Image Enhancement in the Spatial Domain

Image Enhancement

- The principal objective is to process an image so that the result is more suitable than the original image for a specific application

NOTE:

1. For visual interpretation of images, enhancement improves the subjective quality of the image.
2. In image enhancement for machine perception, the analyst is still faced with a certain trial and error before being able to settle on a particular enhancement approach.

Image Enhancement Approaches

- These approaches can be classified as
  - Spatial domain approaches
    - Involves direct manipulation of pixels in an image.

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

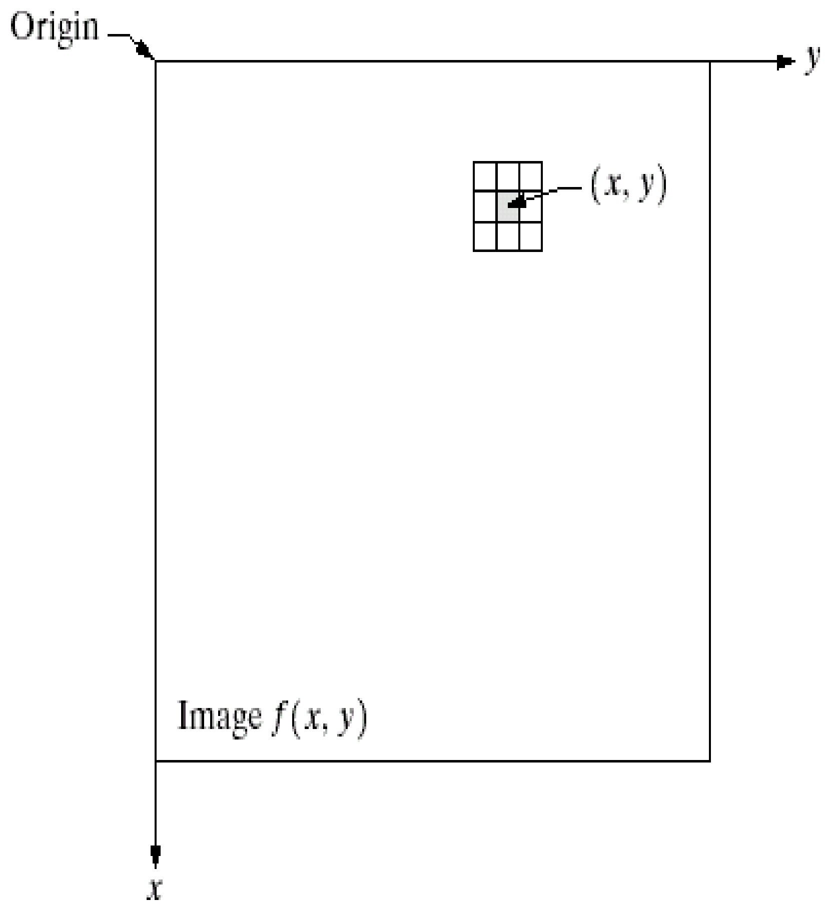
  - Frequency domain approaches
    - Involves modifying the Fourier transform of an image

Spatial Domain Enhancement

- The approaches are further classified as
  - Point processing
    - Modify the gray level of a pixel independent of the nature of its neighbors e.g. thresholding, gray level transformation
  - Neighborhood Processing
    - Small sub-images (masks) are used in local processing to modify each pixel in the image to be enhanced e.g. image sharpening, edge detection.

Spatial Domain Enhancement

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



Gray Level Transformations

- These are among the simplest of all image enhancement techniques
- The transformation (T) maps pixel value r to a pixel value s

  and is denoted by

$$s = T(r)$$

NOTE:

The values of the transformation function are typically stored in a 1D array and the mapping from r to s can be implemented via table lookup.

Gray Level Transformations

Three types of functions often used for image enhancements are

- Linear (negative and identity transformations)
- Logarithmic (log and inverse-log transformations)
- Power-law (*n*th power and *n*th root transformation)

Gray Level Transformations

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

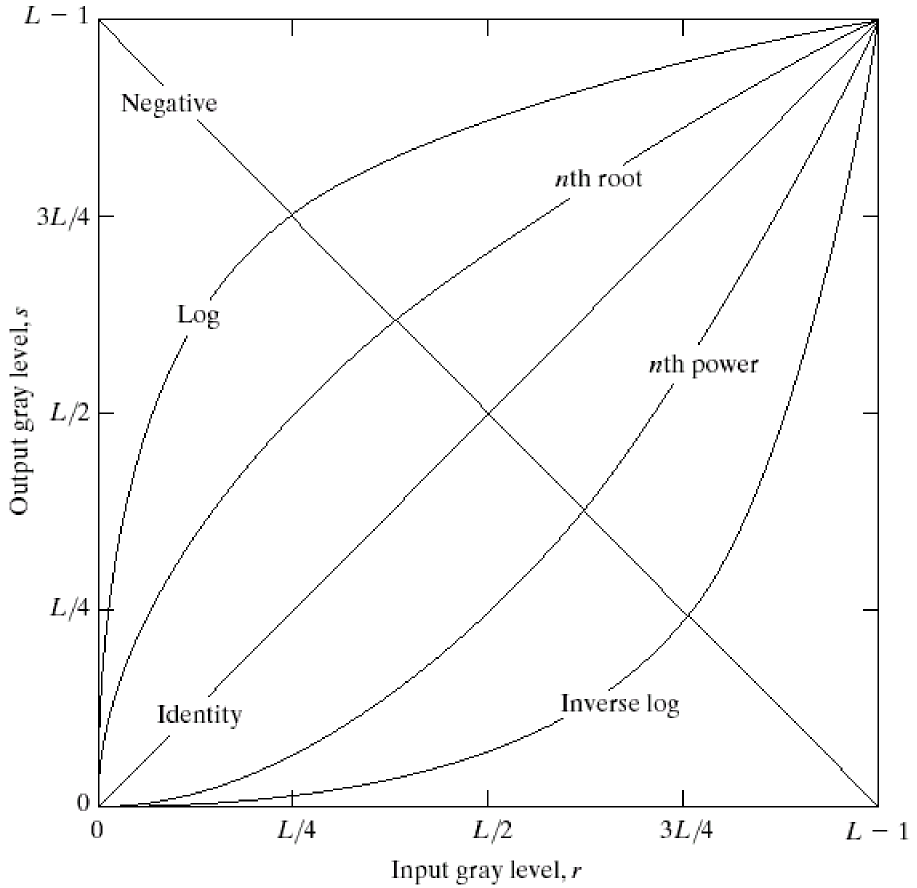


Image Negative

- The negative of an image with gray level in the range [*0*,*L*- 1] is obtained by using

$$s = L - 1 - r$$

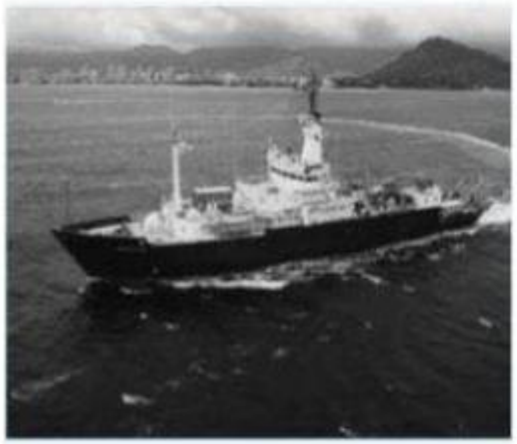
NOTE:

Reversing the intensity levels of an image in this manner produces the equivalent of a photographic negative.

  - This is particularly suited for enhancing white or gray detail embedded in the dark regions of the image, especially when the black areas are dominant.

Image Negative

Image Negative



Original Image

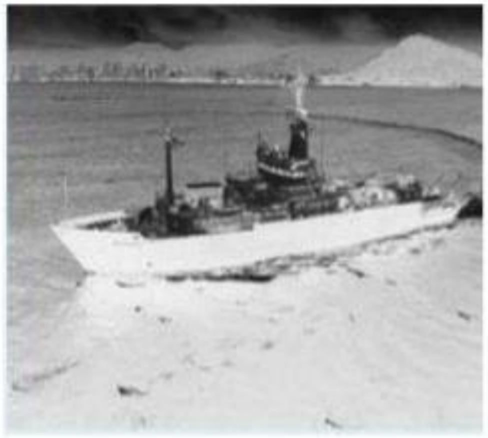


Image negative

Log Transformation

- The general form of the log transformation is

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

  where c is a constant and $r \geq 0$.

- Spreading/compressing gray levels in an image.
- The transformation maps a narrow range of low gray level input values to wider range of output levels and the opposite for higher values of input levels

  NOTE: Used to expand the values of dark pixels while compressing the brighter pixel values. The opposite is true for inverse-log transformation.

  – Compresses the dynamic range of images with large variation in pixel values

Power Law Transformation

- The transformation has the basic form of

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

  where c and γ are constants.
- $\gamma < 1$ maps a narrow range of dark input values to a wider range of output values.
- $\gamma > 1$ does the opposite.

