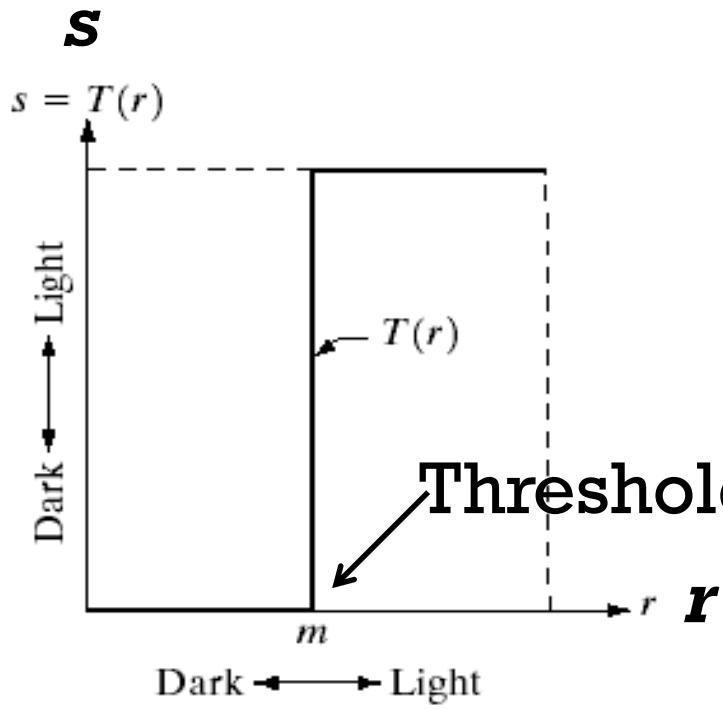
   Modify the shape of histogram

3. Image Averaging

   Noise reduction for still images

Gray-Level Transformations





Thresholding value

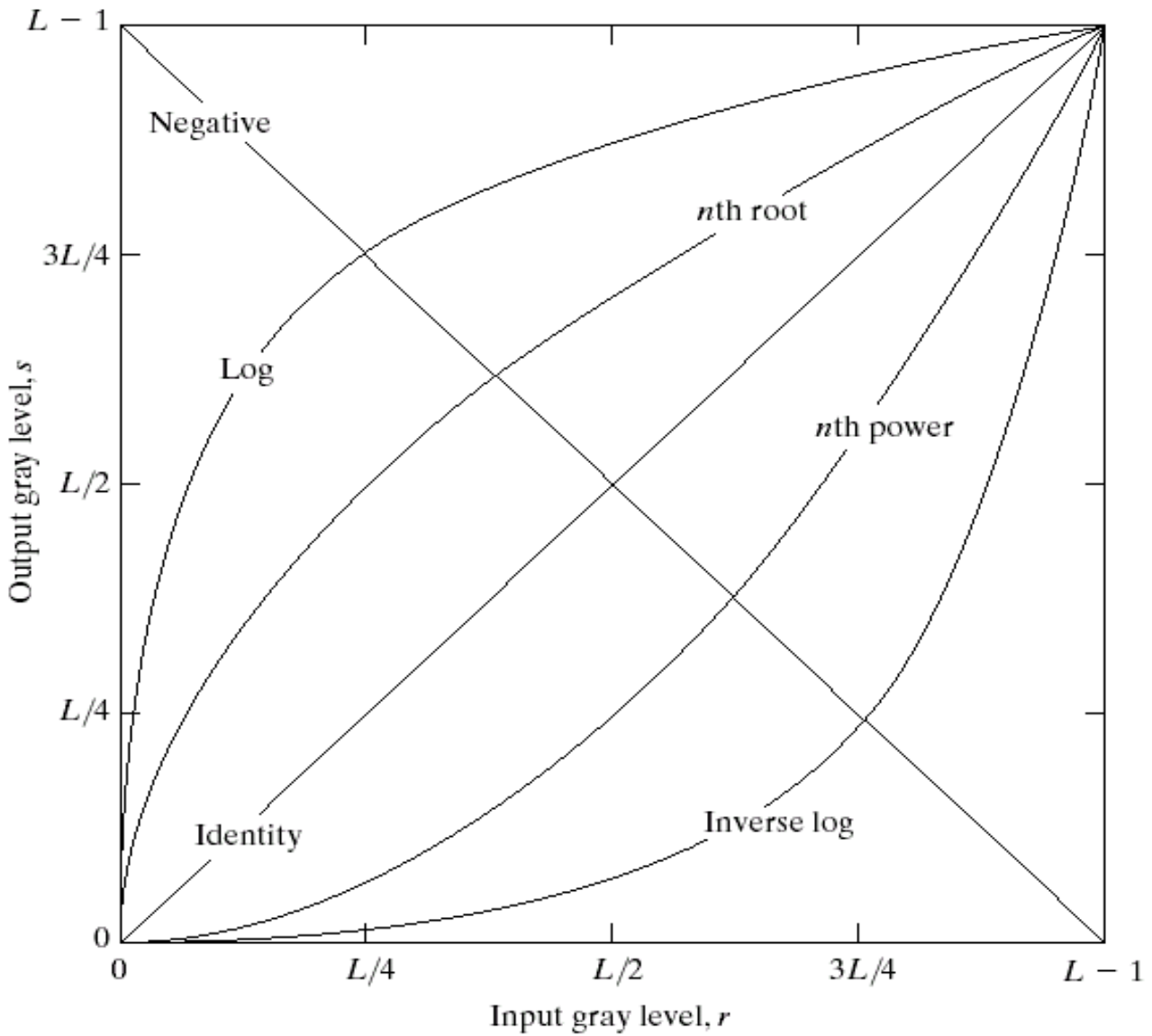
a b

FIGURE 3.2 Gray-level transformation functions for contrast enhancement.

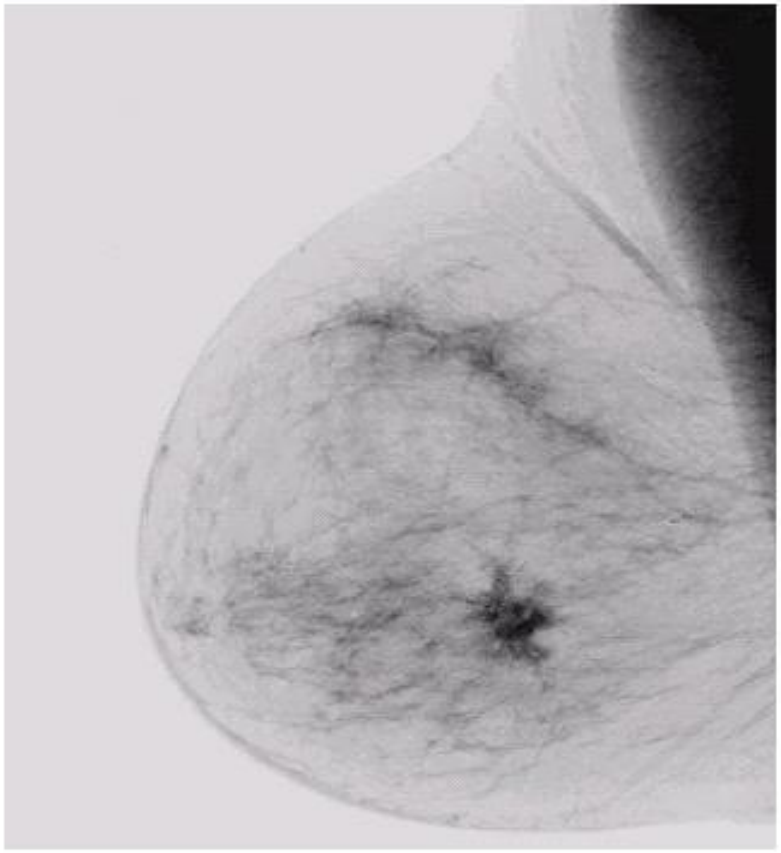
$s = T(r)$, where r : input gray levels,
s : output gray levels

Transformation Functions

FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.



Negative Transformations



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.)

Analysis is made easier by the negative transformation.

Negative Transformations

Inverting a digital image is a **point processing operation.** The output of **image inversion** is a negative of a digital image.

In a digital image the intensity levels vary from **0 to L-1**. The **negative transformation is given by s=L-1-r.**When an image is inverted, each of its pixel value 'r' is subtracted from the maximum pixel value L-1 and the original pixel is replaced with the result 's'.

