DIGITAL IMAGE PROCESSING MODULE-III

PARVATHY LAKSHMY

FACULTY

ELECTRICAL DEPARTMENT

VJTI

IMAGE ENHANCEMENT IN SPATIAL DOMAIN

- The term **Spatial Domain** refers to the **image plane** itself
- Image processing methods in this category are based on direct manipulation of pixels in an image.
- Different from image processing in a transform domain
- It involves **transforming an image** into transform domain, **doing the processing**, obtaining the **inverse transform** to bring results back to the spatial domain.
- Two principle categories of spatial processing are:
- 1) Intensity Transformation 2) Spatial Filtering

IMAGE ENHANCEMENT IN SPATIAL DOMAIN

- Intensity Transformation: **Operate on single pixels of an image**, for the purpose of **contrast manipulation** and **image thresholding**
- Spatial Filtering: Performing operations such as image sharpening, by working in a **neighbourhood of every pixel** in an image.
- Generally Spatial Domain techniques are:

➢ Computationally more efficient

➢ Require less processing resources to implement.

IMAGE ENHANCEMENT IN SPATIAL DOMAIN

- Spatial Domain Processes can be denoted by:

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

- f(x,y)-Input Image
- g(x,y)-Output Image
- T- Operator on f defined over a neighbourhood of point (x,y)
- Operator-Can apply to single image or set of images such as performing pixel-by-pixel sum of sequences of images for noise reduction

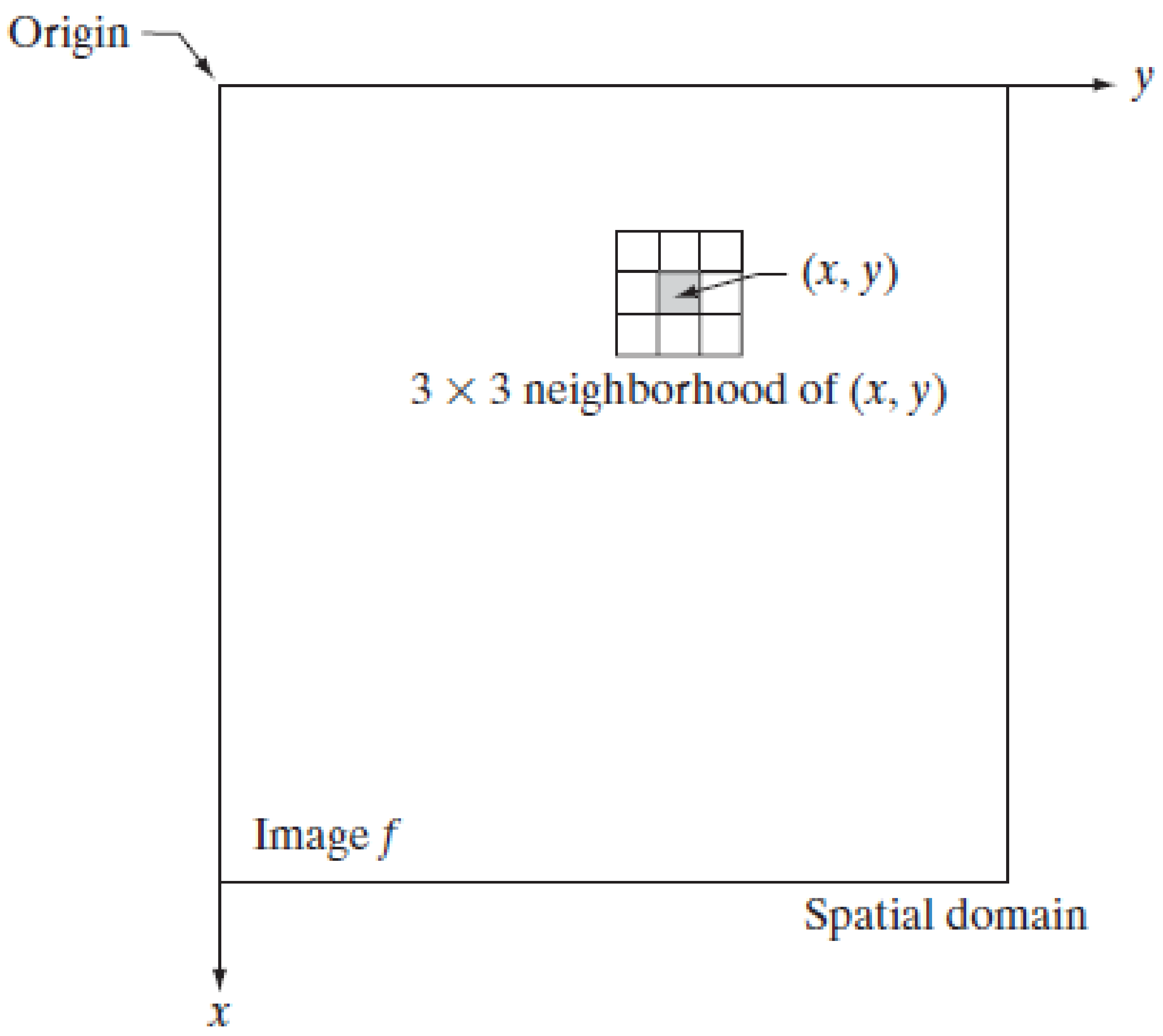


FIGURE 3.1 A 3×3 neighborhood about a point (x, y) in an image in the spatial domain. The neighborhood is moved from pixel to pixel in the image to generate an output image.

IMAGE ENHANCEMENT IN SPATIAL DOMAIN

- Point (x,y)- Arbitrary location in the image and the small region shown containing the point is a neighbourhood of (x,y)
- Typically, neighbourhood is rectangular centred on (x,y) and much smaller in the size than the image.
- The process consists of moving the origin of the neighbourhood from pixel to pixel and applying the operator T to the pixels in the neighbourhood to yield the output at that location
- Thus, for any specific location (x, y), the value of the output image g at those coordinates is equal to the result of applying T to the neighbourhood with origin at (x, y) in f.

IMAGE ENHANCEMENT IN SPATIAL DOMAIN

- For example, suppose that the neighbourhood is a square of size 3X3 and that operator *T* is defined as "compute the average intensity of the neighbourhood."
- Consider an arbitrary location in an image, say (100, 150).
- Assuming that the origin of the neighbourhood is at its centre, the result, g(100,150), at that location is computed as the sum of f(100,150) and its 8-neighbors, divided by 9 (i.e., the average intensity of the pixels encompassed by the neighbourhood).
- The origin of the neighbourhood is then moved to the next location, and the procedure is repeated to generate the next value of the output image *g*.

IMAGE ENHANCEMENT IN SPATIAL DOMAIN

- Typically, the process starts at the top left of the input image and proceeds pixel by pixel in a horizontal scan, one row at a time
- When the origin of neighbourhood is at the border of the image, part of the neighbourhood will reside outside the image.
- Procedure-Either ignore the outside neighbours in computations specified by T or pad image with a border of 0s or some other specified intensity values
- Thickness of the padded border depends on the size of the neighbourhood

IMAGE ENHANCEMENT IN SPATIAL DOMAIN

- This procedure just described is called ***spatial filtering***, in which the neighbourhood, along with a predefined operation, is called a ***spatial filter*** (also referred to as a ***spatial mask*, *kernel*, *template*, or *window*).**
- The type of operation performed in the neighbourhood determines the nature of the filtering process.
- The smallest possible neighborhood is of size 1X1.
- In this case, g depends only on the value of f at a single point (x, y) and T becomes an *intensity* (also called *gray-level* or *mapping*) *transformation function* of the form:

IMAGE ENHANCEMENT IN SPATIAL DOMAIN

$$s = T(r)$$

- Where *s* and *r* are variables denoting, respectively, the intensity of *g* and *f* at any point (*x*, *y*).
- For example if T(r) has the form in the figure below

a b

FIGURE 3.2 Intensity transformation functions. (a) Contrast-stretching function. (b) Thresholding function.

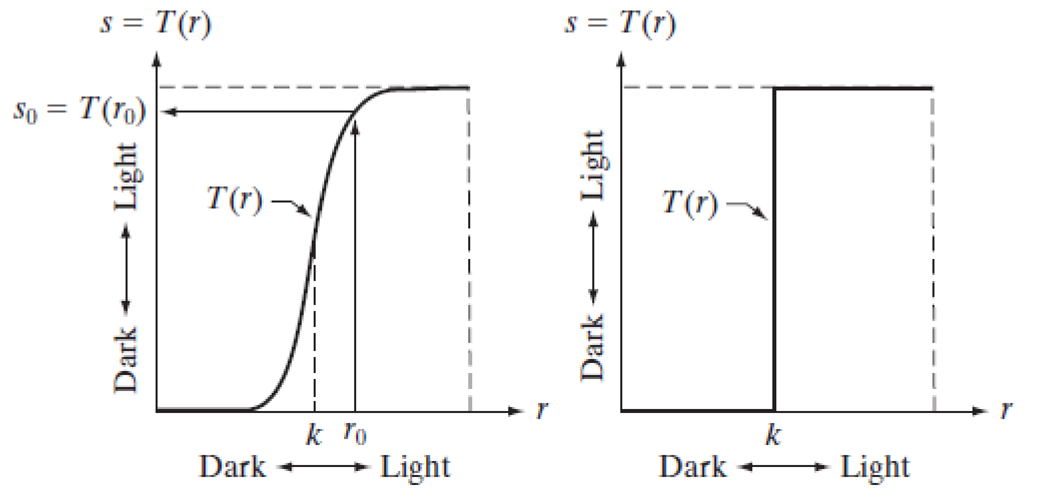


IMAGE ENHANCEMENT IN SPATIAL DOMAIN

- The effect of applying transformation to every pixel f to generate corresponding pixels in g-produce an image of higher contrast than the original by darkening the intensity levels below k and brightening the levels above k
- In this technique, sometimes called *contrast stretching,* values of *r* lower than *k* are compressed by the transformation function into a narrow range of *s*, toward black.
- The opposite is true for values of *r* higher than *k*.
- In the limiting case shown in Figure, *T*(*r*) produces a two-level (binary) image.
- A mapping of this form is called a *thresholding* function.

IMAGE ENHANCEMENT IN SPATIAL DOMAIN

- We use intensity transformations principally for image enhancement.
- Approaches whose results depend only on the intensity at a point sometimes are called ***point processing* techniques**, as opposed to the ***neighborhood processing*** techniques
- **Image Enhancement** is the process of manipulating an image so that the result is more suitable than the original for a specific application.
- Thus, for example, a method that is quite useful for enhancing X-ray images may not be the best approach for enhancing satellite images taken in the infrared band of the electromagnetic spectrum.

IMAGE ENHANCEMENT IN SPATIAL DOMAIN

- There is no general "theory" of image enhancement.
- When an image is processed for visual interpretation, the viewer is the ultimate judge of how well a particular method works.
- When dealing with machine perception, a given technique is easier to quantify.
- For example, in an automated character-recognition system, the most appropriate enhancement method is the one that results in the **best recognition rate**, leaving aside other considerations such as **computational requirements** of one method over another.
- Regardless of the application or method used, however, image enhancement is one of the most visually appealing areas of image processing.

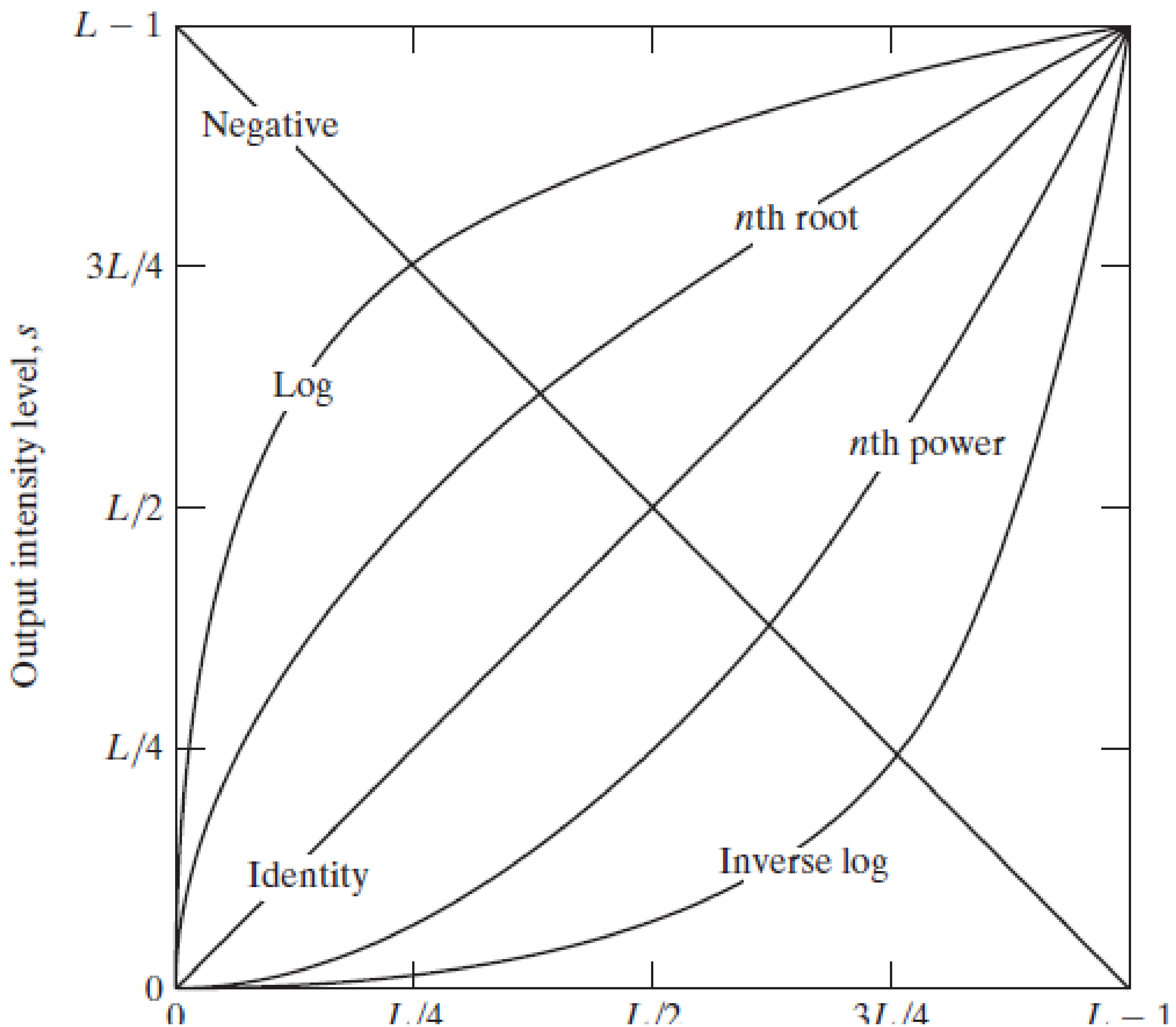
POINT PROCESSING TECHNIQUES

- Intensity transformations-simplest of all image processing techniques
- The values of pixels, before and after processing, will be denoted by r and s, respectively.
- For digital images, values of a transformation function are stored in a one-dimensional array and the mappings from r to s are implemented via table lookups.
- For an 8-bit environment, a lookup table containing the values of T will have 256 entries.

POINT PROCESSING TECHNIQUES

- Common Intensity Transformations for Image Enhancements are:
  - Linear(Negative and Identity Transformations)
  - Logarithmic (log and inverse-log transformation
  - Power-law (nth power and nth root transformations).

FIGURE 3.3 Some basic intensity transformation functions. All curves were scaled to fit in the range shown.



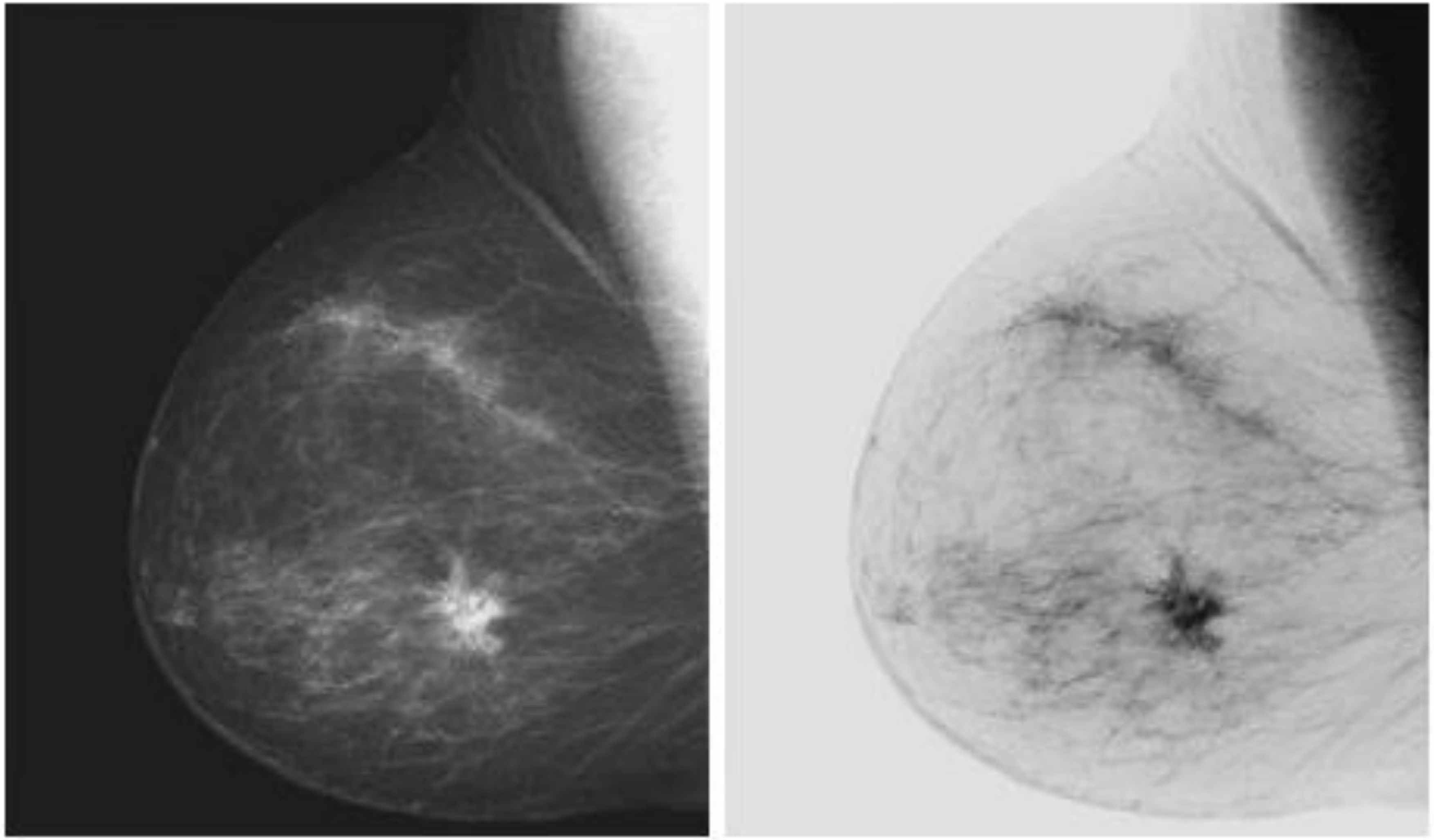
POINT PROCESSING TECHNIQUES

DIGITAL NEGATIVES:

- The negative of an image with intensity levels in the range is obtained by using the negative transformation which is given by the expression:

$$s = L - 1 - r$$

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



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.)

POINT PROCESSING TECHNIQUES

- **LOG TRANSFORMATIONS:**

The general form of the log transformation is:

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

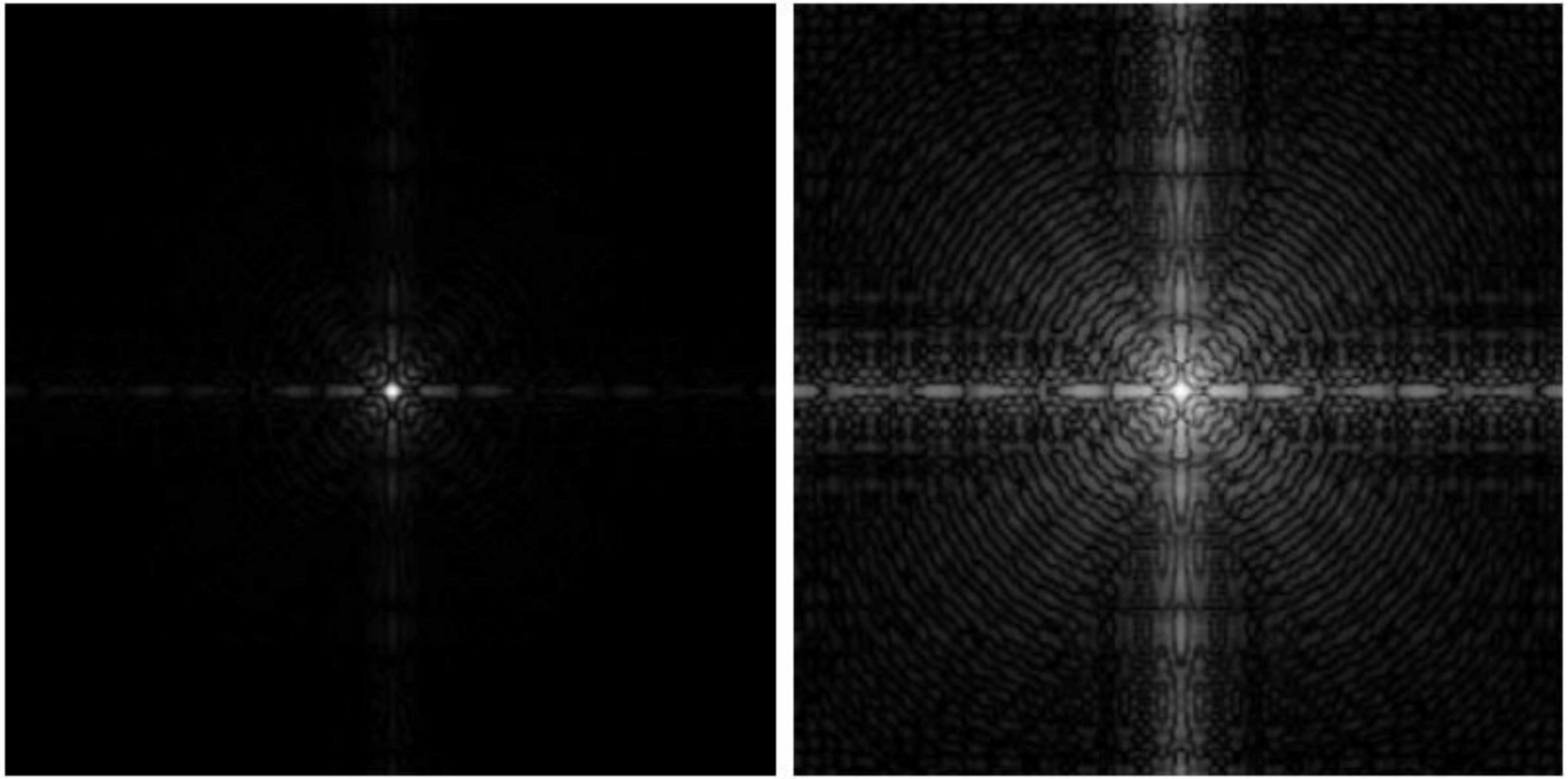
Where c is a constant and assuming $r \geq 0$

- The shape of the log curve in above figure shows that this transformation maps a narrow range of low intensity values in the input into a wider range of output levels.
- We use a transformation of this type to expand the values of dark pixels in an image while compressing the higher-level values.
- The opposite is true of the inverse log transformation.

POINT PROCESSING TECHNIQUES

LOG TRANSFORMATIONS:

- Any curve having the general shape of the log functions would accomplish this spreading/compressing of intensity levels in an image
- The log function has the important characteristic that it 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 in Eq. (3.2-2) with $c = 1$.

POINT PROCESSING TECHNIQUES

- Fig. shows a Fourier spectrum with values in the range $0\ to\ 10^6$.
- When these values are scaled linearly for display in an 8-bit system, the brightest pixels will dominate the display, at the expense of lower (and just as important) values of the spectrum.
- The effect of this dominance is illustrated by the relatively small area of the image in Fig. that is not perceived as black.
- If, instead of displaying the values in this manner, we first apply the equation for log transformations (with in this case) to the spectrum values, then the range of values of the result becomes 0 to 6.2, which is more manageable.

POINT PROCESSING TECHNIQUES

- Figure 3.5(b) shows the result of scaling this new range linearly and displaying the spectrum in the same 8-bit display.
- The wealth of detail visible in this image as compared to an unmodified display of the spectrum is evident from these pictures.
- Most of the Fourier spectra seen in image processing publications have been scaled in just this manner.

POINT PROCESSING TECHNIQUES

POWER LAW(GAMMA) TRANSFORMATIONS:

- Power-law transformations have the basic form:

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

  where c and γ are positive constants

- As in the case of the log transformation, power-law curves with fractional values of γ map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher values of input levels.