Power Law Transformation

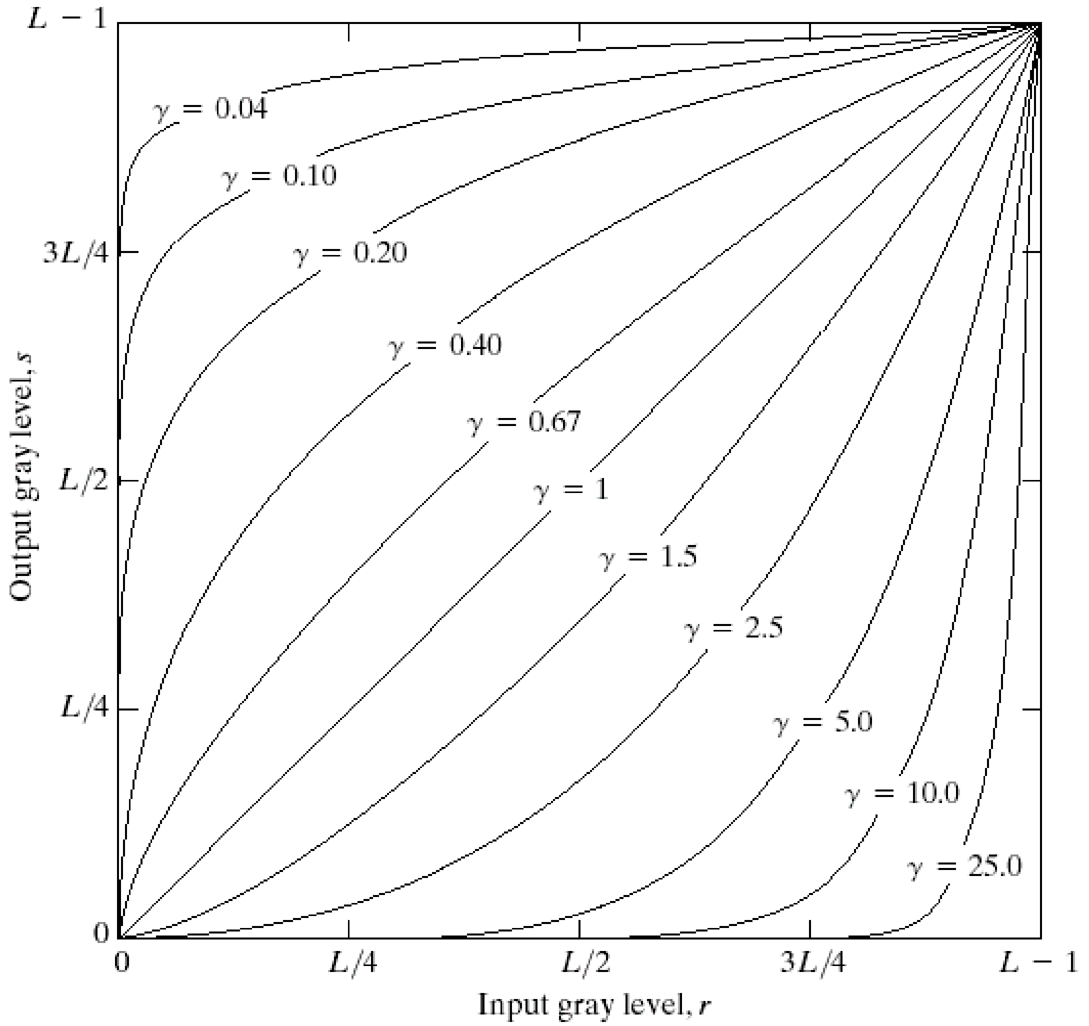


FIGURE 3.6 Plots of the equation $s = cr^\gamma$ for various values of γ ($c = 1$ in all cases).

Power Law Transformation

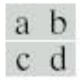
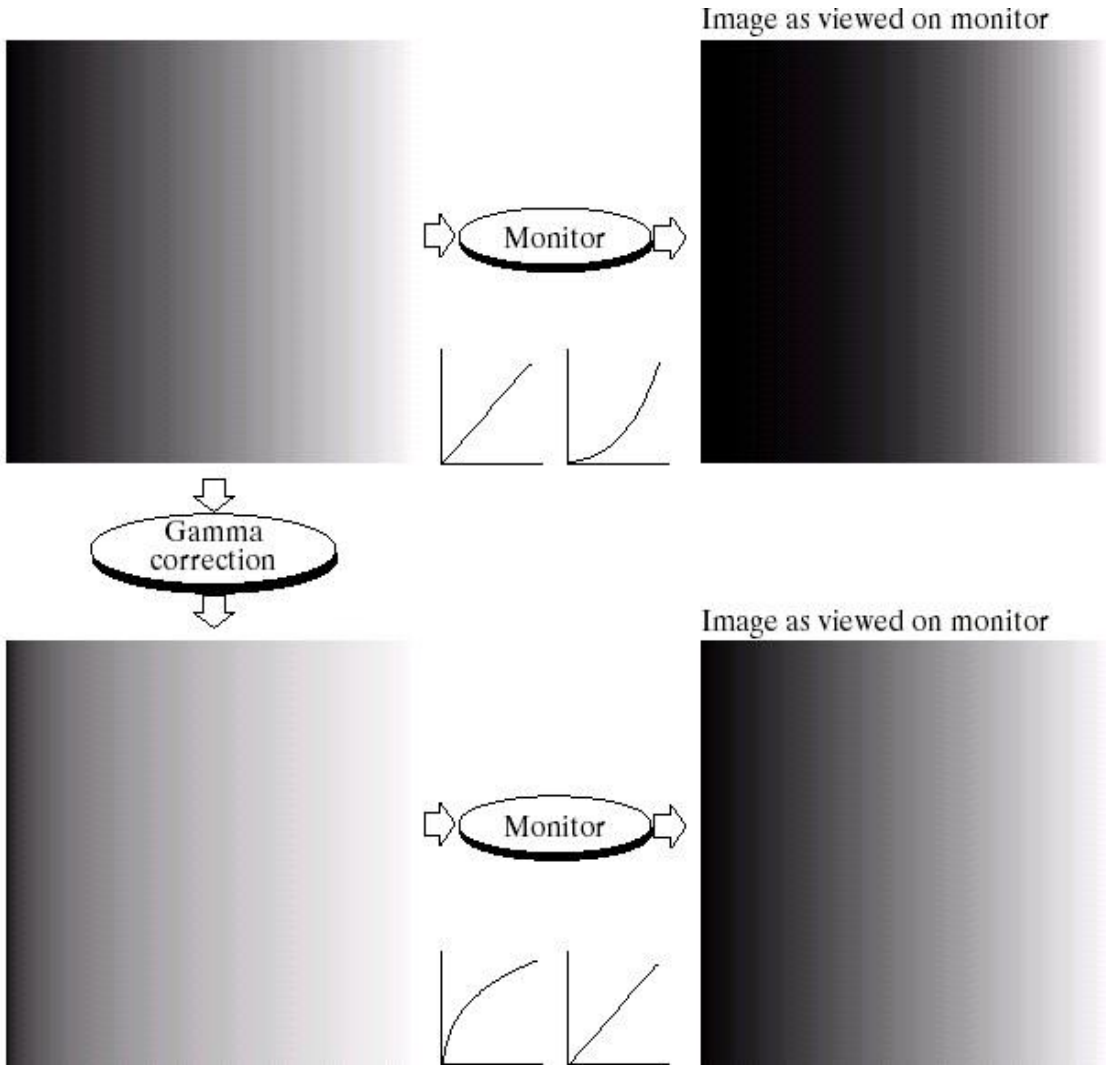
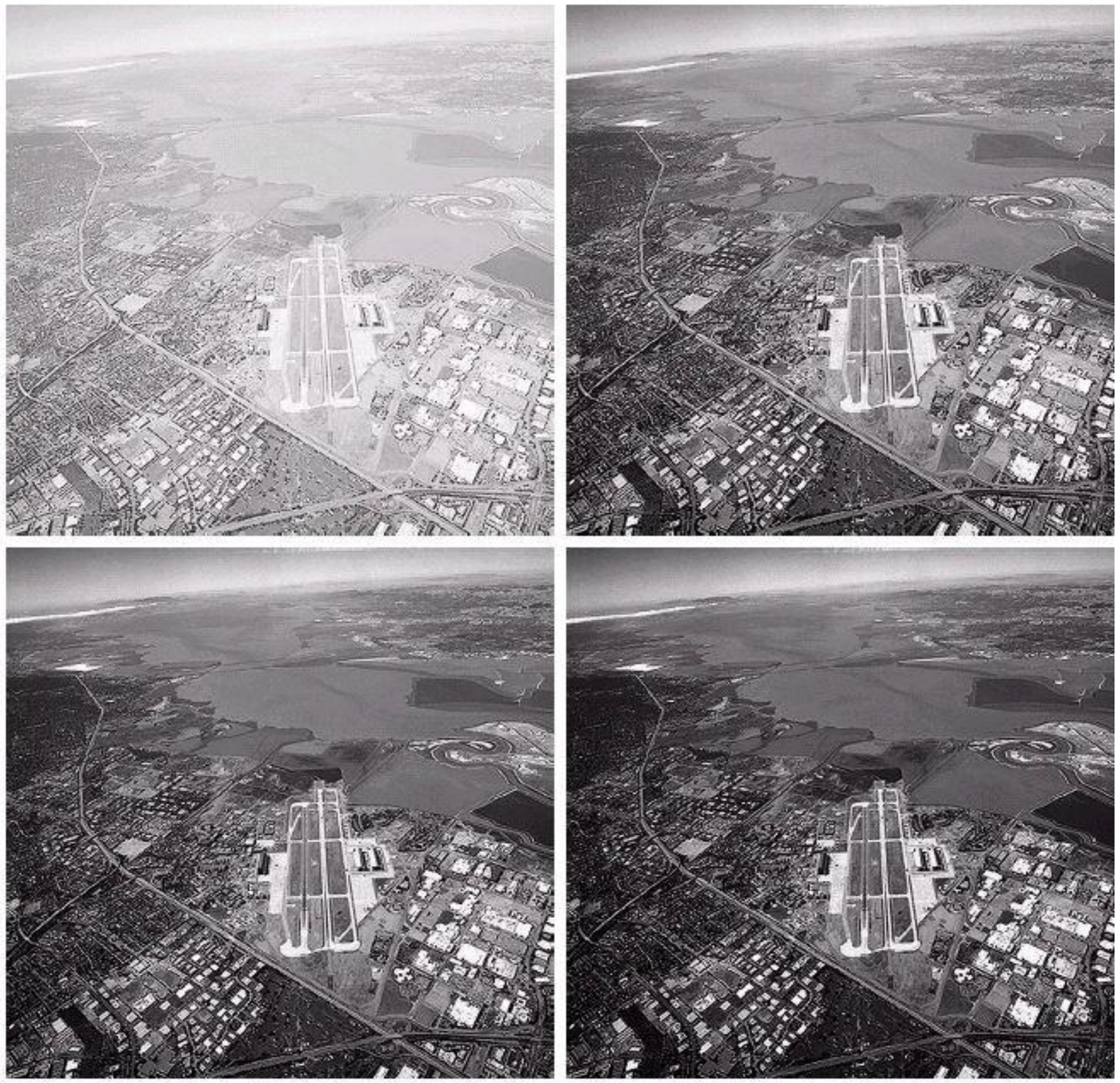