Image inversion or Image negation helps **finding the details from the darker regions of the image.**

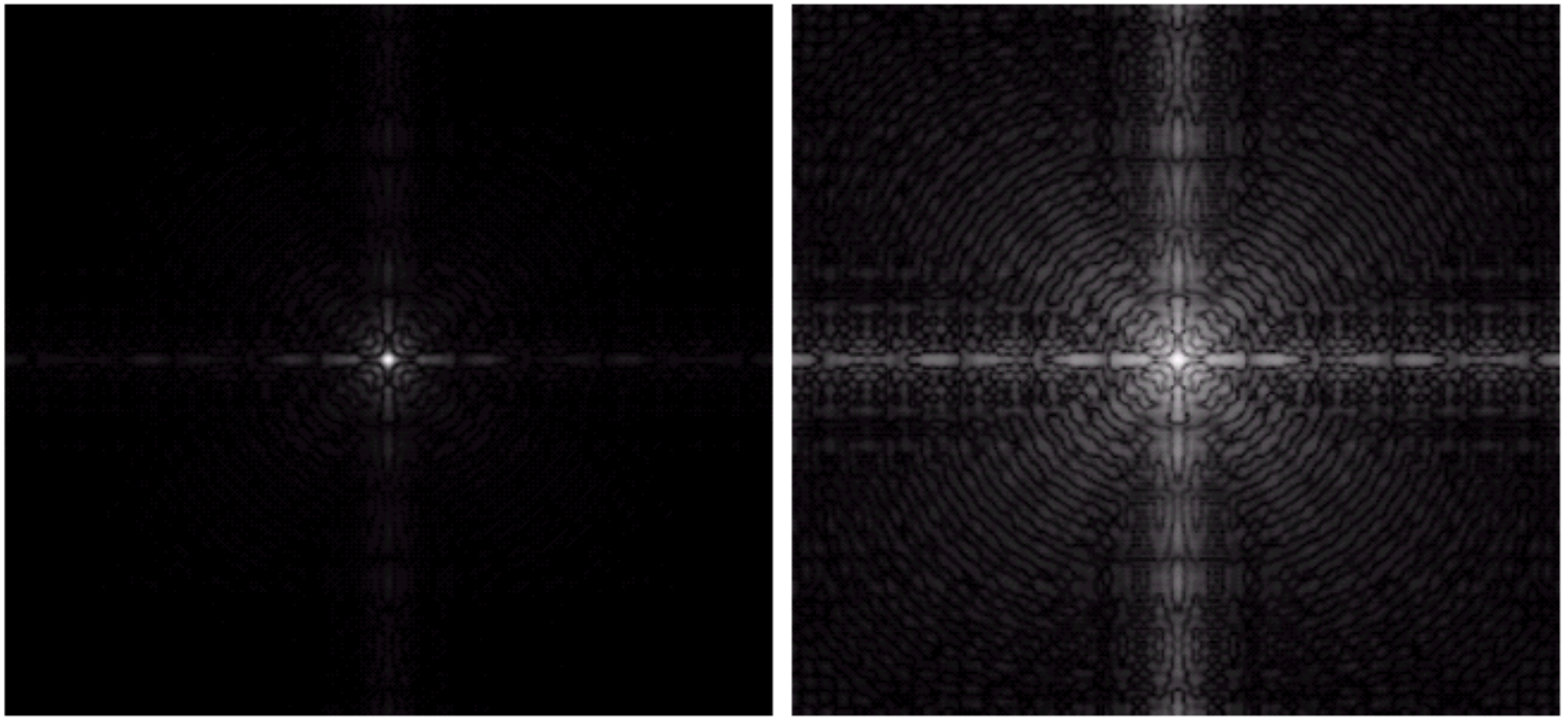
The negative transformation has its applications in the areas of **Medical Imaging, Remote Sensing** and others.

Log Transformations

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

a b

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



Visibility is improved by the log transformation.

Log Transformations

Logarithmic transformation of an image is one of the gray level image transformations. Log transformation of an image means replacing all pixel values, present in the image, with its logarithmic values. Log transformation is used for image enhancement as it expands dark pixels of the image as compared to higher pixel values.

The formula for applying log transformation in an image is,

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

c = 255 / (log (1 + max_input_pixel_value))

When we apply log transformation in an image and any pixel value is '0' then its log value will become infinite. That's why we are adding '1' to each pixel value at the time of log transformation so that if any pixel value is '0', it will become '1' and its log value will be '0'.

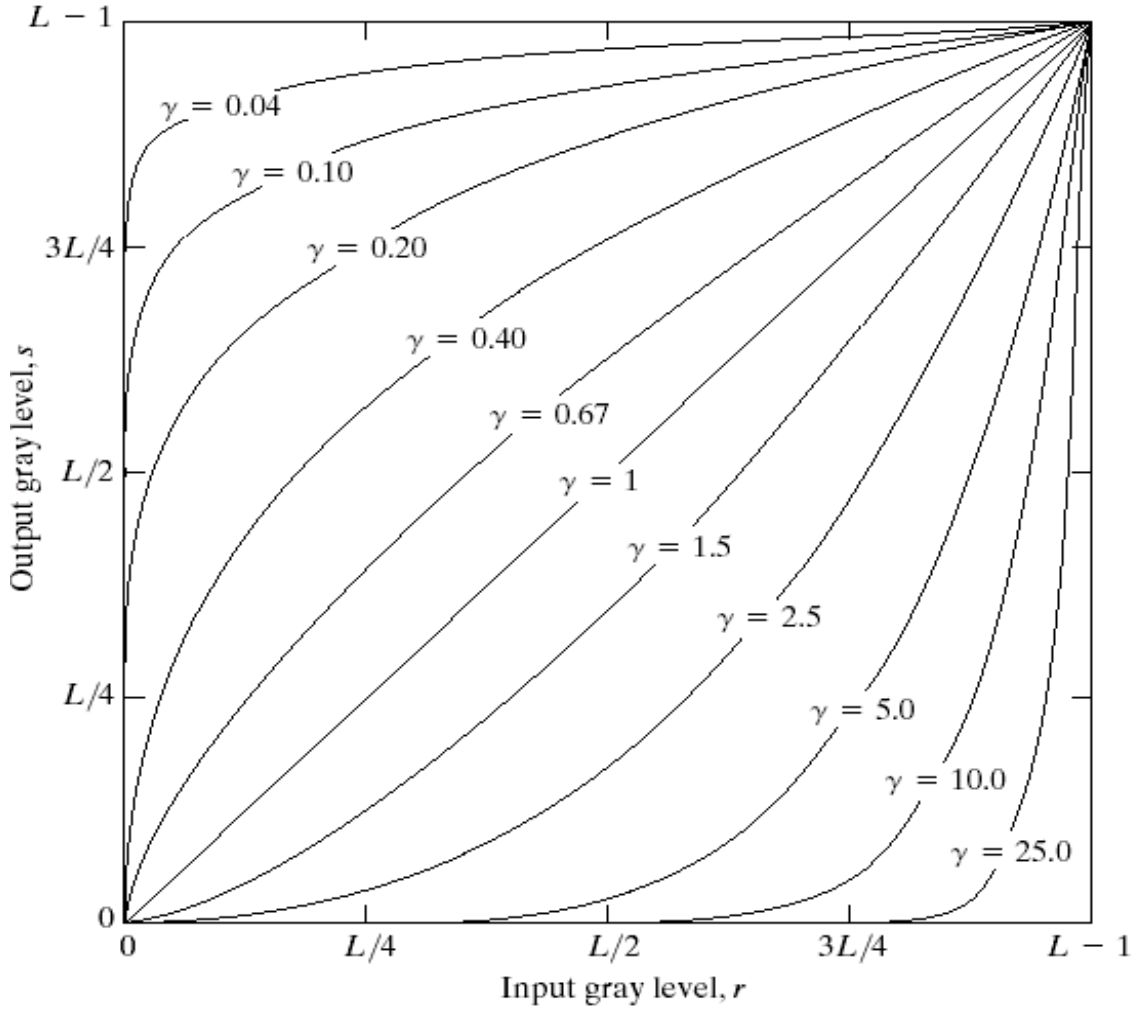
Exercise 1

Determine *c* so that the log transformation can be applied to images with 8-bit gray levels.

Compare the natural logarithm with the base-10 logarithm. Is there any difference in resulting images?

