Image Enhancement

Module - 4

What is Image Enhancement?

- **Image Enhancement:** is the process that improves the quality of the image for a specific application.

Image Enhancement Methods

- **Spatial Domain Methods (Image Plane)**

  Techniques are based on direct manipulation of pixels in an image

- **Frequency Domain Methods**

  Techniques are based on modifying the Fourier transform of the image.

- **Combination Methods**

  There are some enhancement techniques based on various combinations of methods from the first two categories

Spatial Domain Methods

- *Spatial domain* refers to the aggregate of pixels composing an image. Spatial domain methods are procedures that operate directly on these pixels. Spatial domain processes will be denoted by the expression:

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

where, f(x,y) in the input image, g(x,y) is the processed image and ***T*** is as operator on f, defined over some neighborhood of (x,y)

- In addition, ***T*** can operate on a set of input images.

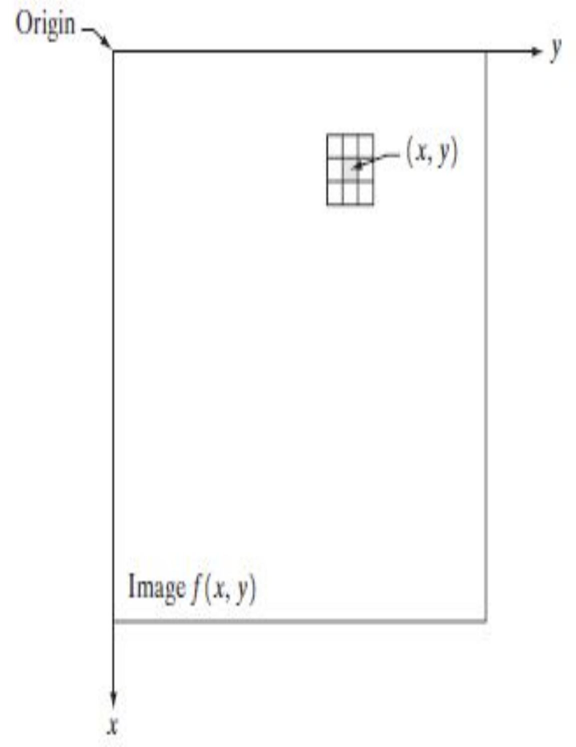
Example:

- The simplest form of ***T***, is when the neighborhood of size 1x1 (that is a single pixel). In this case, g depends only on the value of 'f' at (x,y), and ***T*** becomes a *grey-level* (also called *intensity* or *mapping*) *transformation function* of the form:

  $$s = T\,(r)$$

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

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



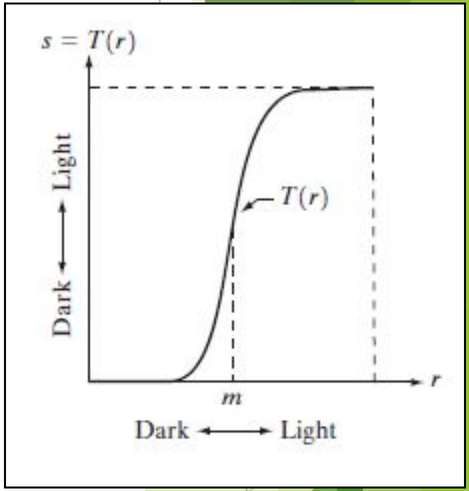
Examples of Enhancement Techniques

- **Contrast Stretching**:

  If T(r) has the form as shown in the figure below, the effect of applying the transformation to every pixel of f to generate the corresponding pixels in g would:

  Produce higher contrast than the original image, by:

  - Darkening the levels below m in the original image
  - Brightening the levels above m in the original image

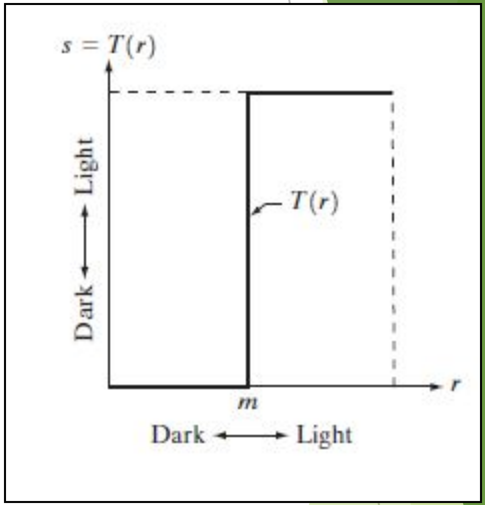


  So, Contrast Stretching: is a simple image enhancement technique that improves the contrast in an image by 'stretching' the range of intensity values it contains to span a desired range of values. Typically, it uses a linear function

Examples of Enhancement Techniques

- **Thresholding**

 Is a limited case of contrast stretching, it produces a two-level (binary) image.



 Some fairly simple, yet powerful, processing approaches can be formulated with grey-level transformations. Because enhancement at any point in an image depends only on the gray level at that point, techniques in this category often are referred to as *point processing*.

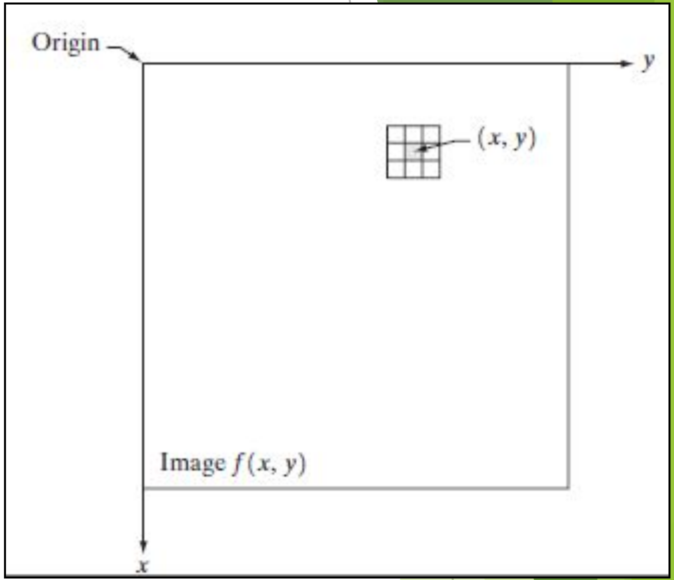
Examples of Enhancement Techniques

- Larger neighborhoods allow considerable more flexibility. The general approach is to use a function of the values of *f* in a predefined neighborhood of (x,y) to determine the value of *g* at (x,y).

- One of the principal approaches in this formulation is based on the use of so-called ***masks*** (also referred to as ***filters***)

Examples of Enhancement Techniques

So, a mask/filter: is a small (say 3X3) 2-D array, such as the one shown in the figure, in which the values of the mask coefficients determine the nature of the process, such as *image sharpening*. Enhancement techniques based on this type of approach often are referred to as



Some Basic Intensity (Gray-level) Transformation Functions

- Grey-level transformation functions (also called, intensity functions), are considered the simplest of all image enhancement techniques.
- The value of pixels, before an after processing, will be denoted by *r* and *s*, respectively. These values are related by the expression of the form:

$$s = T\,(r)$$

where T is a transformation that maps a pixel value r into a pixel value s.

Some Basic Intensity (Gray-level) Transformation Functions

Consider the following figure, which shows three basic types of functions used frequently for image enhancement:

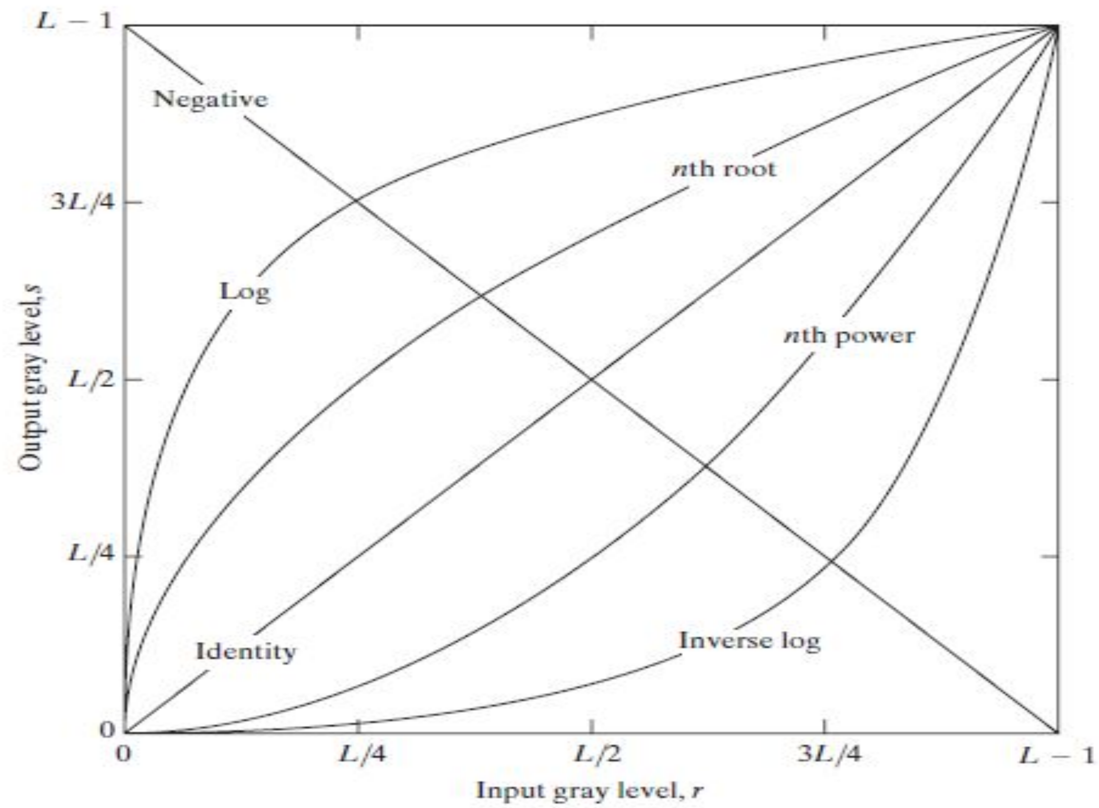


FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.

Prof.

Some Basic Intensity (Gray-level) Transformation Functions

- The three basic types of functions used frequently for image enhancement:
  - Linear Functions:
    - Negative Transformation
    - Identity Transformation
  - Logarithmic Functions:
    - Log Transformation
    - Inverse-log Transformation
  - Power-Law Functions:
    - n^{th} power transformation
    - n^{th} root transformation

Linear Functions

- **Identity Function**
  - Output intensities are identical to input intensities
  - This function doesn’t have an effect on an image, it was included in the graph only for completeness
  - Its expression:

$$s = r$$

Linear Functions

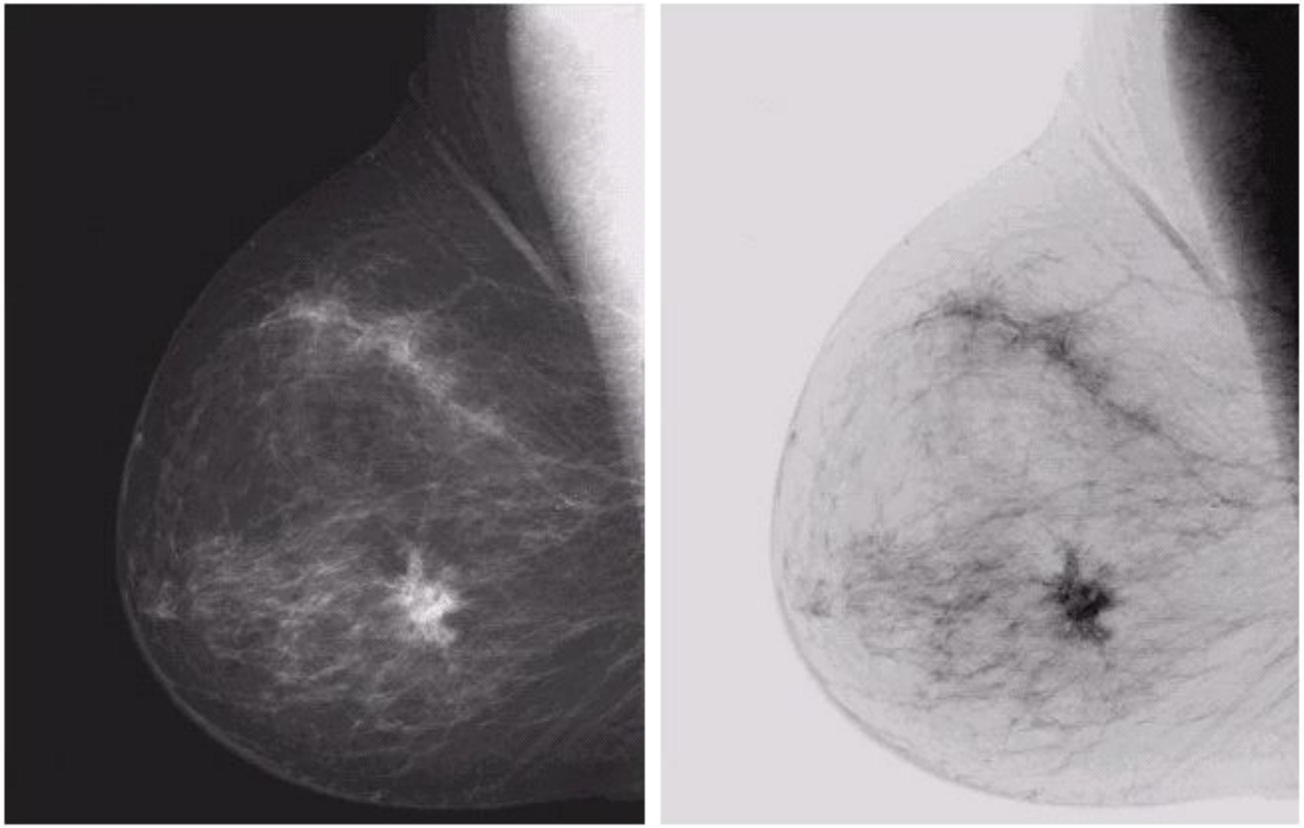
- **Image Negatives (Negative Transformation)**
  - The negative of an image with gray level in the range [0, L-1], where L = Largest value in an image, is obtained by using the negative transformation's expression:

$$s = L - 1 - r$$

Which reverses the intensity levels of an input image, in this manner produces the equivalent of a photographic negative.