FIGURE 3.7
(a) Linear-wedge gray-scale image.
(b) Response of monitor to linear wedge.
(c) Gamma-corrected wedge.
(d) Output of monitor.



Power Law Transformation



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.)

Piece-wise Linear Transformation

- These use piecewise linear functions that offers the advantage that the form of the piecewise functions can be arbitrarily complex.
- The disadvantage of piecewise functions is that their specification requires considerably more user inputs.
- Examples:
  - Contrast stretching
  - Gray level slicing
  - Bit plane slicing

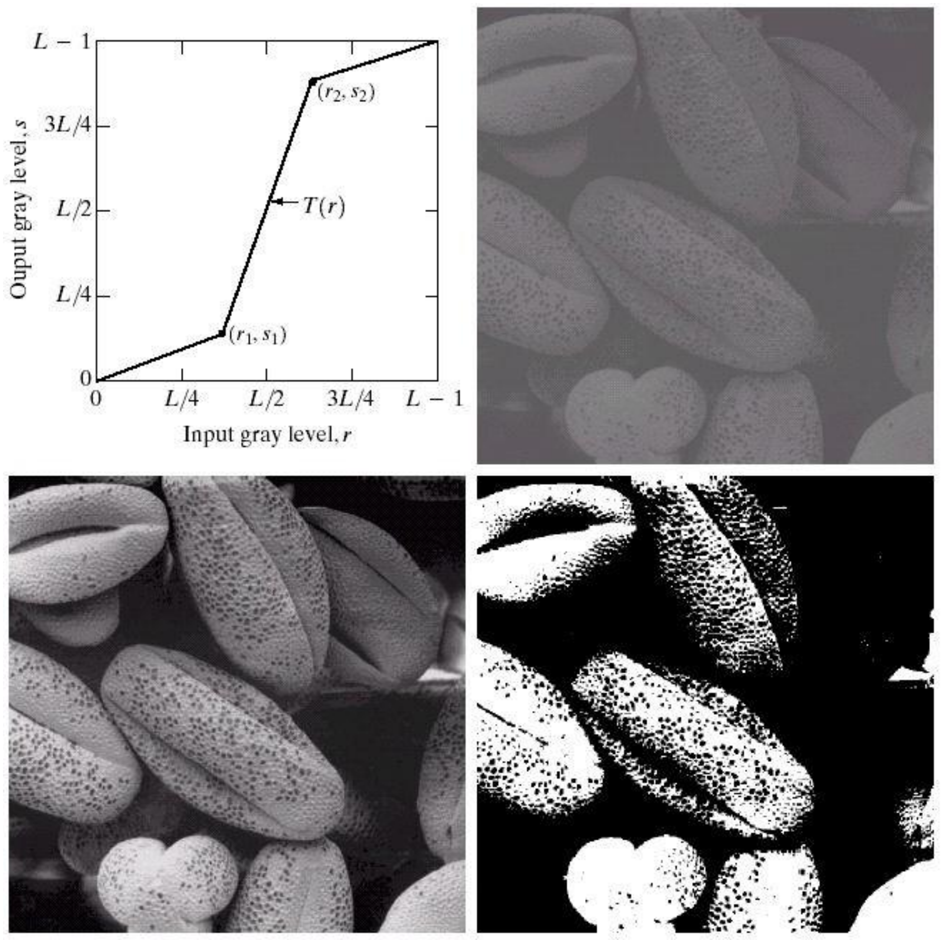
Contrast Stretching

- The idea behind contrast stretching is to increase the dynamic range of the gray levels of interest in the image being processed.
- NOTE:

  The low-contrast image can be caused due to poor illumination, lack of dynamic range of the sensor, wrong setting of the aperture, etc.

  – This general transformation can be modified to become the thresholding function.

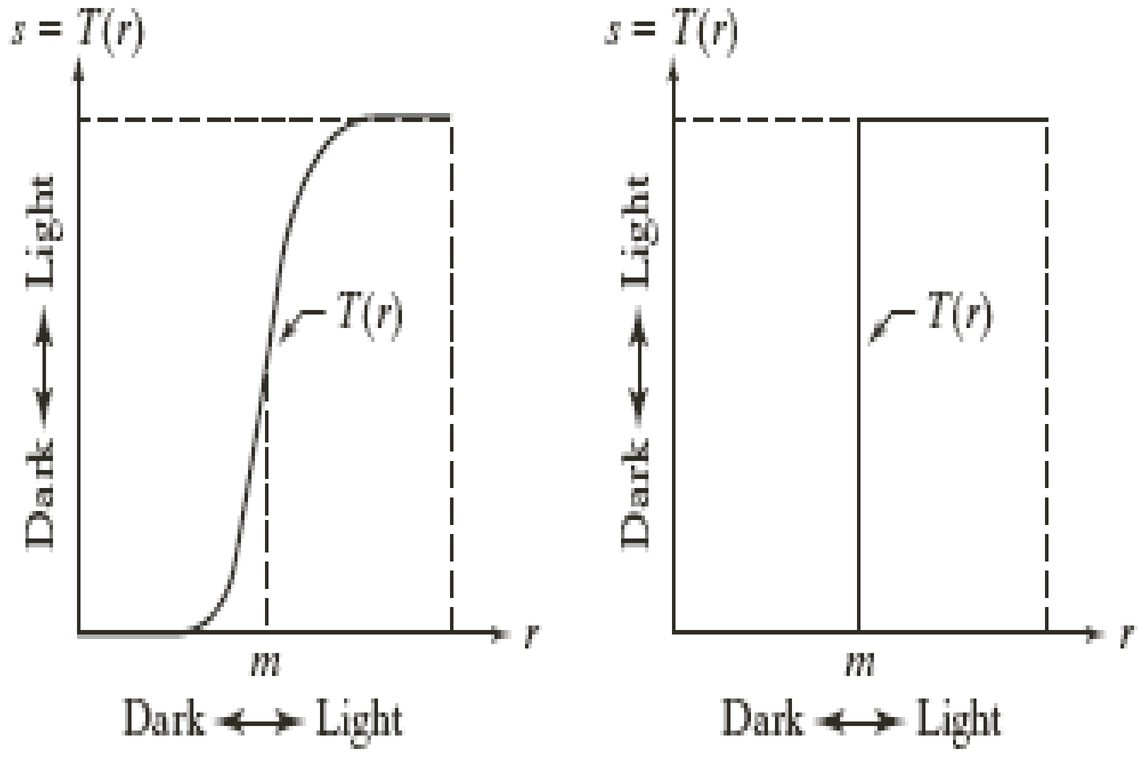
Contrast Stretching



| 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.)

Contrast Stretching



a b

FIGURE 2.4
(a) Contrast-stretching transformation.
(b) Thresholding transformation.

Gray-Level Slicing

- Is used for highlighting a specific area of gray levels in an image.
- This is achieved by:

1. Display a high value for all gray levels in the range of interest and a low value for all other gray levels.
2. Brighten the desired range of gray levels but preserve the background and gray level tonalities in the image.

NOTE: Variations of the above two transformations are also possible.