Power-Law Transformations

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



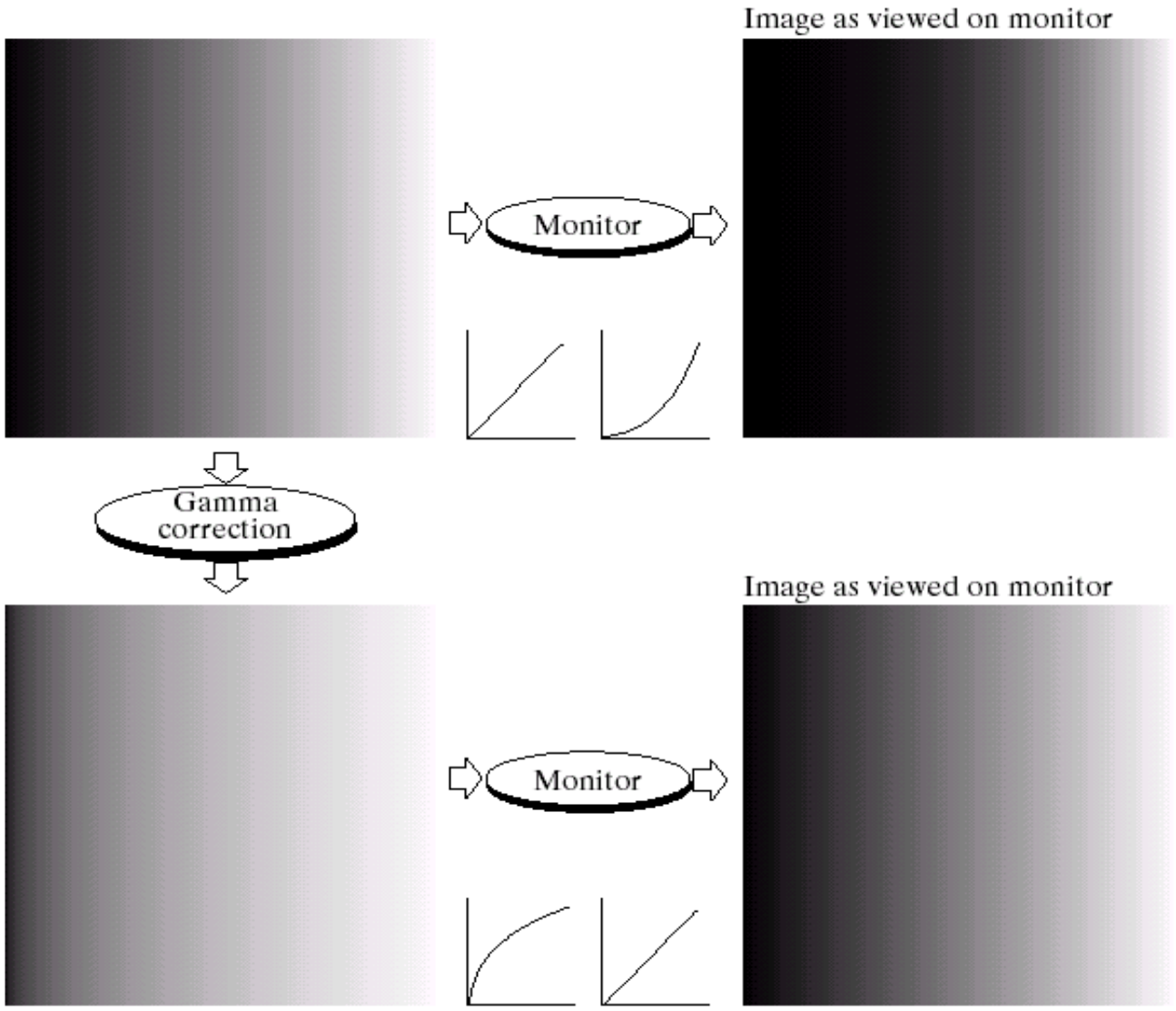
A variety of devices used for image capture, printing, and display respond according to a power law.

The correction of this power-law response is called *gamma correction*.

Gamma Correction

a b
c d

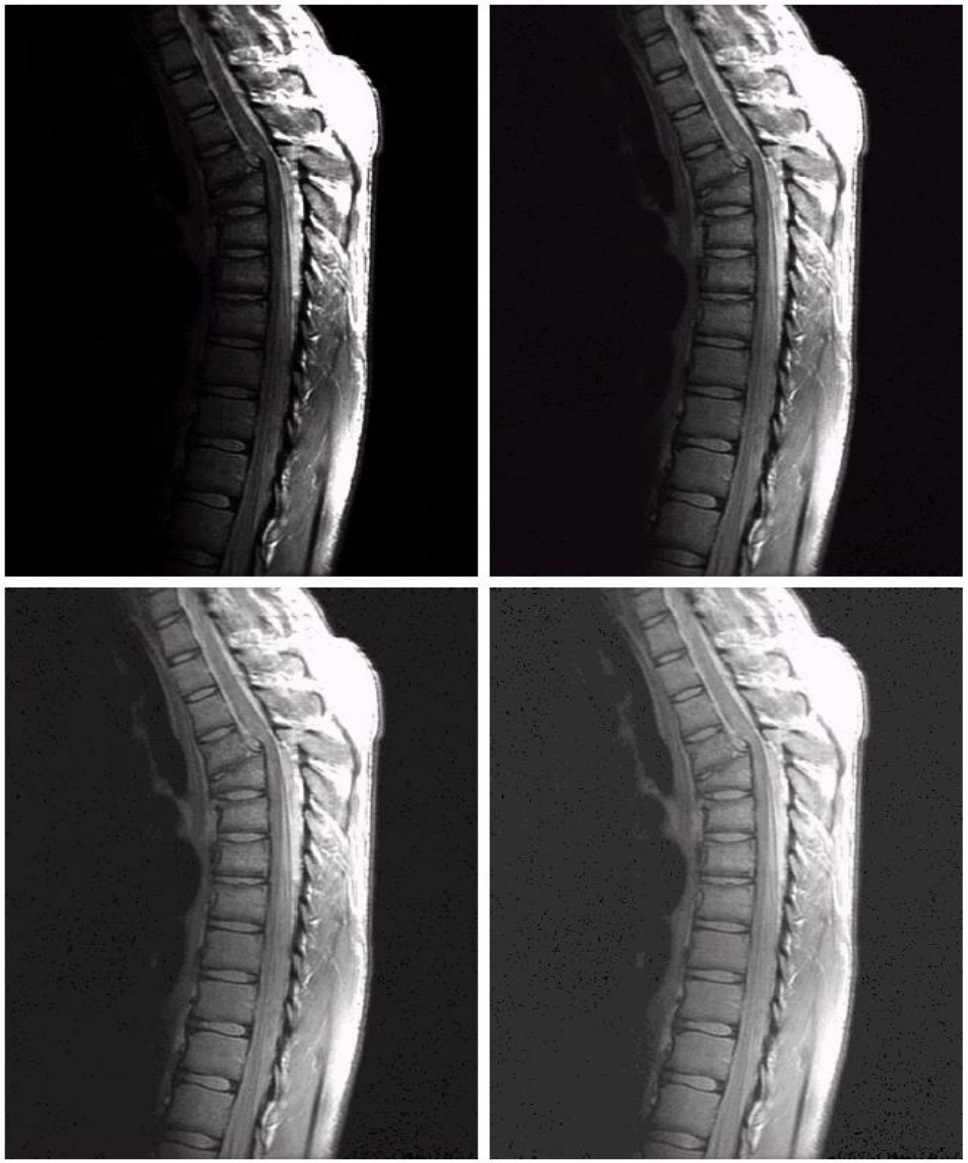
FIGURE 3.7
(a) Linear-wedge gray-scale image. (b) Response of monitor to linear wedge. (c) Gamma-corrected wedge. (d) Output of monitor.



Applications

- Monitors
- Scanners
- Printers

Power-Law Transformations ($\gamma<1$)