  - The negative transformation is suitable for enhancing white or gray detail embedded in dark regions of an image, especially when the black area are dominant in size

Image Negative



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.)

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

Logarithmic Transformations

- **Log Transformation**

  The general form of the log transformation:

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

  Where c is a constant, and $r \geq 0$

  - Log curve maps a narrow range of low gray-level values in the input image into a wider range of the output levels.
  - Used to expand the values of dark pixels in an image while compressing the higher-level values.
  - It compresses the dynamic range of images with large variations in pixel values.

Log Transformation

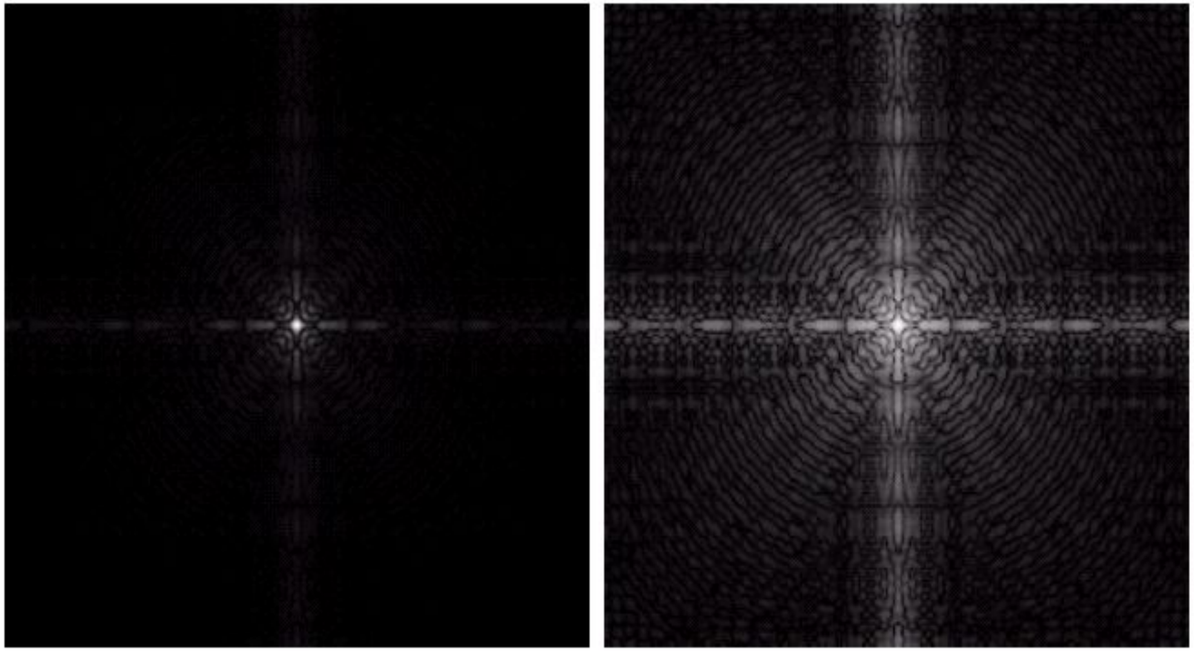
$$s = c \log(1+r)$$

c: constant

- Compresses the dynamic range of images with large variations in pixel values

a b

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



Logarithmic Transformations

- **Inverse Logarithm Transformation**
  - Do opposite to the log transformations
  - Used to expand the values of high pixels in an image while compressing the darker-level values.

Power-Law Transformations

- Power-law transformations have the basic form of:

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

  Where c and γ are positive constants

Power-Law Transformations

- Different transformation curves are obtained by varying γ (gamma)

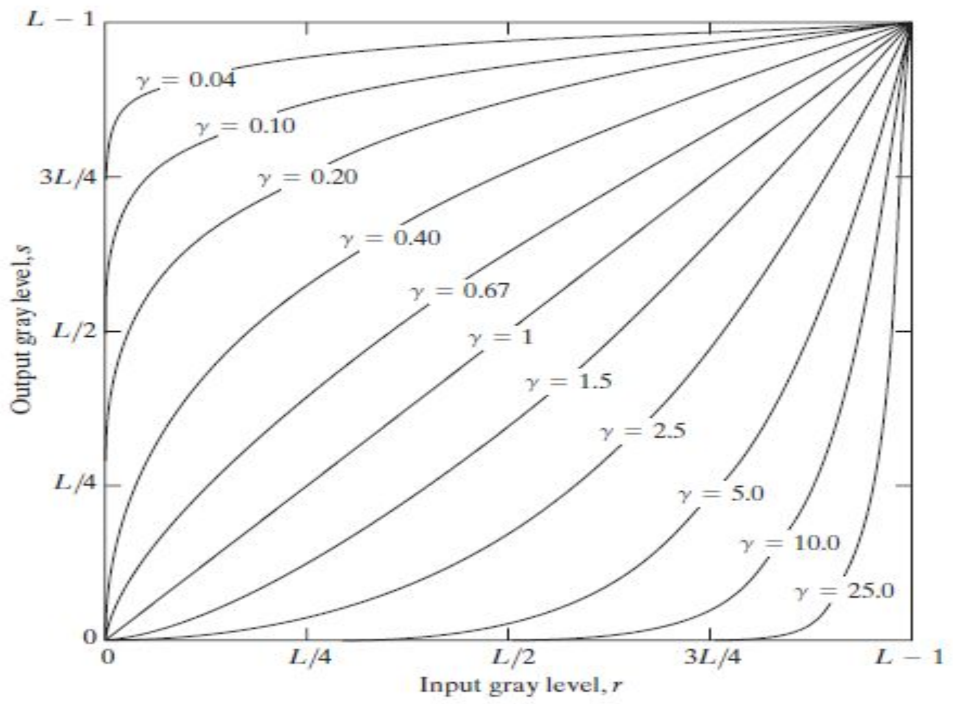


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

Power-Law Transformations

- Variety of devices used for **image capture, printing and display** respond according to a power law. The process used to correct this power-law response phenomena is called ***gamma correction***.

- For example, cathode ray tube (CRT) devices have an intensity-to-voltage response that is a power function, with exponents varying from approximately 1.8 to 2.5.With reference to the curve for g=2.5 in Fig. 3.6, we see that such display systems would tend to produce images that are darker than intended. This effect is illustrated in Fig. 3.7. Figure 3.7(a) shows a simple gray-scale linear wedge input into a CRT monitor. As expected, the output of the monitor appears darker than the input, as shown in Fig. 3.7(b). Gamma correction

  in this case is straightforward. All we need to do is preprocess the input image before inputting it into the monitor by performing the transformation. The result is shown in Fig. 3.7(c).When input into the same monitor, this gamma-corrected input produces an output that is close in appearance to the original image, as shown in Fig. 3.7(d).

Power-Law Transformation

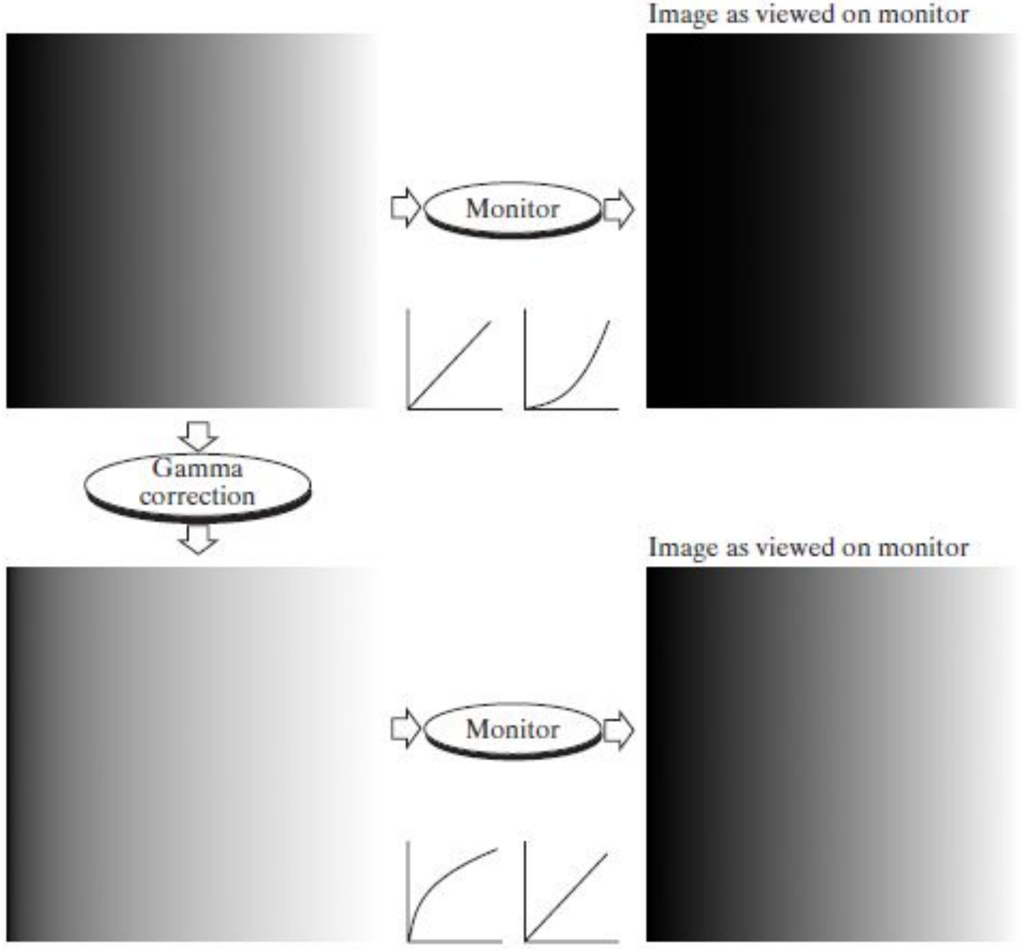
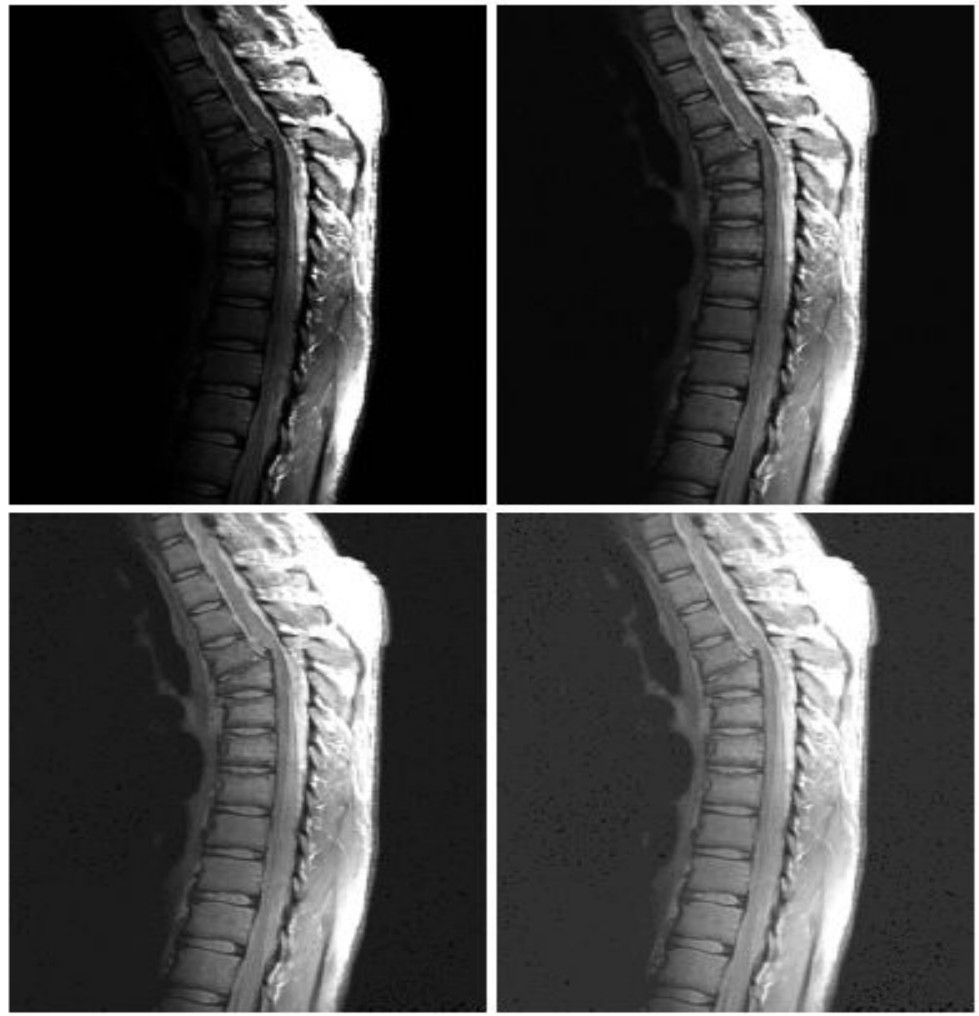


FIGURE 3.7
(a) Linear-wedge gray-scale image.
(b) Response of monitor to linear wedge.
(c) Gamma-corrected wedge.
(d) Output of monitor.

Power-Law Transformation

- In addition to
  gamma
  correction,
  power-law
  transformation
  s
  are useful for
  general-purpos
  e
  contrast
  manipulation.
  See figure 3.8



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 Transformation

- Another illustration of 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.)