POINT PROCESSING TECHNIQUES

POWER LAW(GAMMA) TRANSFORMATIONS:

- Unlike the log function, here a family of possible transformation curves are obtained simply by varying γ.
- In Fig below curves generated with values of $\gamma > 1$ have exactly the opposite effect as those generated with values of $\gamma < 1$.
- The equation above reduces to the identity transformation when $c = \gamma = 1$
- 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**.

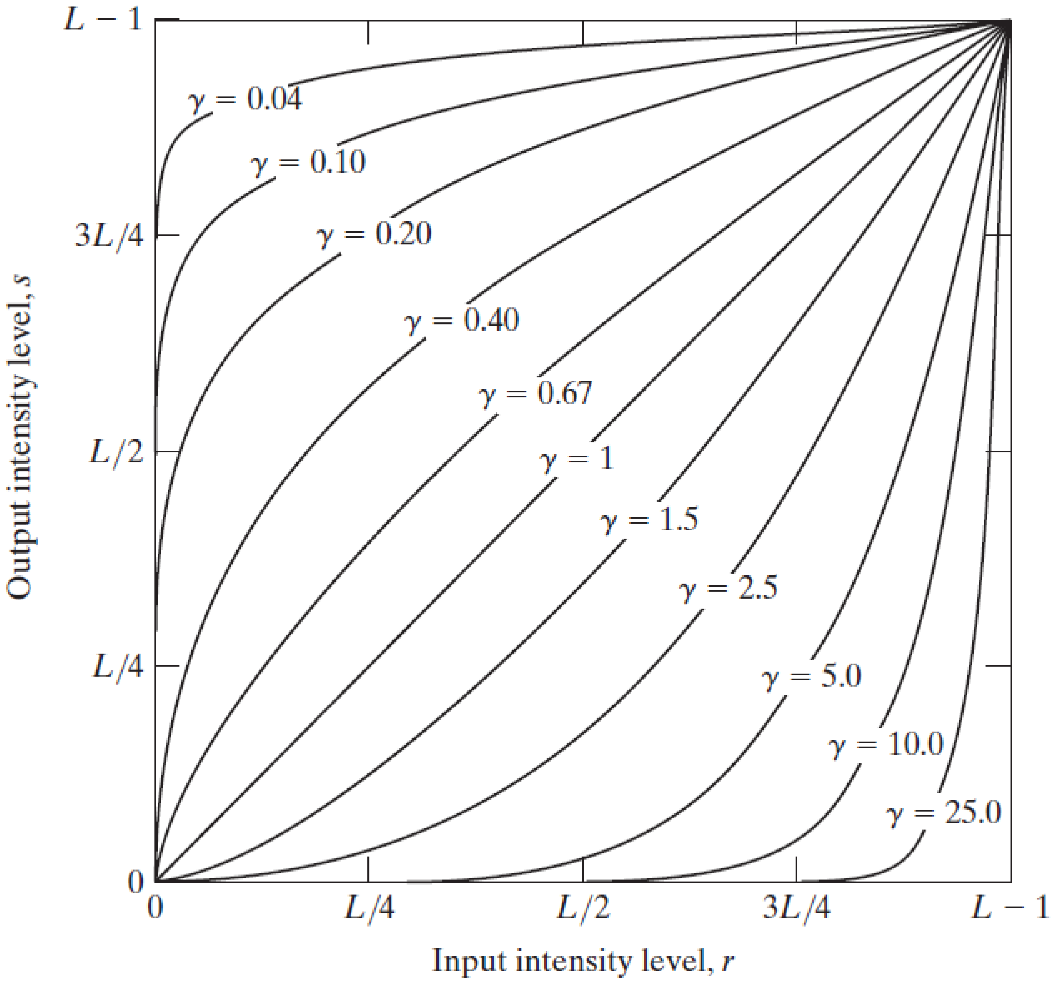
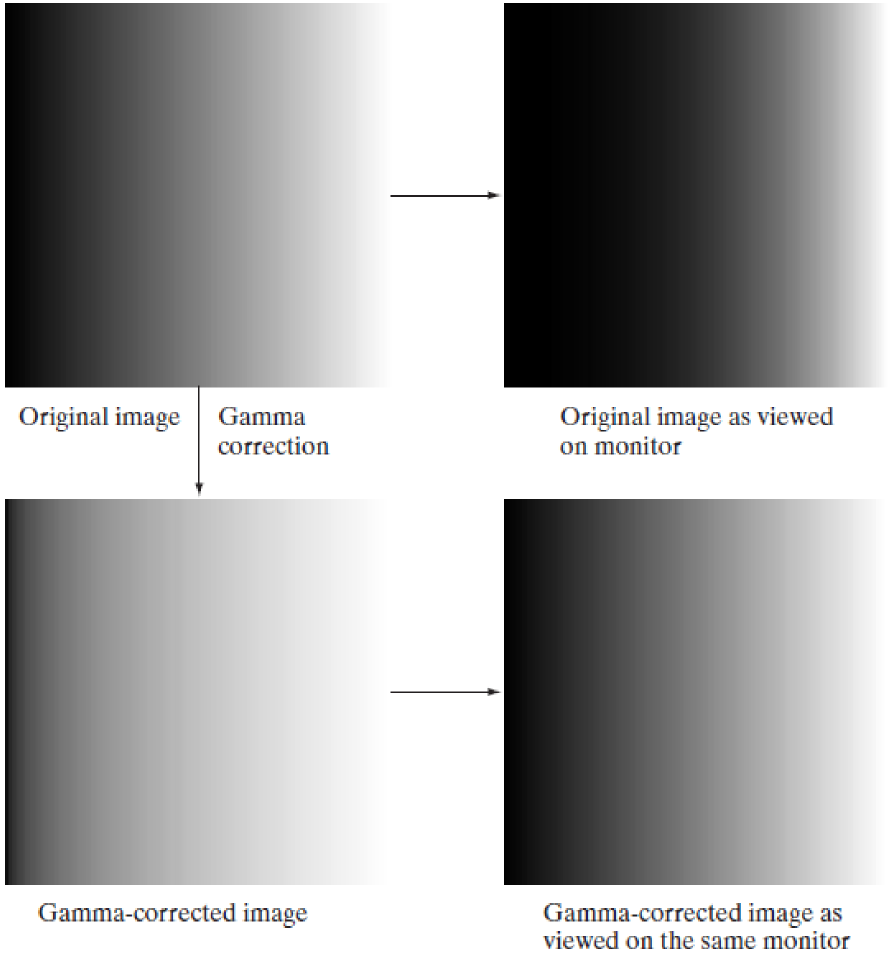


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases). All curves were scaled to fit in the range shown.



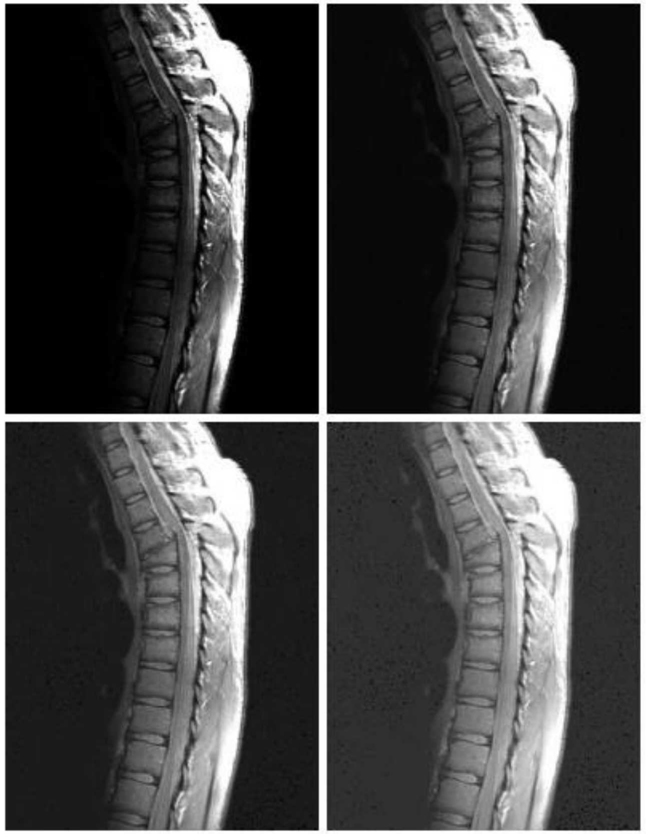
a b
c d

FIGURE 3.7
(a) Intensity ramp image. (b) Image as viewed on a simulated monitor with a gamma of 2.5. (c) Gamma-corrected image. (d) Corrected image as viewed on the same monitor. Compare (d) and (a).

POINT PROCESSING TECHNIQUES

POWER LAW(GAMMA) TRANSFORMATIONS:

- In addition to gamma correction, power-law transformations are useful for general-purpose contrast manipulation.



a b
c d

FIGURE 3.8 (a) Magnetic resonance image (MRI) 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 courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

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

- Practical implementation of some important transformations can be formulated only as piecewise functions.
- One of the simplest piecewise linear functions is a **contrast-stretching** transformation.
- Low-contrast images can result from poor illumination, lack of dynamic range in the imaging sensor, or even the wrong setting of a lens aperture during image acquisition
- ***Contrast stretching*** is a process that expands the range of intensity levels in an image so that it spans the full intensity range of the recording medium or display device

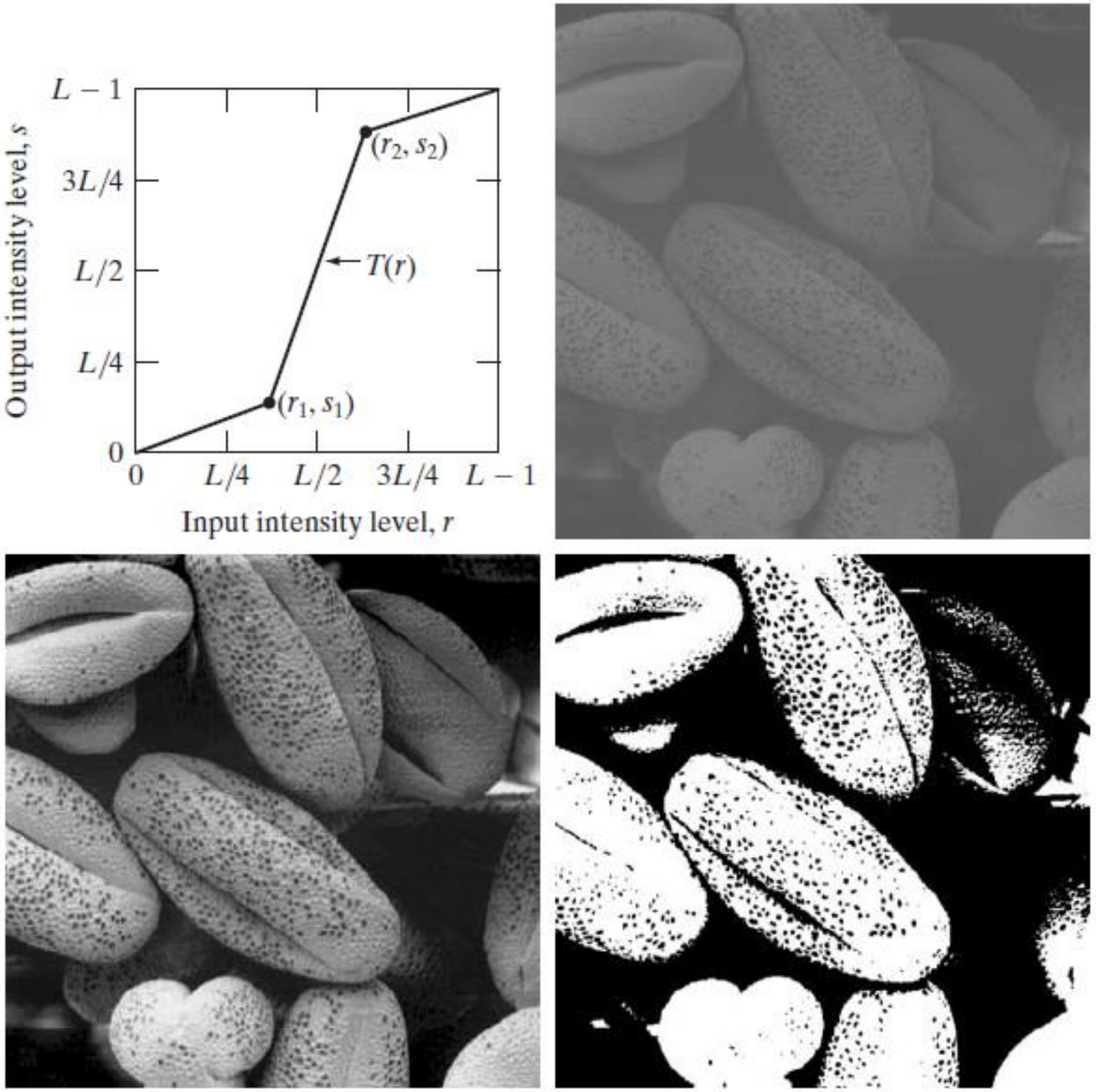


FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

PIECEWISE LINEAR TRANSFORMATION FUNCTIONS

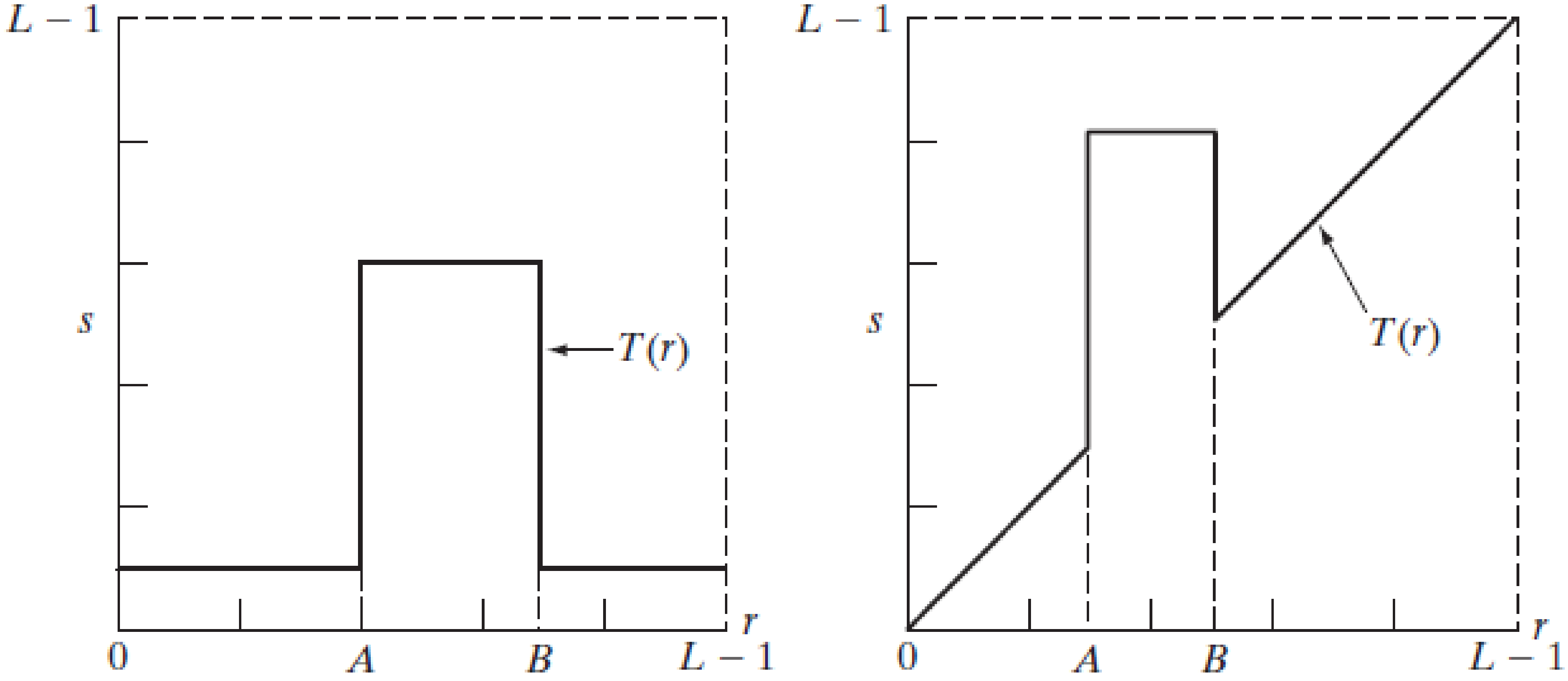
- The locations of points (r_1, s_1) $and(r_2, s_2)$ control the shape of the transformation function.
- If $r_1, = s_1$ and $r_2 = s_2$, the transformation is a linear function that produces no changes in intensity levels.
- If $r_1 = r_2$ and $s_1 = 0$ and $s_2 = L - 1$, the transformation becomes a ***thresholding function*** that creates a binary image, as illustrated in Fig. 3.2(b).
- Figure 3.10(b) shows an 8-bit image with low contrast.
- Figure 3.10(c) shows the result of contrast stretching, obtained by setting (r_1, s_1) = $(r_{min}, 0)$ and $(r_2, s_2) = (r_{max}, L - 1)$

PIECEWISE LINEAR TRANSFORMATION FUNCTIONS

- Where r_{min} and r_{max} denote the minimum and maximum intensity levels in the image, respectively.
- Thus, the transformation function stretched the levels linearly from their original range to the full range [0, L-1]
- Fig. 3.10(d) shows the result of using the thresholding function defined previously, with $(r_1, s_1) = (m, 0)$ and $(r_2, s_2) = (m, L - 1)$, where *m* is the mean intensity level in the image

PIECEWISE LINEAR TRANSFORMATION FUNCTIONS

- **Intensity(Gray level) slicing:**
- Highlighting a specific range of intensities in an image often is of interest.
- Applications include enhancing features such as masses of water in satellite imagery and enhancing flaws in X-ray images.
- The process, often called ***intensity-level slicing***

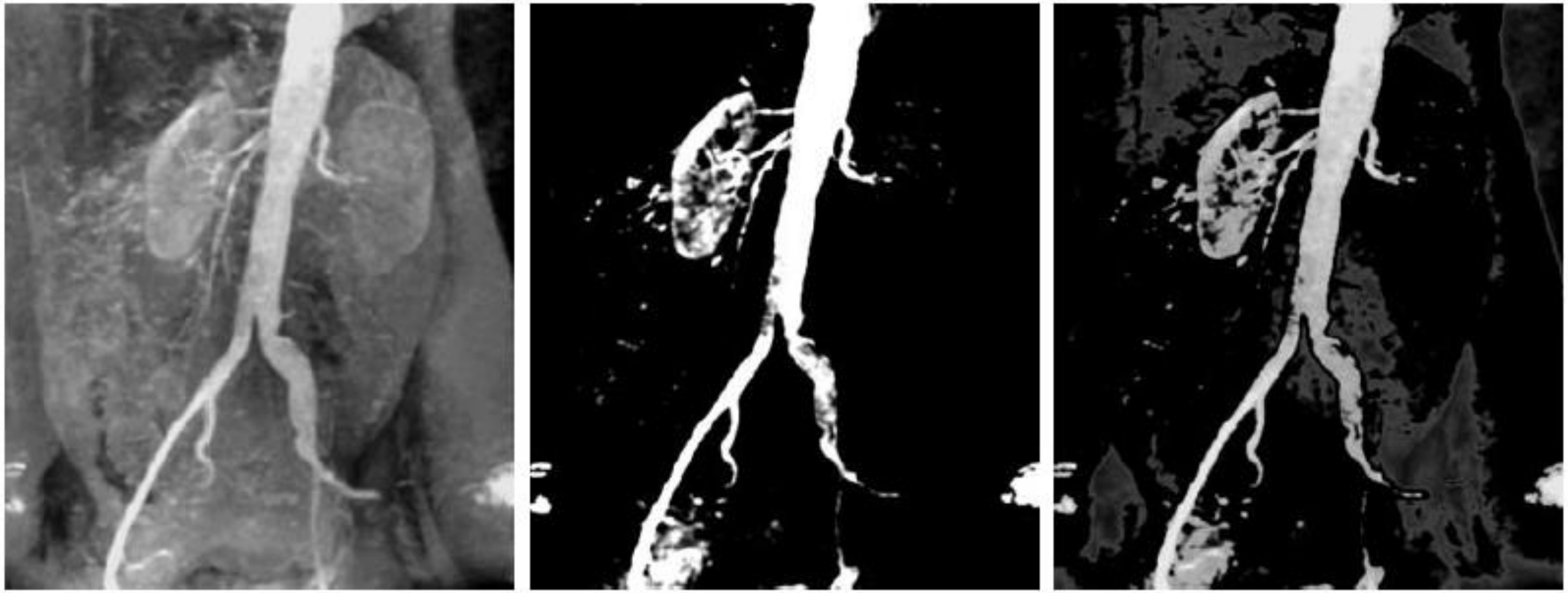


a b

FIGURE 3.11 (a) This transformation highlights intensity range $[A, B]$ and reduces all other intensities to a lower level. (b) This transformation highlights range $[A, B]$ and preserves all other intensity levels.

PIECEWISE LINEAR TRANSFORMATION FUNCTIONS

- It can be implemented in several ways, but most are variations of two basic themes.
- One approach is to display in one value (say, white) all the values in the range of interest and in another (say, black) all other intensities.
- This transformation, shown in Fig. 3.11(a), produces a binary image.
- The second approach, based on the transformation in Fig. 3.11(b), brightens (or darkens) the desired range of intensities but leaves all other intensity levels in the image unchanged.



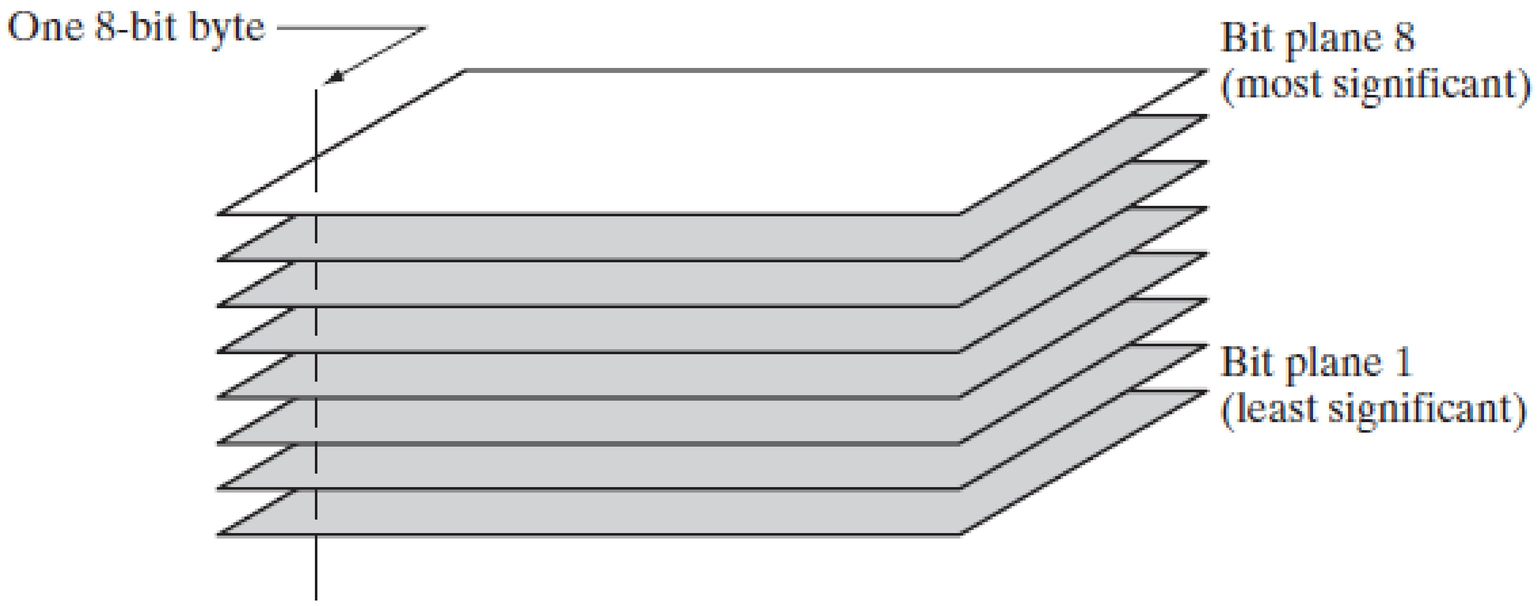
a b c

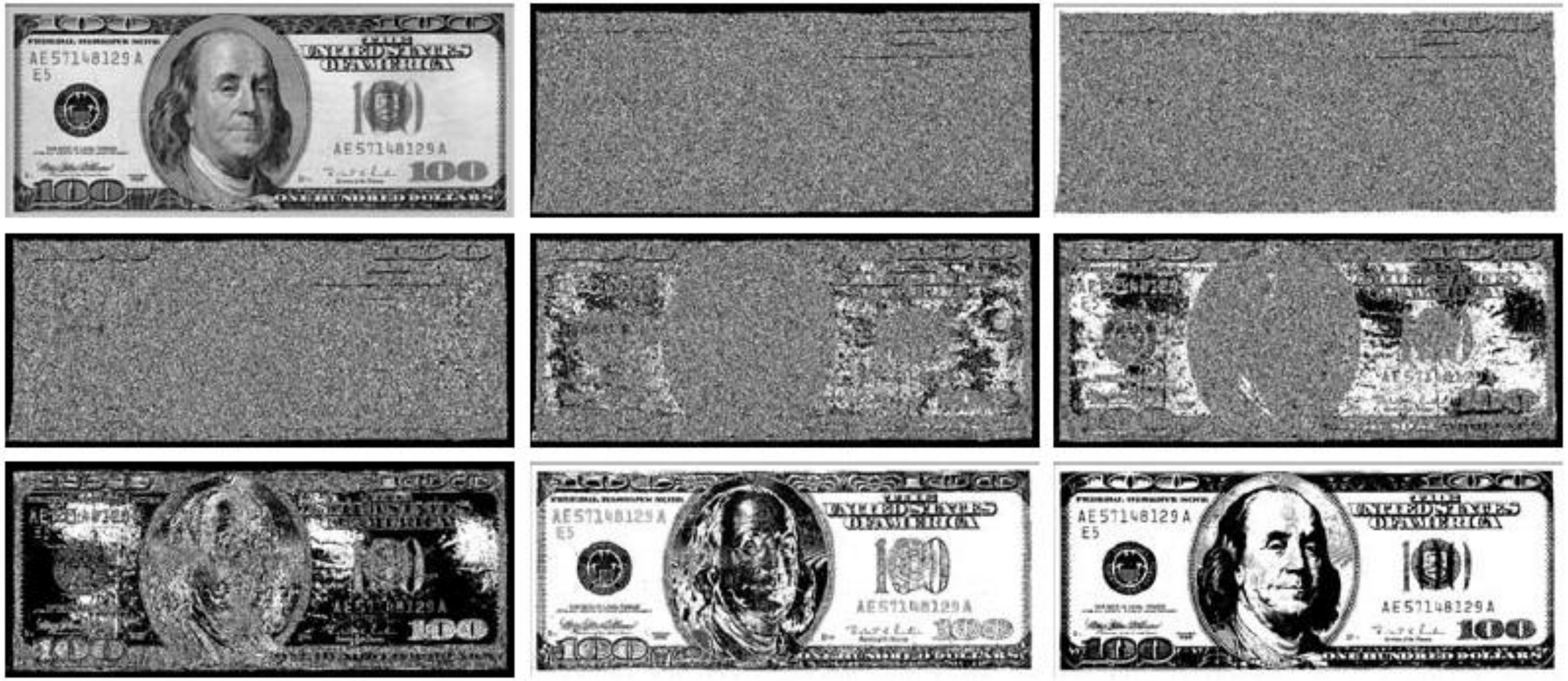
FIGURE 3.12 (a) Aortic angiogram. (b) Result of using a slicing transformation of the type illustrated in Fig. 3.11(a), with the range of intensities of interest selected in the upper end of the gray scale. (c) Result of using the transformation in Fig. 3.11(b), with the selected area set to black, so that grays in the area of the blood vessels and kidneys were preserved. (Original image courtesy of Dr. Thomas R. Gest, University of Michigan Medical School.)