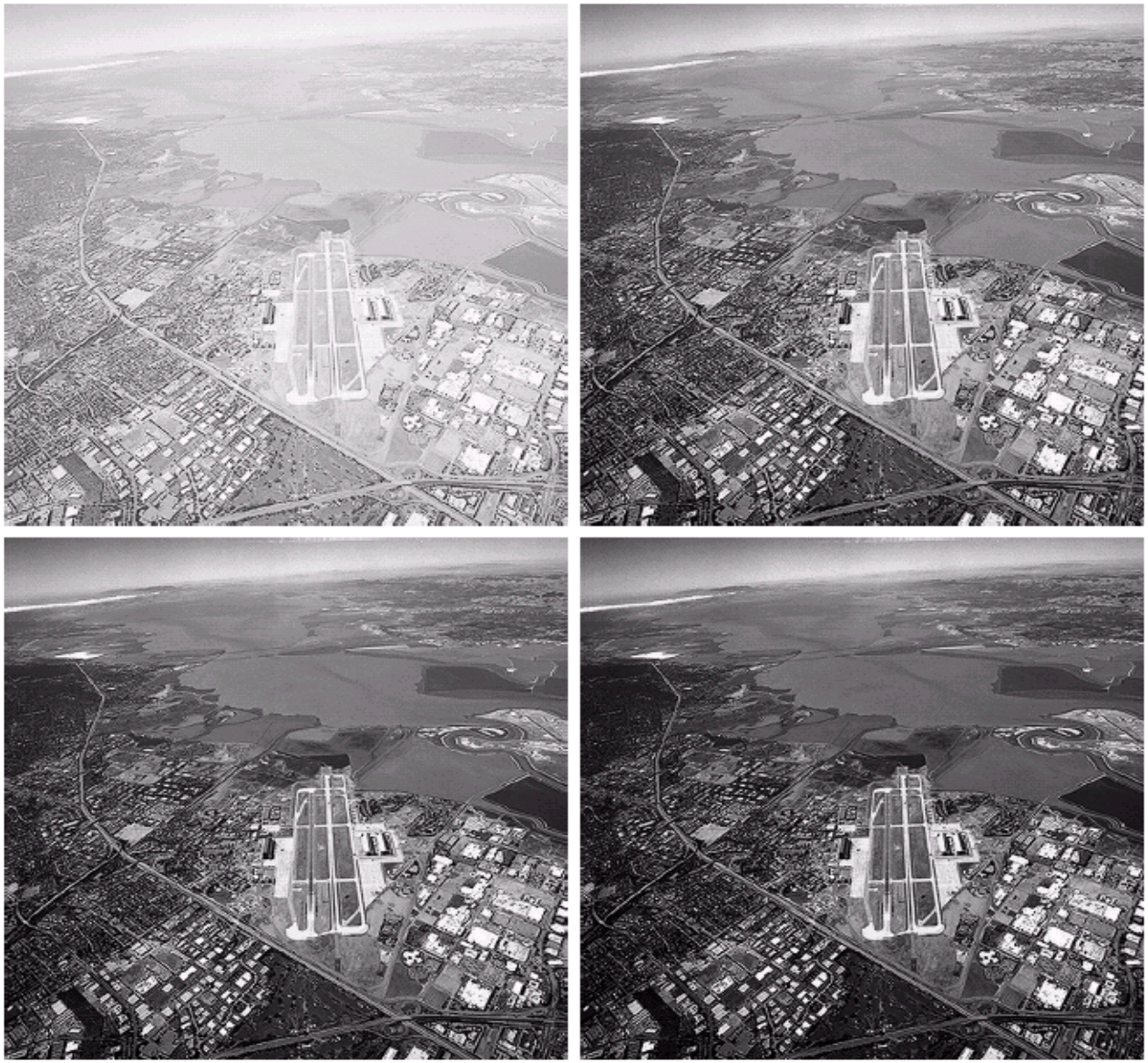
a b
c d

FIGURE 3.8
(a) Magnetic resonance (MR) image of a fractured human spine.
(b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 0.6, 0.4$, and 0.3, respectively. (Original image for this example courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

Power-Law Transformations ($\gamma>1$)

a b
c d

FIGURE 3.9
(a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0$, and 5.0, respectively. (Original image for this example courtesy of NASA.)



Bit-plane Decomposition

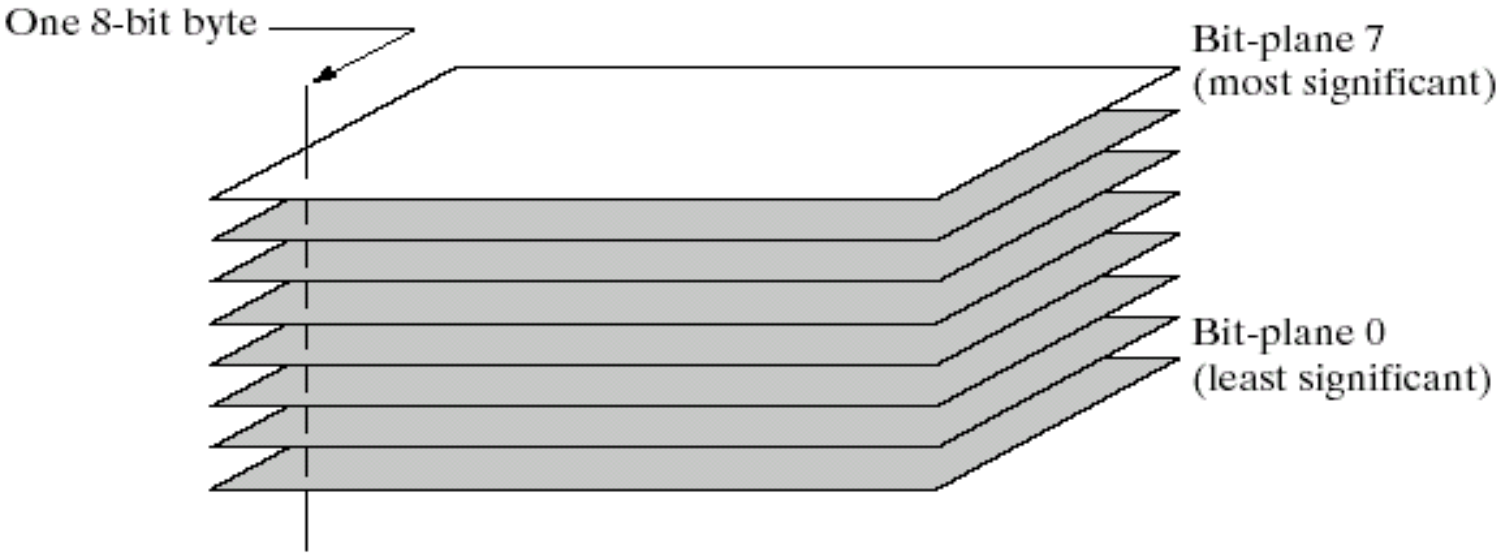
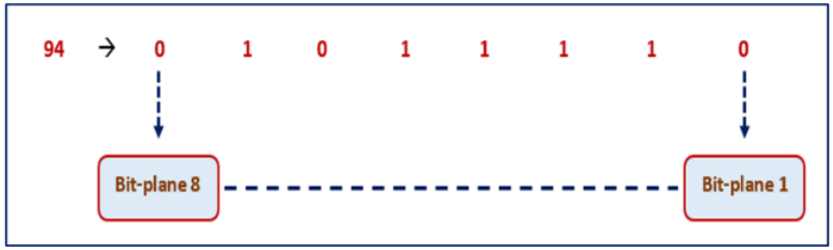
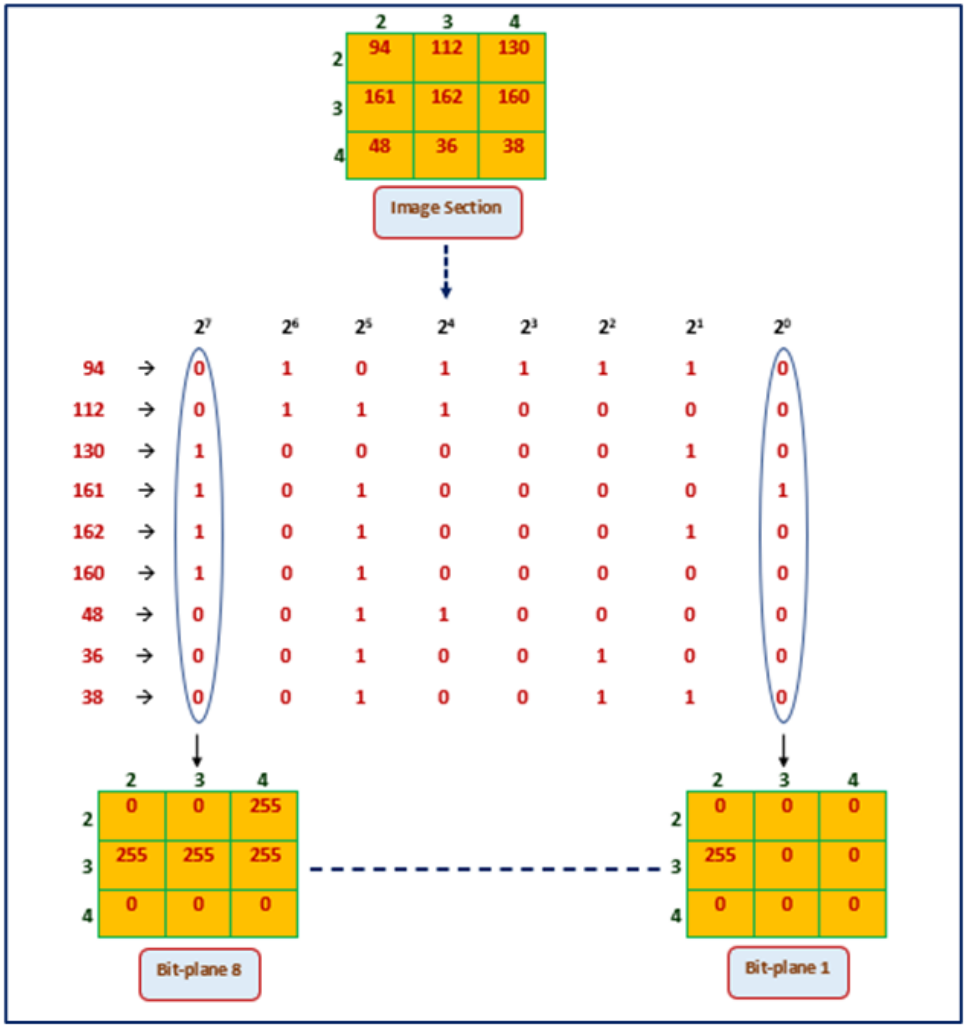


FIGURE 3.12
Bit-plane representation of an 8-bit image.