Piecewise-Linear Transformation Functions

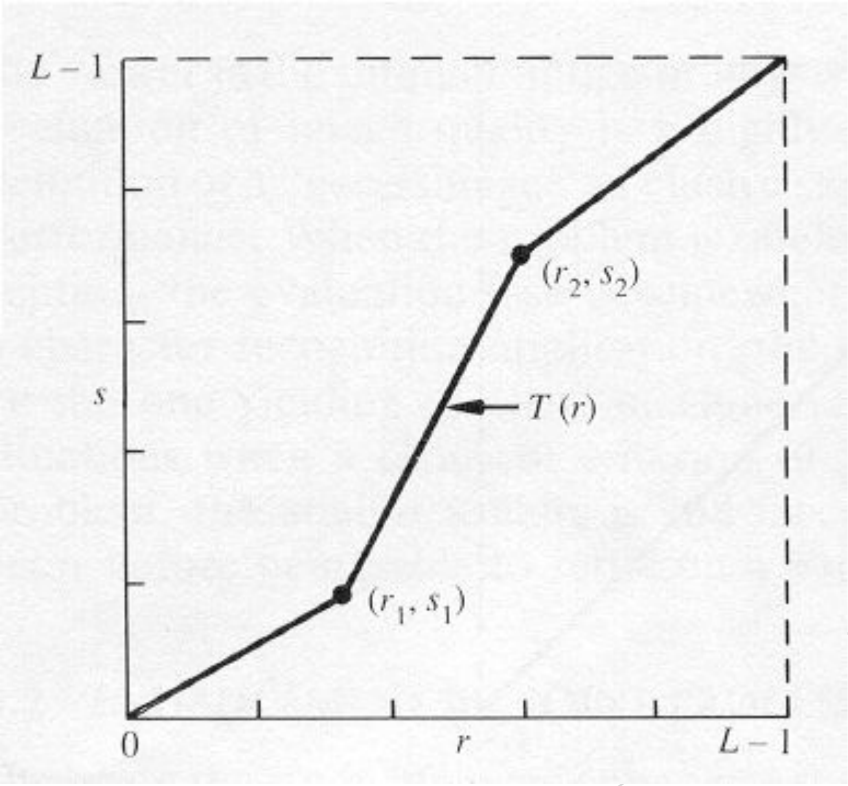
- **Principle Advantage:** Some important transformations can be formulated only as a piecewise function.
- **Principle Disadvantage:** Their specification requires more user input that previous transformations
- **Types of Piecewise transformations are:**
  - Contrast Stretching
  - Gray-level Slicing
  - Bit-plane slicing

Contrast Stretching

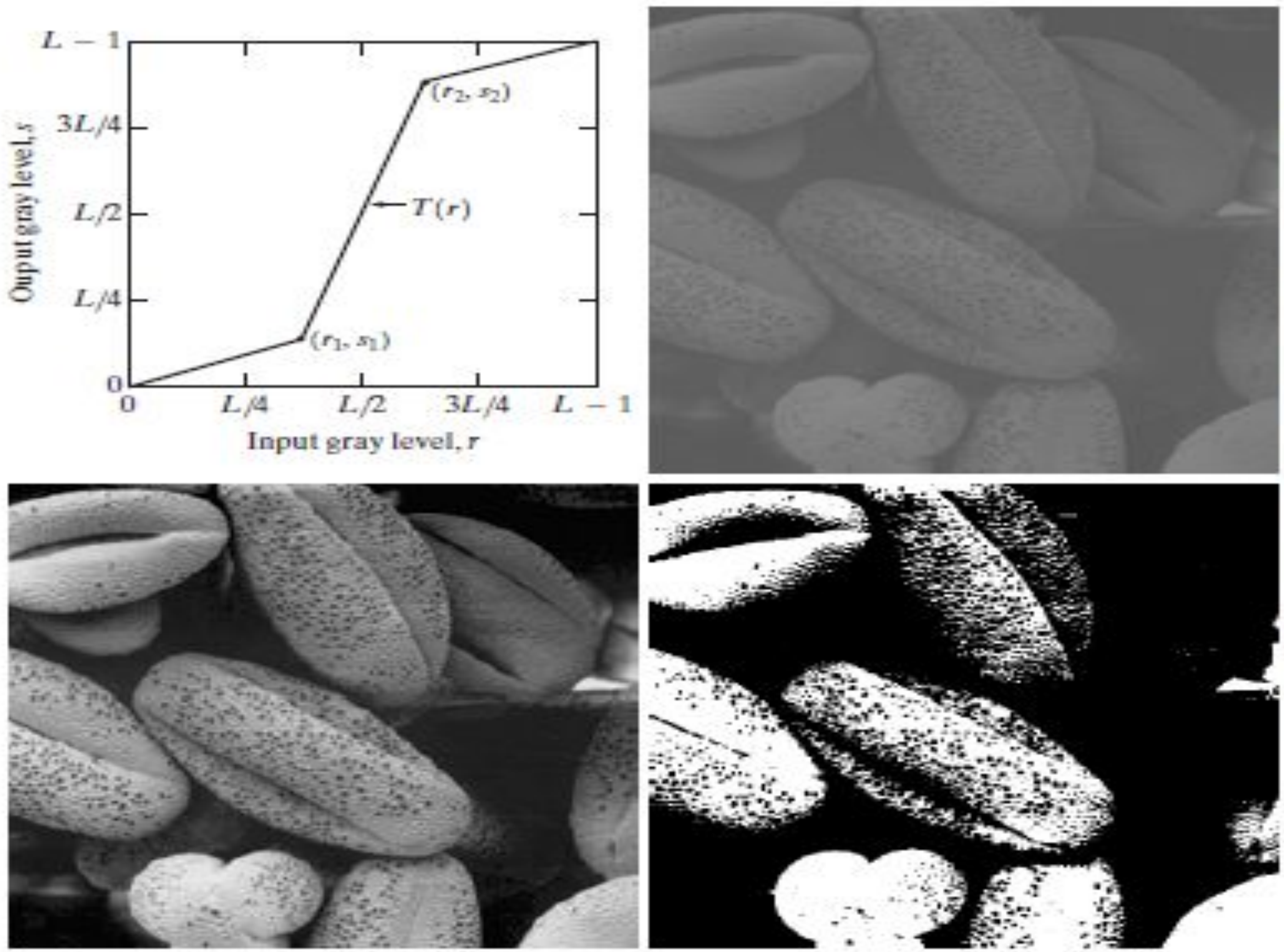
- One of the simplest piecewise linear functions is a contrast-stretching transformation, which is used to enhance the low contrast images.
- Low contrast images may result from:
  - Poor illumination
  - Wrong setting of lens aperture during image acquisition.

Contrast Stretching

- To increase the dynamic range of the gray levels in the image being processed.



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

- Figure 3.10(a) shows a typical transformation used for contrast stretching. The locations of points (r1, s1) and (r2, s2) control the shape of the transformation function.
- If r1 = s1 and r2 = s2, the transformation is a linear function that produces no changes in gray levels.
- If r1 = r2, s1 = 0 and s2 = L-1, the transformation becomes a ***thresholding function*** that creates a binary image. As shown previously in slide 7.
- Intermediate values of (r1, s1) and (r2, s2) produce various degrees of spread in the gray levels of the output image, thus affecting its contrast.
- In general, r1 ≤ r2 and s1 ≤ s2 is assumed, so the function is always increasing.

Contrast Stretching

- Figure 3.10(b) shows an 8-bit image with low contrast.

- Fig. 3.10(c) shows the result of contrast stretching, obtained by setting (r1, s1) = (r_{min}, 0) and (r2, s2) = (r_{max},L-1) where r_{min} and r_{max} denote the minimum and maximum gray levels in the image, respectively. Thus, the transformation function stretched the levels linearly from their original range to the full range [0, L-1].

- Finally, Fig. 3.10(d) shows the result of using the ***thresholding function*** defined previously, with r1=r2=m, the mean gray level in the image.

Gray-level Slicing

- This technique is used to highlight a specific range of gray levels in a given image. It can be implemented in several ways, but the two basic themes are:
  - One approach is to display a high value for all gray levels in the range of interest and a low value for all other gray levels. This transformation, shown in Fig 3.11 (a), produces a binary image.
  - The second approach, based on the transformation shown in Fig 3.11 (b), brightens the desired range of gray levels but preserves gray levels unchanged.
  - Fig 3.11 (c) shows a gray scale image, and fig 3.11 (d) shows the result of using the transformation in Fig 3.11 (a).

Gray-level Slicing

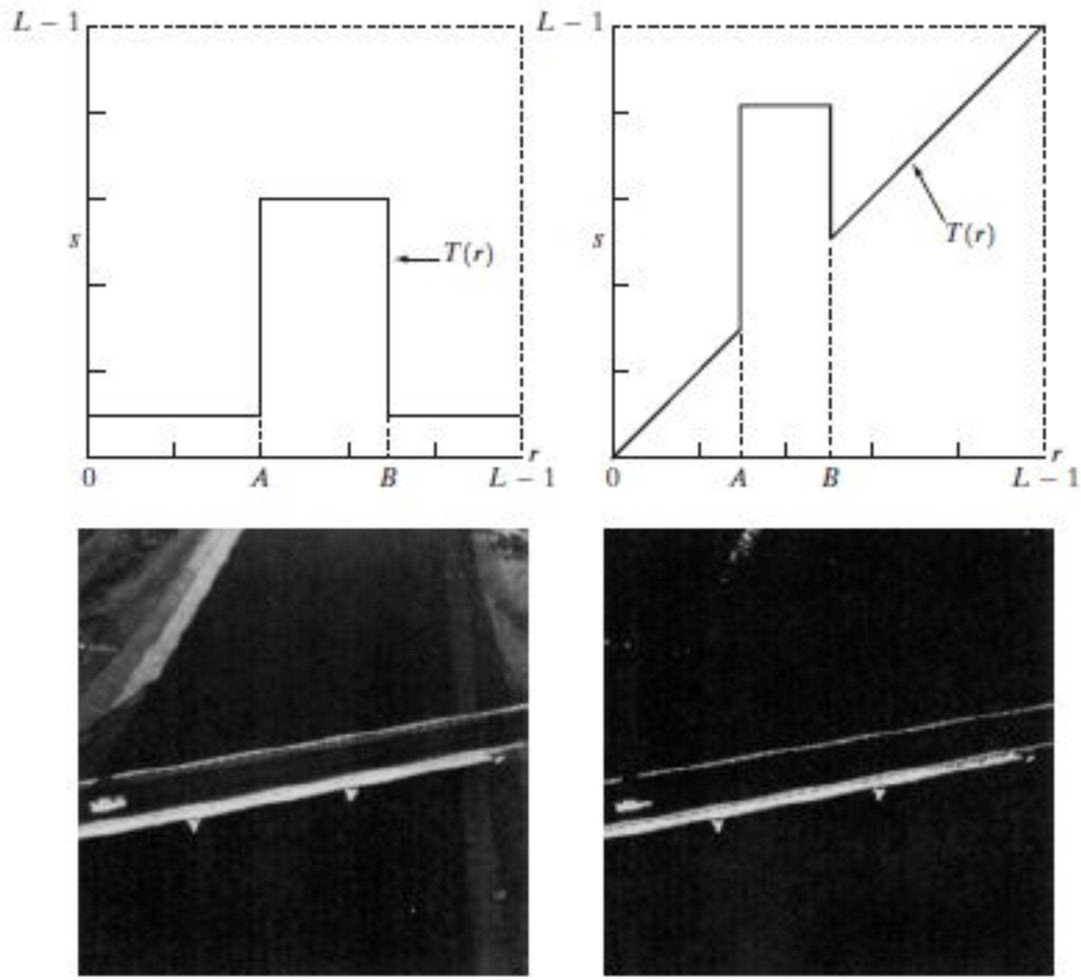


FIGURE 3.11 (a) This transformation highlights range [A, B] of gray levels and reduces all others to a constant level. (b) This transformation highlights range [A, B] but preserves all other levels. (c) An image. (d) Result of using the transformation in (a).

Bit-plane Slicing

- Pixels are digital numbers, each one composed of bits. Instead of highlighting gray-level range, we could highlight the contribution made by each bit.
- This method is useful and used in image compression.

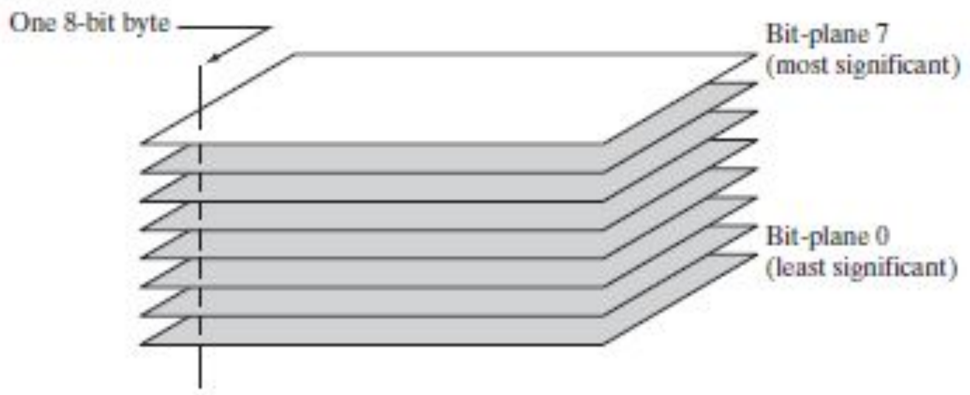


FIGURE 3.12 Bit-plane representation of an 8-bit image.

- Most significant bits contain the majority of visually significant data.

Bit-Plane Slicing

- To highlight the contribution made to the total image appearance by specific bits.
  - i.e. Assuming that each pixel is represented by 8 bits, the image is composed of 8 1-bit planes.
  - Plane 0 contains the least significant bit and plane 7 contains the most significant bit.
  - Only the higher order bits (top four) contain visually significant data. The other bit planes contribute the more subtle details.