Gray-Level Slicing

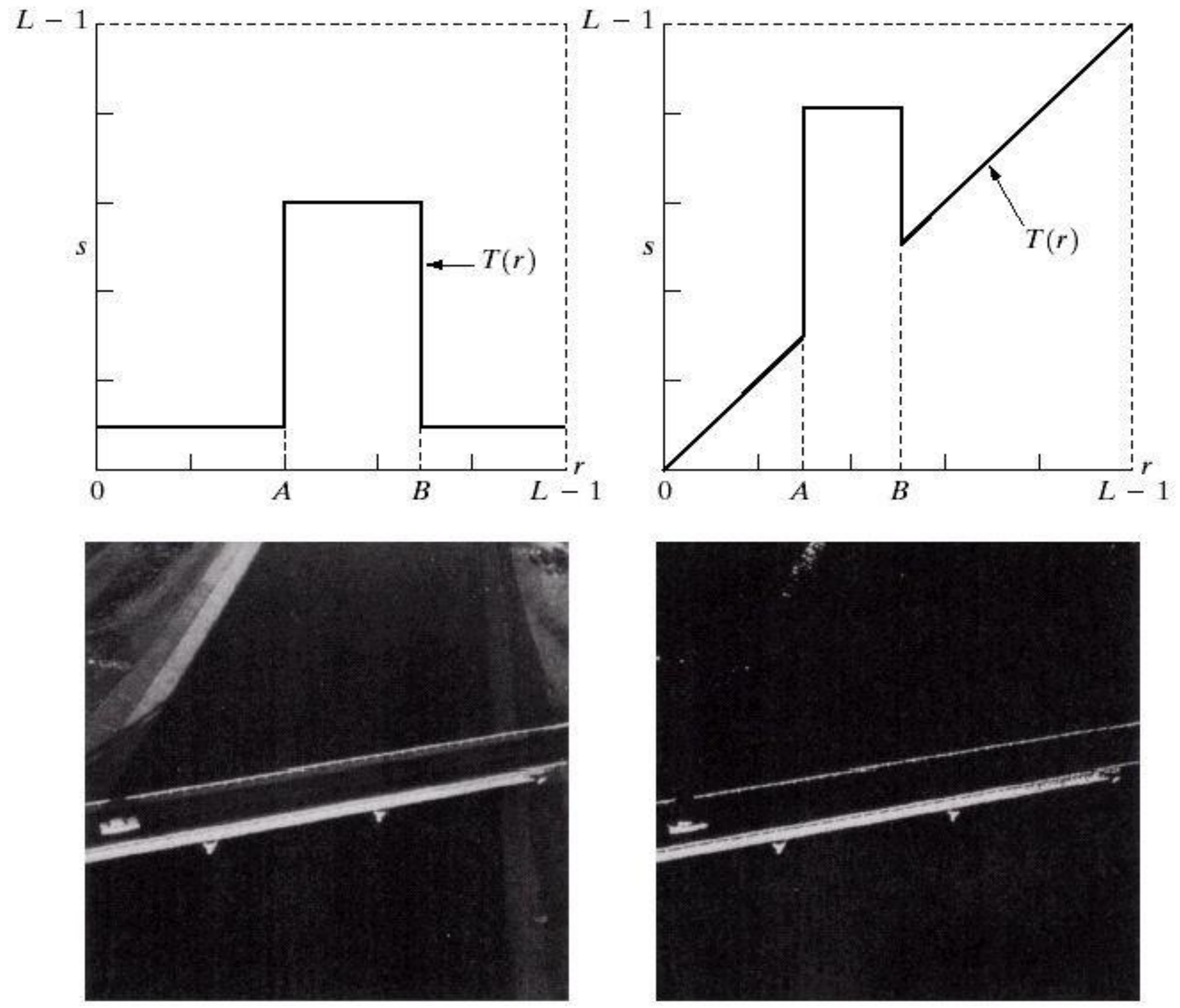


FIGURE 3.11 (a) This transformation highlights range $[A, B]$ of gray levels and reduces all others to a constant level. (b) This transformation highlights range $[A, B]$ but preserves all other levels. (c) An image. (d) Result of using the transformation in (a).

Bit Plane Slicing

- Highlighting the contribution made to the total image appearance by specific bits.
  - For an 8-bit image, eight 1-bit planes images, ranging from bit plane 0 (LSB) to bit plane 7 (MSB) are constructed.

NOTE: Higher bits contain the majority of the visually significant data and lower bit planes contribute to more subtle details in the image.

Bit Plane Slicing

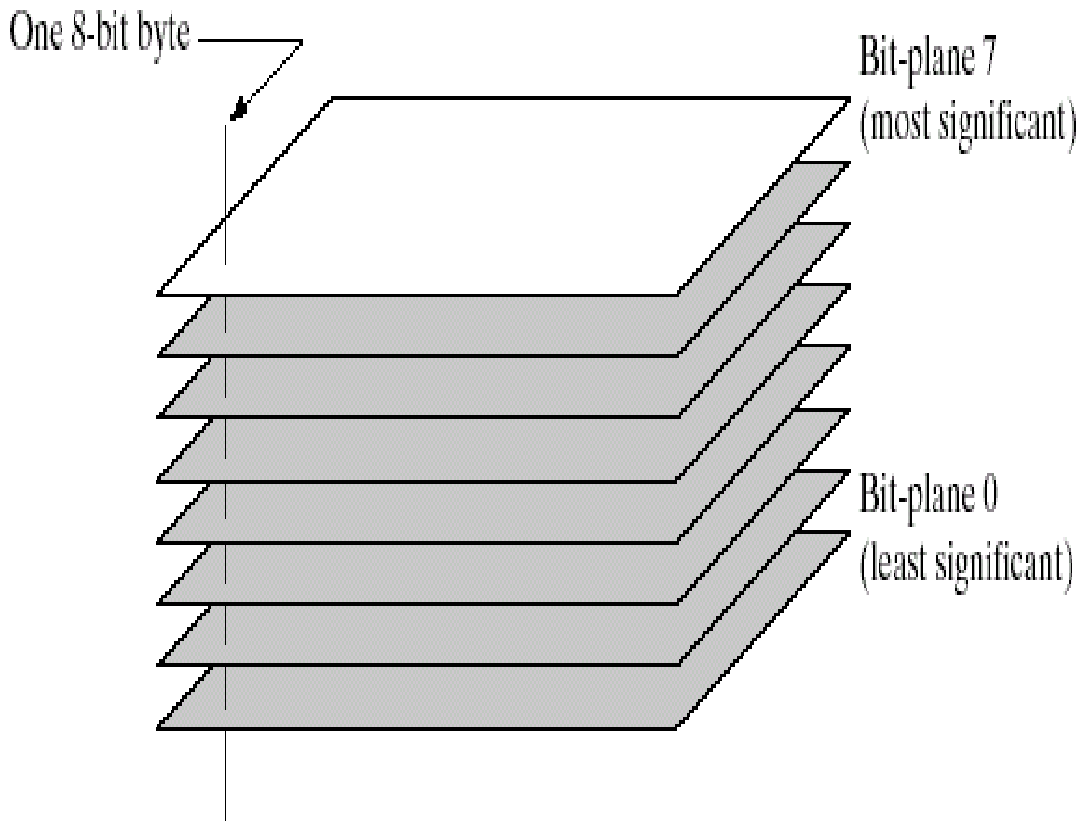


FIGURE 3.12 Bit-plane representation of an 8-bit image.

Bit Plane Slicing

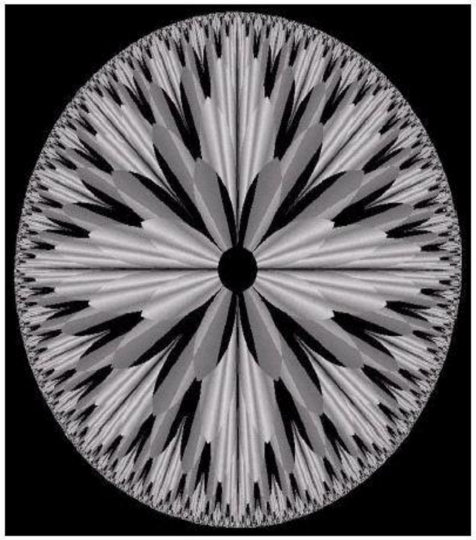


FIGURE 3.13 An 8-bit fractal image. (A fractal is an image generated from mathematical expressions). (Courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA.)