Separating a digital image into its bit planes is useful for analyzing the relative importance played by each bit of the image. Also, this type of decomposition is useful for image compression.

Bit-plane Decomposition





Bit-plane Decomposition

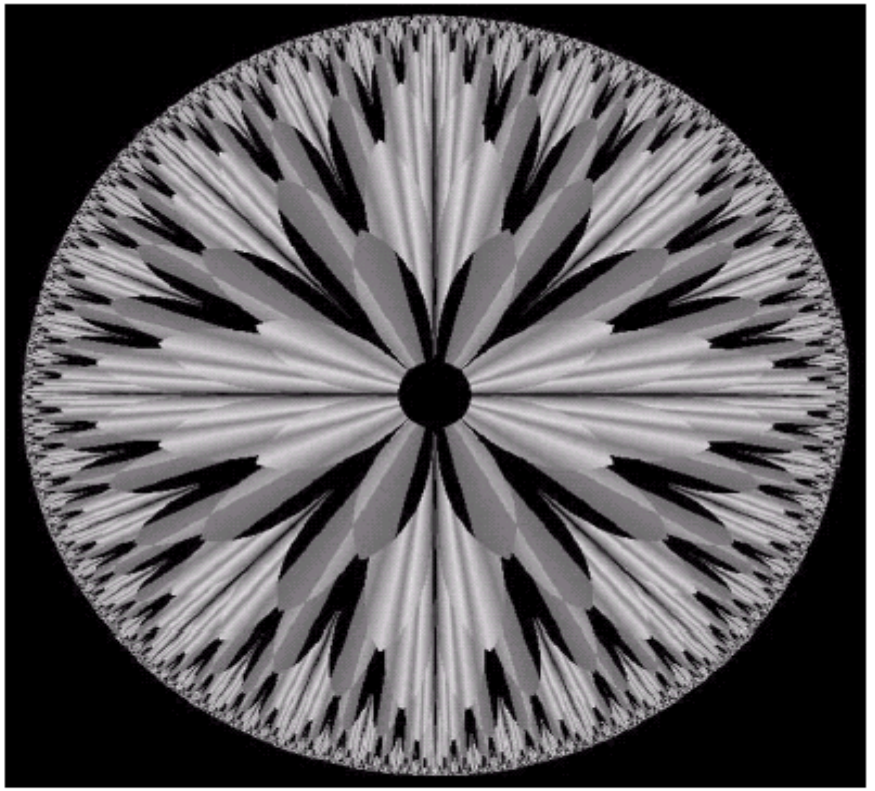


FIGURE 3.13 An 8-bit fractal image. (A fractal is an image generated from mathematical expressions). (Courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA.)

Bit-plane Decomposition

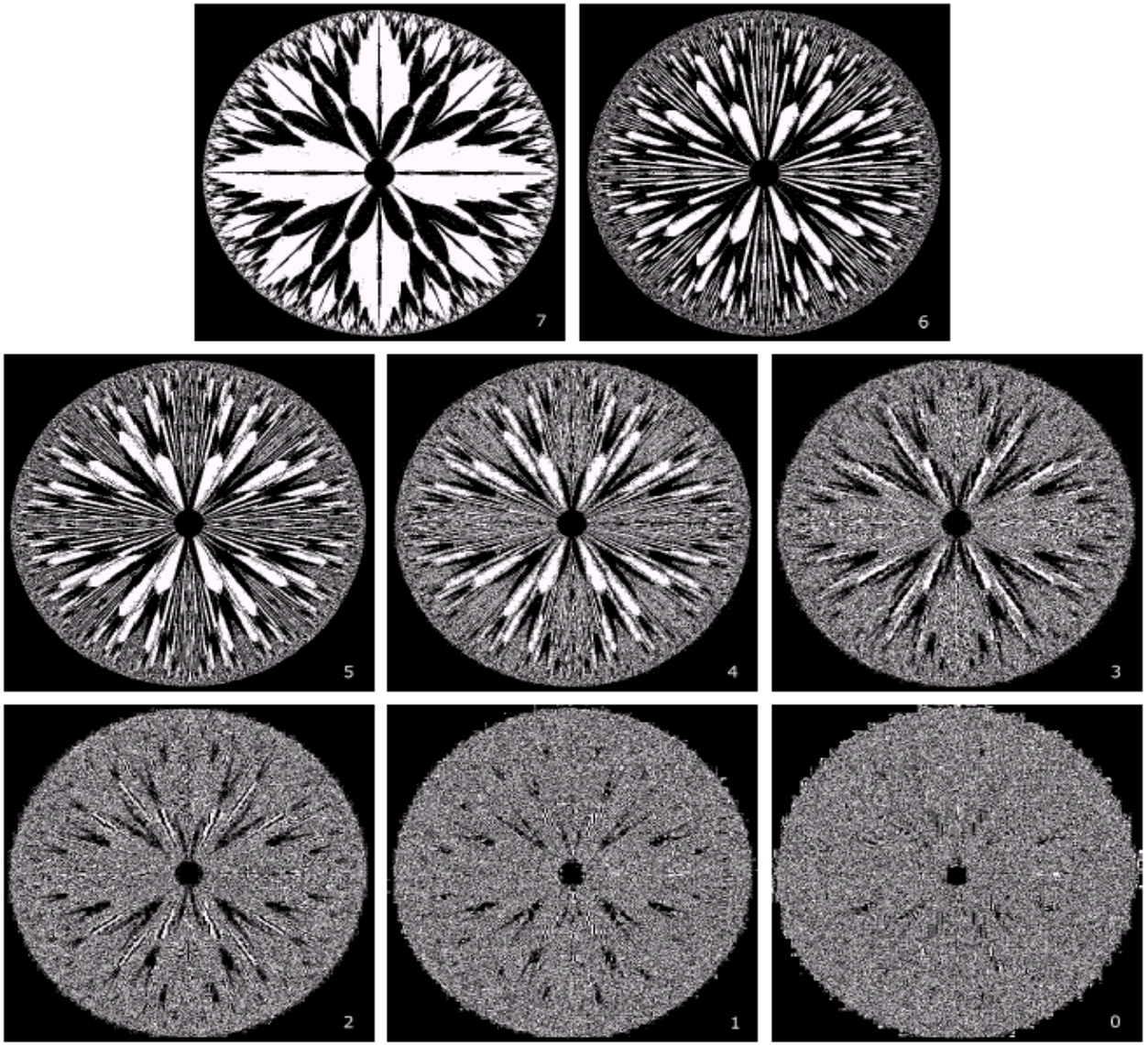
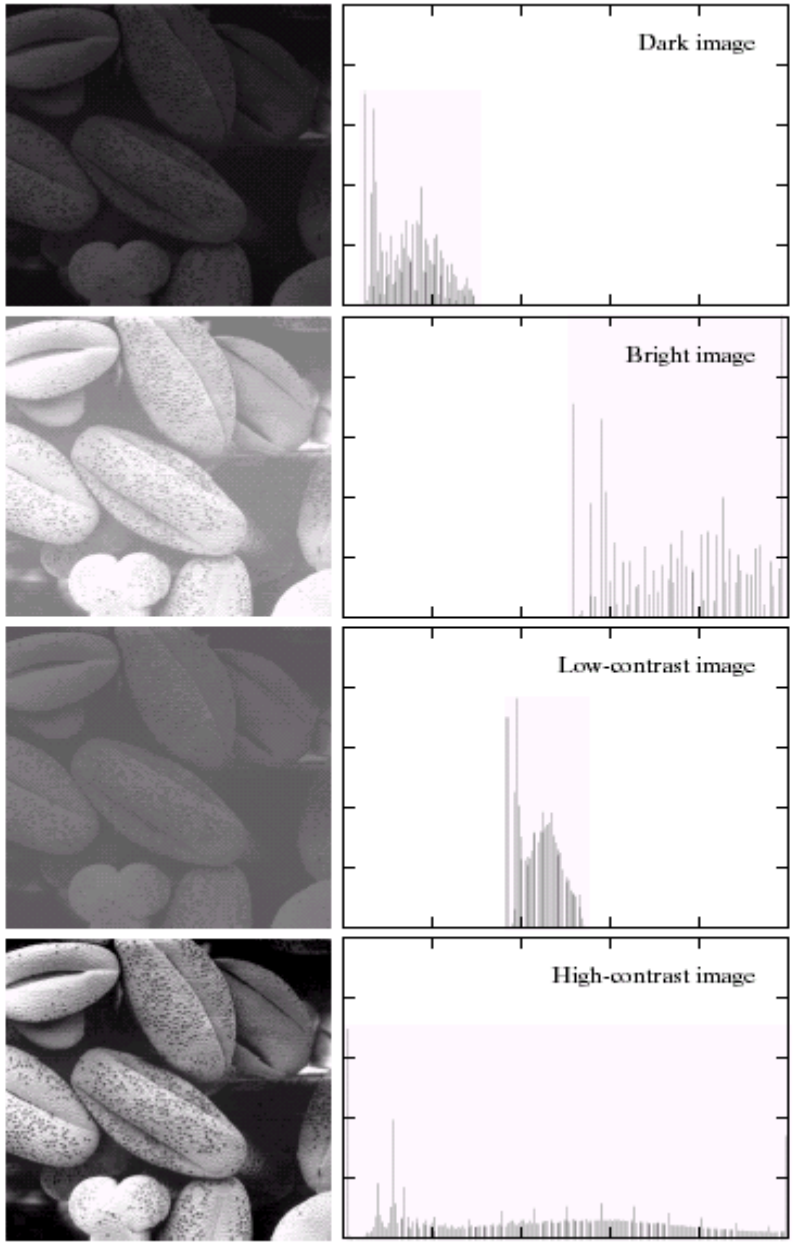