PIECEWISE LINEAR TRANSFORMATION FUNCTIONS

- **Bit Plane Slicing:**
- Pixels are digital numbers composed of bits.
- For example, the intensity of each pixel in a 256-level gray-scale image is composed of 8 bits (i.e., one byte).
- Instead of highlighting intensity-level ranges, we could highlight the contribution made to total image appearance by specific bits.
- As Fig. 3.13 shows that an 8-bit image may be considered as being composed of eight 1-bit planes, with plane 1 containing the lowest-order bit of all pixels in the image and plane 8 all the highest-order bits.

FIGURE 3.13
Bit-plane representation of an 8-bit image.





a b c
d e f
g h i

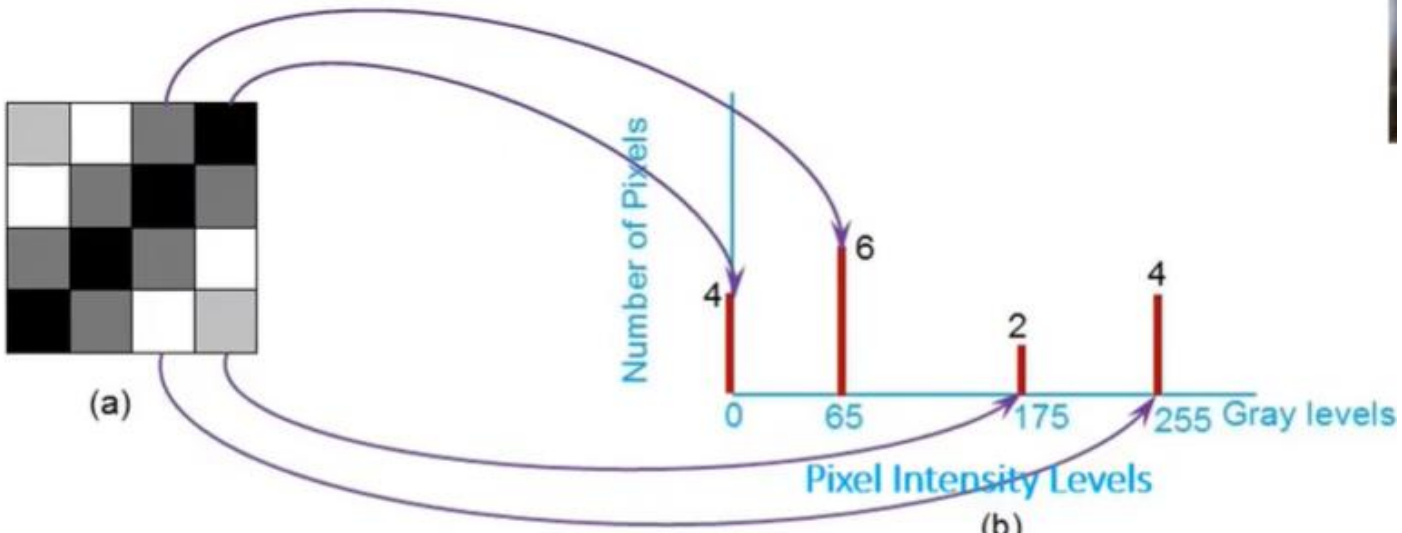
FIGURE 3.14 (a) An 8-bit gray-scale image of size 500 × 1192 pixels. (b) through (i) Bit planes 1 through 8, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image.

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).

HISTOGRAM OF A DIGITAL IMAGE

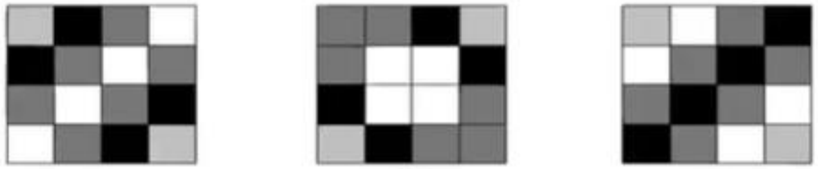
- The ***histogram*** of a digital image with intensity levels in the range [0,L-1] is a discrete function $\mathbf{h(r_k)=n_k}$
- where r_k is the *k*th intensity value
- n_k- number of pixels in the image with intensity r_k



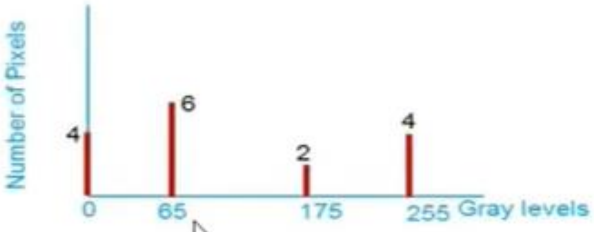
Number of Pixels
4
6
2
4
0
65
175
255 Gray levels
Pixel Intensity Levels

(a)

(b)

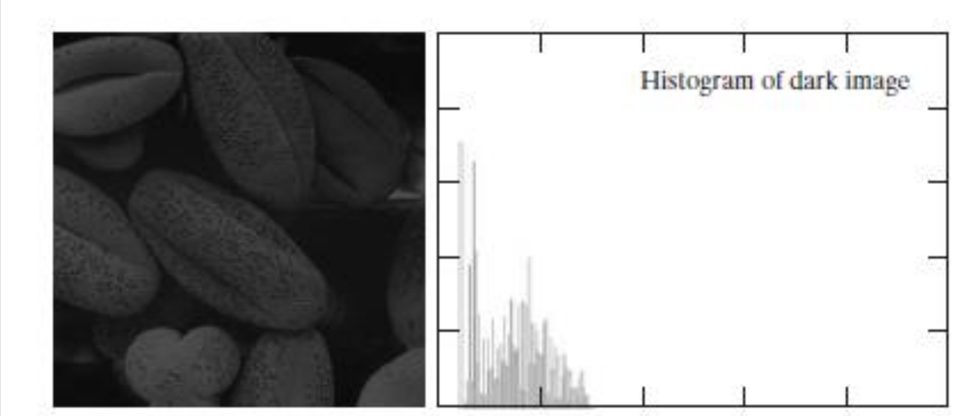


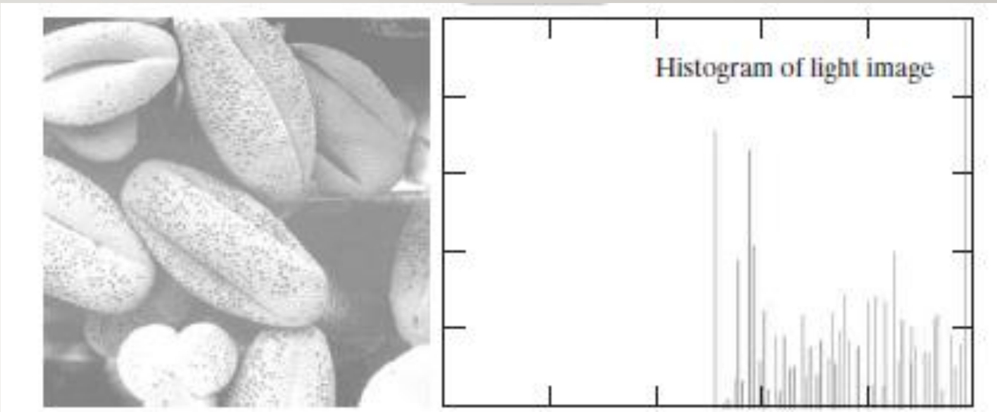
(a) (b) (c)

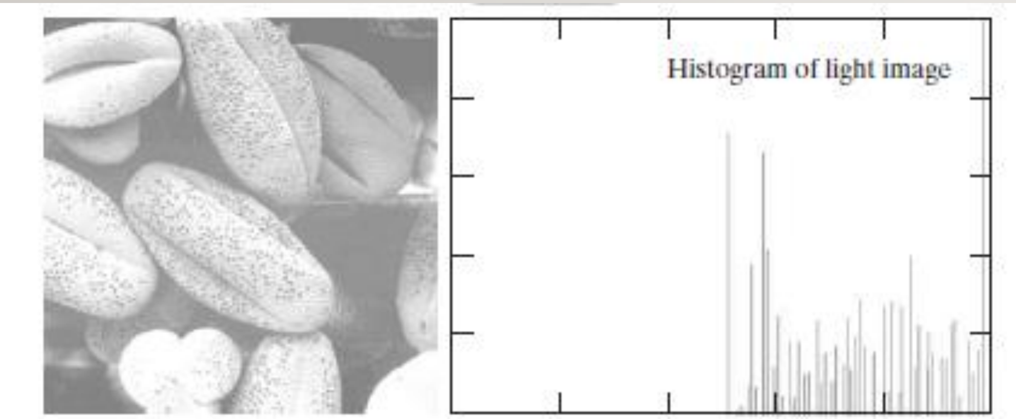


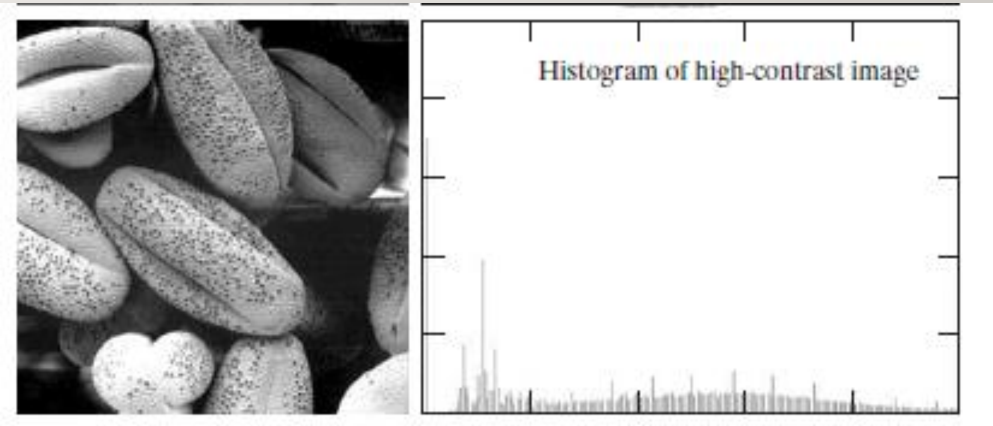
Number of Pixels
4
6
2
4
0
65
175
255 Gray levels

(d)









The right side of the figure shows the histograms corresponding to these images.
The horizontal axis of each histogram plot corresponds to intensity values r_k

HISTOGRAM OF A DIGITAL IMAGE

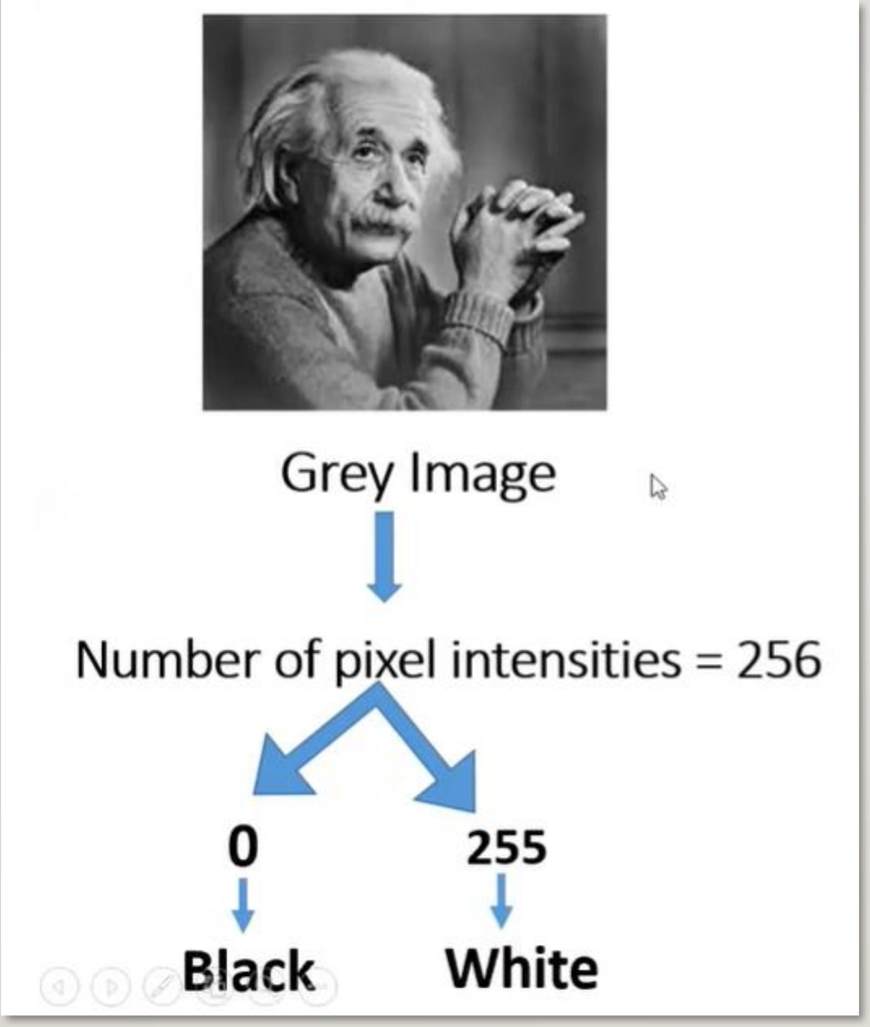
- The vertical axis corresponds to values of $h(r_k)=n_k$ or $p(r_k)=n_k/MN$ if the values are normalized.
- Thus, histograms may be viewed graphically plots of $h(r_k)=n_k$ versus r_k or $p(r_k)=n_k/MN$ versus r_k
- Dark image that the components of the histogram are concentrated on the low (dark) side of the intensity scale.
- Similarly, the components of the histogram of the light image are biased toward the high side of the scale.

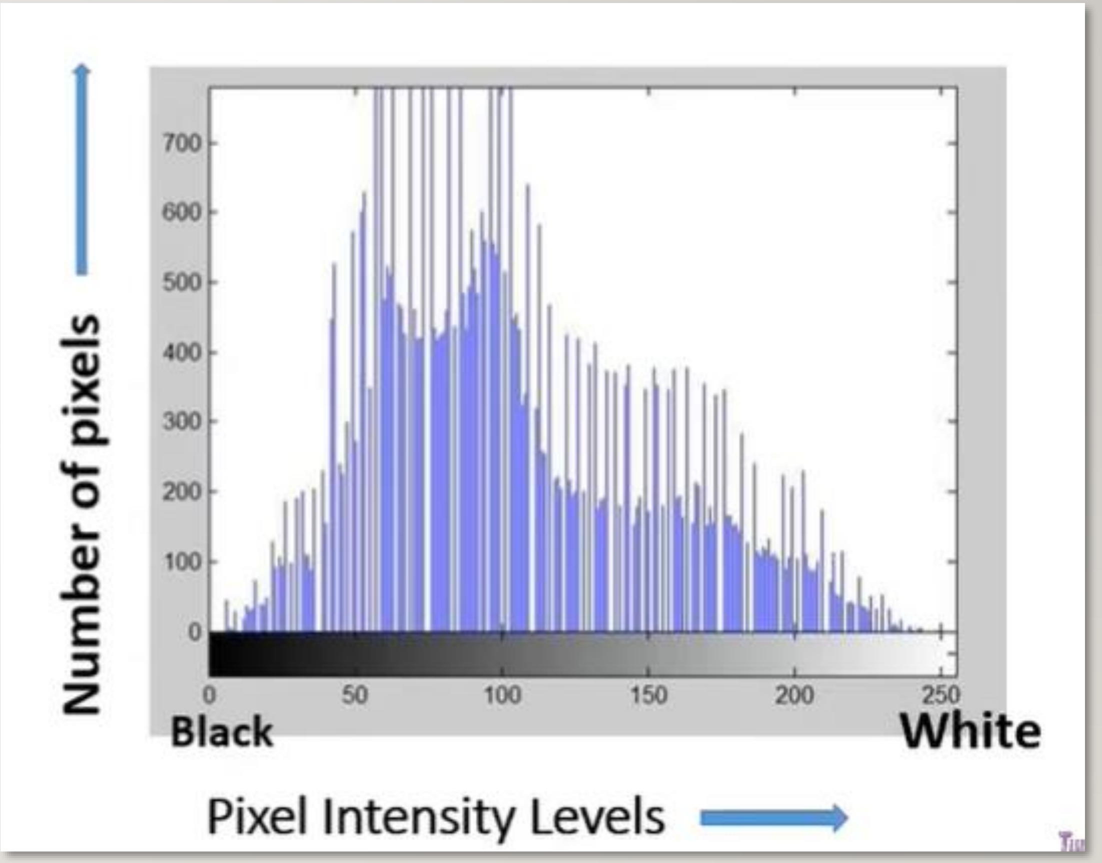
HISTOGRAM OF A DIGITAL IMAGE

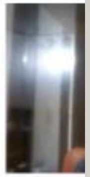
- Low Contrast Image: Narrow histogram located towards the middle of the intensity scale.

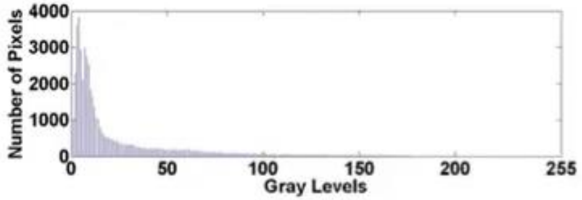
(For a monochrome image this implies a dull, washed-out gray look)

- High Contrast Image: Cover a wide range of the intensity scale

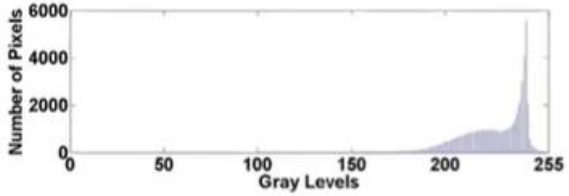

Grey Image
Number of pixel intensities = 256
0
255
Black
White


Number of pixels
700
600
500
400
300
200
100
0
0
50
100
150
200
250
Black
White
Pixel Intensity Levels



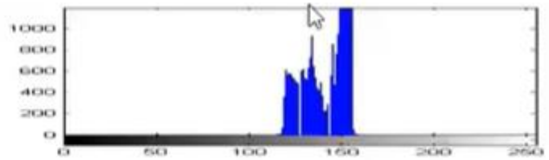


More Black Pixels

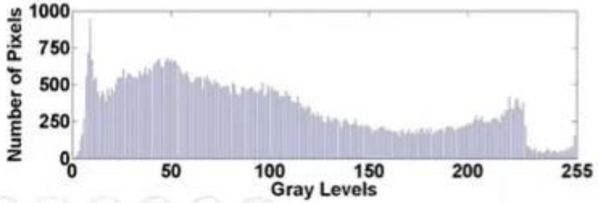


(a)