Illustration

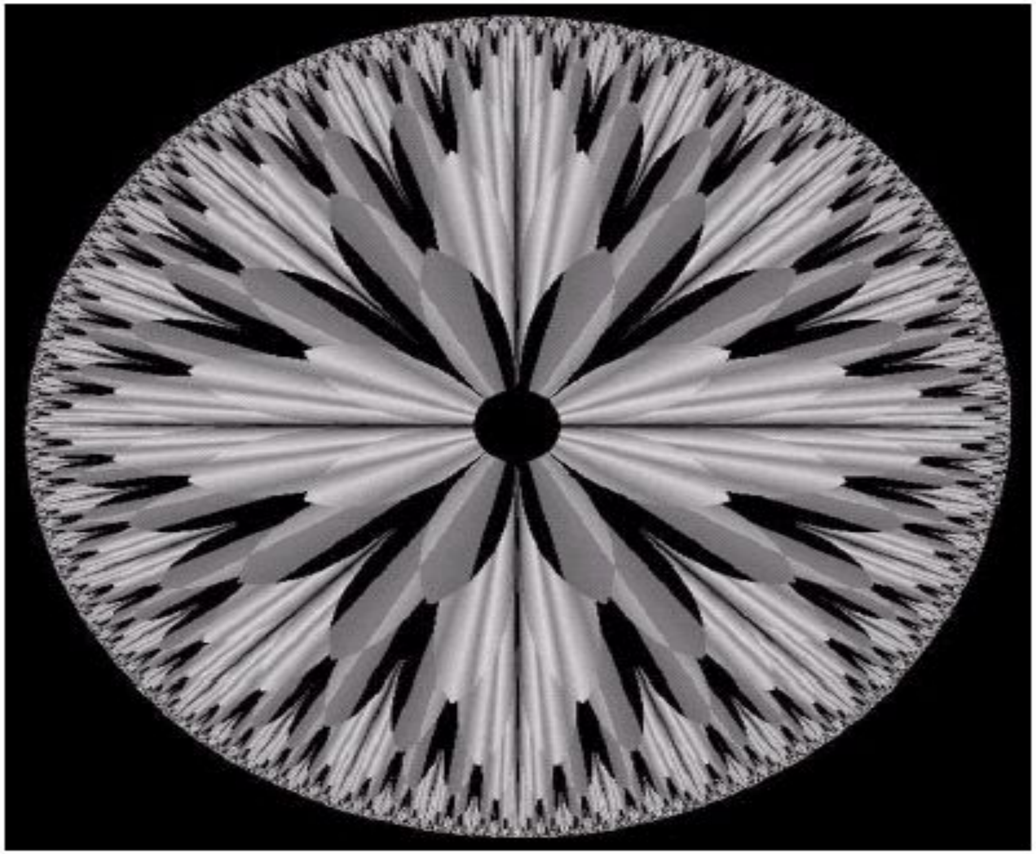


FIGURE 3.13 An 8-bit fractal image. (A fractal is an image generated from mathematical expressions). (Courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA.)

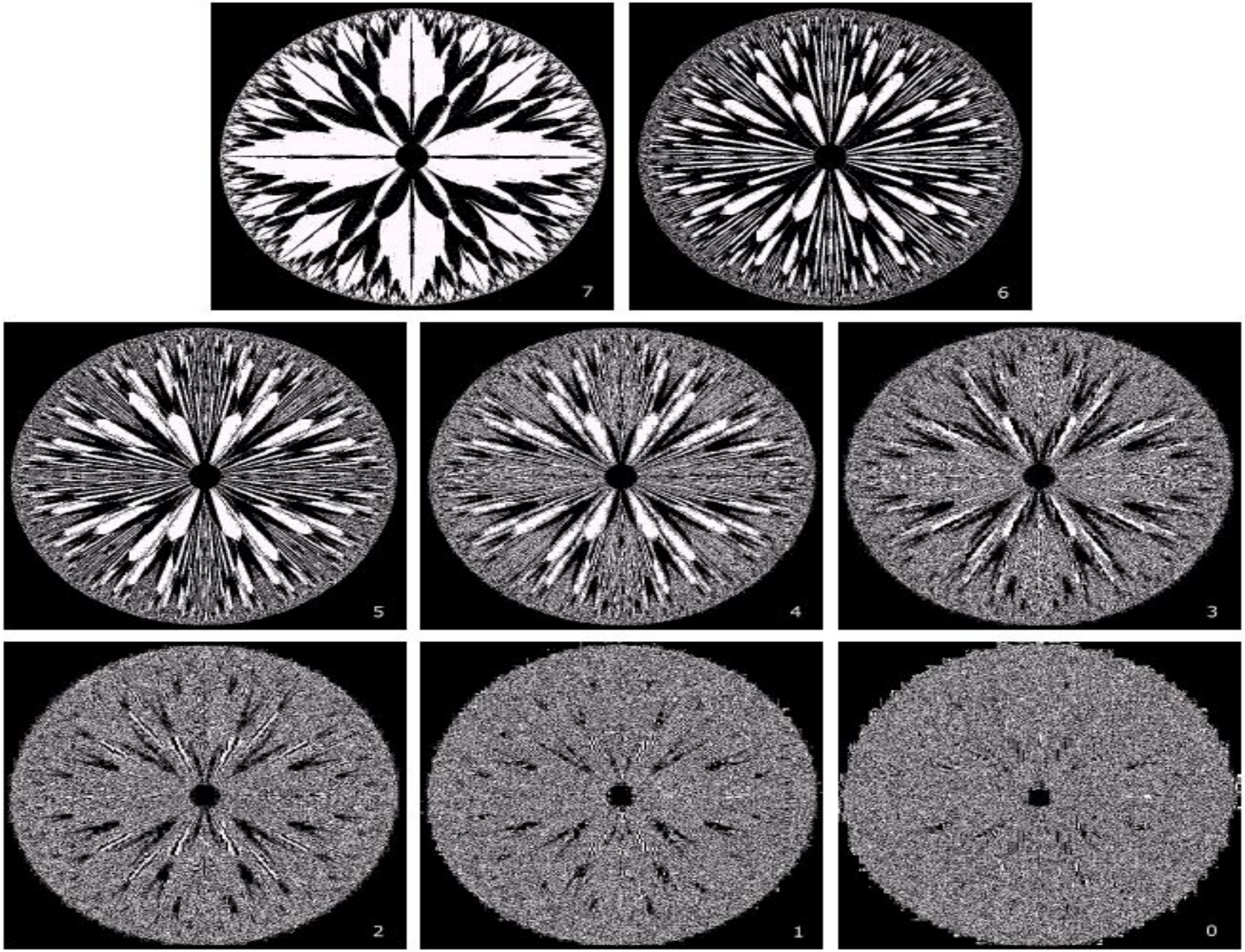


FIGURE 3.14 The eight bit planes of the image in Fig. 3.13. The number at the bottom, right of each image identifies the bit plane.

Histogram Processing

- The histogram of a digital image with gray levels from 0 to L-1 is a discrete function $h(r_k)=n_k$, where:
  - r_k is the kth gray level
  - n_k is the # pixels in the image with that gray level
  - n is the total number of pixels in the image
  - k = 0, 1, 2, …, L-1
- Normalized histogram: $p(r_k)=n_k/n$
  - sum of all components = 1

Types of processing:

- Histogram equalization
- Histogram matching (specification)
- Local enhancement

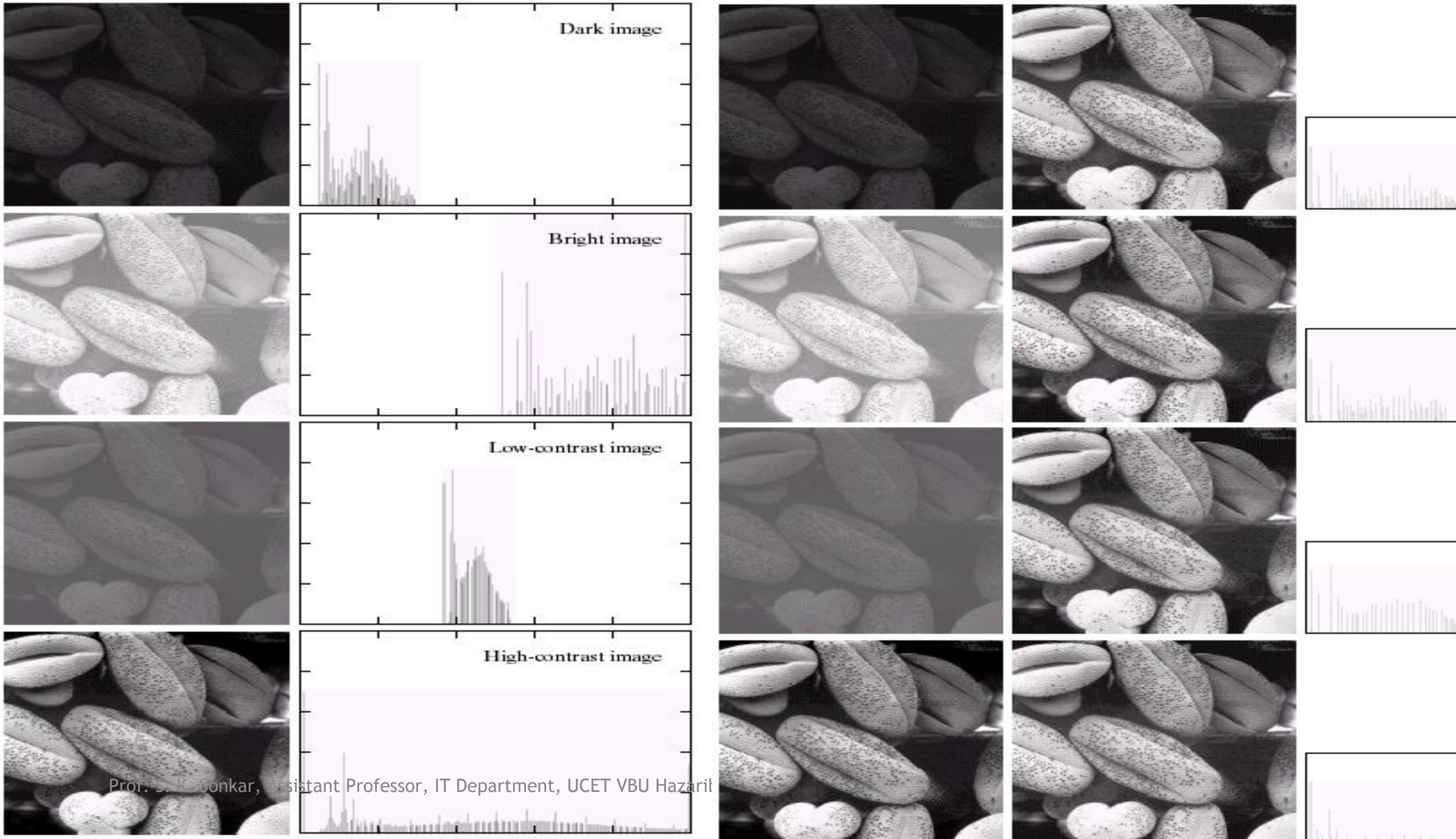
Histogram Equalization

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

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

Image Histogram

Image Histogram Equalized



Histogram Matching (or Specification)

- Histogram equalization does not allow interactive image enhancement and generates only one result: an approximation to a uniform histogram.

- Sometimes though, we need to be able to specify particular histogram shapes capable of highlighting certain gray-level ranges.

Method

- Specify the desired density function and obtain the transformation function G(z):

$$v = G(z) = \sum_{0}^{z} p_z(w) \approx \sum_{i=0}^{z} \frac{n_i}{n}$$

p_z: specified desirable PDF for output

– Apply the inverse transformation function $z=G^{-1}(s)$ to the levels obtained in step 1.

Image Smoothing or Averaging

- A noisy image:

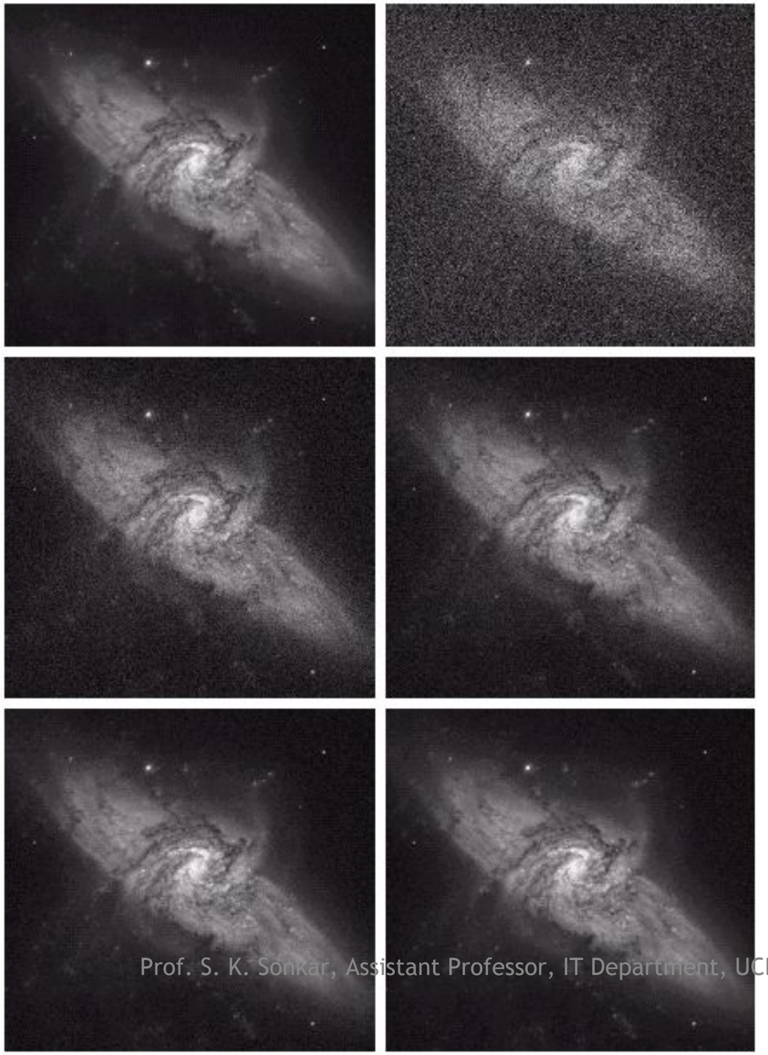
$$g(x, y) = f(x, y) + n(x, y)$$

- Averaging M different noisy images:

$$\bar{g}(x, y) = \frac{1}{M}\sum_{i=1}^{M} g_i(x, y)$$

- As M increases, the variability of the pixel values at each location decreases.
  - This means that g(x,y) approaches f(x,y) as the number of noisy images used in the averaging process increases.