FIGURE 3.14 The eight bit planes of the image in Fig. 3.13. The number at the bottom, right of each image identifies the bit plane.

Histogram of Digital Images



The histogram of a digital image with gray levels in the range $[0, L-1]$ is a discrete function $h(r_k)=n_k$, where r_k is the k^{th} gray level and n_k is the number of pixels in the image having gray level r_k.

It is reasonable to conclude that an image whose gray levels spread widely and uniformly appears better. Can we stretch the histogram?

Preparation for Histogram Equalization

Probability:

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

where *n* denotes the total number of pixels in the image.

Cumulative Histogram:

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

Histogram Equalization

The cumulative histogram is used as a gray-level transformation function.

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

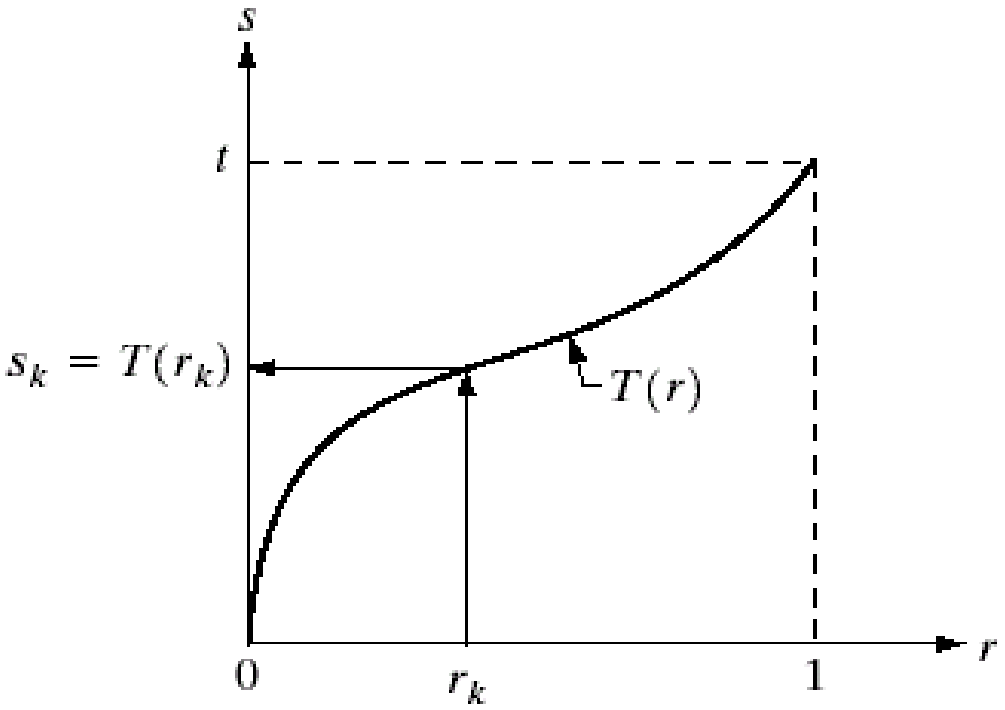
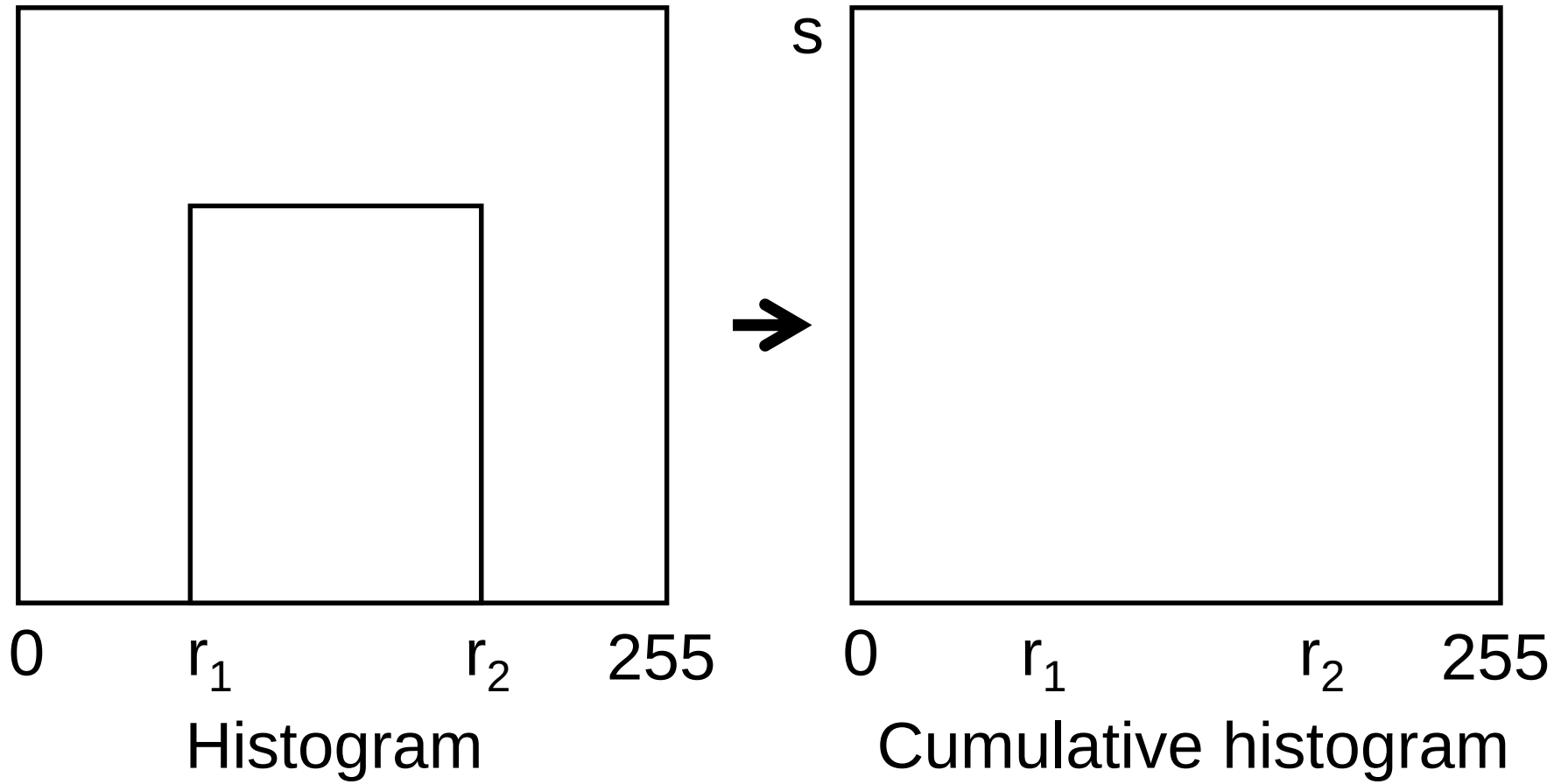