More White Pixels

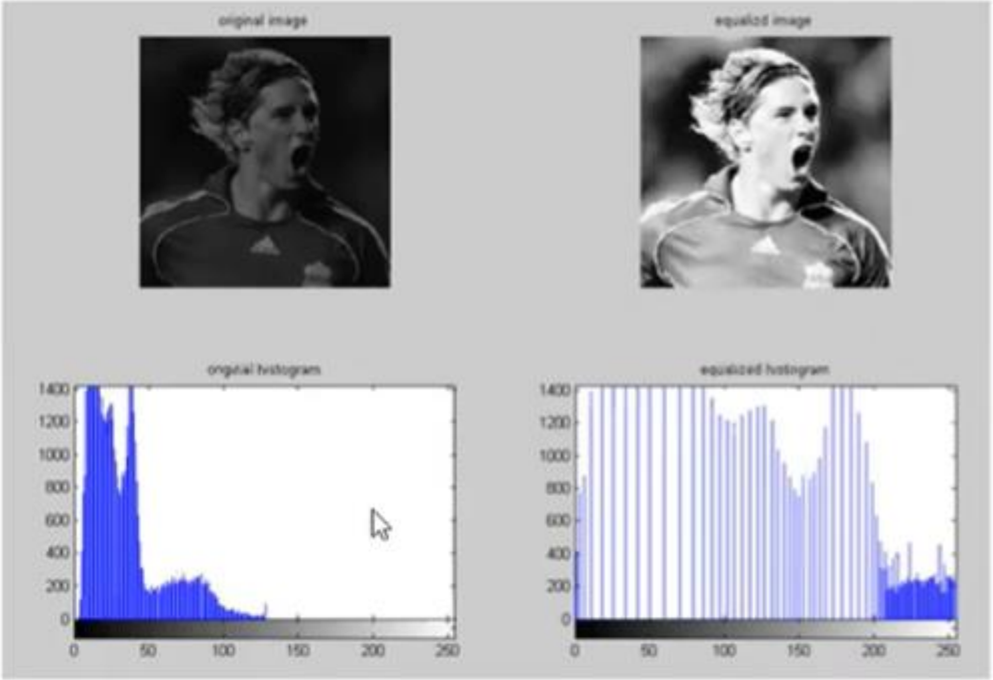


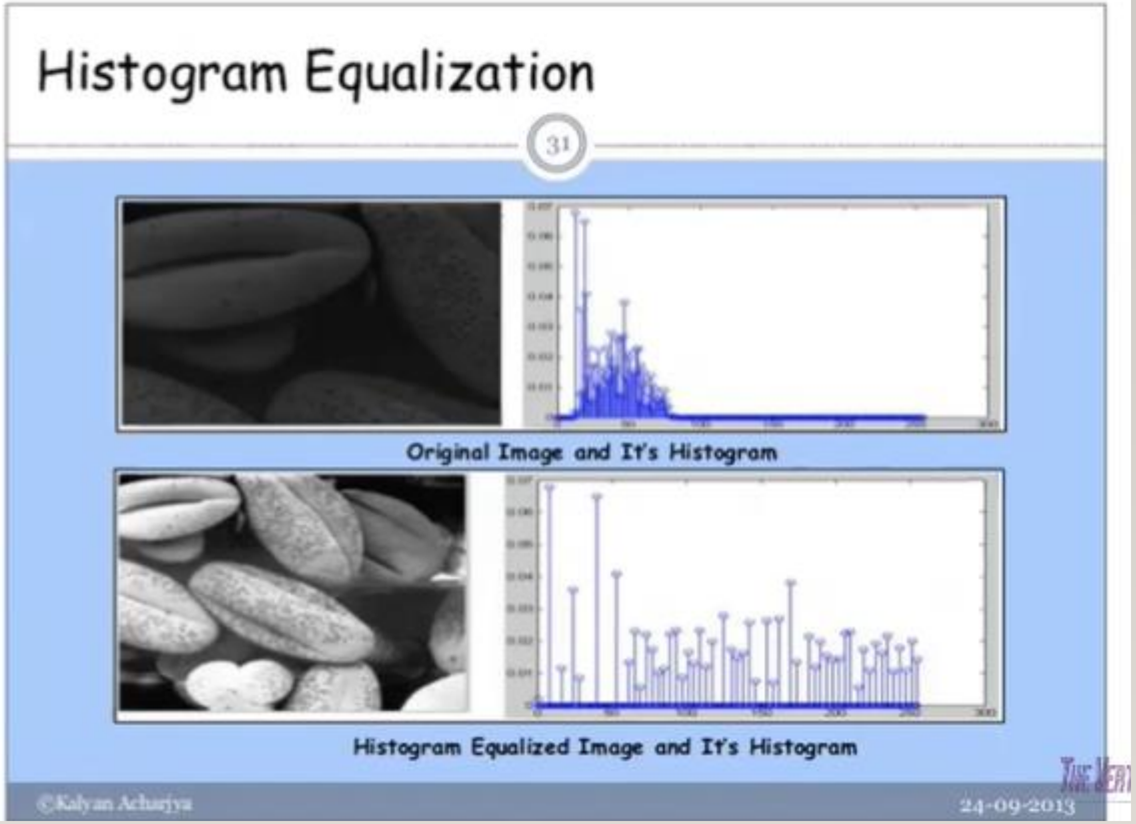
Dull Image



Pixels of all intensity levels

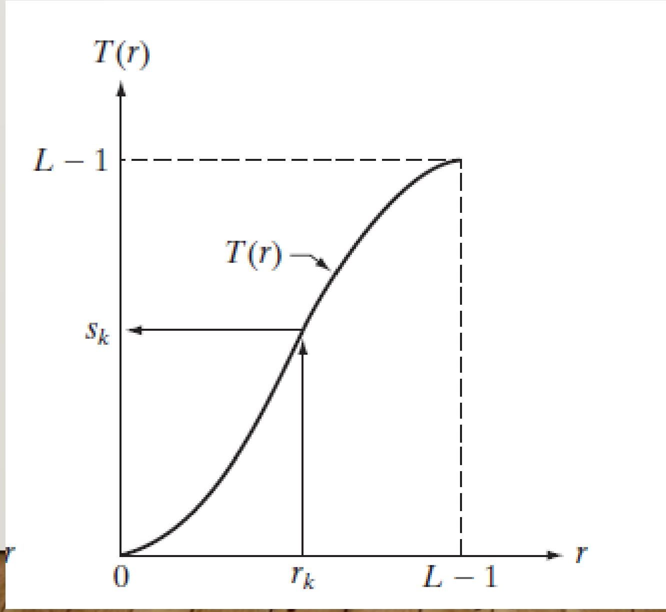
Histogram Applications





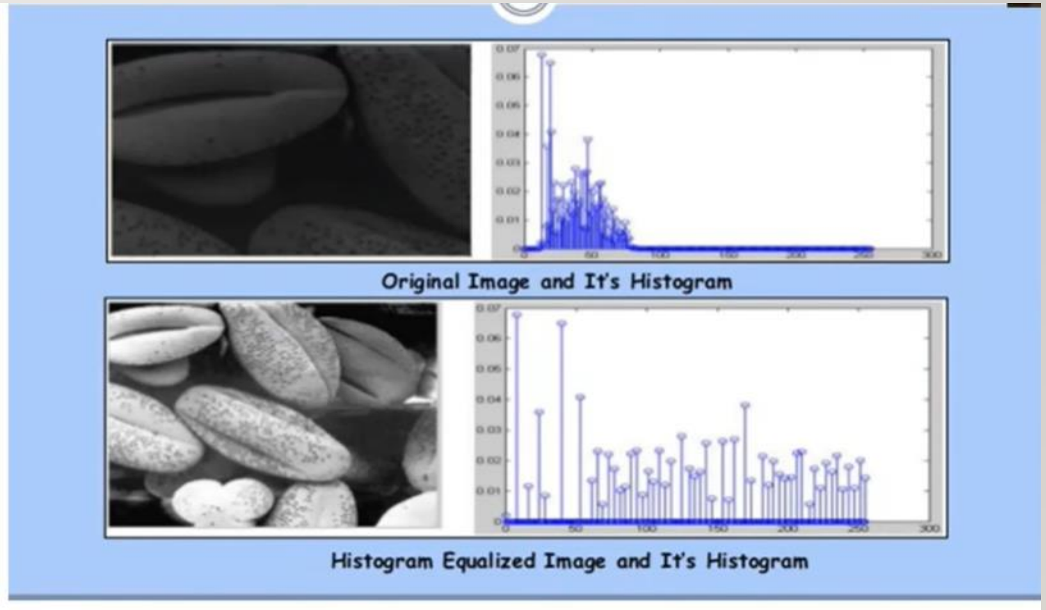
HISTOGRAM EQUALIZATION

- Let the variable *r* denote the intensities of an image to be processed.
- r ranges from [0,L-1] where r=0 represents black and r=L-1 represents white
- Gray level transformation is singular and monotonically increasing.

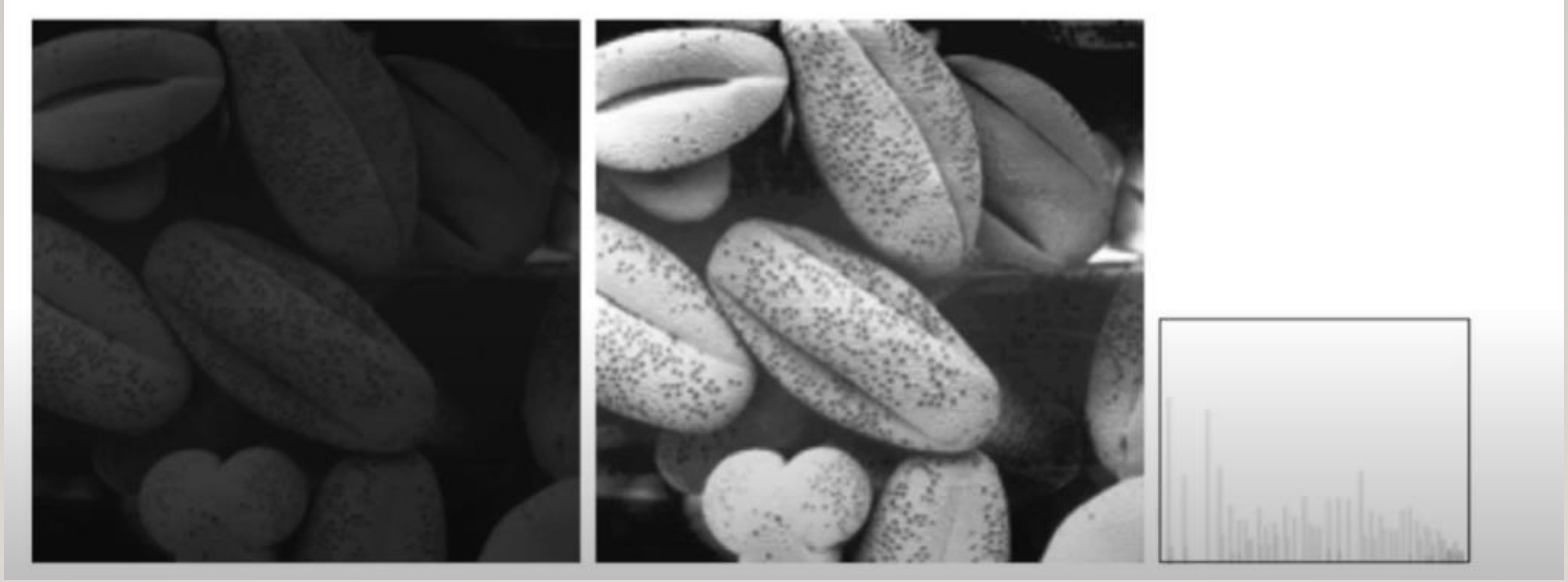


HISTOGRAM EQUALISATION

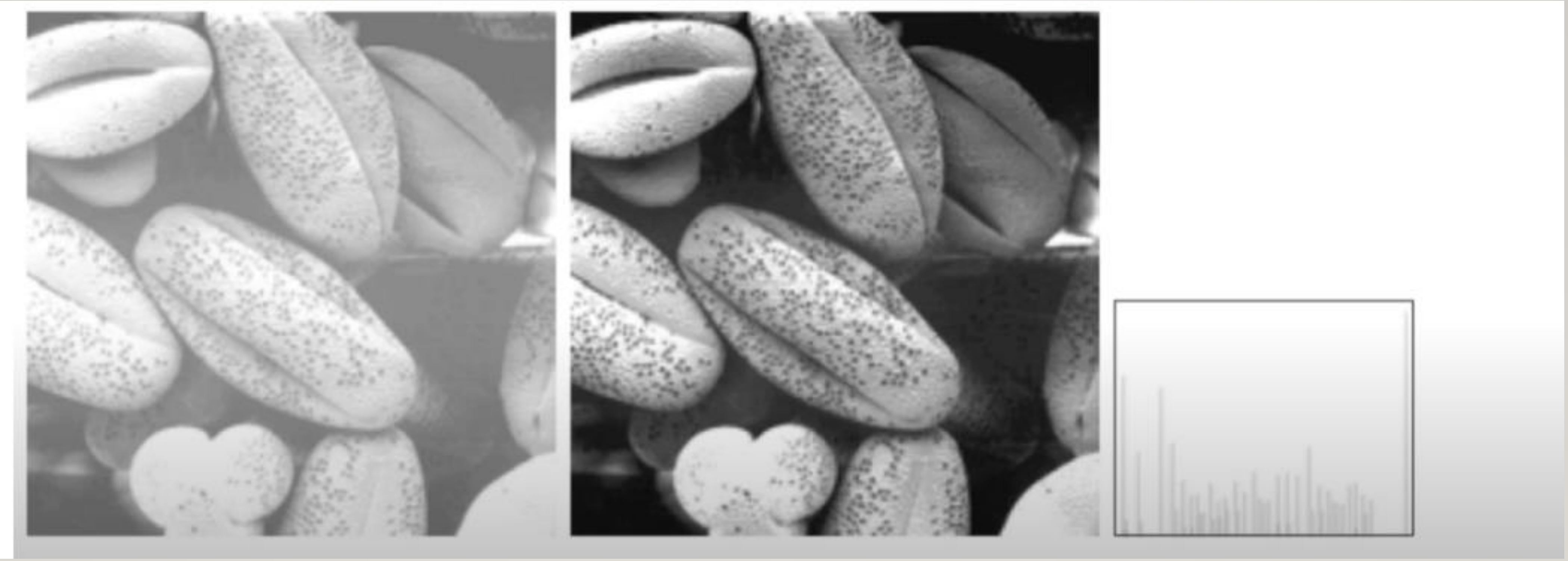
- Process that spreads out the intensity levels equally in an image so that they are evenly distributed across their range
- In this technique the histogram of the image is as flat as possible
- Produce visually pleasing results across wider range of image



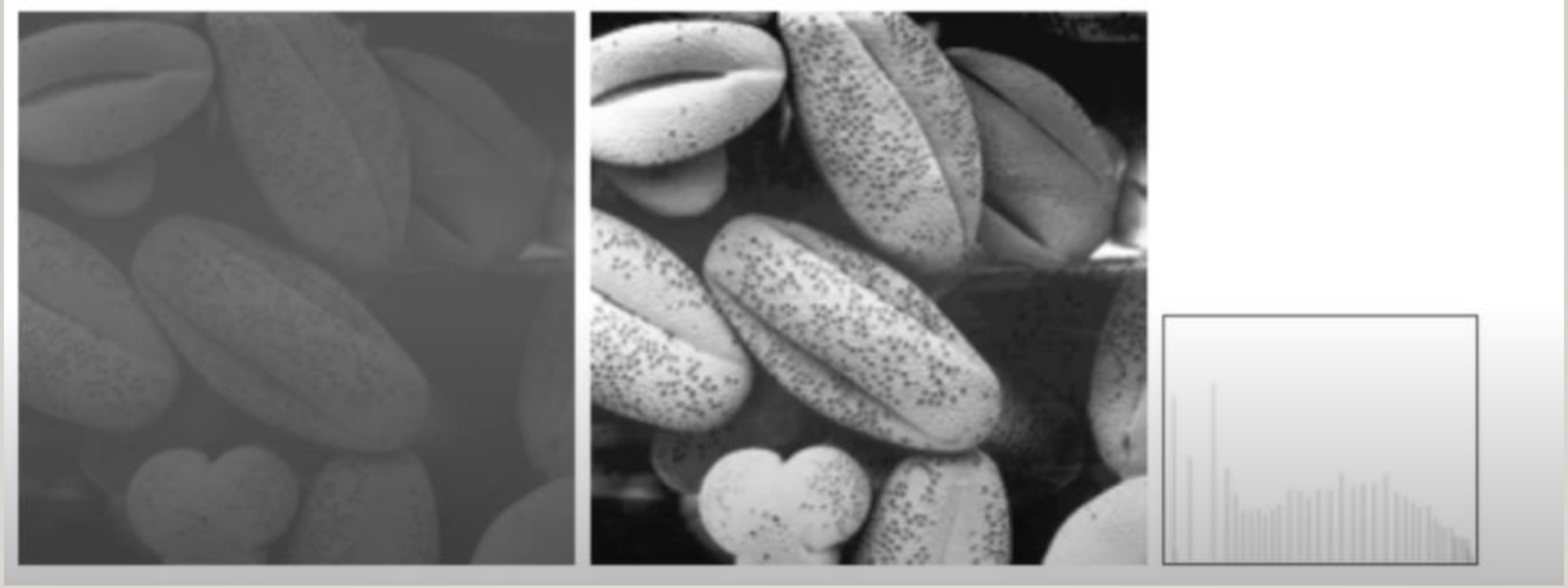
Results of Histogram Equalization-Dark Image



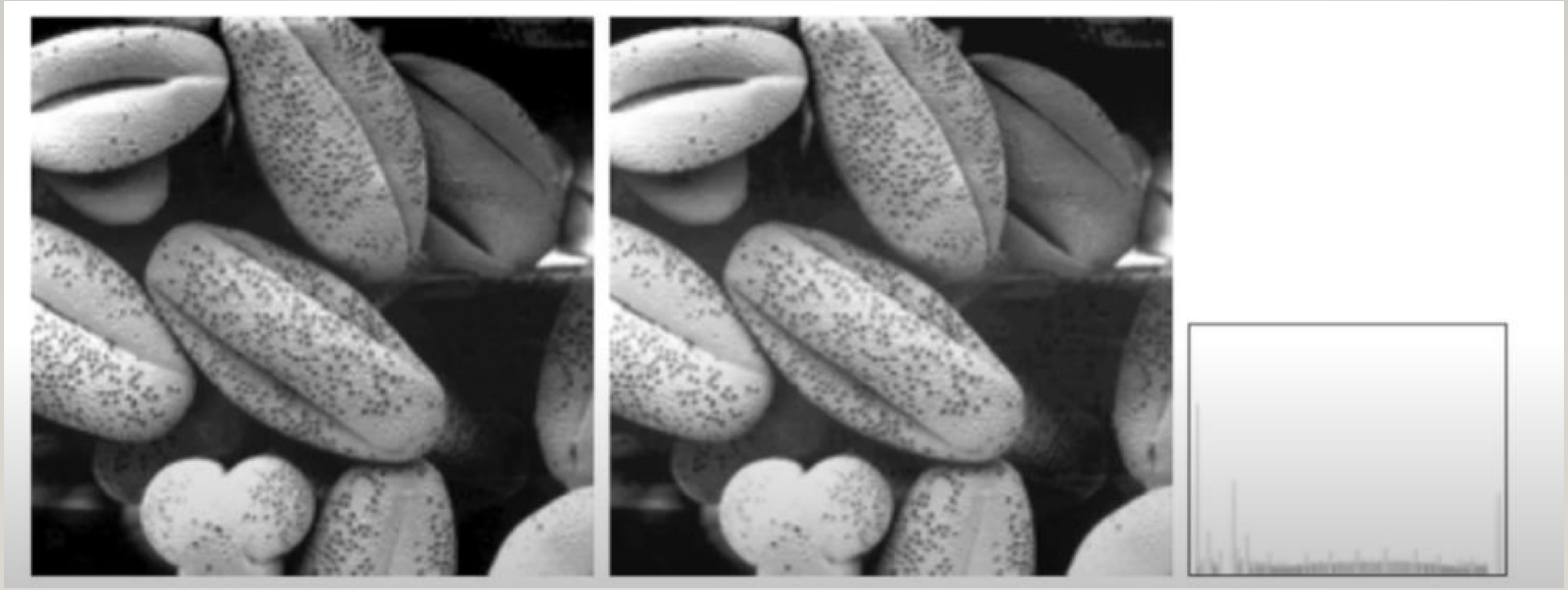
Results of Histogram Equalization-Light Image



Results of Histogram Equalization-Low Contrast Image



Results of Histogram Equalization- High Contrast Image



Example 5.1 *Perform histogram equalisation of the image* $\begin{bmatrix} 4 & 4 & 4 & 4 & 4 \\ 3 & 4 & 5 & 4 & 3 \\ 3 & 5 & 5 & 5 & 3 \\ 3 & 4 & 5 & 4 & 3 \\ 4 & 4 & 4 & 4 & 4 \end{bmatrix}$.

Solution The maximum value is found to be 5. We need a minimum of 3 bits to represent the number. There are eight possible gray levels from 0 to 7. The histogram of the input image is given below:

Gray level	0	1	2	3	4	5	6	7
Number of pixels	0	0	0	6	14	5	0	0

Step 1 Compute the running sum of histogram values.

The running sum of histogram values is otherwise known as cumulative frequency distribution.

Gray level	0	1	2	3	4	5	6	7
Number of pixels	0	0	0	6	14	5	0	0
Running sum	0	0	0	6	20	25	25	25

Step 2 Divide the running sum obtained in Step 1 by the total number of pixels. In this case, the total number of pixels is 25.

Gray level	0	1	2	3	4	5	6	7
Number of pixels	0	0	0	6	14	5	0	0
Running sum	0	0	0	6	20	25	25	25
Running Sum/Total number of pixels	0/25	0/25	0/25	6/25	20/25	25/25	25/25	25/25

Step 3 Multiply the result obtained in Step 2 by the maximum gray-level value, which is 7 in this case.

Gray level	0	1	2	3	4	5	6	7
Number of pixels	0	0	0	6	14	5	0	0
Running Sum	0	0	0	6	20	25	25	25
Runningsum / Total number of pixels	0/25	0/25	0/25	6/25	20/25	25/25	25/25	25/25
Multiply the above result by maximum gray level	$\frac{0}{25} * 7$	$\frac{0}{25} * 7$	$\frac{0}{25} * 7$	$\frac{6}{25} * 7$	$\frac{20}{25} * 7$	$\frac{25}{25} * 7$	$\frac{25}{25} * 7$	$\frac{25}{25} * 7$

The result is then rounded to the closest integer to get the following table:

Gray level	0	1	2	3	4	5	6	7
Number of pixels	0	0	0	6	14	5	0	0
Running Sum	0	0	0	6	20	25	25	25
Running Sum/Total number of pixels	0/25	0/25	0/25	6/25	20/25	25/25	25/25	25/25
Multiply the above result by maximum gray level	0	0	0	2	6	7	7	7

Step 4 Mapping of gray level by a one-to-one correspondence:

Original gray level	Histogram equalised values
0	0
1	0
2	0
3	2
4	6
5	7
6	7
7	7

The original image and the histogram equalised image are shown side by side.

$$\begin{bmatrix} 4 & 4 & 4 & 4 & 4 \\ 3 & 4 & 5 & 4 & 3 \\ 3 & 5 & 5 & 5 & 3 \\ 3 & 4 & 5 & 4 & 3 \\ 4 & 4 & 4 & 4 & 4 \end{bmatrix} \xrightarrow[\text{Equalisation}]{\text{Histogram}} \begin{bmatrix} 6 & 6 & 6 & 6 & 6 \\ 2 & 6 & 7 & 6 & 2 \\ 2 & 7 & 7 & 7 & 2 \\ 2 & 6 & 7 & 6 & 2 \\ 6 & 6 & 6 & 6 & 6 \end{bmatrix}$$

Original image **Histogram equalised image**

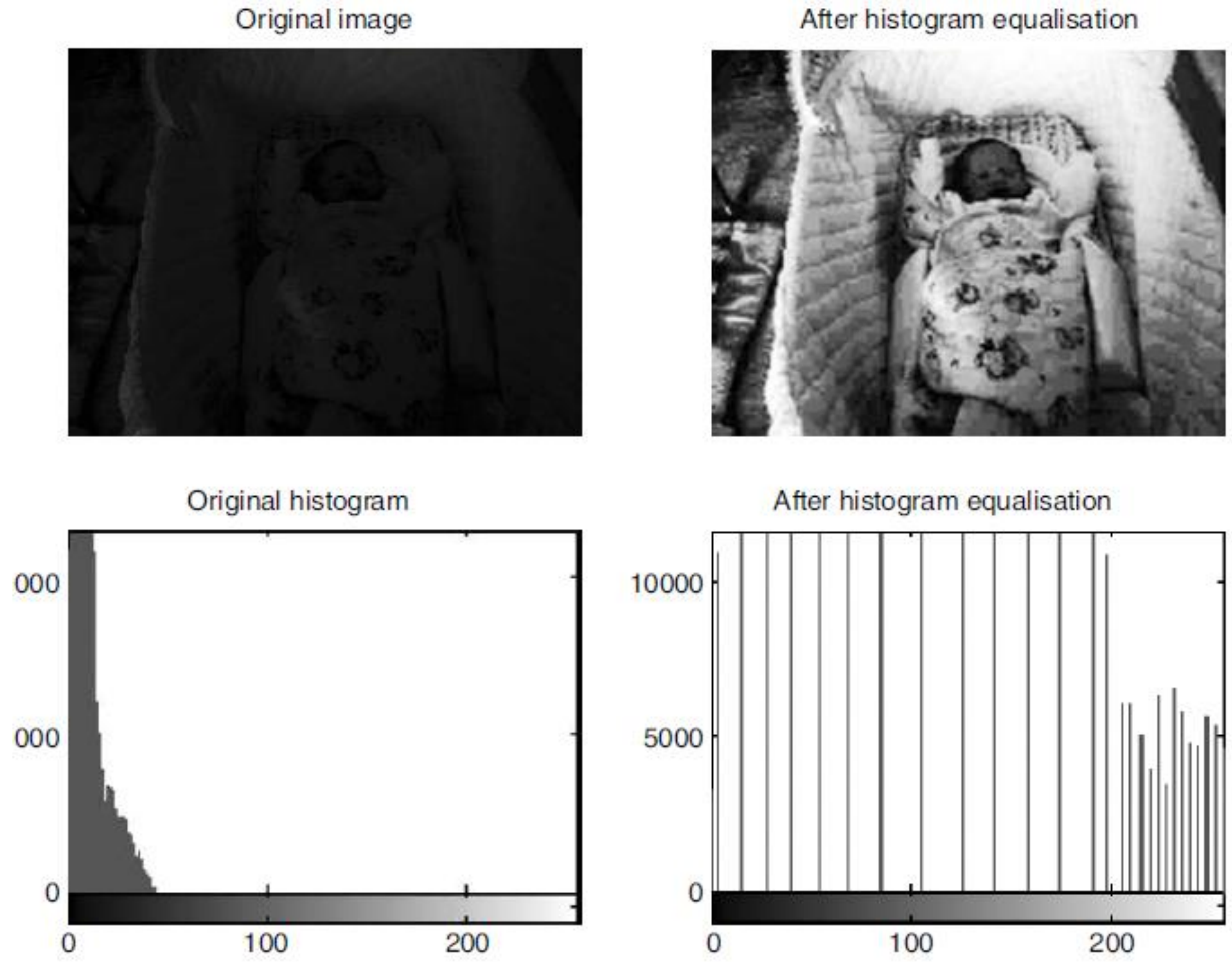


Fig. 5.11 *Results of histogram equalisation*

HISTOGRAM SPECIFICATION(MATCHING)

- When automatic enhancement is desired, Histogram equalisation is a good approach because the results from this technique are predictable, and the method is simple to implement.
- It is useful sometimes 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*.**

HISTOGRAM MATCHING-PROCEDURE

- Goal: Take an input image and generate an output image based upon the shape of a specific(or reference) histogram.