Example



a b
c d
e f

FIGURE 3.30 (a) Image of Galaxy Pair NGC 3314. (b) Image corrupted by additive Gaussian noise with zero mean and a standard deviation of 64 gray levels. (c)–(f) Results of averaging $K = 8, 16, 64$, and 128 noisy images. (Original image courtesy of NASA.)

Limitations of Averaging Filter

1. It leads to blurring of an image. Blurring affects feature localization.
2. When averaging operation applied to an image corrupted by impulse noise then the impulse noise is attenuated and diffused but not removed.
3. A single pixel, with a very distinct value can affect the mean value of all the pixels in its neighbourhood significantly.

Spatial Filtering

- Use of spatial masks for image processing (spatial filters)
- Linear and nonlinear filters
- Low-pass filters eliminate or attenuate high frequency components in the frequency domain (sharp image details), and result in **image blurring.**

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

$a=(m-1)/2$ and $b=(n-1)/2$,
m x n (odd numbers)

For $x=0,1,\ldots,M-1$ and $y=0,1,\ldots,N-1$

The basic approach is to sum products between the mask coefficients and the intensities of the pixels under the mask at a specific location in the image:

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_9 z_9$$ (for a 3 x 3 filter)

Neighborhood Averaging

Each point in the smoothed image, $\hat{F}(x, y)$ is obtained from the average pixel value in a neighbourhood of (x, y) in the input image.

For example, if we use a 3×3 neighbourhood around each pixel we would use the mask

$$\begin{matrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{matrix}$$

Spatial Domain Enhancement

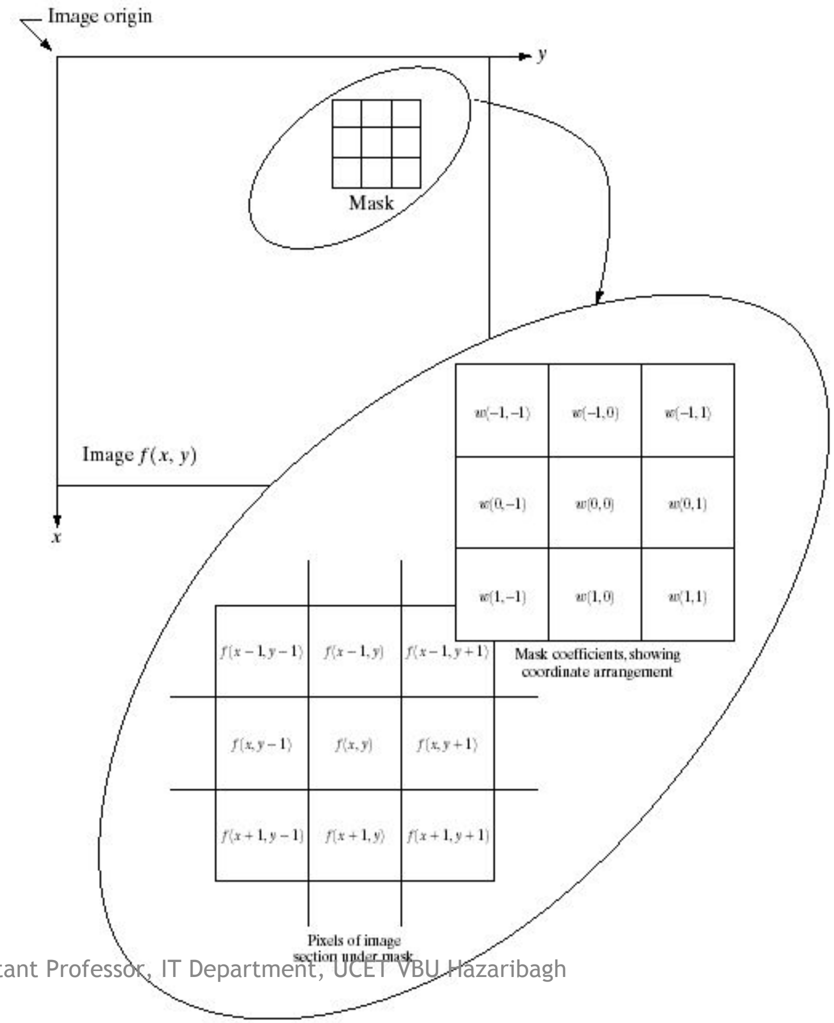


FIGURE 3.32 The mechanics of spatial filtering. The magnified drawing shows a 3 × 3 mask and the image section directly under it; the image section is shown displaced out from under the mask for ease of readability.

General Spatial Filter

FIGURE 3.33 Another representation of a general 3 × 3 spatial filter mask.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Smoothing(Averaging) 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.

Non-linear Filter

- Median filtering (nonlinear)
  - Used primarily for noise reduction *(eliminates isolated spikes)*
  - The gray level of each pixel is replaced by the median of the gray levels in the neighborhood of that pixel (instead of by the average as before).

Illustration

Sharpening Filters

- The main aim in image sharpening is to highlight fine detail in the image
- With image sharpening, we want to enhance the high-frequency components

Frequency Domain filter

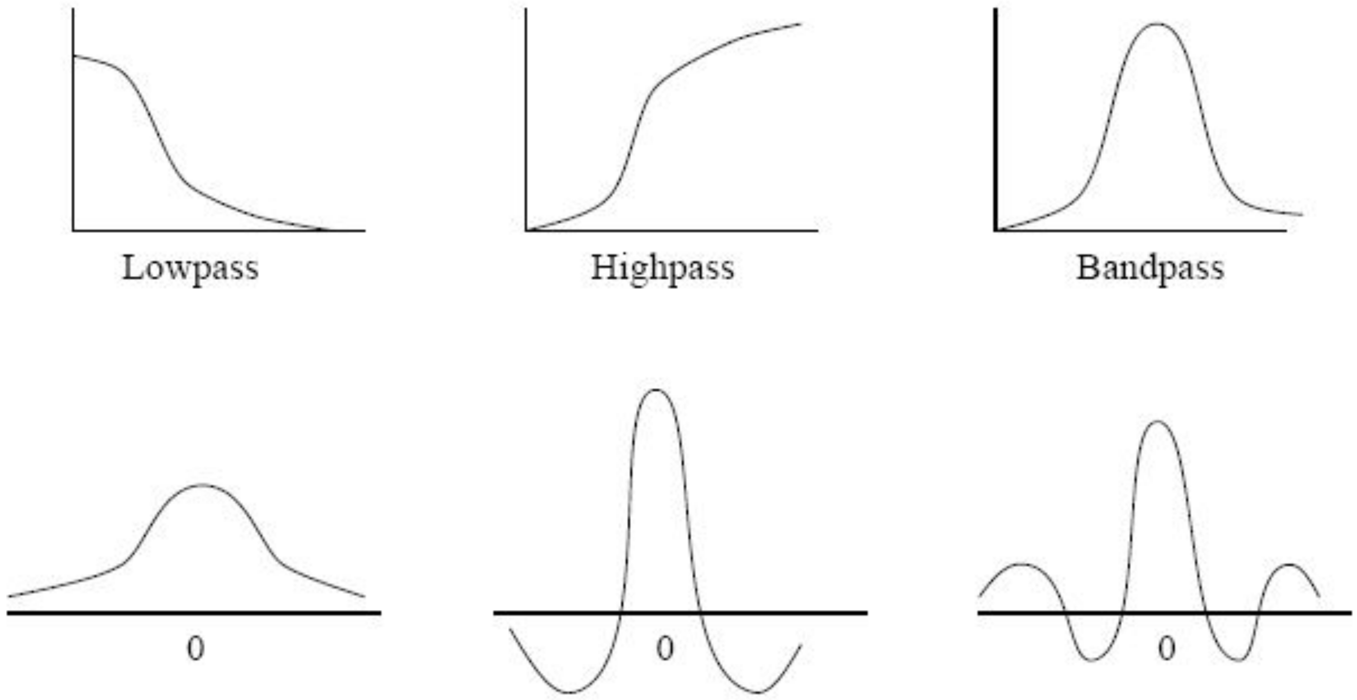


Figure 4: Frequency domain filters (top) and their corresponding spatial domain counterparts (bottom).

Frequency Domain filter

- Frequency refers to the rate of repetition of some periodic event.
- In Image processing spatial frequency refers to the variation of image brightness with its position in space.
- Fourier Transform is a tool to obtain the frequency components.
- If we multiply each element of the fourier coefficient by a suitably chosen weighting function then we can accentuate certain frequency components and attenuate others. The corresponding changes in the spatial domain can be seen after an inverse transform is computed.
- This selective enhancement or suppression of frequency components is termed as Fourier Filtering or Frequency domain filtering.

Frequency Domain filter

- The spatial representation of image data describes the adjacency relationship between pixels.
- On the other hand, the frequency domain representation clusters the image data according to their frequency distribution.

Filtering

- Let input image be $f(m,n)$ and filter kernel be $h(m,n)$

Filtering in spatial domain=$f(m,n)*h(m,n)$

Filtering in frequency domain=$F(k,l) \times H(k,l)$

Image Enhancement in Frequency domain

a b

FIGURE 4.3
(a) Image of a 20×40 white rectangle on a black background of size 512×512 pixels.
(b) Centered Fourier spectrum shown after application of the log transformation given in Eq. (3.2-2). Compare with Fig. 4.2.