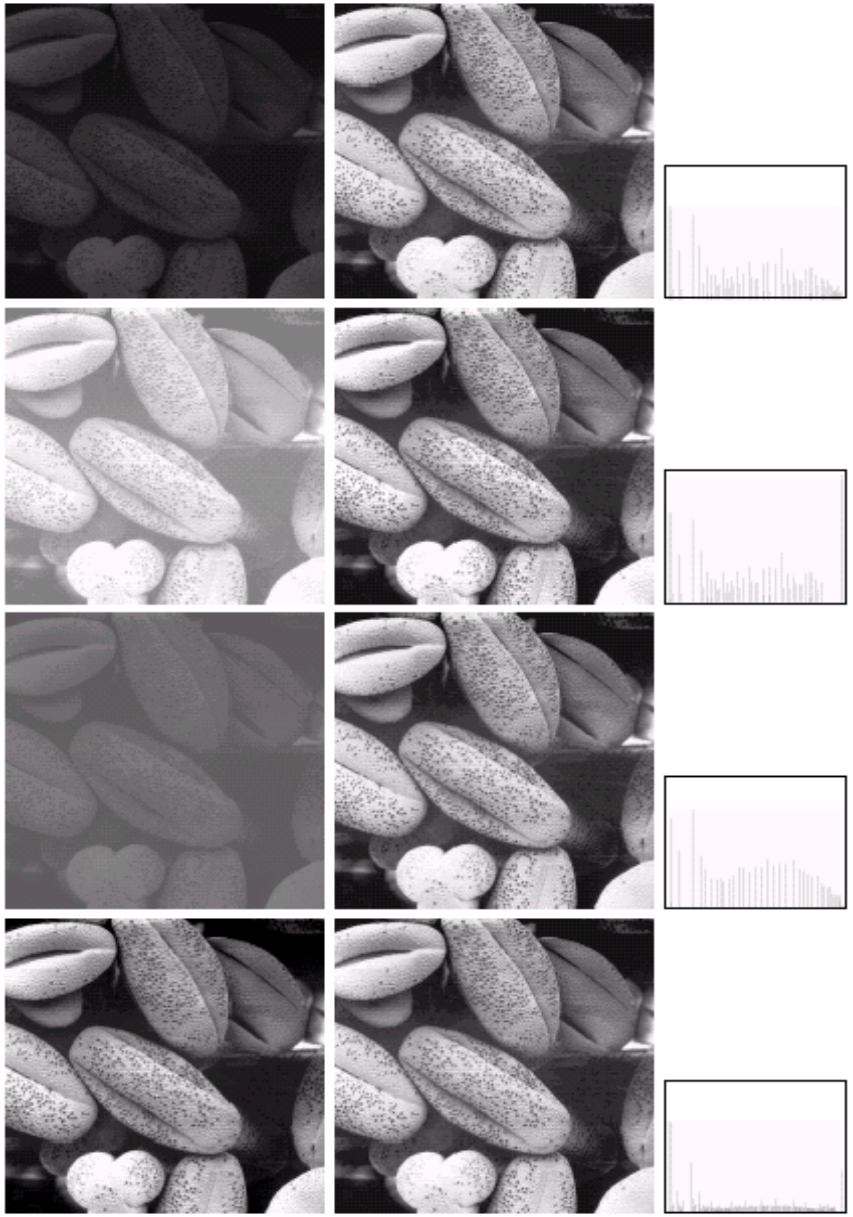
FIGURE 3.16 A gray-level transformation function that is both single valued and monotonically increasing.

Exercise 2

Plot the cumulative histogram for the histogram below.



Histogram Equalization Results



Histogram equalization automatically determines a gray-level transformation function that produces an output image with a uniform histogram.

Histogram equalization improves image contrast, but the appearance of the image is not always natural.

Image Averaging

Noisy image = Original image + Zero mean Noise

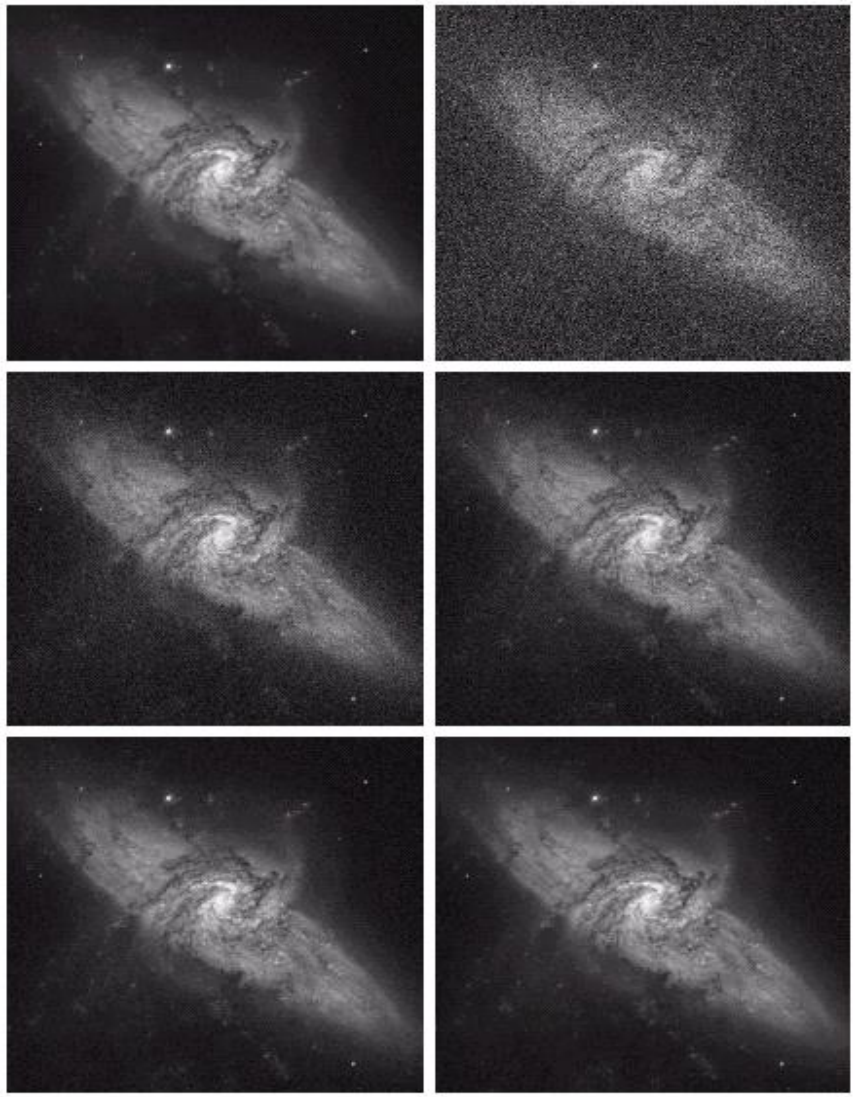
$$g(x,y) = f(x,y) + \eta(x,y)$$

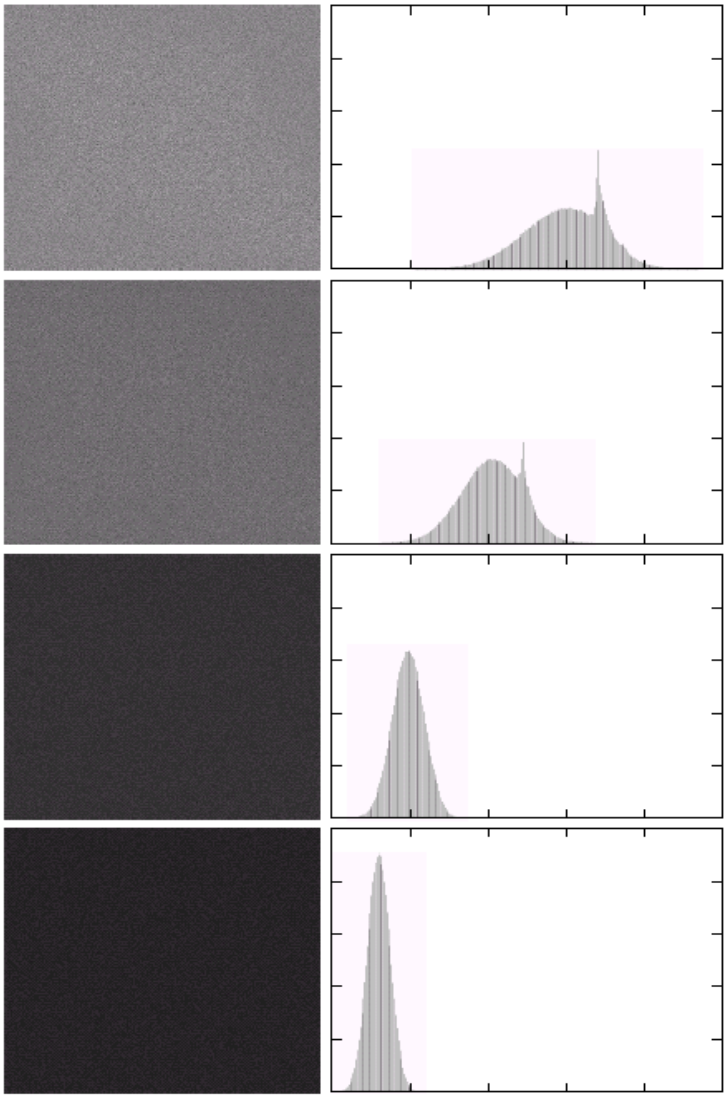
Image Averaging: $$\bar{g}(x,y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x,y)$$