Example : Given histogram (a) & (b), modify histogram (a) as given by histogram (b)

(a)

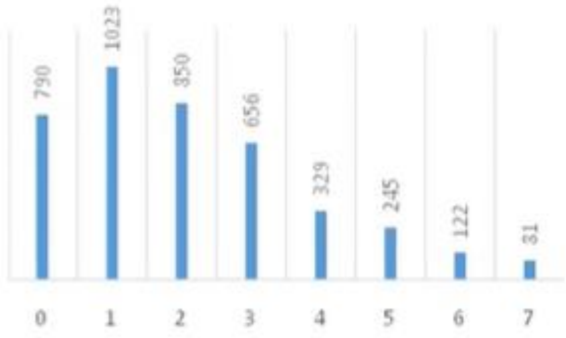
Gray level.	0	1	2	3	4	5	6	7
No. of pixels	790	1023	850	656	329	245	122	81

(b)

Gray level.	0	1	2	3	4	5	6	7
No. of pixels	0	0	0	614	819	1230	819	614

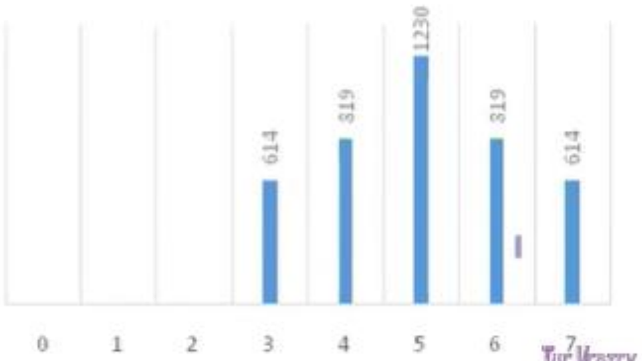
(a)

Gray level.	0	1	2	3	4	5	6	7
No. of pixels	790	1023	850	656	329	245	122	81



(b)

Gray level.	0	1	2	3	4	5	6	7
No. of pixels	0	0	0	614	819	1230	819	614



THE VERTEX

Equalize Histogram(a)

Gray level	nk	PDF	CDF	Sk x 7	Round off	New nk.
0	790	0.19	0.19	1.33	1	790
1	1023	0.25	0.44	3.08	3	1023
2	850	0.21	0.65	4.55	5	850
3	656	0.16	0.81	5.67	6	656+329	985
4	329	0.08	0.89	6.23	6	
5	245	0.06	0.95	6.65	7	245 +122+81	448
6	122	0.03	0.98	6.86	7	
7	81	0.02	1	7	7	
	N=4096					

Equalize Histogram(b)

Gray level	nk	PDF	CDF	Sk x 7	Round off
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	614	0.149	0.149	1.05	1
4	819	0.20	0.35	2.50	3
5	1230	0.30	0.65	4.55	5
6	819	0.20	0.85	5.97	6
7	614	0.15	1	7	7
	N=4096				

Mapping

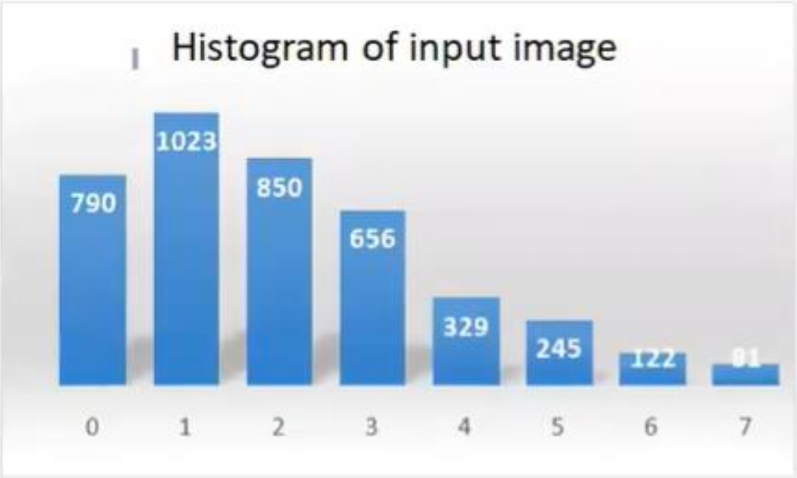
First and last columns of histogram (b)

Gray level	Round off
0	0
1	0
2	0
3	1
4	3
5	5
6	6
7	7

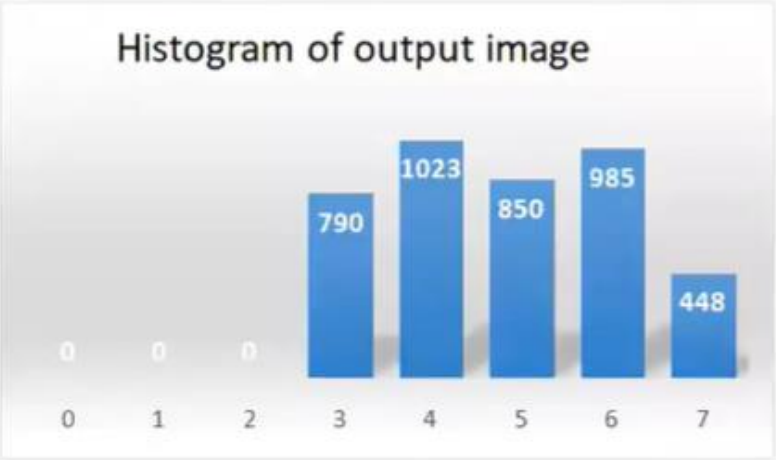
Last two columns of histogram (a)

Round off	New nk.
1	790
3	1023
5	850
6	
6	985
7	
7	448
7	

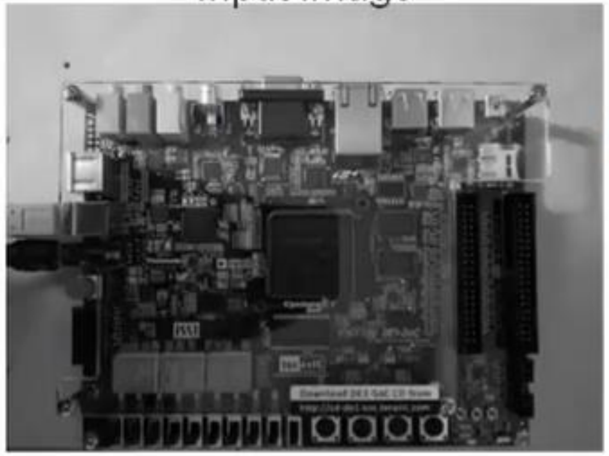
Gray level.	0	1	2	3	4	5	6	7
No. of pixels	0	0	0	790	1023	850	985	448



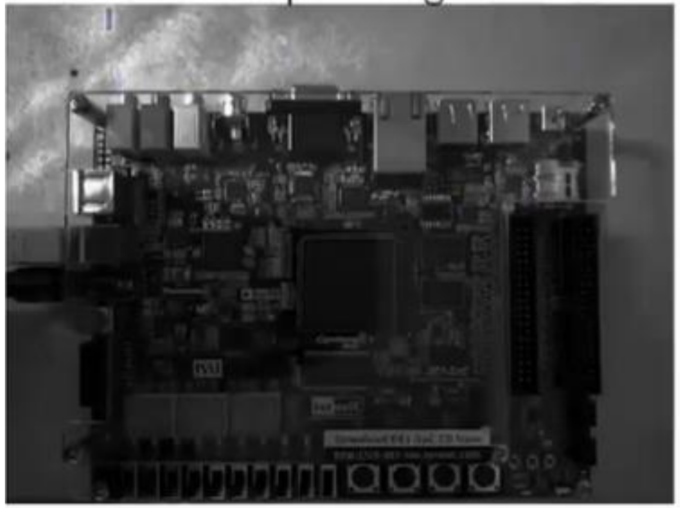
Plot histogram for modified image.



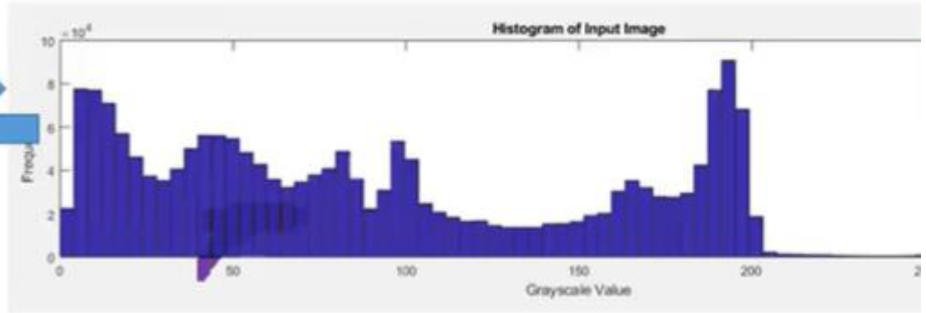
Input image



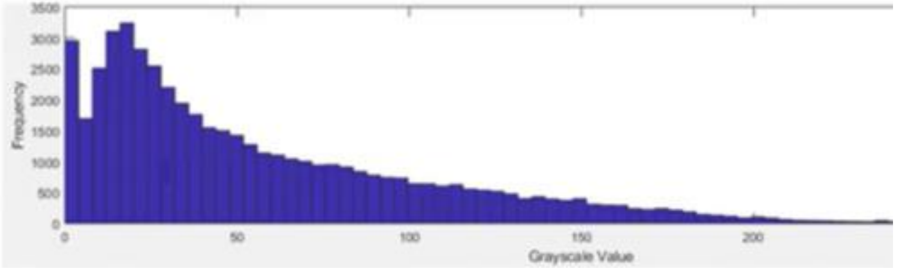
Output image



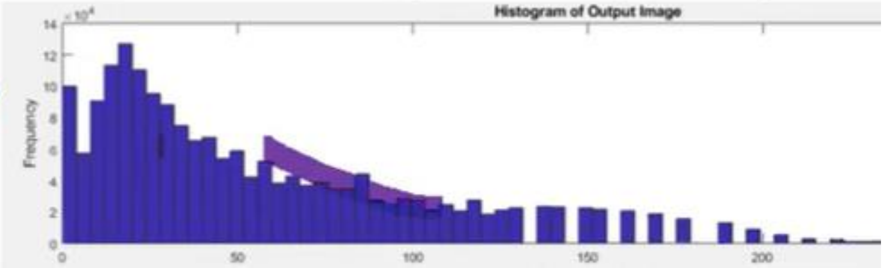
Histogram of input image



Reference histogram that emphasize the lowe



Histogram of output image



NEIGHBOURHOOD PROCESSING-MECHANICS OF SPATIAL FILTERING

- Some operations work with values of image pixels in the neighbourhood and corresponding values of a sub image with same dimensions as the neighbourhood.
- Sub image is called a filter mask, kernel, template or window.
- Values in the filter sub image is referred to as coefficients rather than pixels.

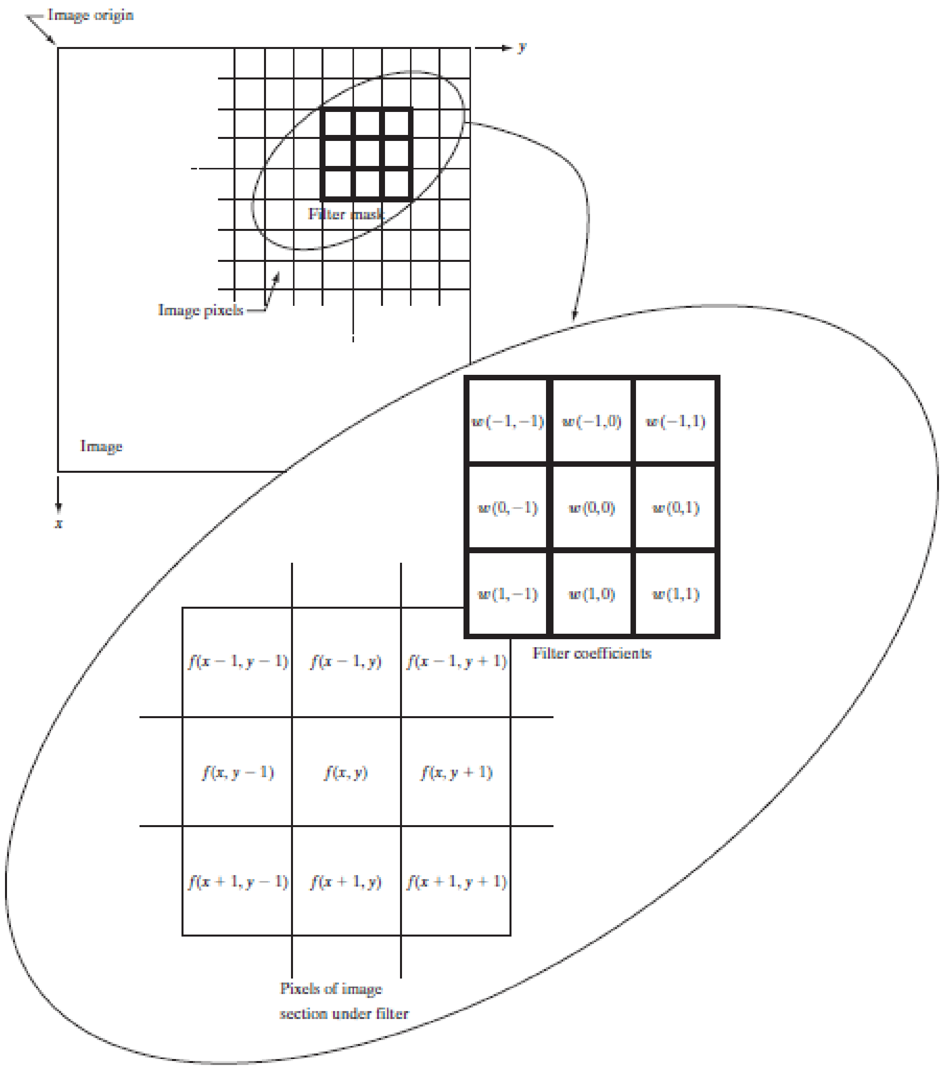


FIGURE 3.28 The mechanics of linear spatial filtering using a 3×3 filter mask. The form chosen to denote the coordinates of the filter mask coefficients simplifies writing expressions for linear filtering.

NEIGHBOURHOOD PROCESSING-MECHANICS OF SPATIAL FILTERING

- Process involves moving the filter mask from point to point in an image
- At each point (x,y), response of the filter is calculated using a predefined relationship
- Linear Spatial Filtering-Response=Sum of products of filter coefficients and corresponding image pixels in area spanned by filter mask.
- For a 3X3 neighbourhood, at any point (x, y) in the image, the response of the filter is the sum of products of the filter coefficients and the image pixels encompassed by the filter:
- Centre coefficient of the filter, , aligns with the pixel at location (x, y).

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

NEIGHBOURHOOD PROCESSING-MECHANICS OF SPATIAL FILTERING

- In general, linear spatial filtering of an image of size MXN with a filter of size is given by the expression:

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

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

- where *x* and *y* are varied so that each pixel in w visits every pixel in *f*.

NEIGHBOURHOOD PROCESSING-MECHANICS OF SPATIAL FILTERING

- The process of linear filtering is similar to the frequency domain concept called convolution.
- Hence Linear spatial filtering is also referred to as "convolving a mask with an image"
- Similarly filter masks are sometimes called convolution masks or convolution kernels.
- Non linear spatial filters also operate on neighbourhood and the mechanics of sliding is the same.
- Filtering option is based conditionally on values of neighbourhood pixel but often do not follow the sum of products operation followed in linear spatial filters

NEIGHBOURHOOD PROCESSING-MECHANICS OF SPATIAL FILTERING

- Eg: Noise reduction can be achieved using nonlinear spatial filtering whose basic function is to compute median-gray level value in the neighbourhood on which the filter is located.
- Computation of the median is a nonlinear operation as is computation of variance.
- **When centre of pixel approaches the border of the image?**
- Consider a square mask of nXn.
- At least one edge of the mask will coincide with the border of the image when the centre of the mask is at a distance of (n-1)/2 pixels away from the border of the image.

NEIGHBOURHOOD PROCESSING-MECHANICS OF SPATIAL FILTERING

- If centre of mask moves any closer to the border, one or more rows or columns will be located outside the image plane.
- **Solution:**
- Limit the centre of the mask to be at a distance no less than (n-1)/2 pixels from the border.
- Result in filtered image will be smaller but all pixels in filtered image will be processed with full mask
- Other approaches-Zero padding. Padding rows and columns of the image with “0’s”

LOW PASS FILTER-SMOOTHING SPATIAL FILTERS

- An image is smoothed by decreasing the disparity between the pixel values by averaging nearby pixels
- Using a low pass filter-tends to retain low frequency information within an image and reducing the high frequency information.
- High Frequency Information-Edges in an image

TYPES OF LPF

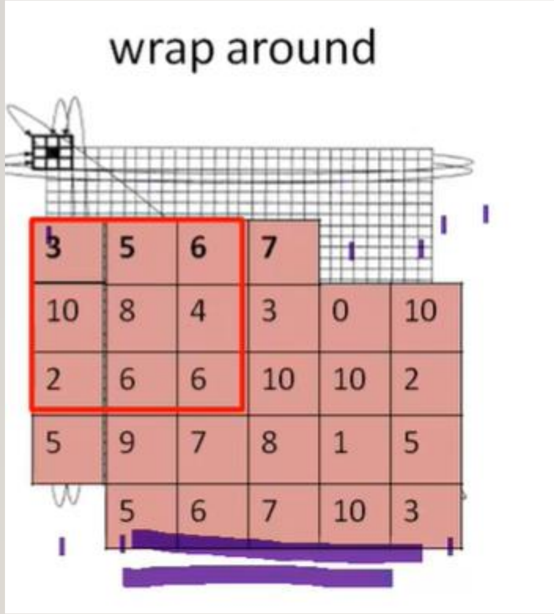
Averaging(Mean) Filter:

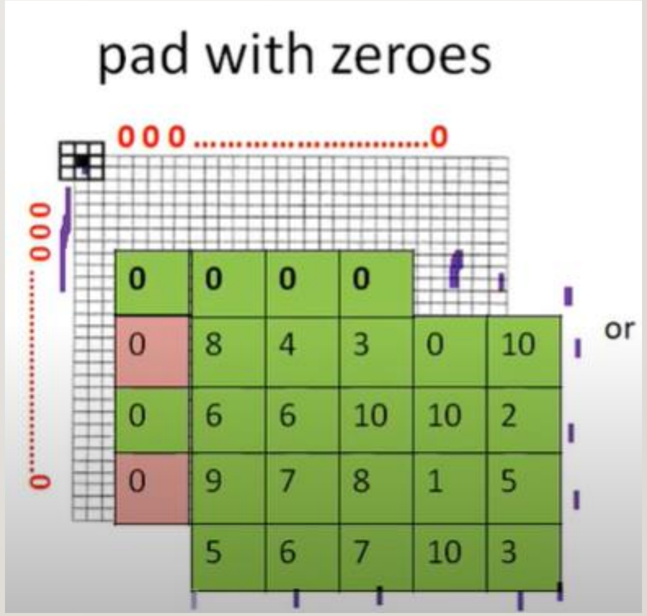
- Used as a method of smoothing images reducing the amount of intensity variation between one pixel and reducing noise in images

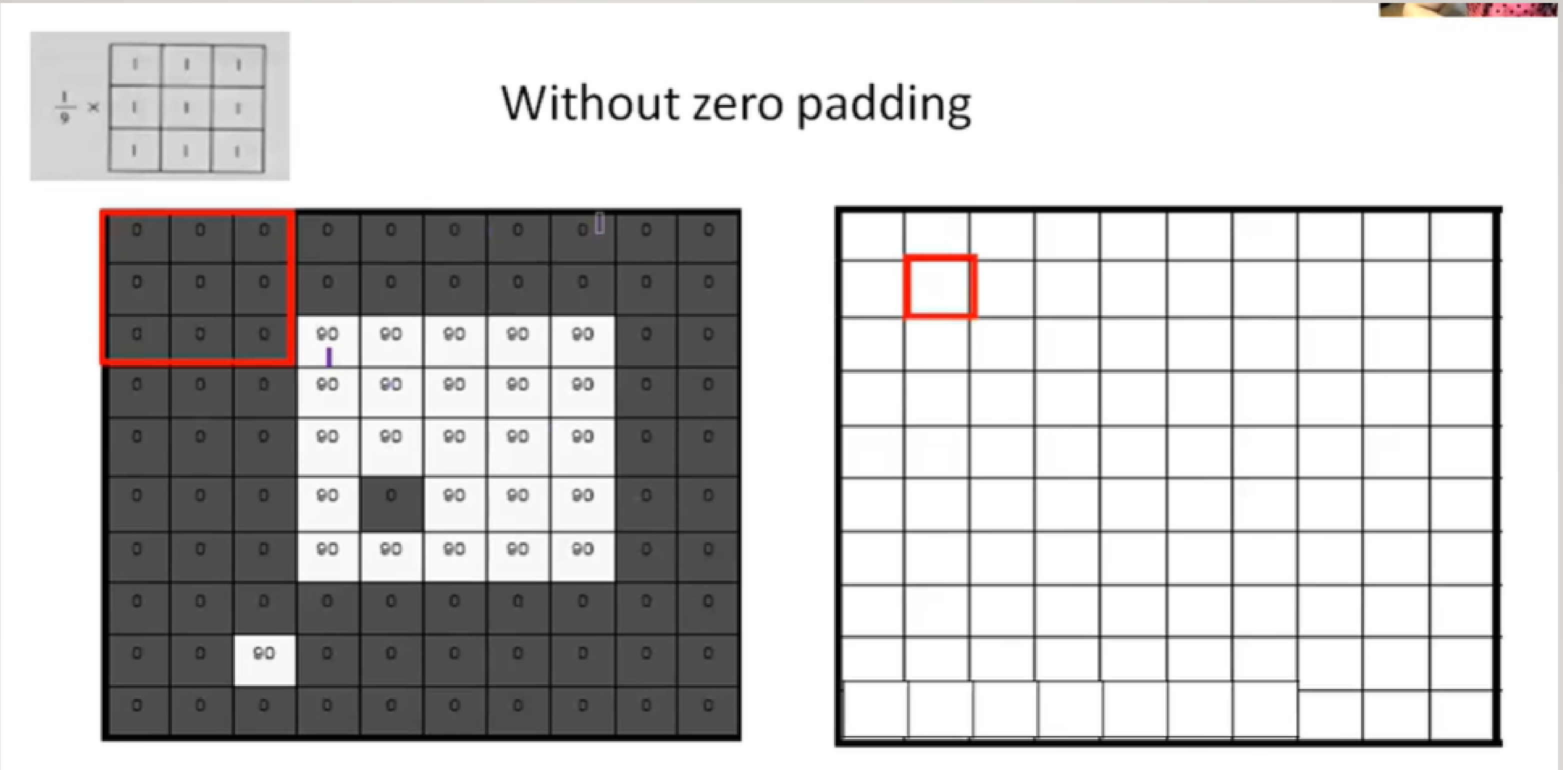
$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} \quad \text{Or} \quad 1/9 \; \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

TYPES OF LPF

Handling with boundaries:







AVERAGING(MEAN) FILTER EXAMPLE

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

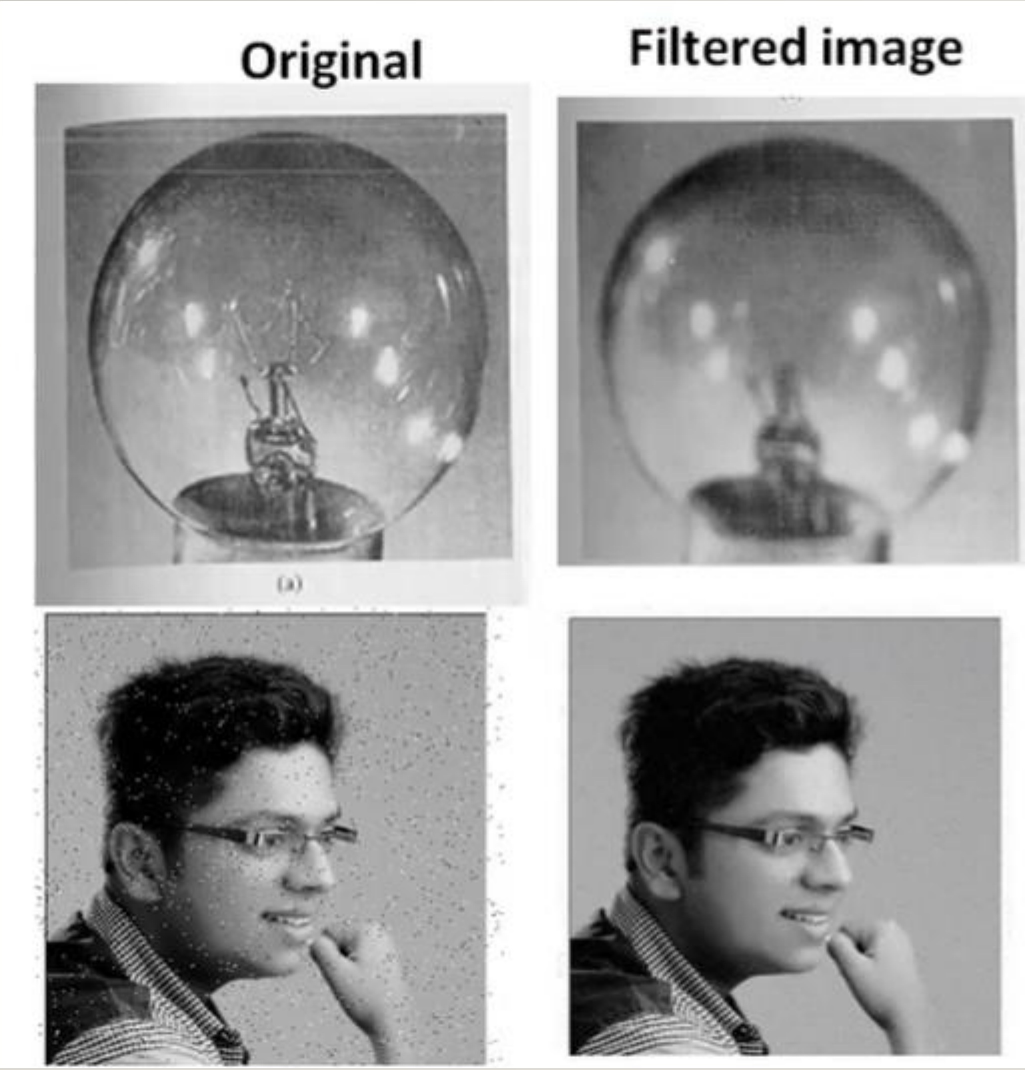
Without zero padding

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

	0	10	20						

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

	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	60	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	



Smoothing Effect of mean filter

Mean filter has effect on salt and pepper noise but not much. It just makes it blurred

MEDIAN FILTER

- Median filtering is a nonlinear operation often used in image processing to reduce salt and pepper noise.
- Assume 3X3 empty mask
- Place empty mask at left hand corner
- Arrange a pixels in ascending order
- Choose median values from these 9 values
- Place median at the centre
- Move the mask in a similar fashion from left to right and top to bottom

Example

Apply median filter on following image.