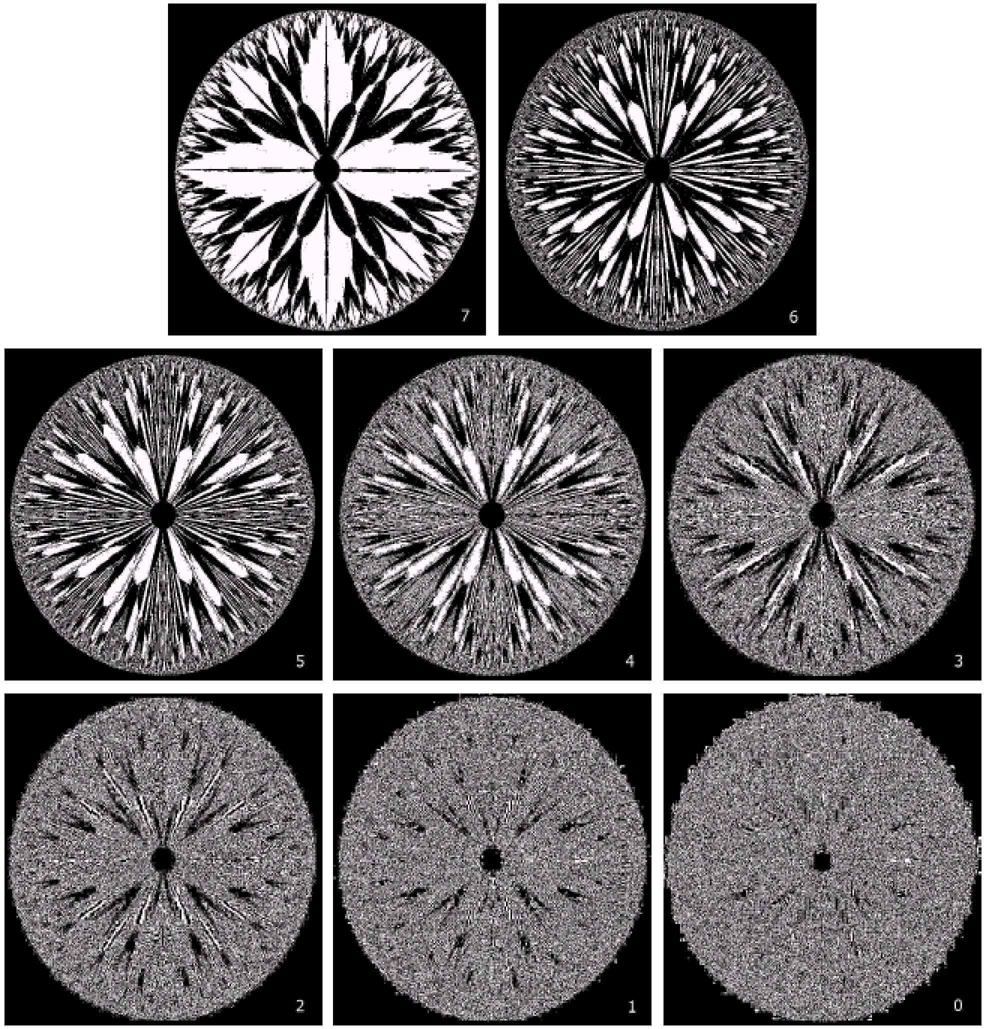


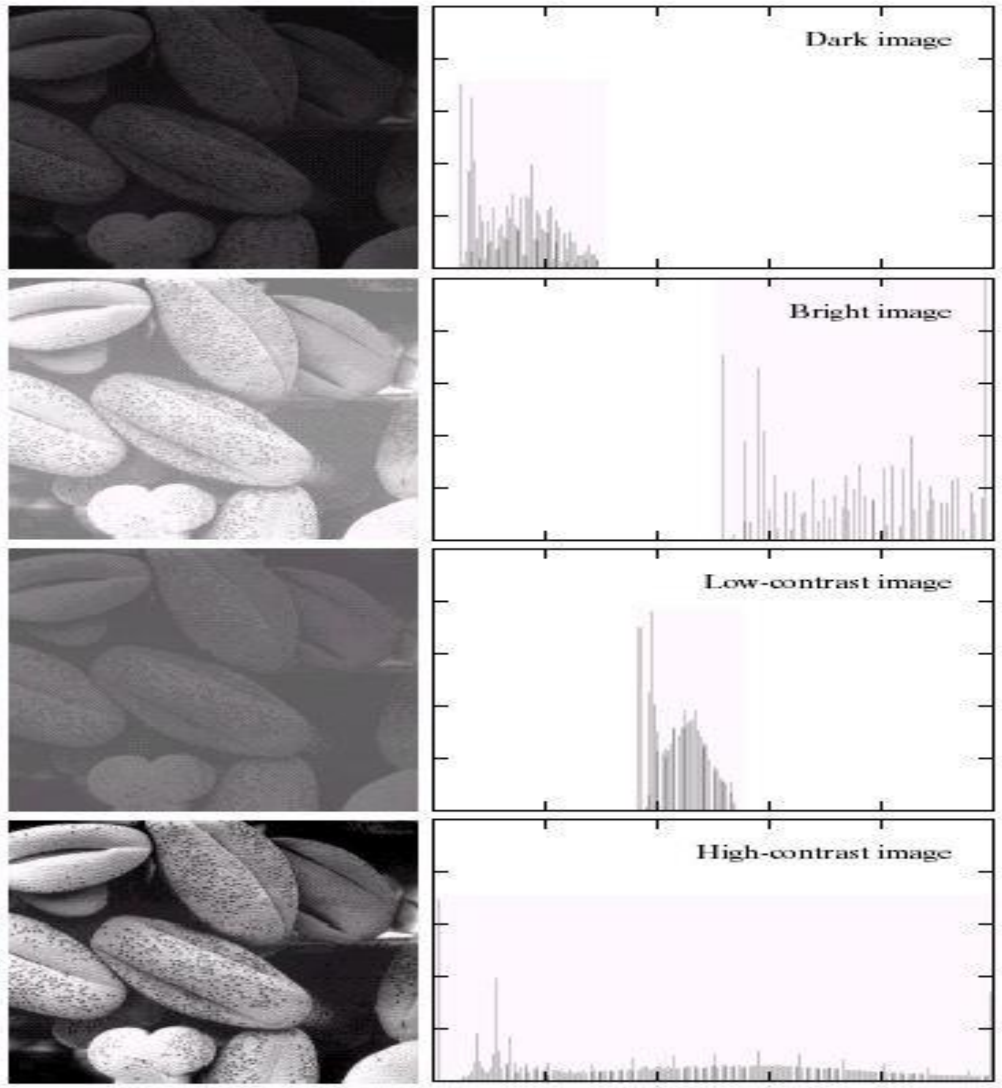
FIGURE 3.14 The eight bit planes of the image in Fig. 3.13. The number at the bottom, right of each image identifies the bit plane.

Histogram processing

- Histogram of an image '*h*' is a function that gives the number of occurrences of the gray levels in an image '*f*' i.e. *h(k)* is the number of occurrence of the gray '*k*' in the image '*f*'
- A normalized histogram is obtained by dividing *h(k)* by the total number of pixels in the image.

  NOTE: The normalized histogram can be looked upon as the probability of occurrence of the various gray levels in an image.

Histogram Processing



Four basic images types viz. dark, light, low contrast and high contrast images and their corresponding histograms

Histogram Equalization

- It is reasonable to expect that an image whose pixels tend to occupy the entire range of possible gray levels and tend to be distributed uniformly, will have an appearance of high contrast and will exhibit a large variety of gray tones.
- The probability of occurrence of gray level r_k in an image is approximated by

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

Histogram Equalization

- Histogram equalization is given by

$$s_k = T(r_k) = \sum p_r(r_j)$$

$$= \sum n_k/n \qquad k=0,1,\ldots,L\text{-}1$$

- It has a tendency of spreading the histogram of the input image so the levels in the output image spans a fuller range of gray scale.

  NOTE: $T(r_k)$ depends on $p_r(r_k)$, but the resulting $p_s(s_k)$ is always uniform like independent of the form of $p_r(r_k)$.

Computing Histogram Equalization

- The probability of occurrence of gray level r_k in an image is approximated by

$$p_r(r_k) = n_k / n \qquad k=0,1,\dots,L\text{-}1$$

- Histogram equalization is given by

$$s_k = T(r_k) = \sum p_r(r_j)$$
$$= \sum n_k/n \qquad k=0,1,\dots,L\text{-}1$$

The transformation function $T(r)$ satisfies the following conditions:

– $T(r)$ is single-valued and monotonically increasing in the interval $0 \le r \le 1$

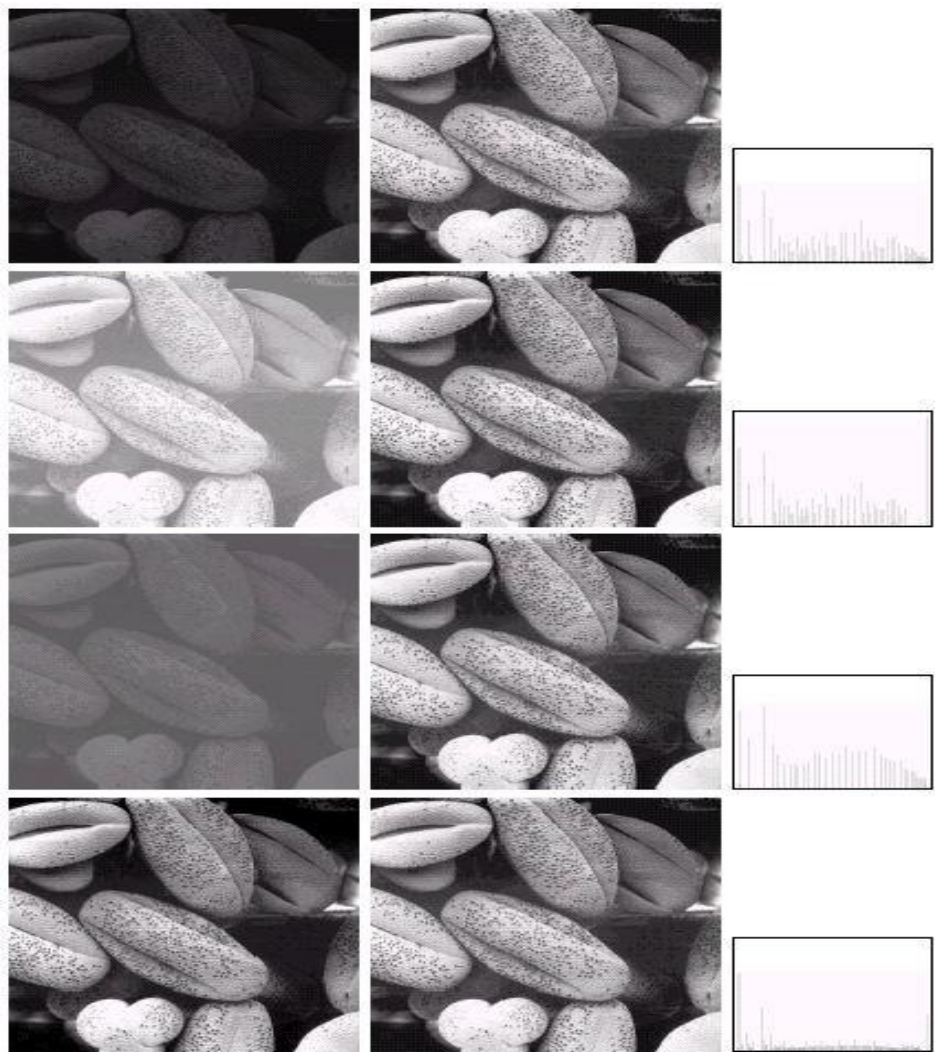
– $0 \le T(r) \le 1$ for $0 \le r \le 1$