$$E\left\{\bar{g}(x,y)\right\} = f(x,y)$$

$$\sigma^2_{\bar{g}(x,y)} = \frac{1}{K}\sigma^2_{\eta(x,y)}$$

Image Averaging Results





a b

FIGURE 3.31
(a) From top to bottom: Difference images between Fig. 3.30(a) and the four images in Figs. 3.30(c) through (f), respectively.
(b) Corresponding histograms.

Exercise 3

Discuss the limitations of the image averaging technique to improve image quality.

Assignment from Chapter 2

1. Apply gray-level transformations to your facial image, and obtain the negative and also an image of better appearance. (Chap2_1.m)
2. Apply histogram equalization to your facial image and comment on the result. (Chap2_1.m)
3. Decompose your facial images into 8-bit planes, and comment on the result. (Chap2_2.m)
4. Generate a set of noisy images by adding Gaussian noise to your image separately. Average the noisy images and comment on the result. (Chap2_3.m)

Root Mean Square Error, E_{RMS}

$$E_{RMS} = \sqrt{\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left(f'(x,y) - f(x,y)\right)^2}$$

Expression in MATLAB:

```
rms=sqrt( sum(sum( (f '-f).*(f '-f) ))/M/N );
plot(rms)
```

MATLAB Commands

- img=imread(‘Lena.bmp’); %Image reading
- buf1=imadjust(img,[0.2 0.8],[0 1],0.8); %Gray-level transformation
- buf2=histeq(img); %Histogram equalization
- imhist(buf1) %Display histogram
- nimg=imnoise(img,’gaussian’,0,0.01); %Noise addition

Image Enhancement Results

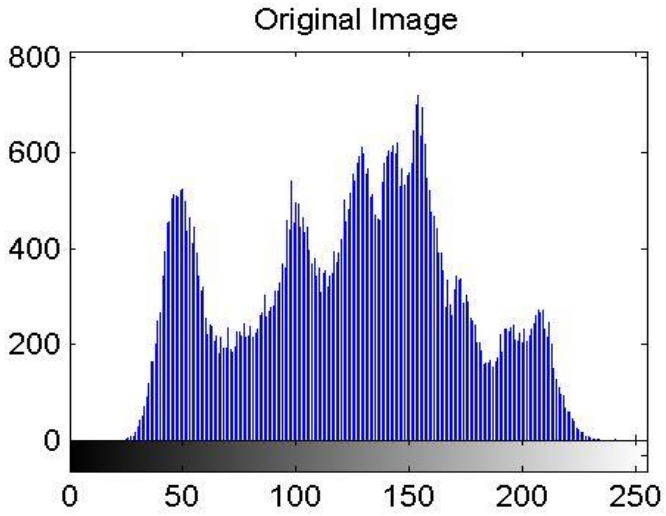
Original Image

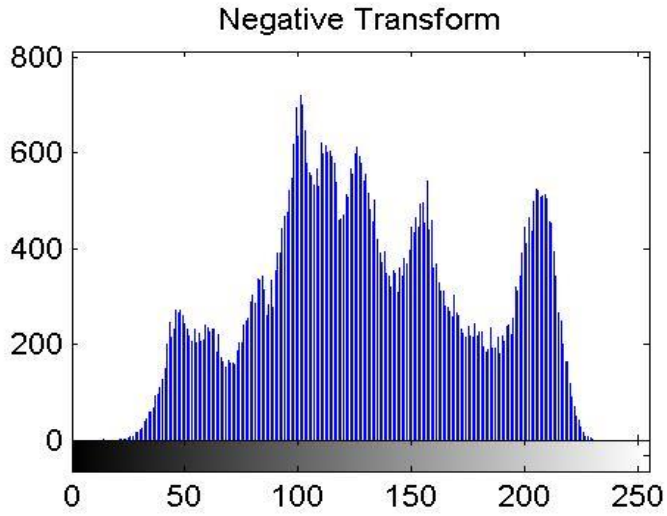
Negative Transform

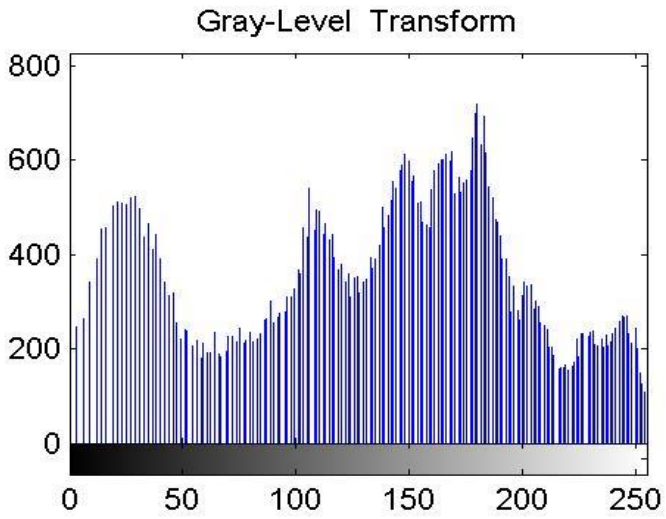
Gray-Level Transform

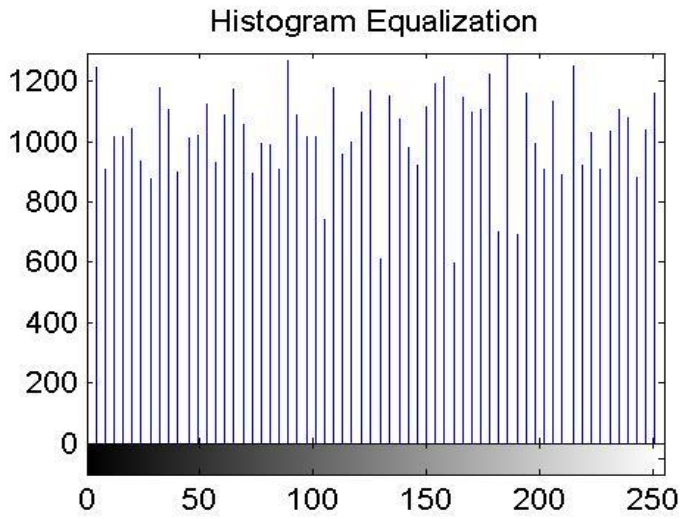
Histogram Equalization

Histograms of Resulting Images









Bit-plane Decomposition

MSB

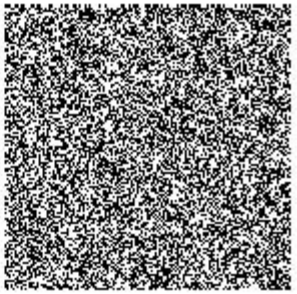
7th bit

6th bit

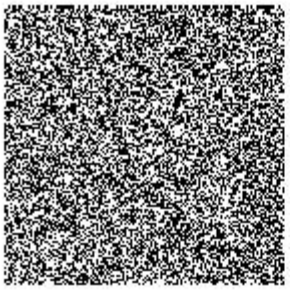
5th bit

4th bit

3rd bit



2nd bit



LSB

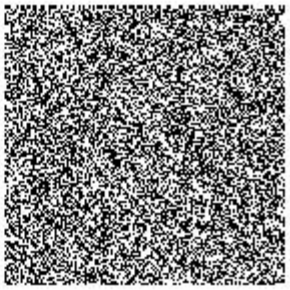


Image Averaging Results

Original image

Noisy image

Average of two

Average of three

Average of four

Average of five

Image Averaging Results

Average of six

Average of seven

Average of eight

Average of nine

Average of ten

RMS Error, E_{RMS}