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	250	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

- **Solution.** Arrange a pixel in ascending order.

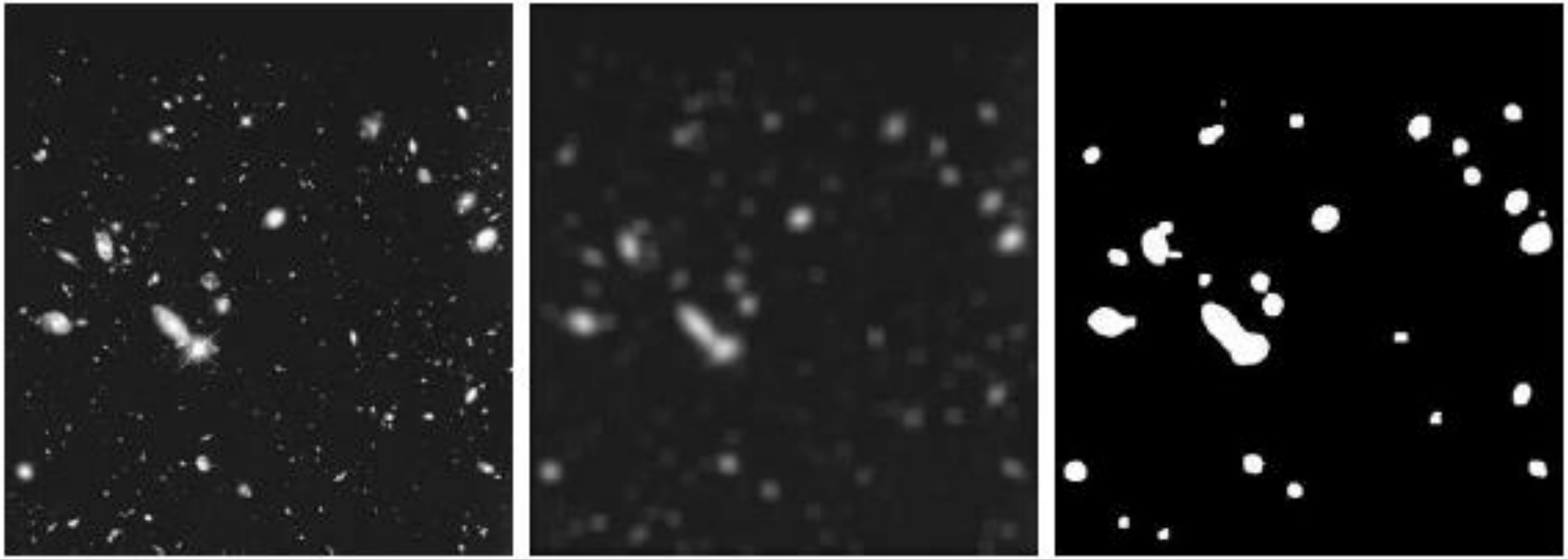
(10, 10, 10, 10, 10, 10, 10, 10, 250)

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	250	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10
10	250	10	10	10	10	10	10
10	10	10	10	10	10	10	10
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50
50	50	50	50	50	50	50	50

We can see that salt & pepper noise is removed.

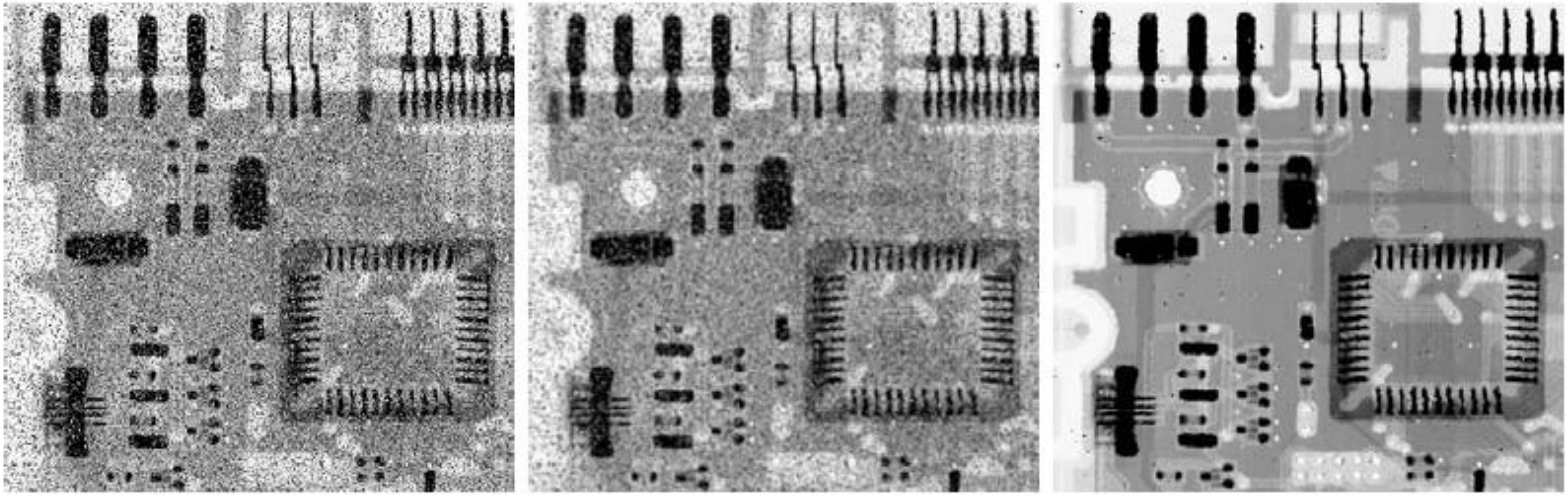


a b c

FIGURE 3.34 (a) Image of size 528×485 pixels from the Hubble Space Telescope. (b) Image filtered with a 15×15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

ORDER STATISTICS FILTER

- Order-statistic filters are nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and then replacing the value of the centre pixel with the value determined by the ranking result.
- The best-known filter in this category is the **median filter**, which, as its name implies, replaces the value of a pixel by the median of the intensity values in the neighbourhood of that pixel.
- Median filters are particularly effective in the presence of **impulse noise**, also called salt-and-pepper noise because of its appearance as white and black dots superimposed on an image.



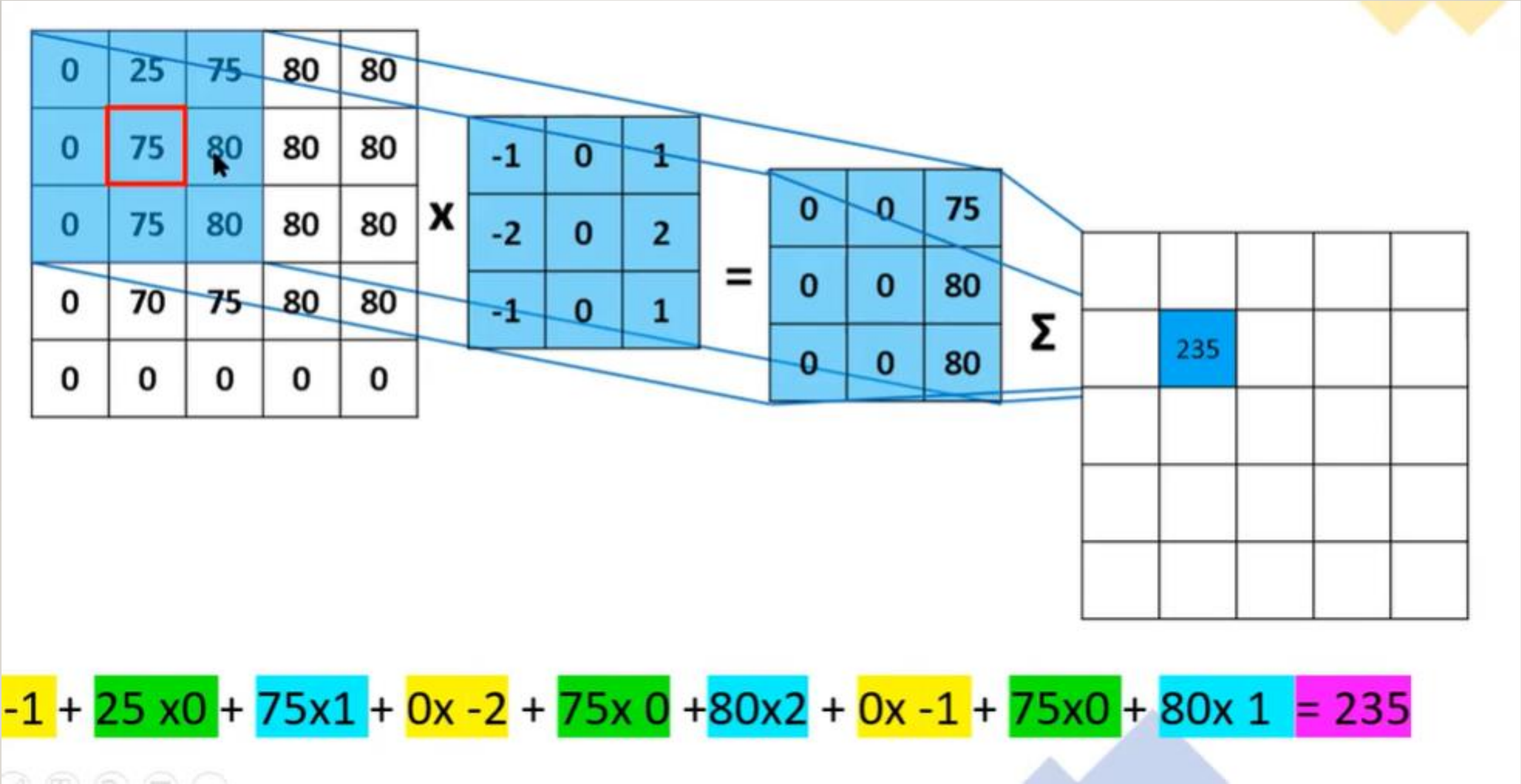
a b c

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

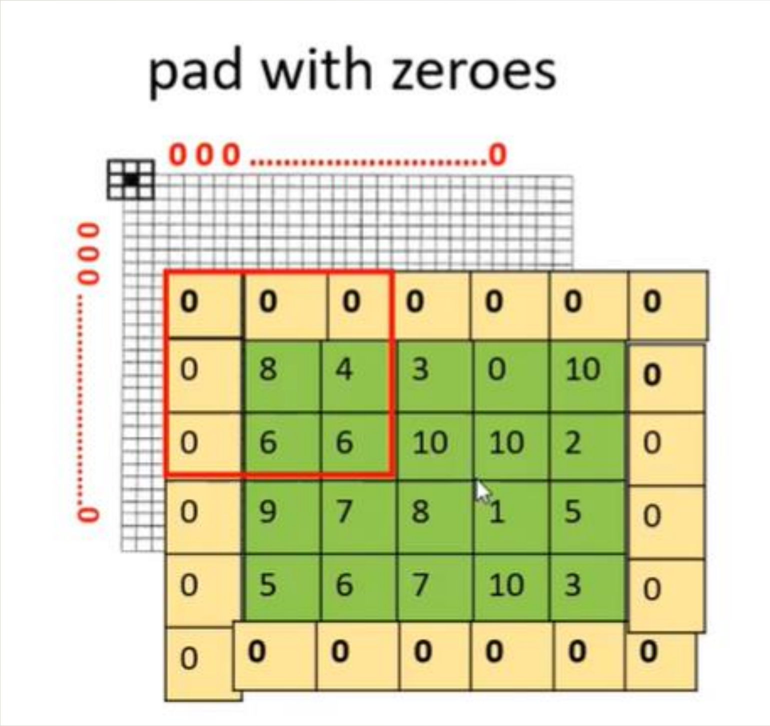
HIGH PASS FILTER

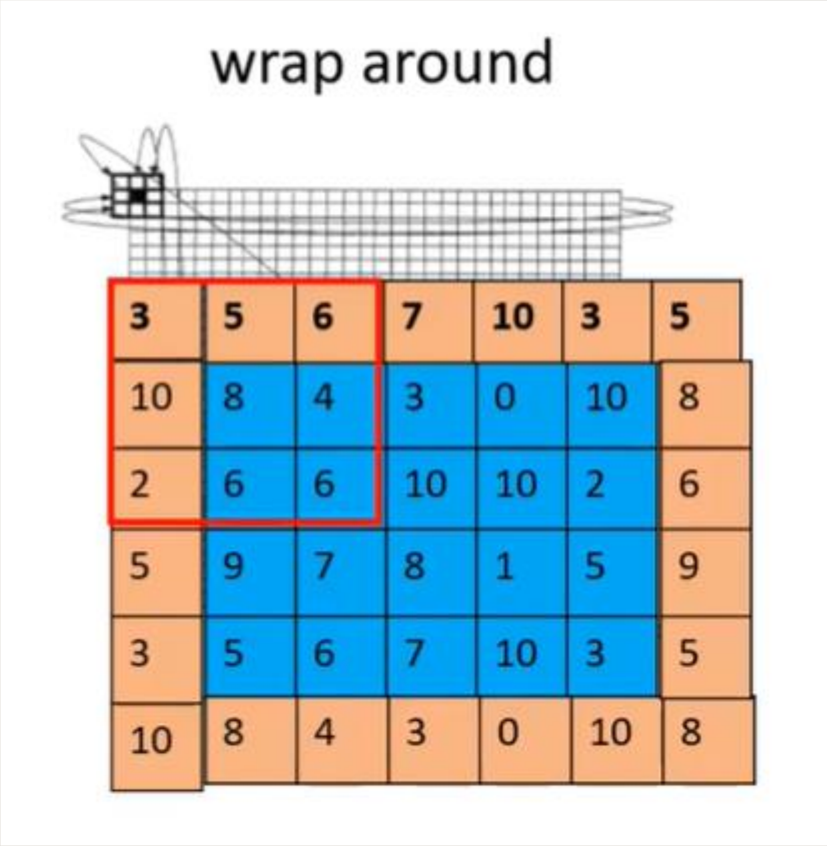
- Can be used to make an image appear sharper.
- These filters emphasize fine details in the image
- Works exactly the same way as low pass filtering: just uses a different convolution kernel

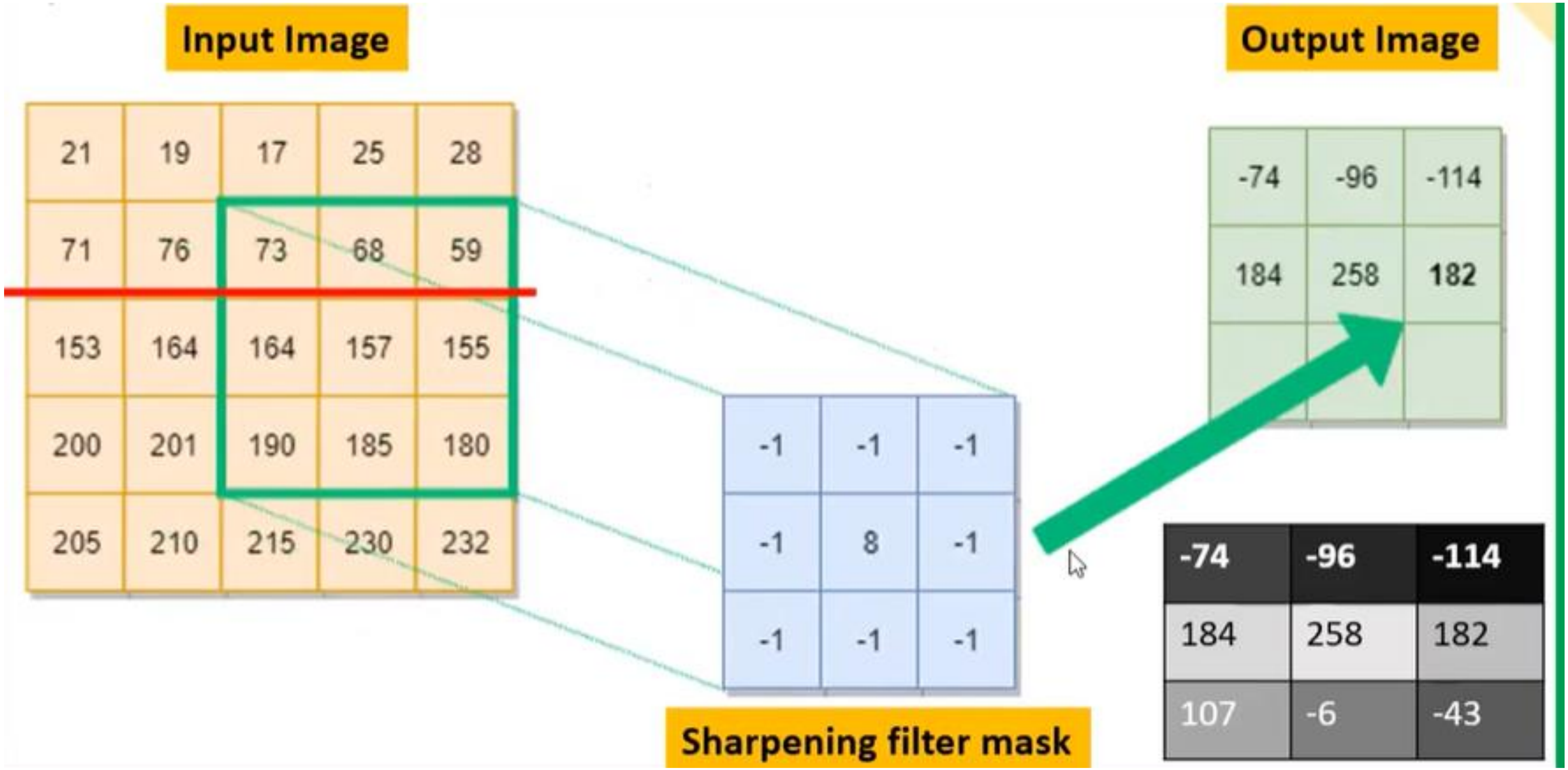
General Procedure



HANDLING PIXEL BOUNDARIES







21	19	17	25	28
71	76	73	68	59
153	164	164	157	155
200	201	190	185	180
205	210	215	230	232

-1	-1	-1
-1	8	-1
-1	-1	-1

-74	-96	-114
184	258	182

-74	-96	-114
184	258	182
107	-6	-43

COMPARISON

Low-Pass Filter Kernels:

Purpose:

- To blur an image and reduce noise by attenuating high-frequency details.

Kernel Structure:

- Coefficients are generally positive.
- The sum of all coefficients in the kernel equals 1 (or a positive value).
- The kernel emphasizes similarity between neighbouring pixels.
- **Effect**: Results in a smoother image with reduced sharp features

COMPARISON

High-Pass Filter Kernels

Purpose:

- To sharpen an image and detect edges by attenuating low-frequency details and enhancing differences between adjacent pixels.

Kernel Structure:

- Contains a positive value in the centre, surrounded by negative values.
- The sum of all coefficients in the kernel is typically zero.
- These coefficients are designed to amplify sudden changes in pixel intensity.
- **Effect**: Produces a sharper image by highlighting edges and can amplify noise

HIGH BOOST FILTER

- A **High-boost filter** is an image sharpening technique that enhances the high-frequency components such as edges and fine details, while retaining the original image's low-frequency content.
- It is often used to emphasize subtle details in images or restore blurred images.

$$Hb = A \cdot I - G$$

I: Original Image

G: Smoothed version of the image which is usually obtained using a low-pass filter

A: Amplification factor i.e., boosting constant when A=1, it reduces to a high-pass filter.

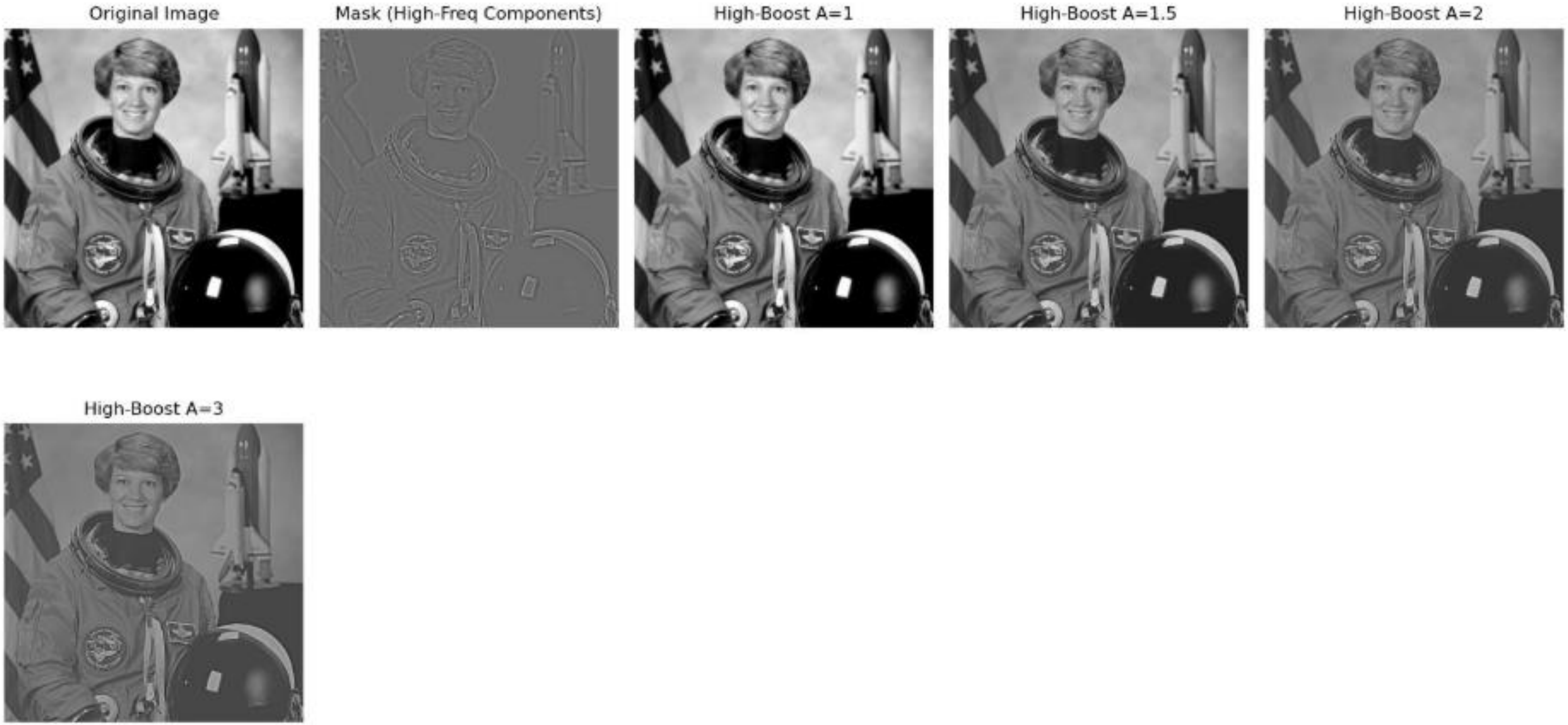
HIGH BOOST FILTER

- The High - Boost Filter exhibits some properties which are mentioned as follows −
- When **A > 1** then the filter enhances the high-frequency components while retaining the low-frequency components of the original image.
- When **A = 1** then the filter reduces to a standard high-pass filter by emphasizing only edges and details.
- High-boost filters provide a flexible sharpening mechanism where we can control the degree of sharpening through A.

STEPS TO APPLY HIGH BOOST FILTERING

- To apply the High - Boost Filter to an image we have to follow certain steps. Here are the steps to be followed –
- **Smooth the image:** Firstly we have to smooth the image by using a low pass filter to extract low-frequency components.
- **Subtract the smoothed image:** Next we have to subtract the smoothed image from the original image to extract the high-frequency components.
- **Add back the original image:** Finally we have to add back the original image multiplied by the amplification factor to retain and amplify the low-frequency content.