Histogram Equalization - Example

Equalize a 10 x 10 3-bit image with the following histogram

Gray Level	No. of Pixels
0	20
1	5
2	10
3	15
4	25
5	10
6	15
7	0

Histogram Equalization



Local Histogram Equalization

- The equalization method discussed so far is *global*, they use the information of the entire image for its operation.
- However, it is sometimes necessary to enhance details over small areas or regions in an image.
- This is achieved by using a transformation function that based on properties in the neighborhood of every pixel in the image.

Local Histogram Equalization

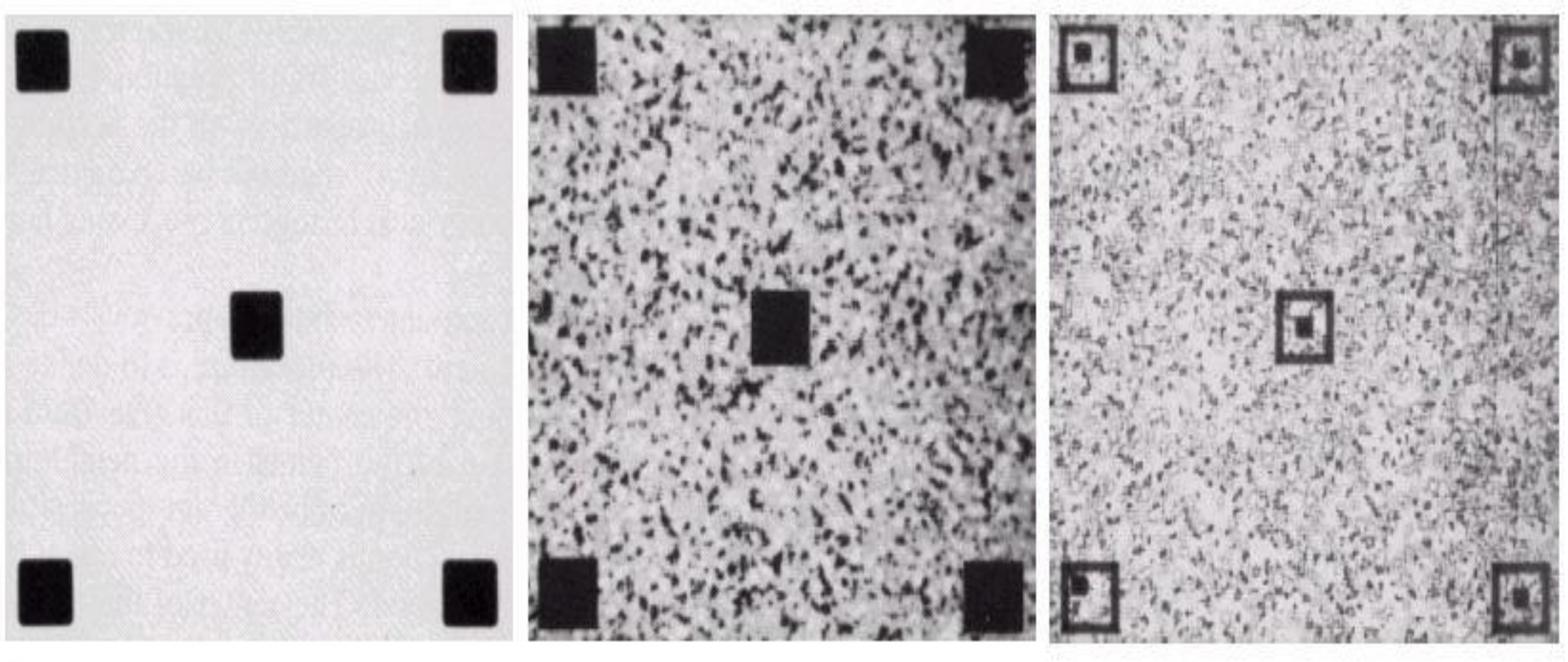


FIGURE 3.23 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization using a 7 × 7 neighborhood about each pixel.

Arithmetic/ Logical Operations

- These operations are performed on a pixel-by-pixel basis between two or more images.

  NOTE: Logical NOT can be performed on a single image.
- The actual implementation of these operations is dependent on the hardware and/or software being used i.e. it can be done sequentially or in parallel over all the pixels.

Arithmetic Operations

- These include:
  - Addition: used in image averaging to remove noise.
  - Subtraction: used for removing static background

NOTE: The dynamic range of the processed images is different from the dynamic range of the input images and one has to take care of this while creating the final image for display.