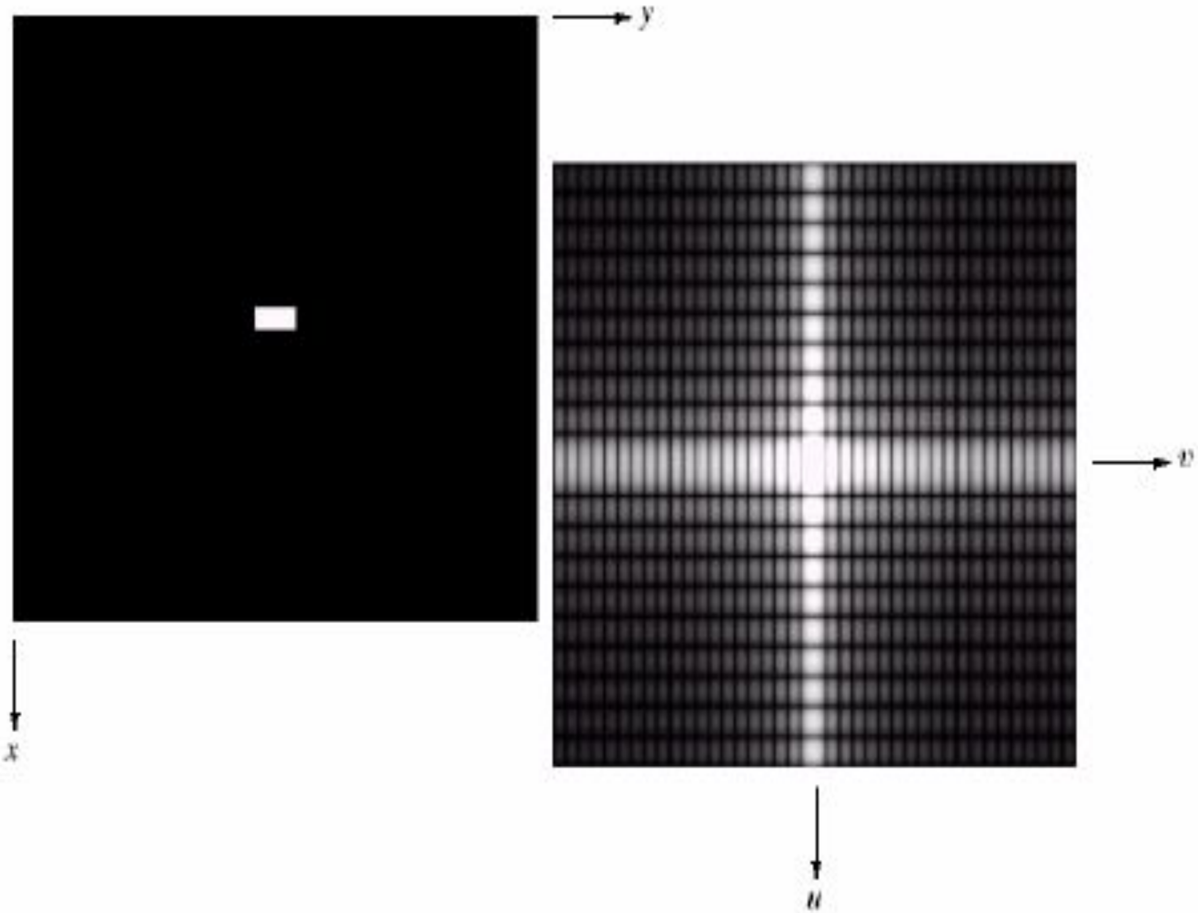


Image Enhancement in Frequency domain

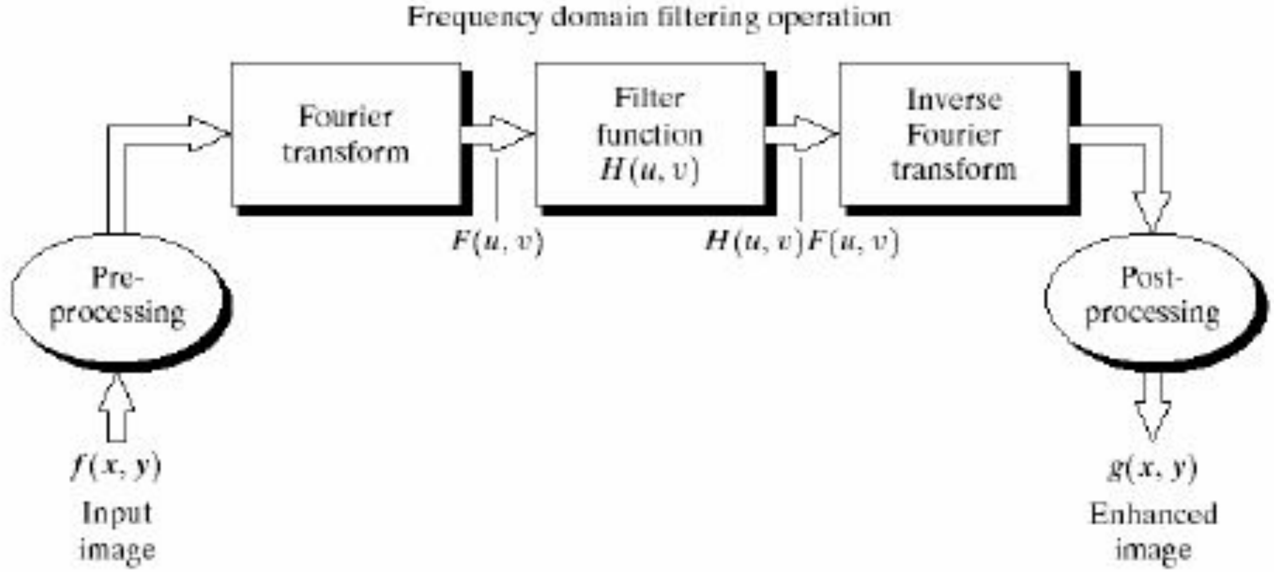


FIGURE 4.5 Basic steps for filtering in the frequency domain.

Image Enhancement in Frequency domain

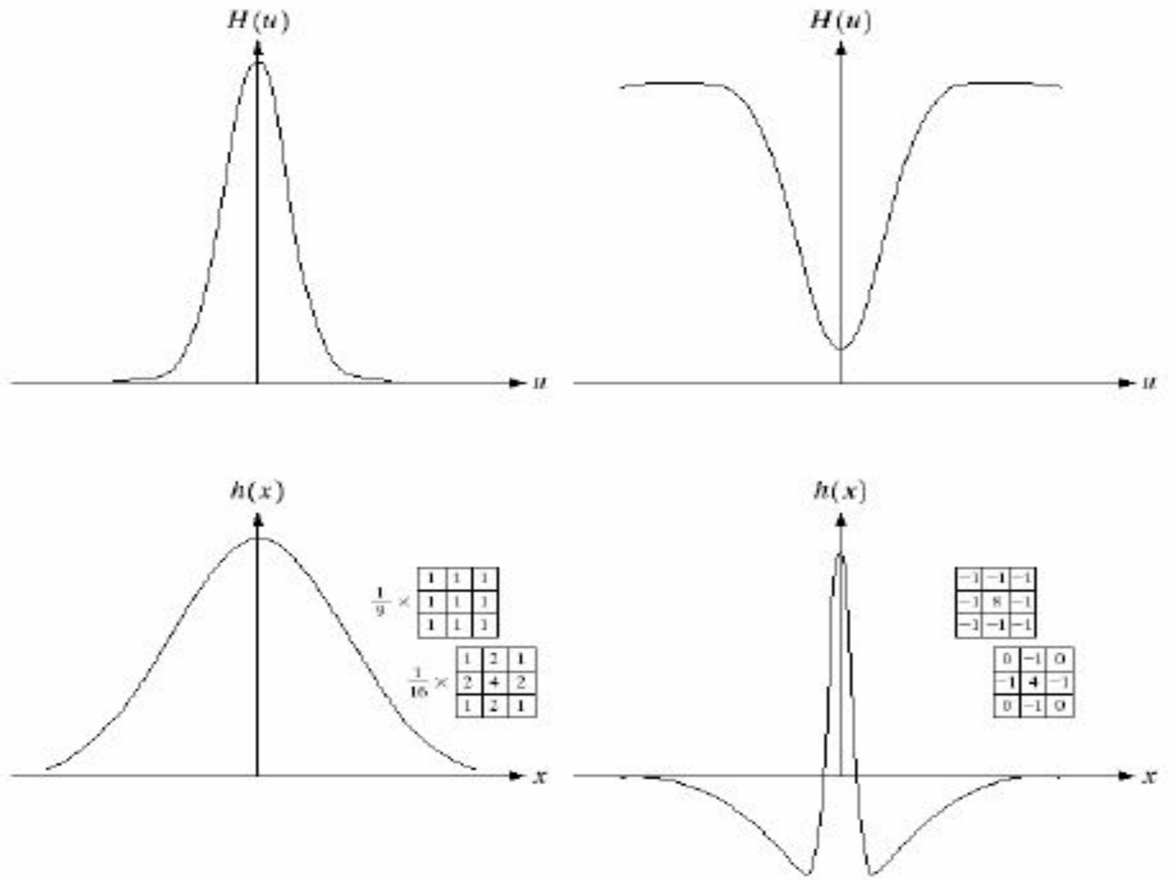


FIGURE 4.9 (a) Gaussian frequency domain lowpass filter. (b) Gaussian frequency domain highpass filter. (c) Corresponding lowpass spatial filter. (d) Corresponding highpass spatial filter. The masks shown are used in Chapter 3 for lowpass and highpass filtering.

Ideal LPF

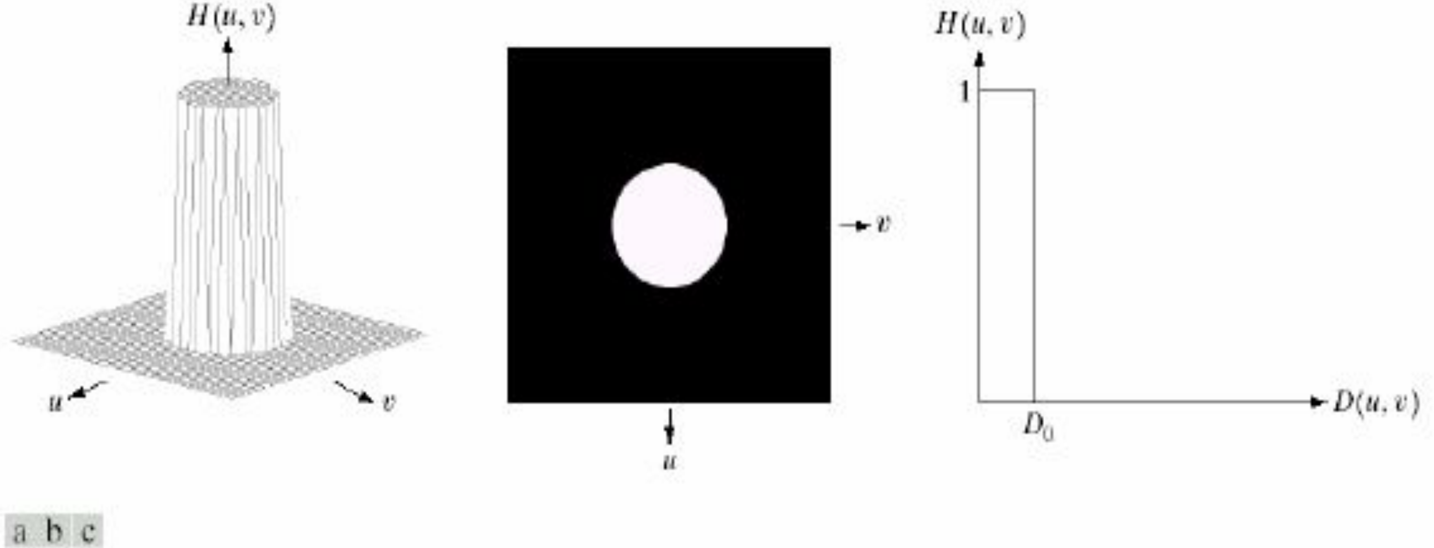


FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

Image Enhancement in Frequency domain

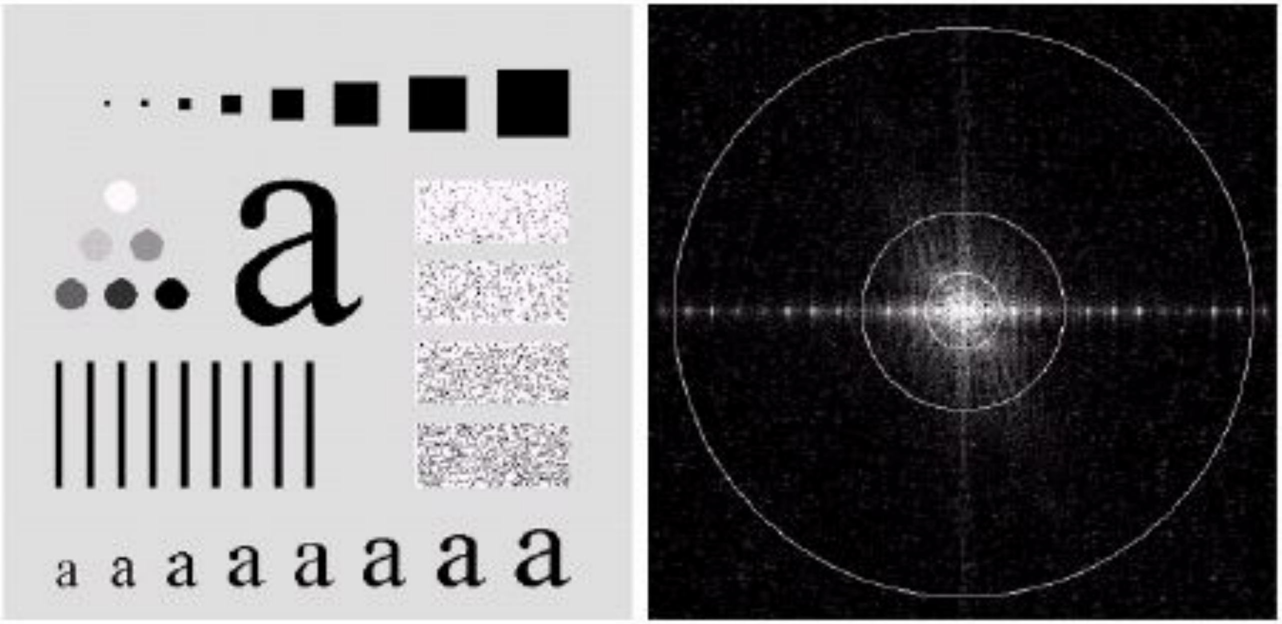
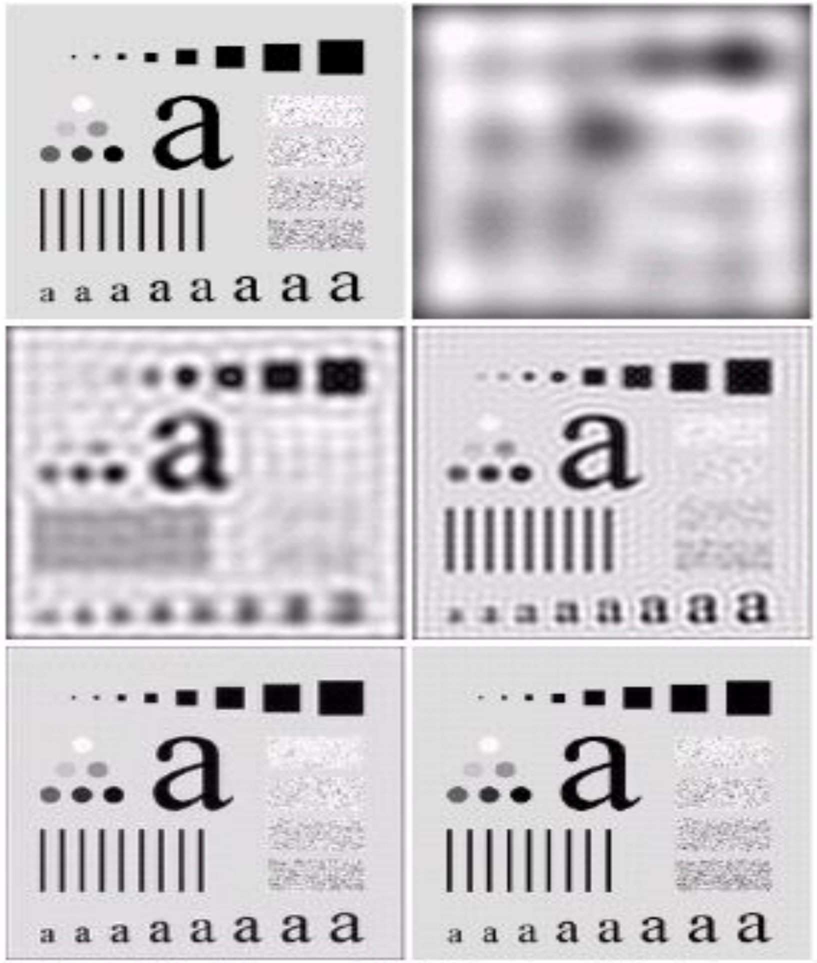


FIGURE 4.11 (a) An image of size 500 × 500 pixels and (b) its Fourier spectrum. The superimposed circles have radii values of 5, 15, 30, 80, and 230, which enclose 92.0, 94.6, 96.4, 98.0, and 99.5% of the image power, respectively.