Original Image
Smoothed Image (Low-Pass Filter)
High-Boost Filtered Image (A = 1.5)


Original Image
Mask (High-Freq Components)
High-Boost A=1
High-Boost A=1.5
High-Boost A=2
High-Boost A=3


Original Image
High-Pass Filter
High-Boost Filter (A=2)

FREQUENCY DOMAIN FILTERING

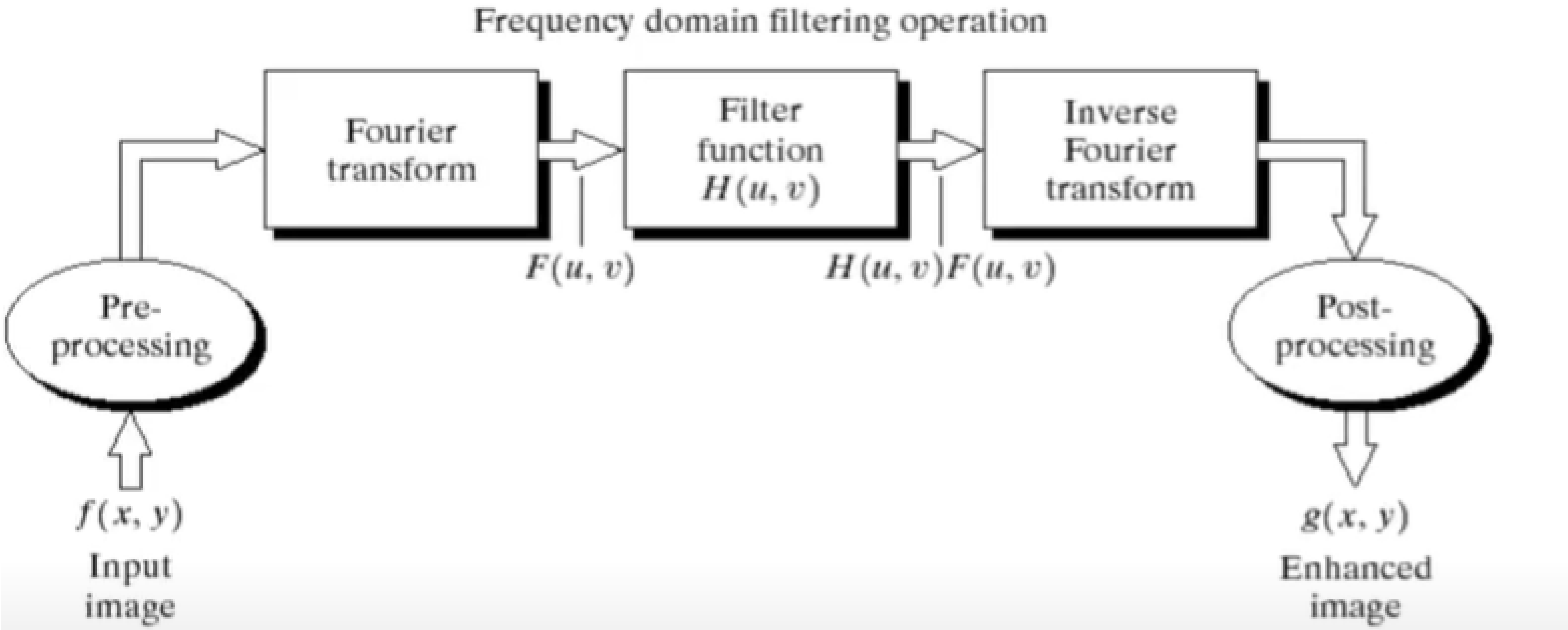
- Frequency domain is the space defined by the values of the Fourier transform and its frequency variables (u,v)
- Filtering in Frequency domain is straightforward
- Consists of the following steps:

➢ Multiply the input image by $(-1)^{x+y}$ to centre the transform

➢ Compute F(u,v), the DFT of the image

➢ Multiply F(u,v) by a filter function H(u,v)

➢ Compute Inverse DFT of the above result


Frequency domain filtering operation
Fourier transform
Filter function $H(u, v)$
Inverse Fourier transform
$F(u, v)$
$H(u, v)F(u, v)$
Pre-processing
Post-processing
$f(x, y)$
Input image
$g(x, y)$
Enhanced image

FREQUENCY DOMAIN FILTERING

- H(u,v) is called a filter(or filter transfer function) because it supresses certain frequencies in the transform while leaving others unchanged.
- The basic modelling for filtering is

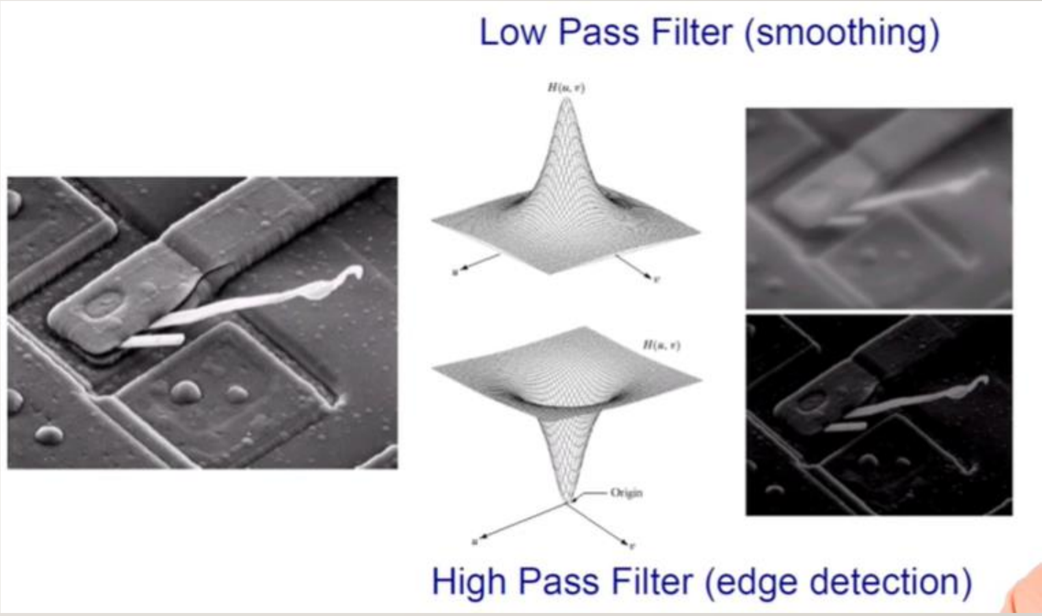
$$G(u,v) = H(u,v)F(u,v)$$

Where F(u,v) is the fourier transform of the image being filtered and H(u,v) is the filter transform function.

Smoothing is achieved in frequency domain by dropping out the high frequency components.

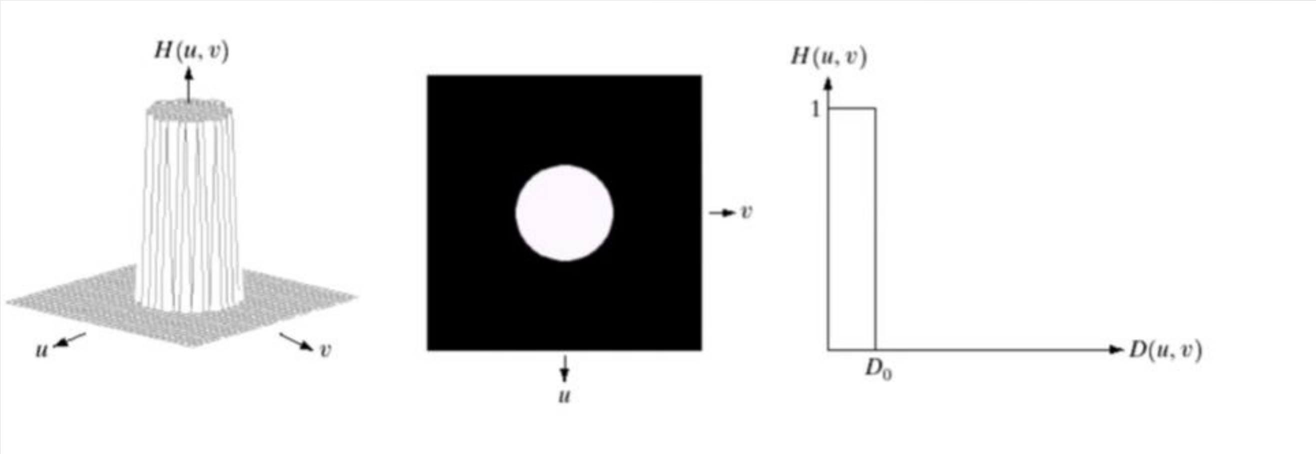
FREQUENCY DOMAIN FILTERING

- Low pass(LP) filters-only pass low frequencies drop high frequencies
- High pass(HP) filters-only pass frequencies above a minimum value



IDEAL LOW PASS FILTER

- Simply cut off all high frequency components that are a specified distance D0 from the origin of the transform



IDEAL LOW PASS FILTER

- Transfer function of ideal low pass filter is given by

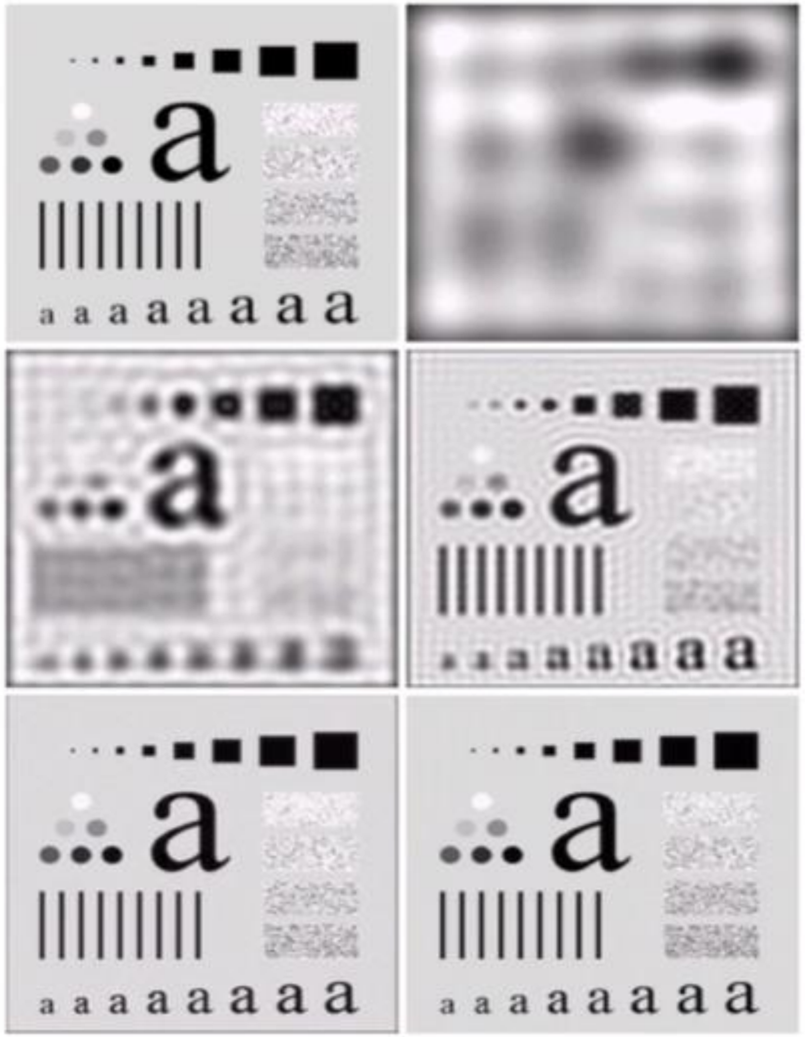
$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \le D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$$

Where D(u,v) is given as

$$D(u,v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}$$

Original image

Result of filtering with ideal low pass filter of radius 5



Result of filtering with ideal low pass filter of radius 15

Result of filtering with ideal low pass filter of radius 30

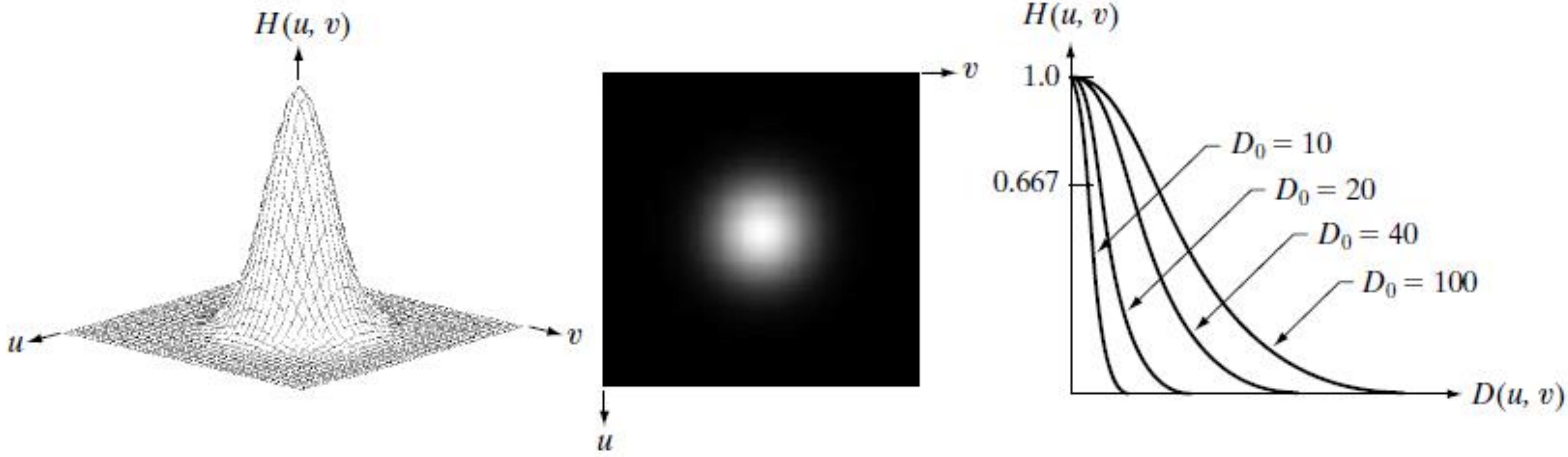
Result of filtering with ideal low pass filter of radius 80

Result of filtering with ideal low pass filter of radius 230

GAUSSIAN LOW PASS FILTERS

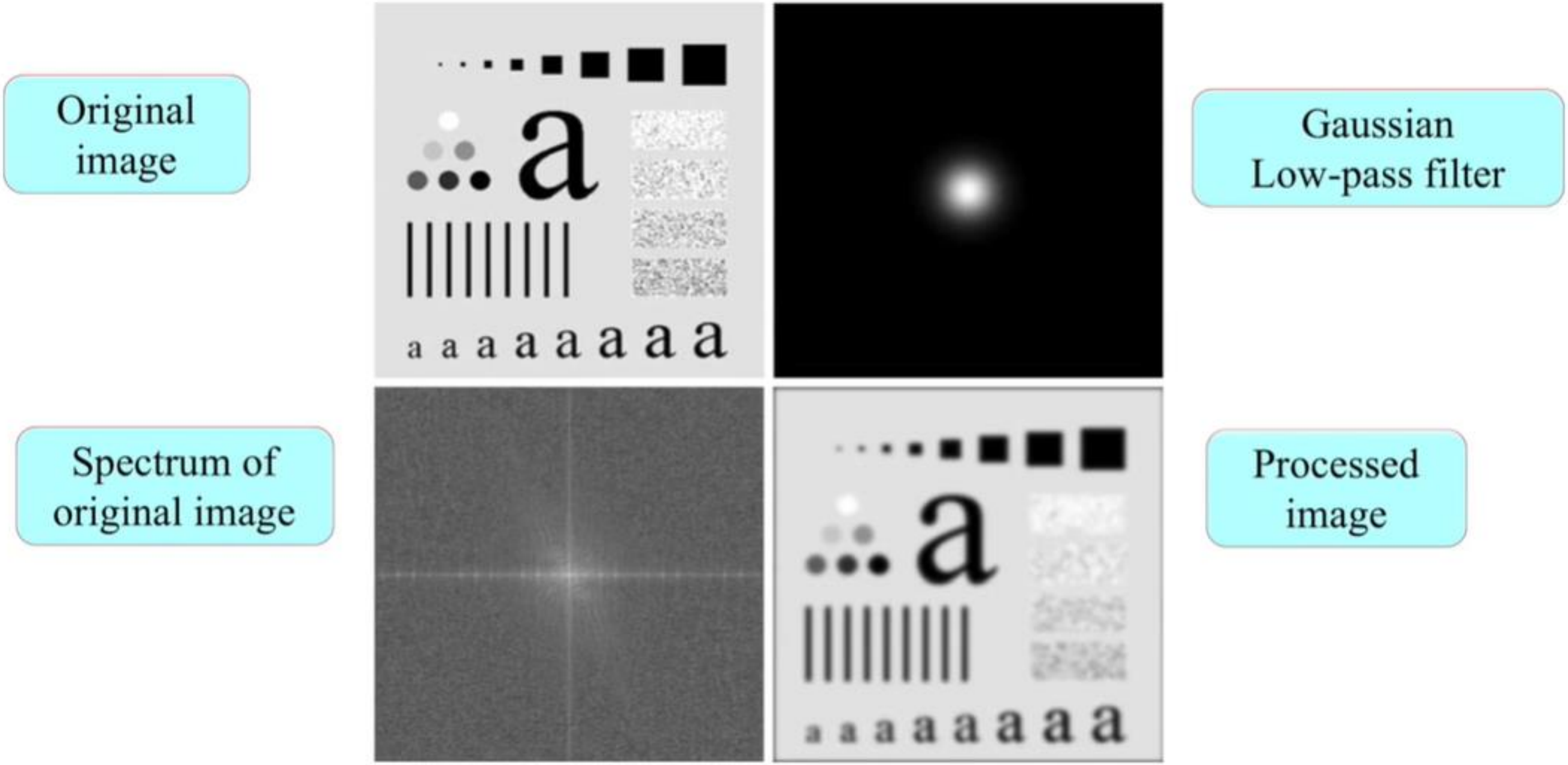
- The transfer function of a gaussian low pass filter

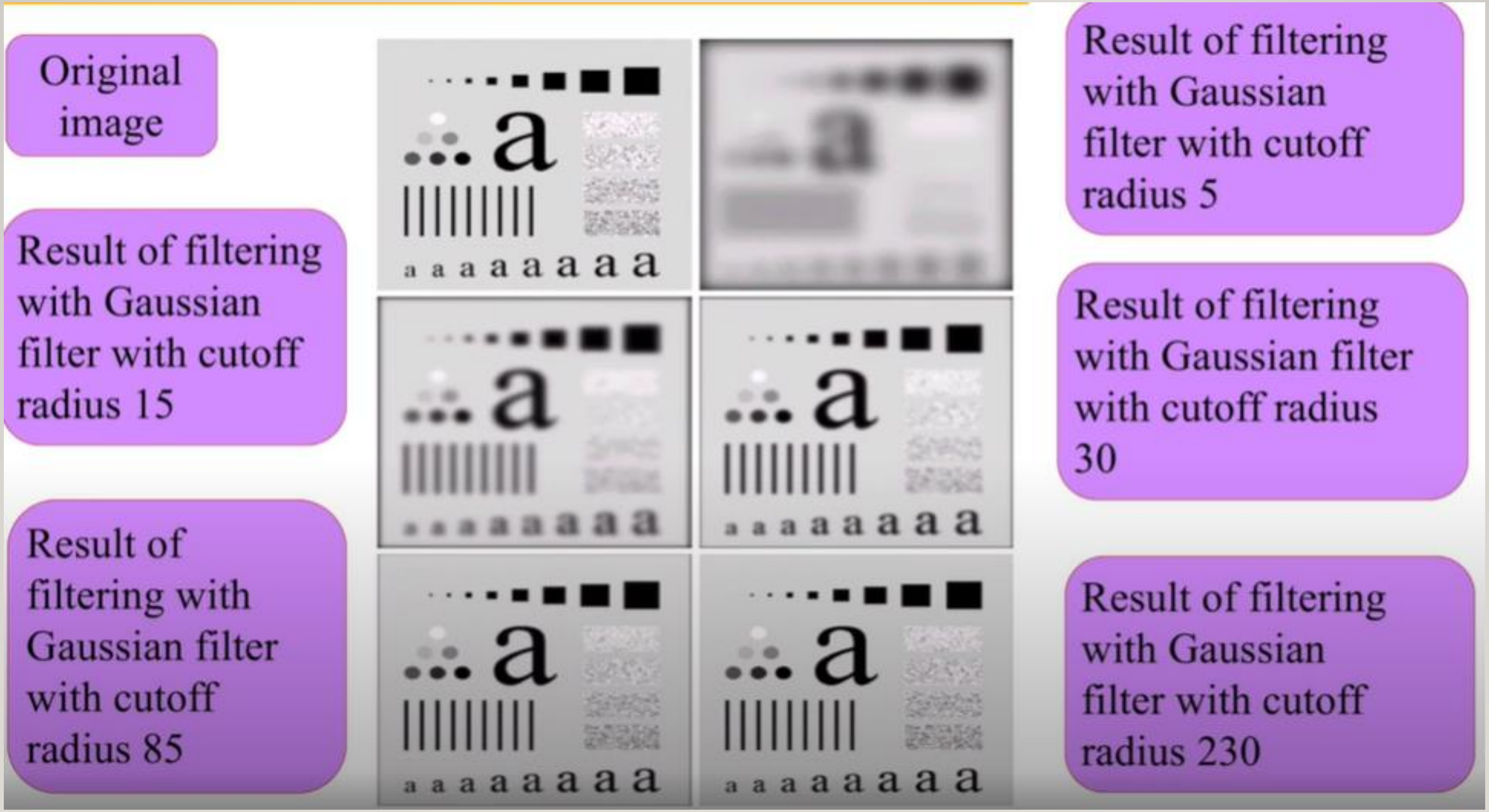
$$H(u,v) = e^{-D^2(u,v)/2D_0^2}$$



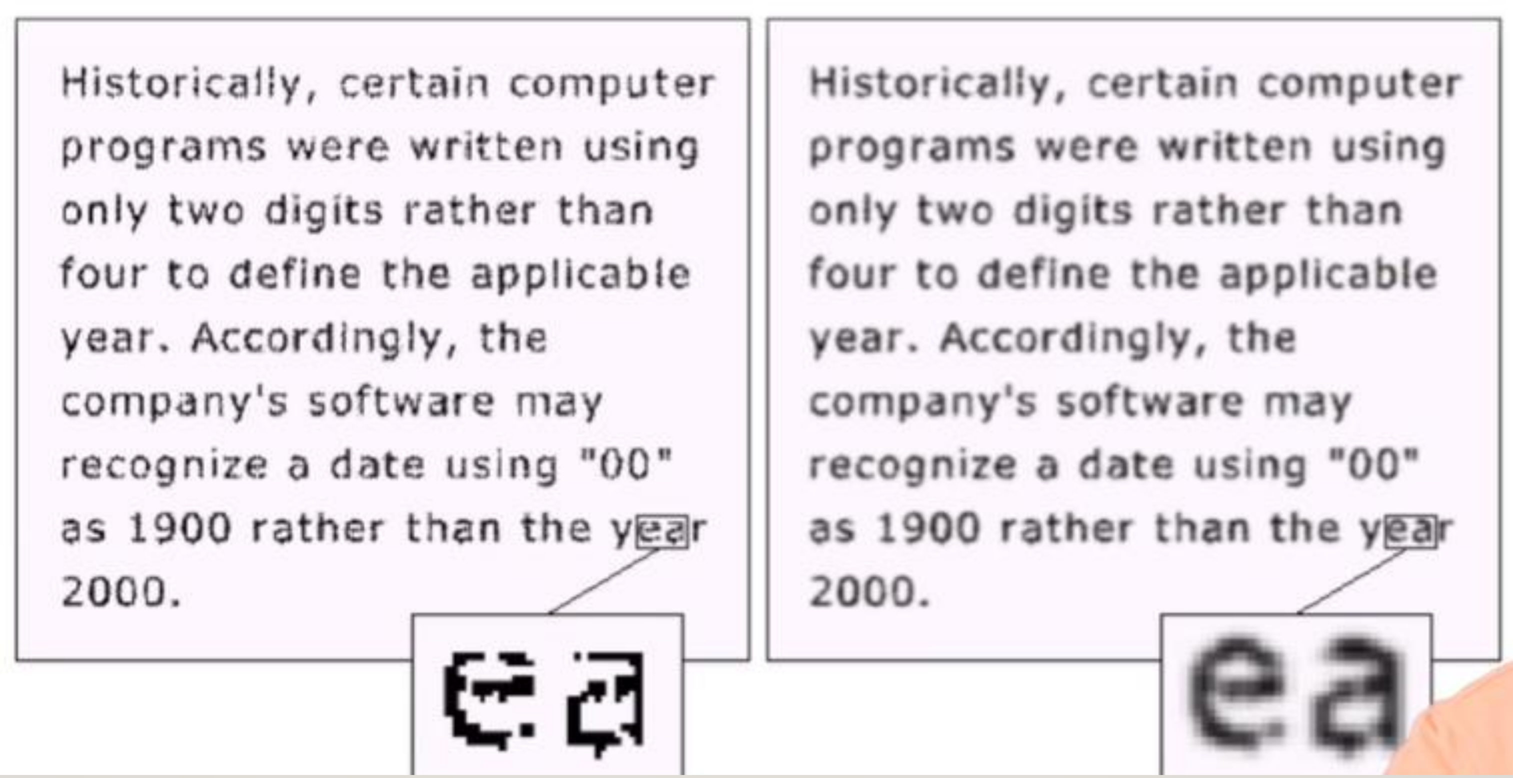
a b c

FIGURE 4.47 (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.