Arithmetic Operations

a b
c d

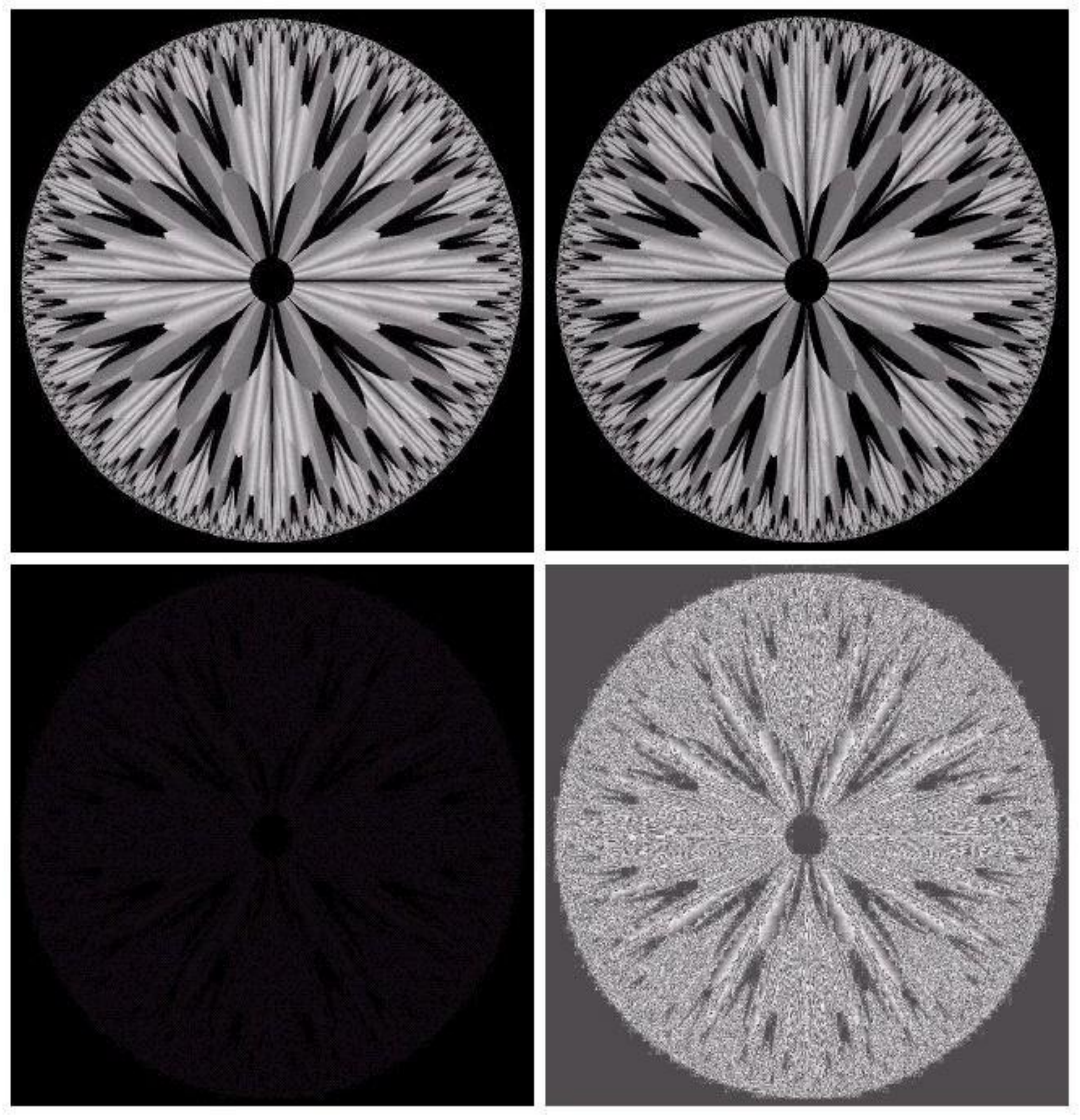


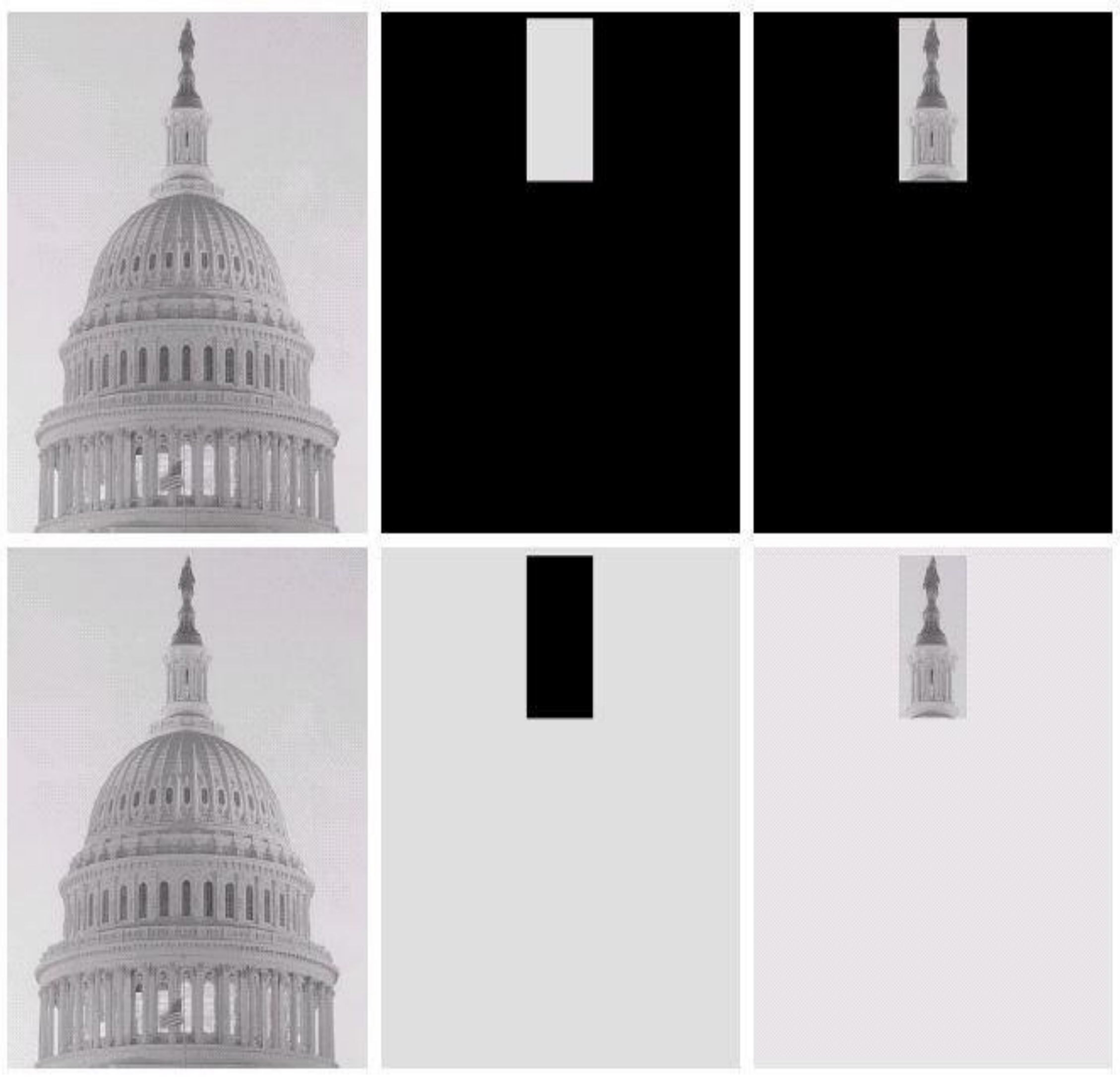
FIGURE 3.28
(a) Original fractal image.
(b) Result of setting the four lower-order bit planes to zero.
(c) Difference between (a) and (b).
(d) Histogram-equalized difference image.
(Original image courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA).

Logical Operations

- These operations are performed on a pixel-by-pixel basis between two or more images.

  NOTE: Logical NOT can be performed on a single image.
- The actual implementation of these operations is dependent on the hardware and/or software being used i.e. it can be done sequentially or in parallel over all the pixels.

Logical Operations



a	b	c
d	e	f

FIGURE 3.27 (a) Original image. (b) AND image mask. (c) Result of the AND operation on images (a) and (b). (d) Original image. (e) OR image mask. (f) Result of operation OR on images (d) and (e).

Spatial Filtering

- Some neighborhood operators work with the
  - Values of the image pixels in the neighborhood and
  - The corresponding values of a subimage that has the same dimension as the neighborhood.

NOTE: The subimage is called a *filter*, *mask*, *kernel*, *template* or *window*. The values in a filter subimage are referred to as *coefficients*, rather than pixels.

Spatial Filtering

- The process consists of simply moving the filter mask from point to point in an image.
- At each point, the *response* of the filter at that point is calculated using a predefined relationship.

  NOTE: For linear filtering, the response *R* is given by the sum of products of the filter coefficients and corresponding image pixels in the area spanned by the filter mask.

Spatial Filtering

- General linear filtering of an image of size M x N with a filter mask of size m x n is

$$R = g(x,y) = \Sigma\,\Sigma\, w(s,t)f(x+s,y+t)$$

where, $x = 0,1,2,\ldots,M-1;$ $\qquad y = 0,1,2,\ldots,N-1$

For $m = n = 3$ we get

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

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

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

- Linear spatial filtering is often referred to as “convolving a mask with an image”.

Spatial Filtering

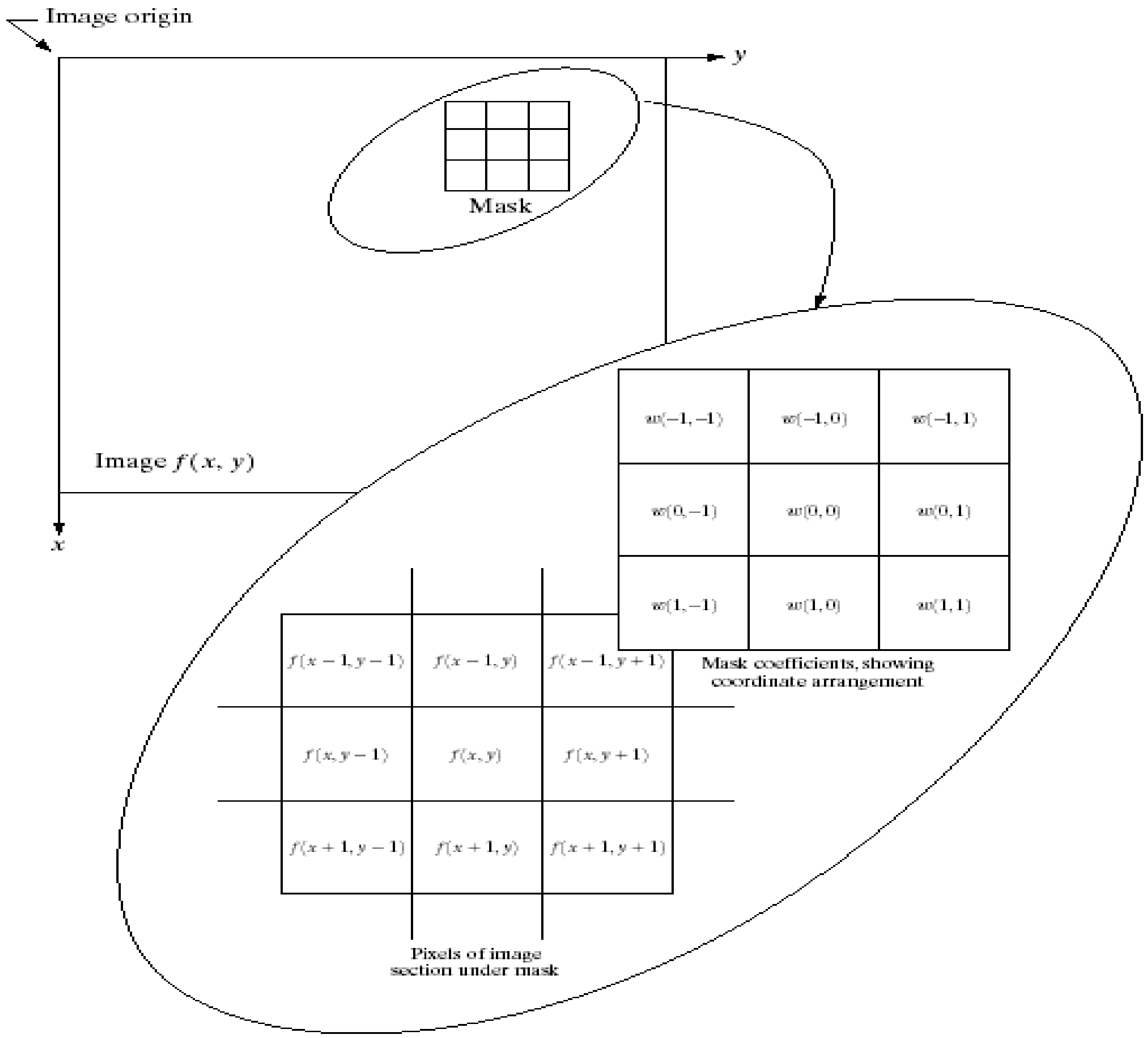


FIGURE 3.32 The mechanics of spatial filtering. The magnified drawing shows a 3×3 mask and the image section directly under it; the image section is shown displaced out from under the mask for ease of readability.