Contd...



a b
c d
e f

FIGURE 4.12 (a) Original image. (b)–(f) Results of ideal lowpass filtering with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). The power removed by these filters was 8, 5.4, 3.6, 2, and 0.5% of the total, respectively.

Contd…

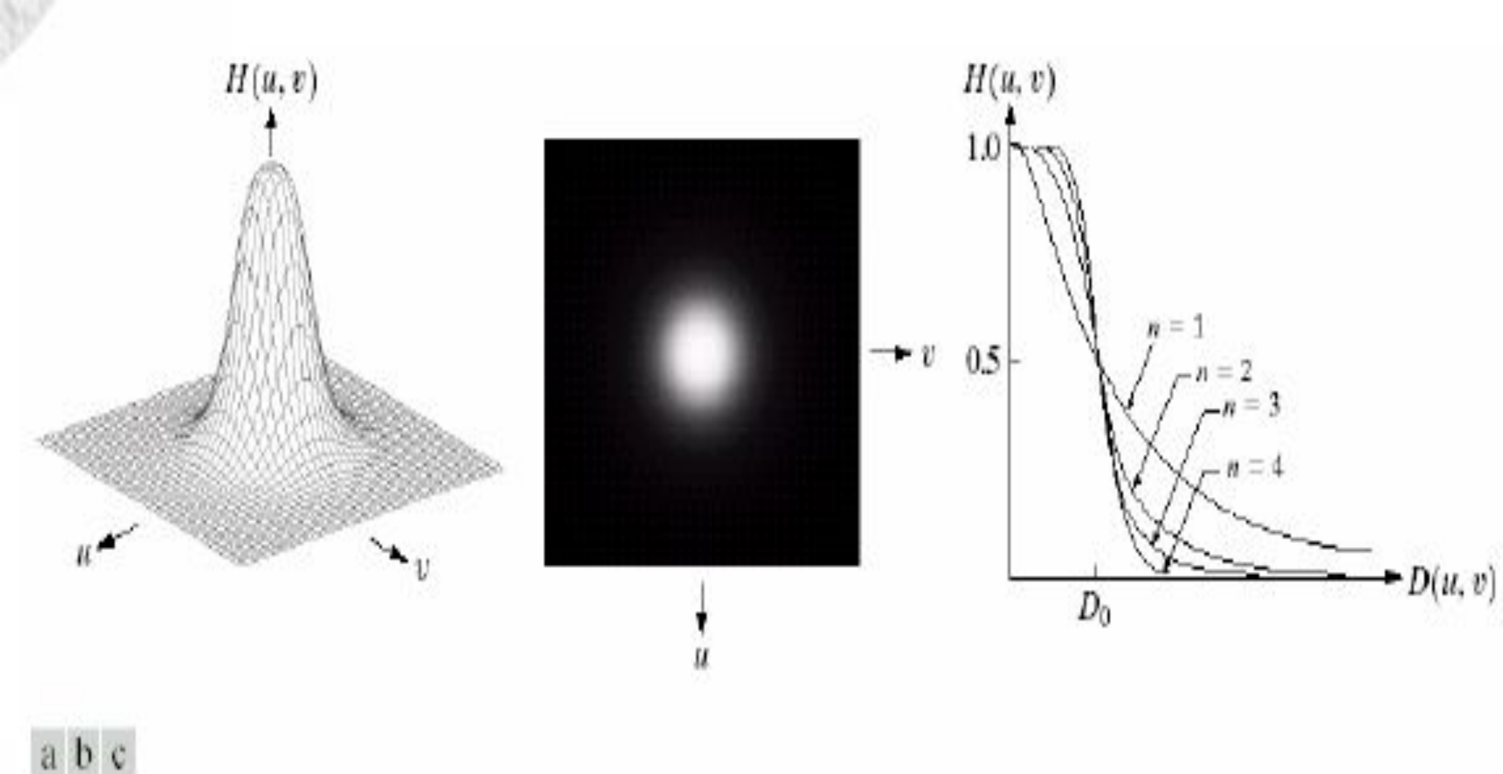


FIGURE 4.14 (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

Contd..

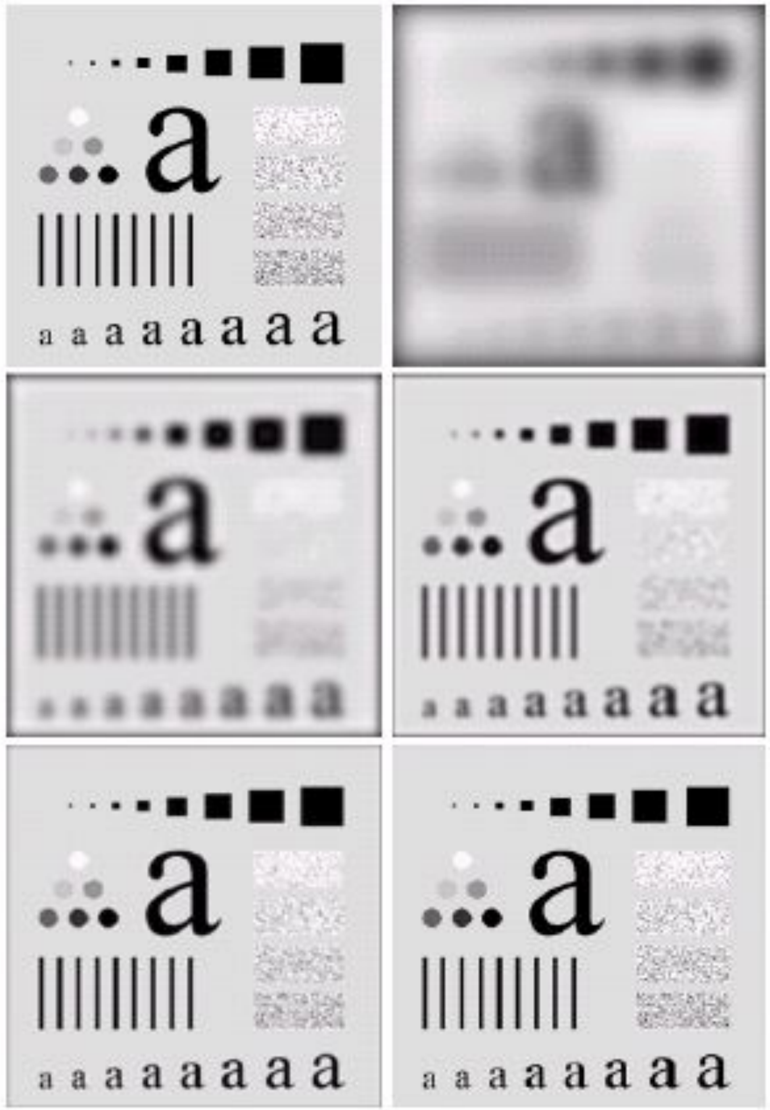
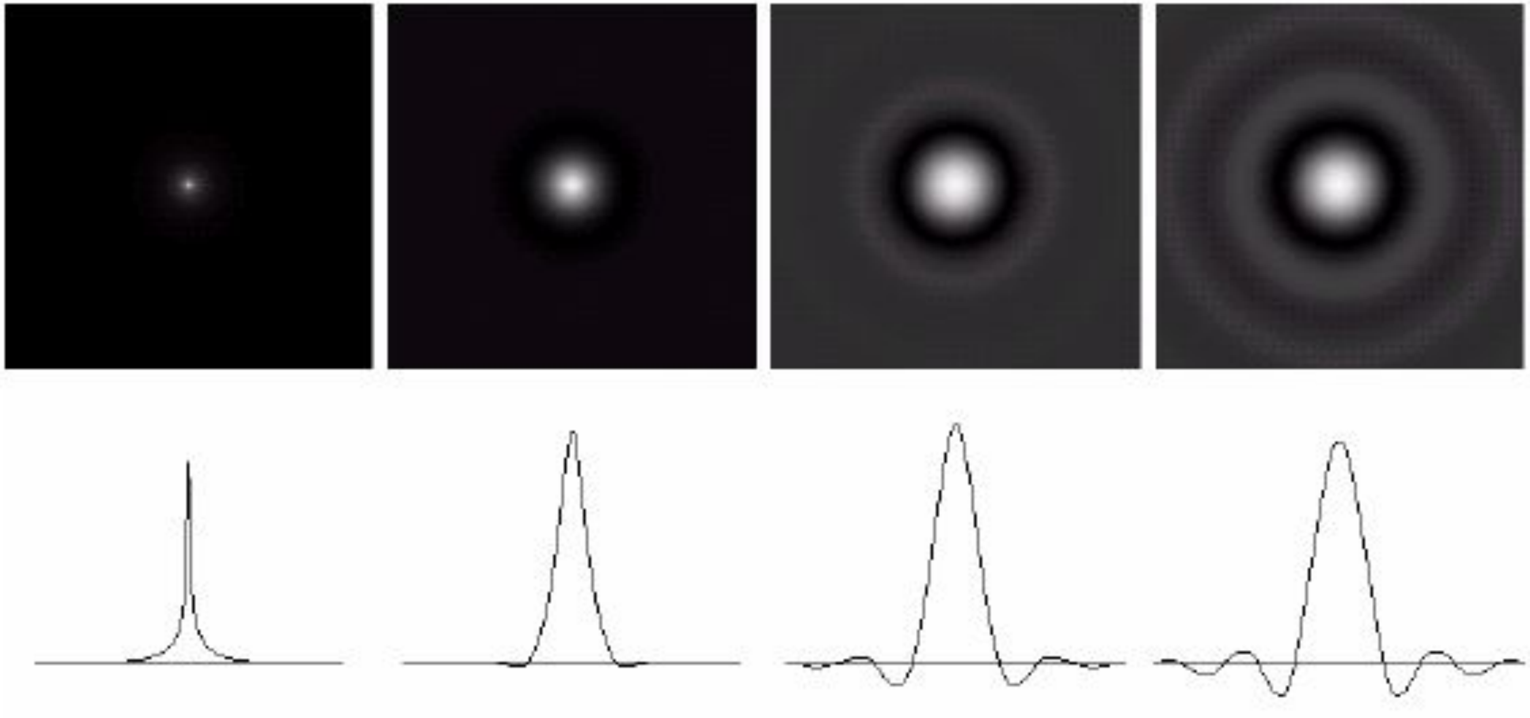


FIGURE 4.15 (a) Original image. (b)–(f) Results of filtering with BLPFs of order 2, with cutoff frequencies at radii of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Fig. 4.12.

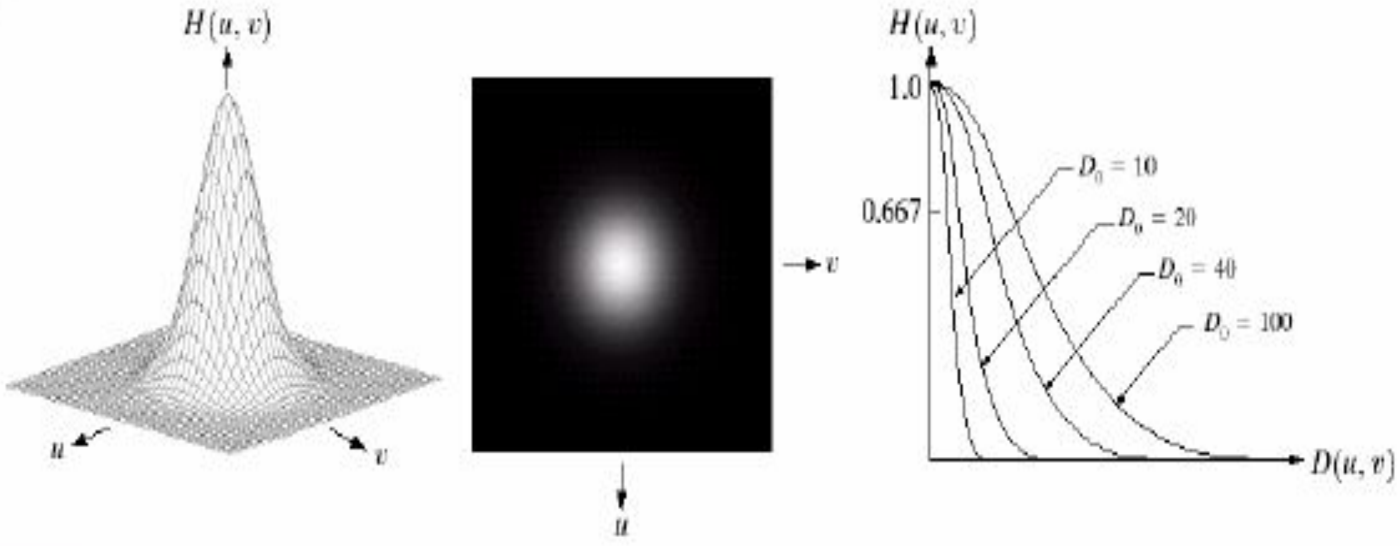
Contd...



a b c d

FIGURE 4.16 (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding gray-level profiles through the center of the filters (all filters have a cutoff frequency of 5). Note that ringing increases as a function of filter order.

Contdd..



a b c

FIGURE 4.17 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.

Contd..