Smoothing Spatial Filters

- These are used for
  - Blurring: removal of small details in image.
  - Noise reduction.
- Smoothing linear filter output is the average of the pixels contained in the neighborhood defined by the filter mask.

  NOTE: These are sometimes called *averaging filters* or *low pass filters*.
- *Box filter* is a spatial filter with all coefficients equal

$$R = g(x,y) = 1/mn \, \Sigma \, \Sigma \, f(x+s,y+t)$$

Smoothing Spatial Filters

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

a b

FIGURE 3.34 Two 3×3 smoothing (averaging) filter masks. The constant multipli er in front of each mask is equal to the sum of the values of its coefficients, as is required to compute an average.

Smoothing Spatial Filters

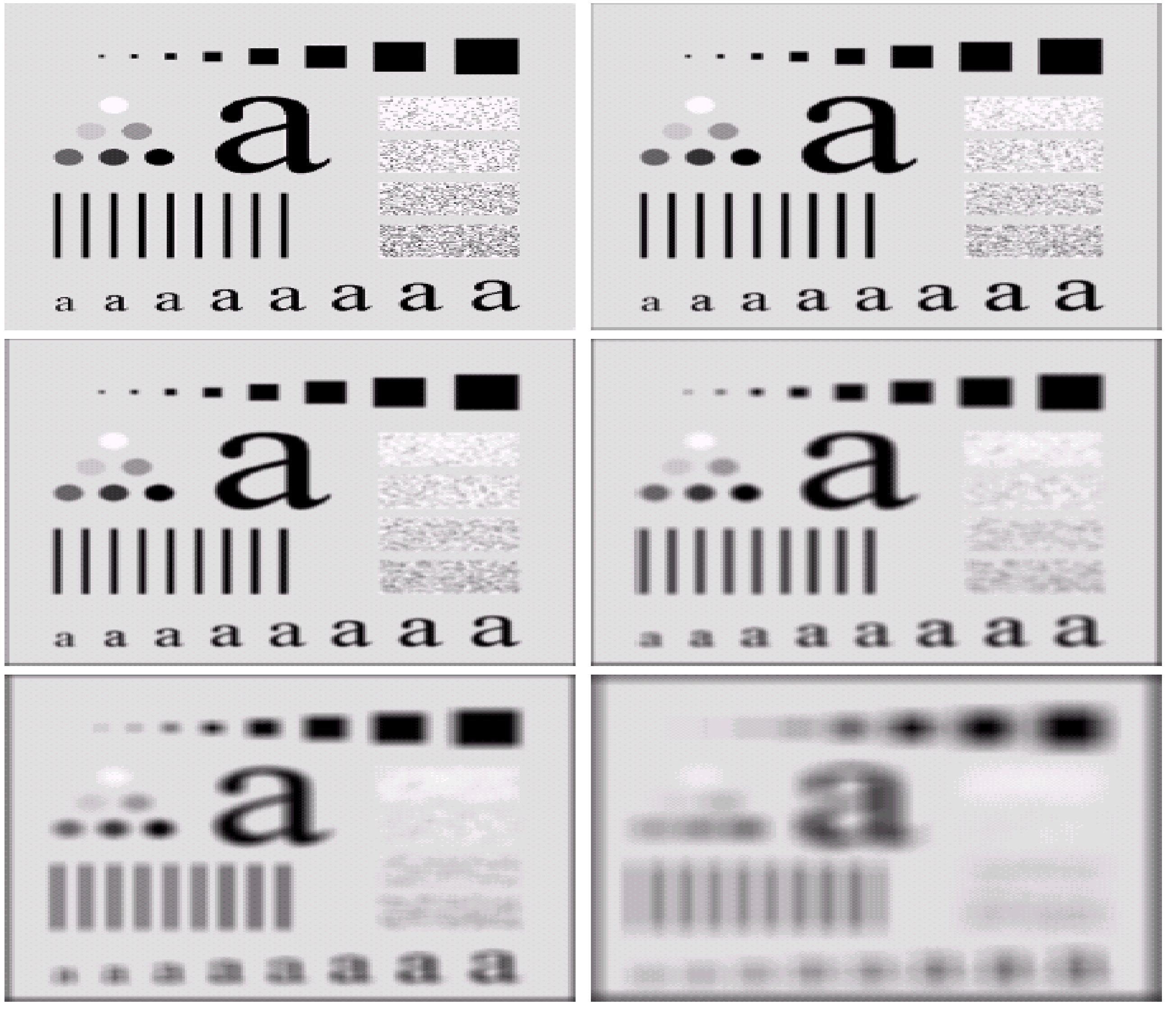
- Weighted averaging filter in which all coefficients are not equal

$$R = g(x,y) = \frac{\Sigma\,\Sigma\, w(s,t)\, f(x+s,y+t)}{\Sigma\,\Sigma\, w(s,t)}$$

where, $x = 0,1,\ldots,M\text{-}1;$ $y = 0,1,\ldots,N\text{-}1$

NOTE: The term, $\Sigma\ \Sigma\ w(s,t)$, in the denominator is called the *normalizing (scale) factor* and is applied to the entire image *after* the filtering process is completed.

Smoothing Spatial Filters



Smoothing Filters Applications

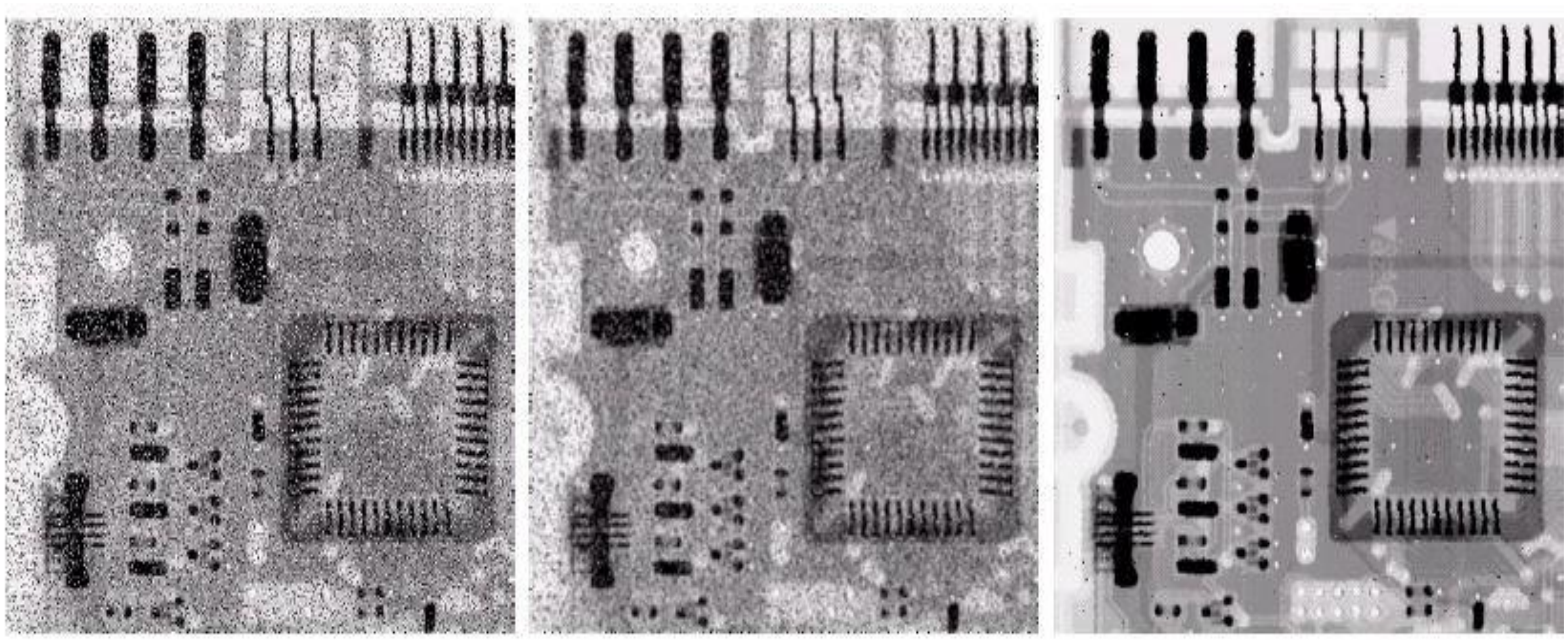
- The spatial low pass filtering can be used as a preprocessing step to
  - Extract the largest, brightest objects in an image.
  - Reducing *irrelevant* detail in an image.
  - Reducing random noise in an image.
  - Smoothing of false contours that result from using insufficient number of gray levels.

Order Statistics Filter

- These are non-linear spatial filters whose response is based on the ordering (ranking) of the pixels contained in the image area encompassed by the filter.

  NOTE: The value of the center pixel is replaced by the value determined by the ranking result.
- Examples
  - Median filter.
  - Max filter.
  - Min filter.

Order Statistics Filter



a b c

FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3 × 3 averaging mask. (c) Noise reduction with a 3 × 3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)