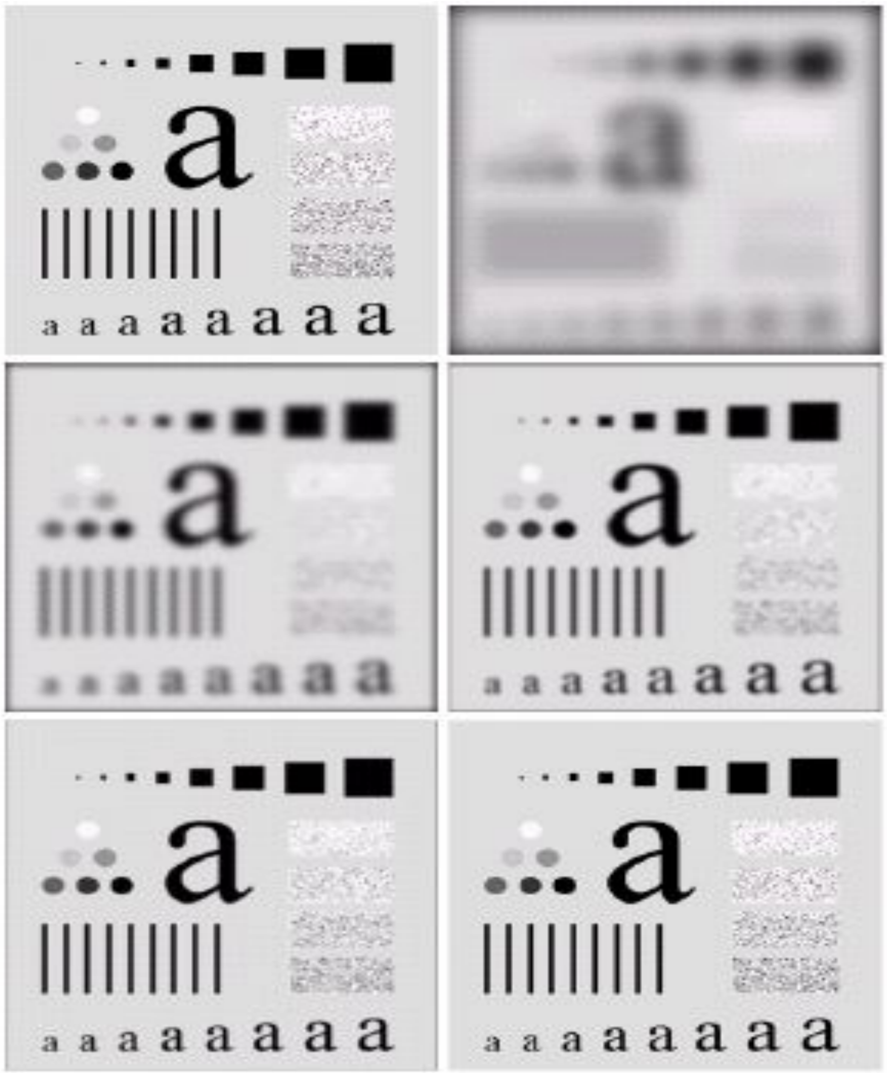
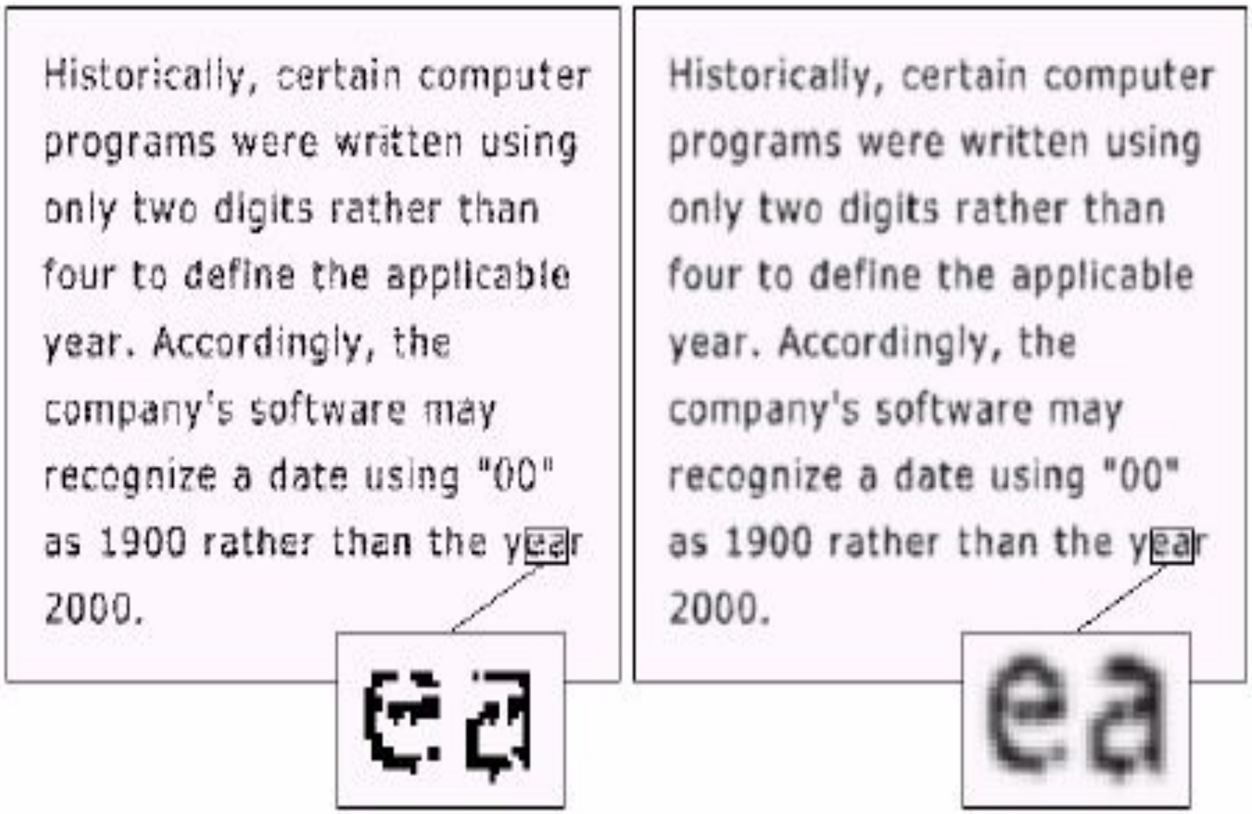


FIGURE 4.18 (a) Original image. (b)–(f) Results of filtering with Gaussian lowpass filters with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in

Contd..

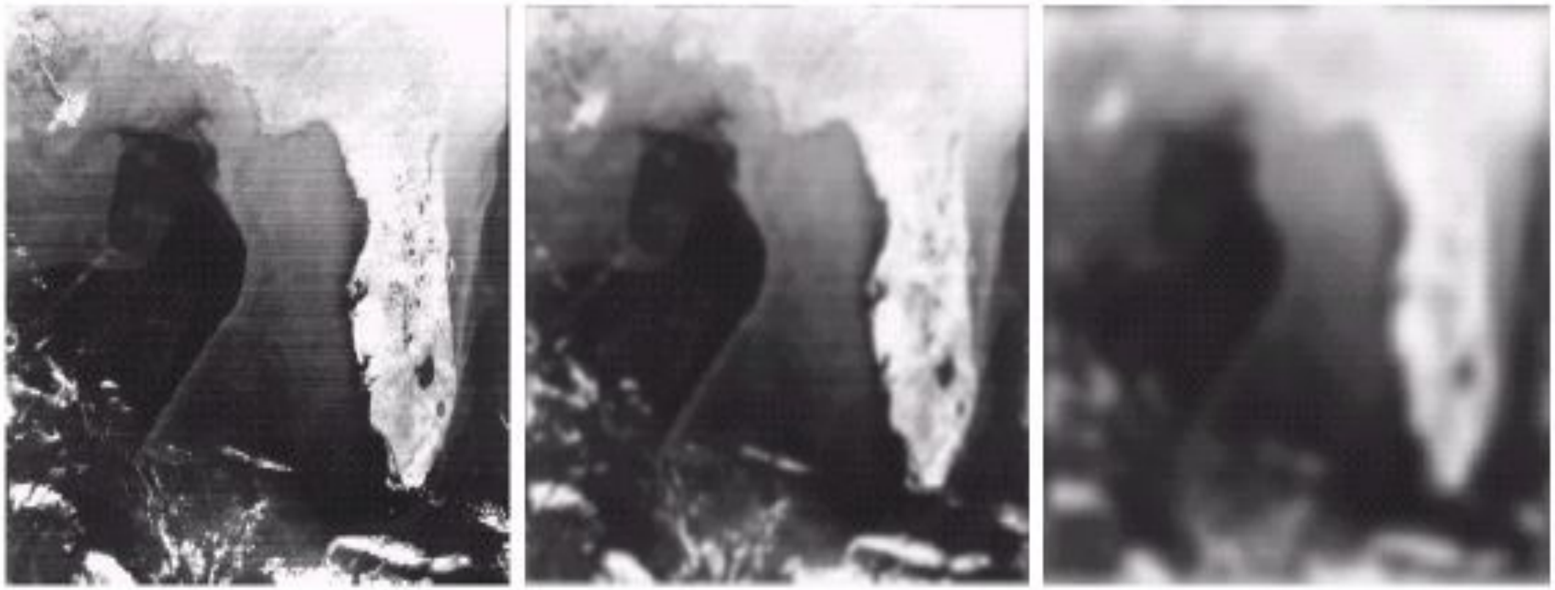


a b

FIGURE 4.19 (a) Sample text of poor resolution (note broken characters in magnified view). (b) Result of filtering with a GLPF (broken character segments were joined).

Contd…

Contd...



a b c

FIGURE 4.21 (a) Image showing prominent scan lines. (b) Result of using a GLPF with $D_0 = 30$. (c) Result of using a GLPF with $D_0 = 10$. (Original image courtesy of NOAA.)

Contd...

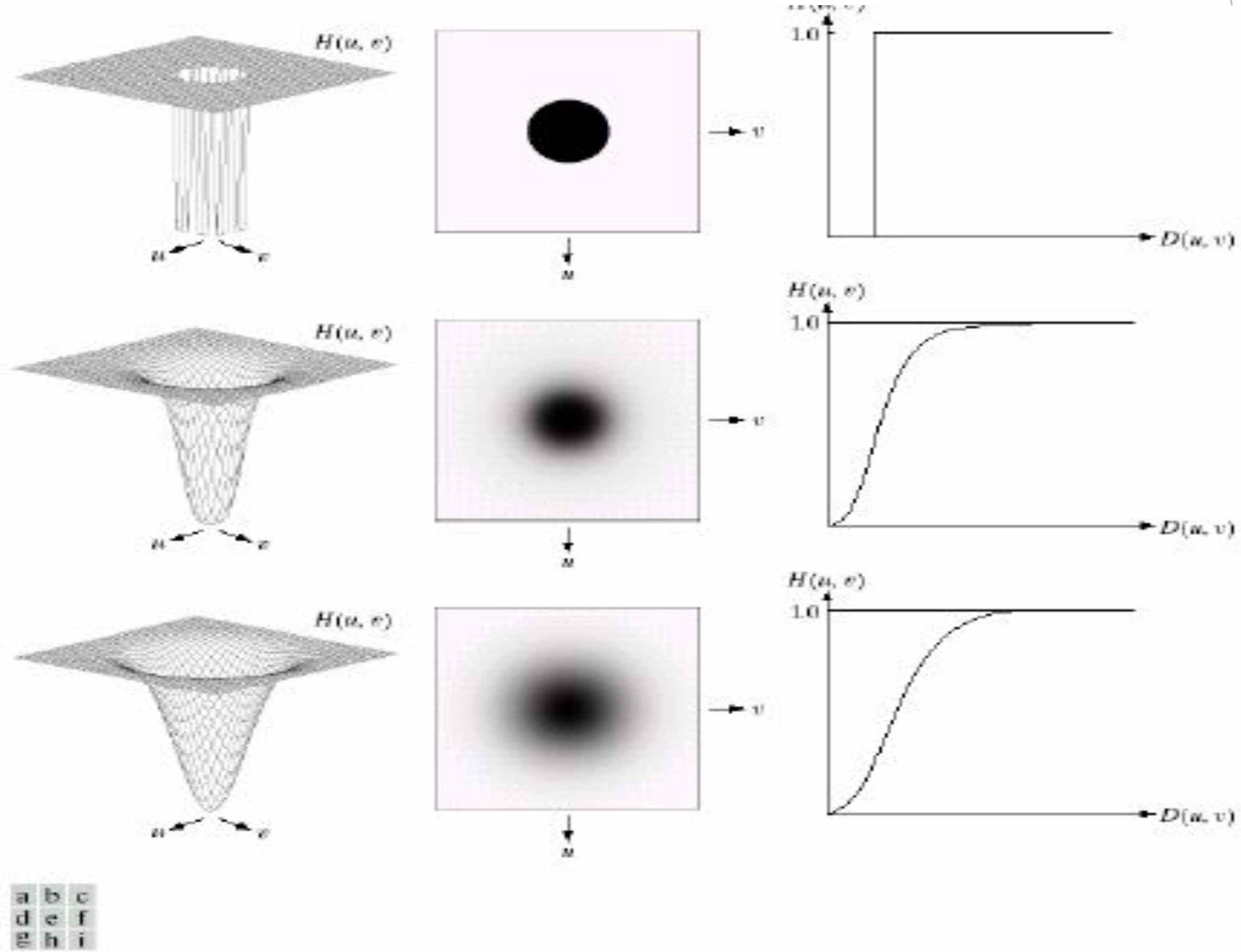


FIGURE 4.22 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

Derivatives

- First derivative

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

- Second derivative

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

Observations

- 1st order derivatives produce thicker edges in an image
- 2nd order derivatives have stronger response to fine detail
- 1st order derivatives have stronger response to a gray level step
- 2nd order derivatives produce a double response at step changes in gray level

A simple spatial filter that achieves image sharpening is given by

$$\begin{array}{ccc} -1/9 & -1/9 & -1/9 \\ -1/9 & 8/9 & -1/9 \\ -1/9 & -1/9 & -1/9 \end{array}$$

- Since the sum of all the weights is zero, the resulting signal will have a zero DC value

Frequency Domain Methods

- We simply compute the Fourier transform of the image to be enhanced, multiply the result by a filter (rather than convolve in the spatial domain), and take the inverse transform to produce the enhanced image.
- Low pass filtering involves the elimination of the high frequency components in the image. It results in blurring of the image

Frequency Domain Methods

Figure 5: Transfer function for an ideal low pass filter.