Original image
Gaussian Low-pass filter
Spectrum of original image
Processed image


Original image
Result of filtering with Gaussian filter with cutoff radius 5
Result of filtering with Gaussian filter with cutoff radius 15
Result of filtering with Gaussian filter with cutoff radius 30
Result of filtering with Gaussian filter with cutoff radius 85
Result of filtering with Gaussian filter with cutoff radius 230

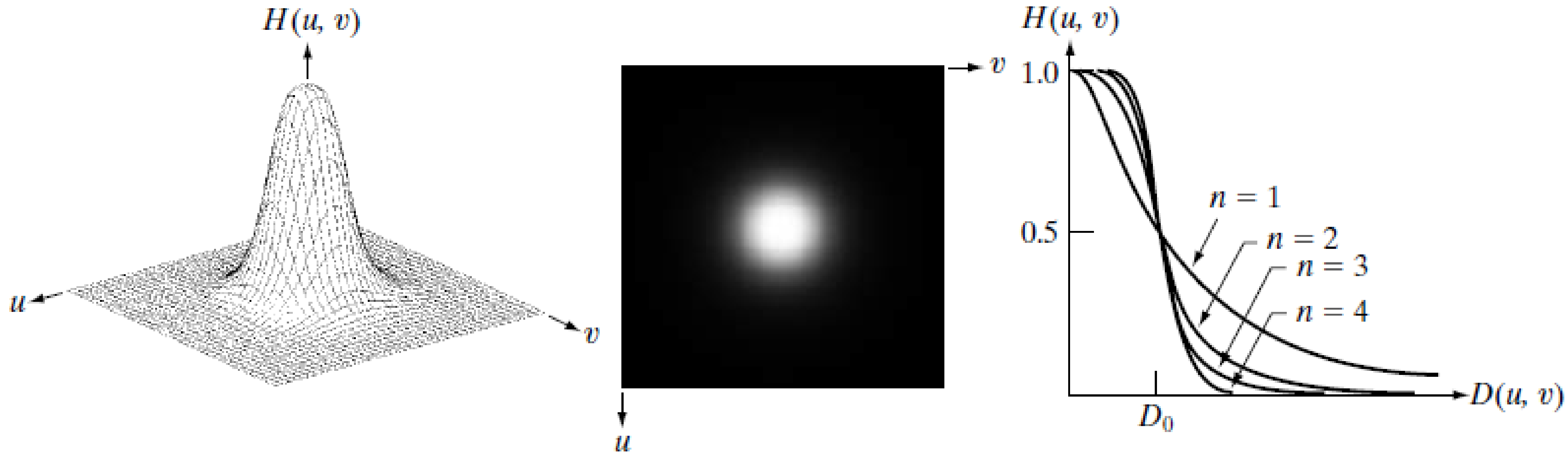
Applications



BUTTERWORTH LOW PASS FILTER

- The transfer function of a Butterworth lowpass filter (BLPF) of order *n*, and
- with cutoff frequency at a distance D_0 from the origin, is defined as

$$H(u, v) = \frac{1}{1 + [D(u, v)/D_0]^{2n}}$$



a b c

FIGURE 4.44 (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.

Original image

Result of filtering with Butterworth filter of order 2 and cutoff radius 5

Result of filtering with Butterworth filter of order 2 and cutoff radius 15

Result of filtering with Butterworth filter of order 2 and cutoff radius 30

Result of filtering with Butterworth filter of order 2 and cutoff radius 80

Result of filtering with Butterworth filter of order 2 and cutoff radius 230

COMPARISON OF LOW PASS FILTERS

Result of filtering with ideal low pass filter of radius 15

Result of filtering with Butterworth filter of order 2 and cutoff radius 15

Result of filtering with Gaussian filter with cutoff radius 15

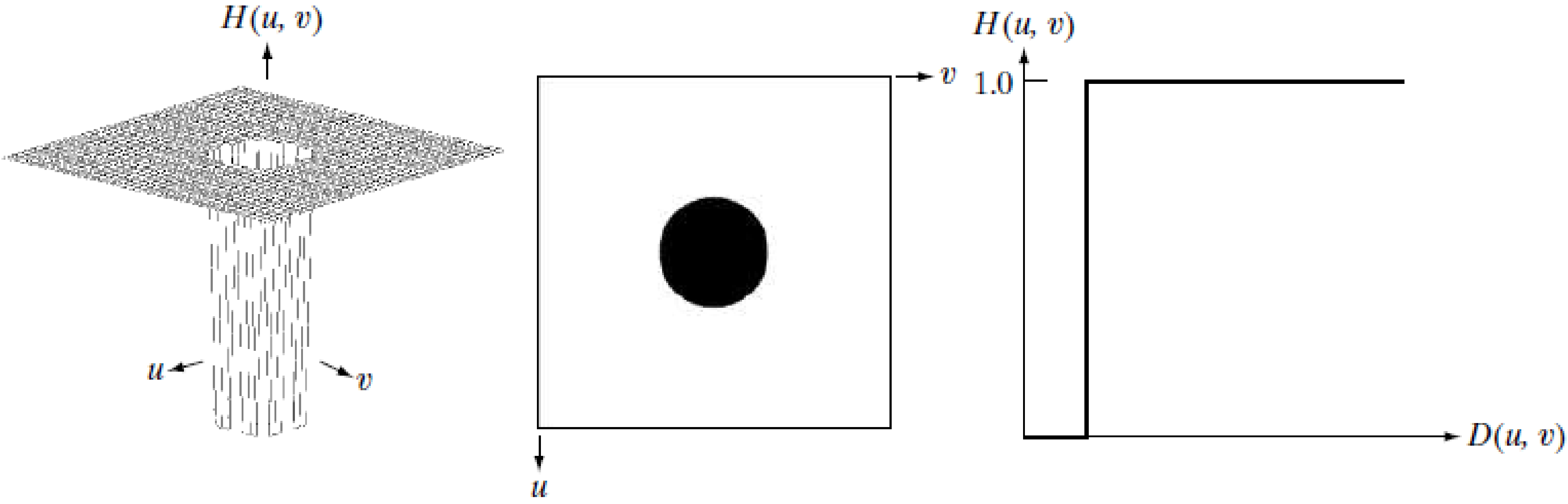
IDEAL HIGH PASS FILTERS

- Edges and fine details in images are associated with high frequency components
- High pass filters-pass only high frequencies and drop low ones
- High pass frequencies are reverse of low pass filters

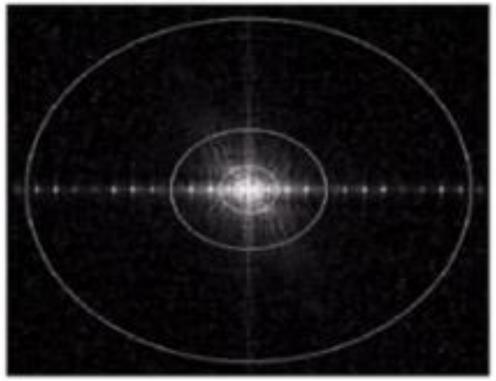
$$H_{hp}(u, v) = 1 - H_{lp}(u, v)$$

- The ideal high pass filter is given by

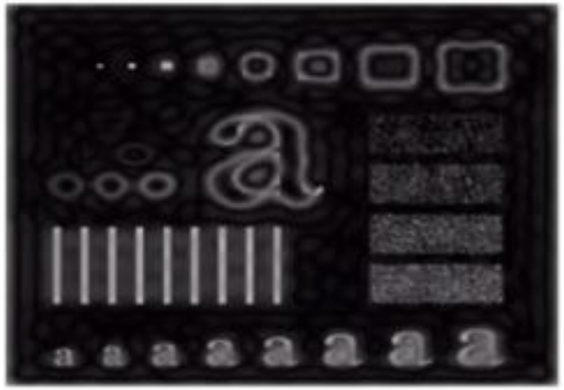
$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$


H(u, v)
u
v
v
u
H(u, v)
1.0
D(u, v)

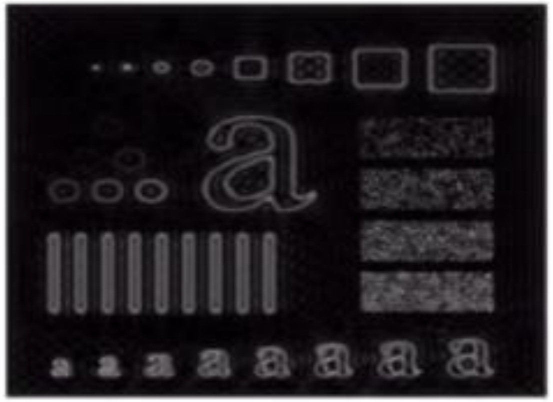
Original Image of 500 x 500



Fourier Spectrum



Result of ideal high-pass filtering with $D0 = 15$



Result of ideal high-pass filtering with $D0 = 30$

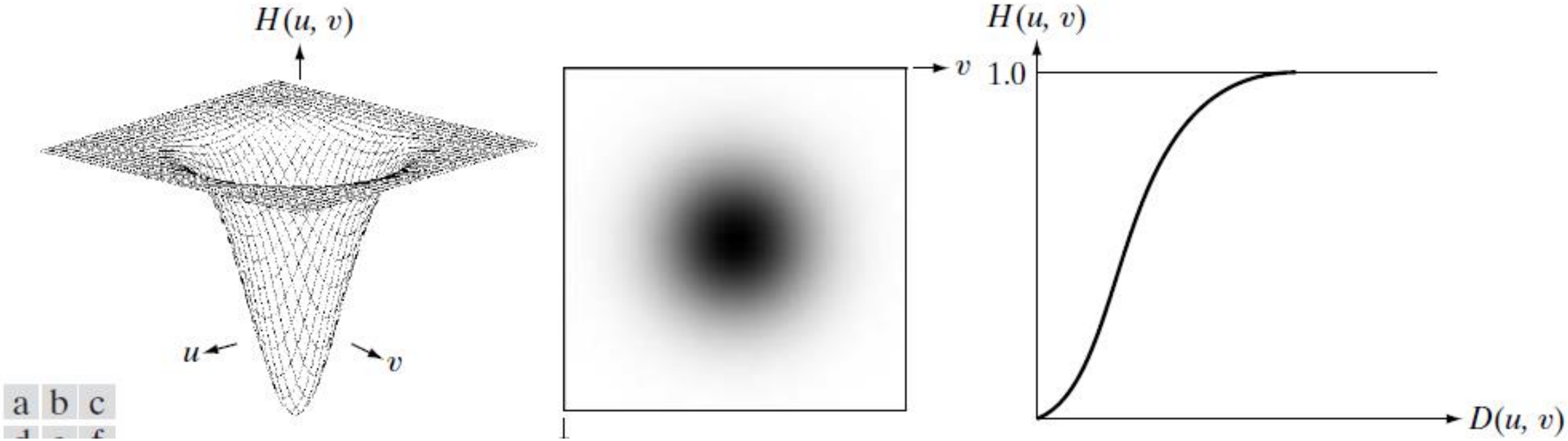
Result of ideal high pass filtering with $D0 = 80$

GAUSSIAN HIGH PASS FILTERS

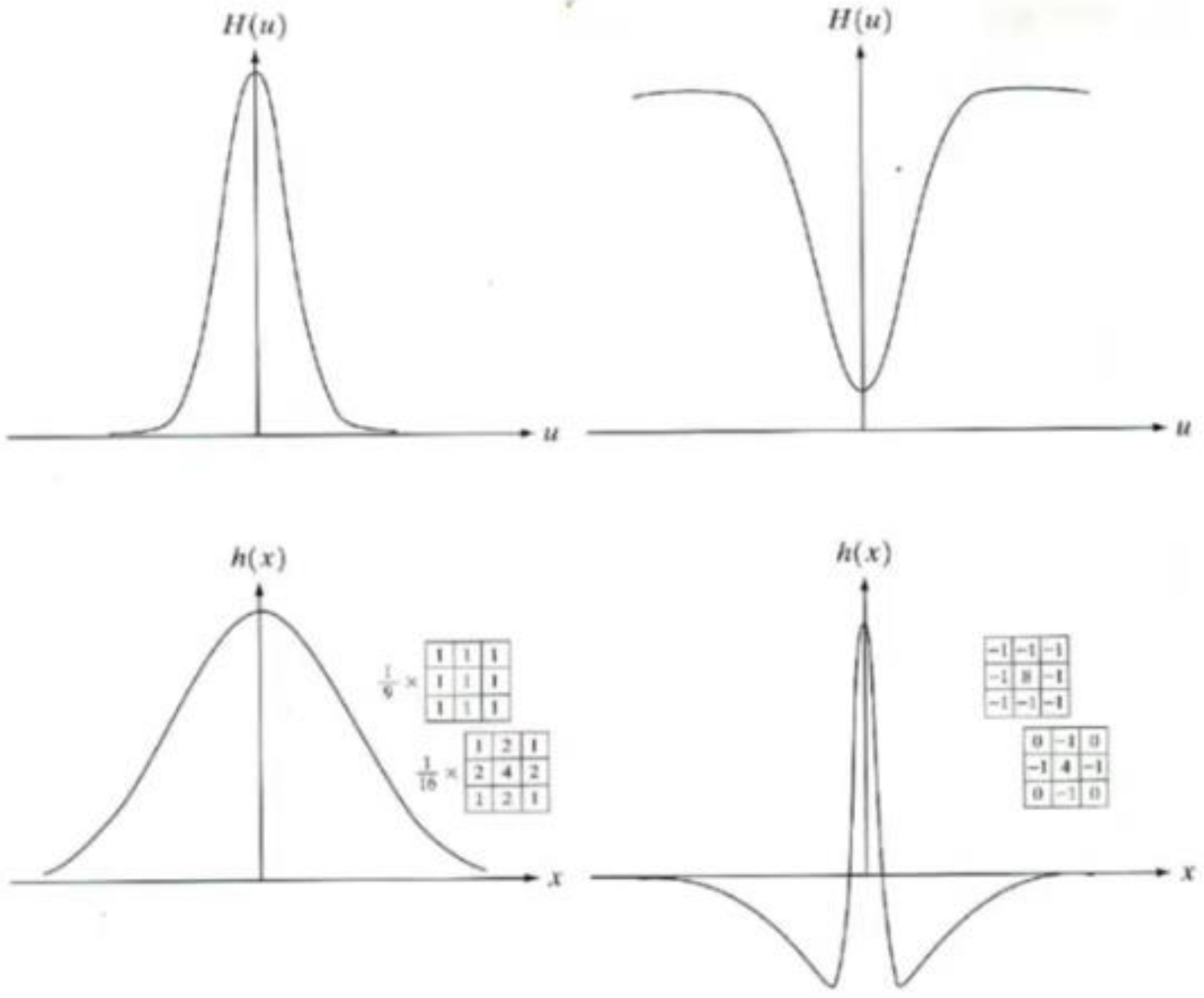
- Gaussian high pass filter is given as:

$$H(u, v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

- D_0 is the cutoff distance as before

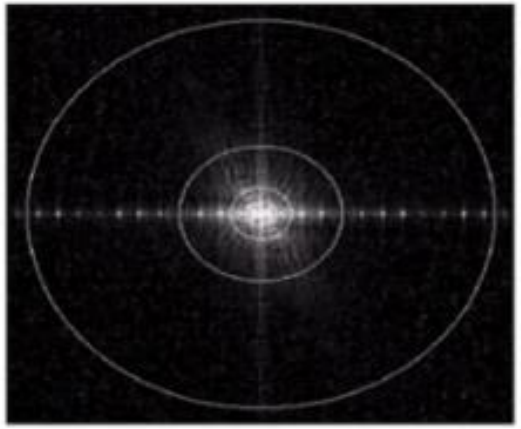


H(u, v)
u
v
a b c
v
H(u, v)
1.0
D(u, v)

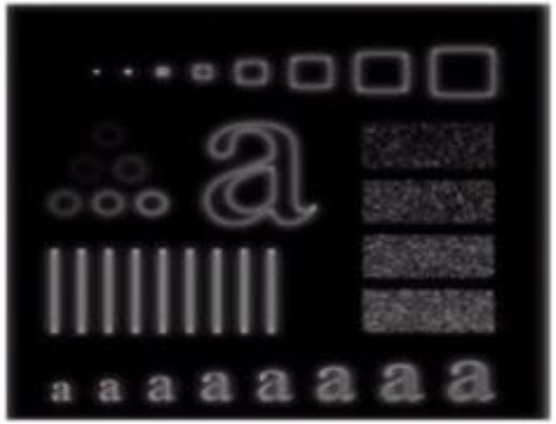


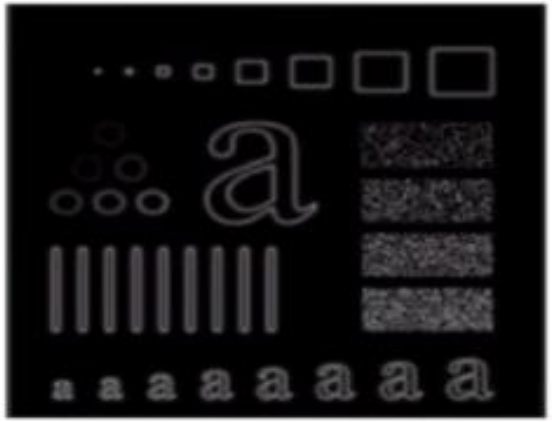
(a) Gaussian Frequency Domain Low-pass Filter
(b) Gaussian Frequency Domain High-pass Filter
(c) Corresponding Low-pass spatial filter
(d) Corresponding High-pass spatial filter

Original Image of
500 x 500



Fourier S





Gaussian High-pass
$D0 = 15$

Gaussian High-pass
$D0 = 30$

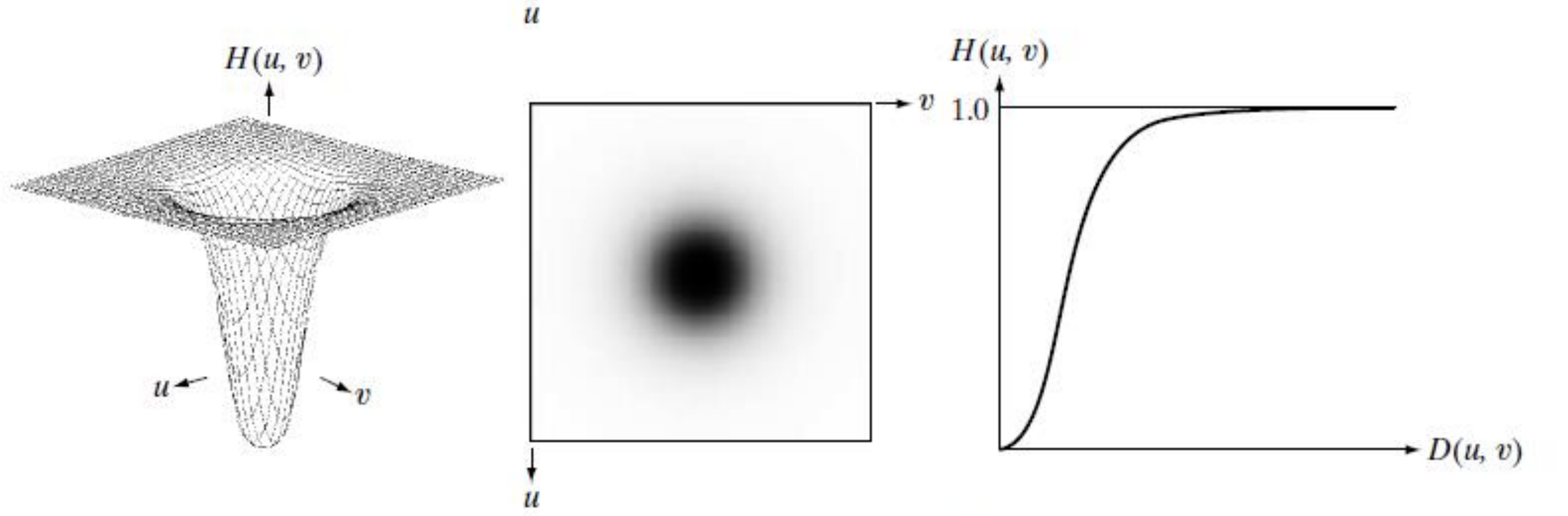
Gaussian High-pass
$D0 = 80$

BUTTERWORTH HIGH PASS FILTER

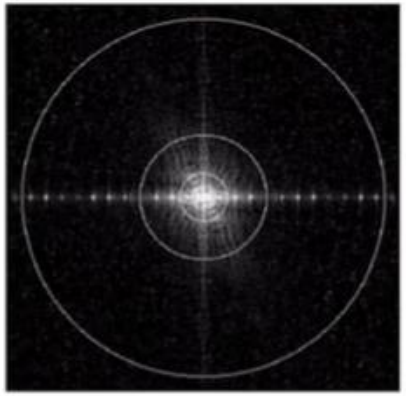
- Butterworth high pass filter is given by:

$$H(u, v) = \frac{1}{1 + [D_0/D(u, v)]^{2n}}$$

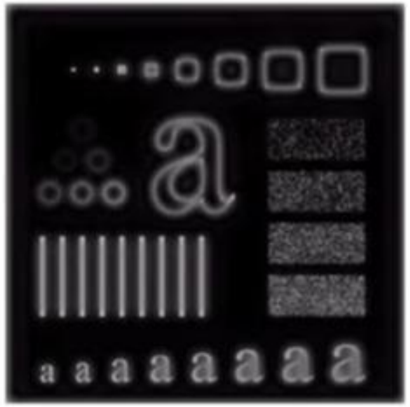
Where n is the order and D_0 is the cut off distance as before.


H(u, v)
u
v
u
v
u
H(u, v)
1.0
D(u, v)

Original Image
of 500 x 500



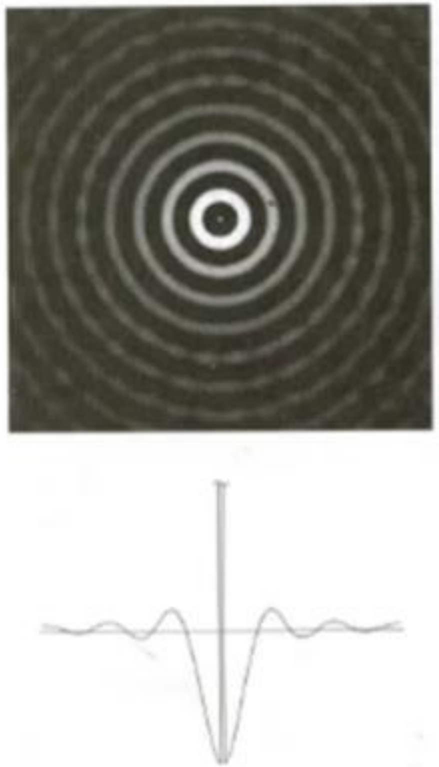
Fourier S



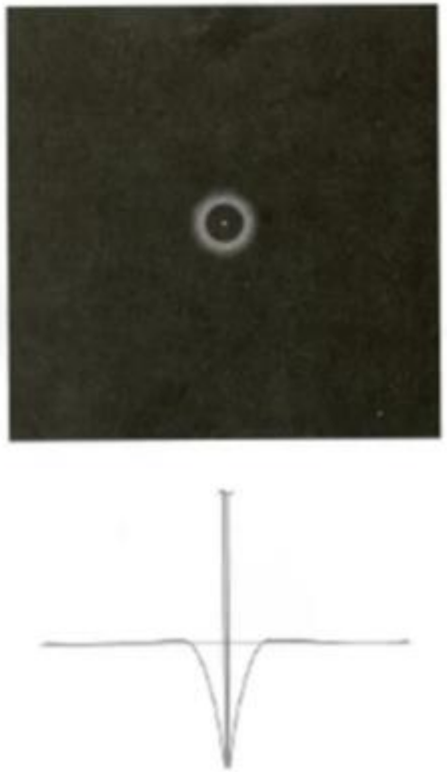
Butterworth
of order 2
$D0 = 15$

Butterworth of
order 2
$D0 = 30$

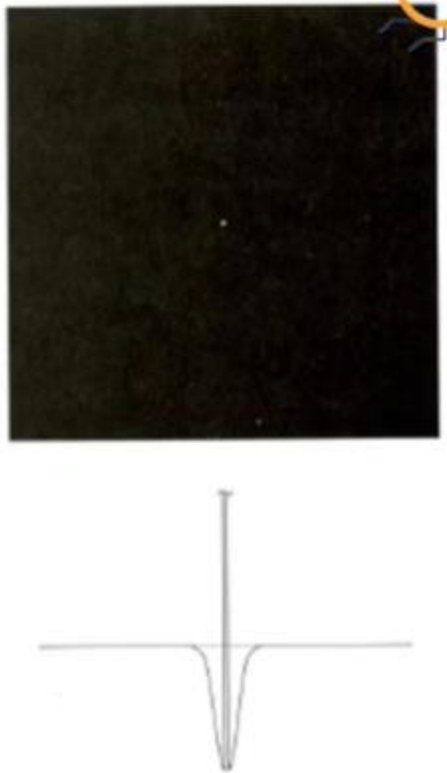
Butterworth
of order 2
$D0 = 80$



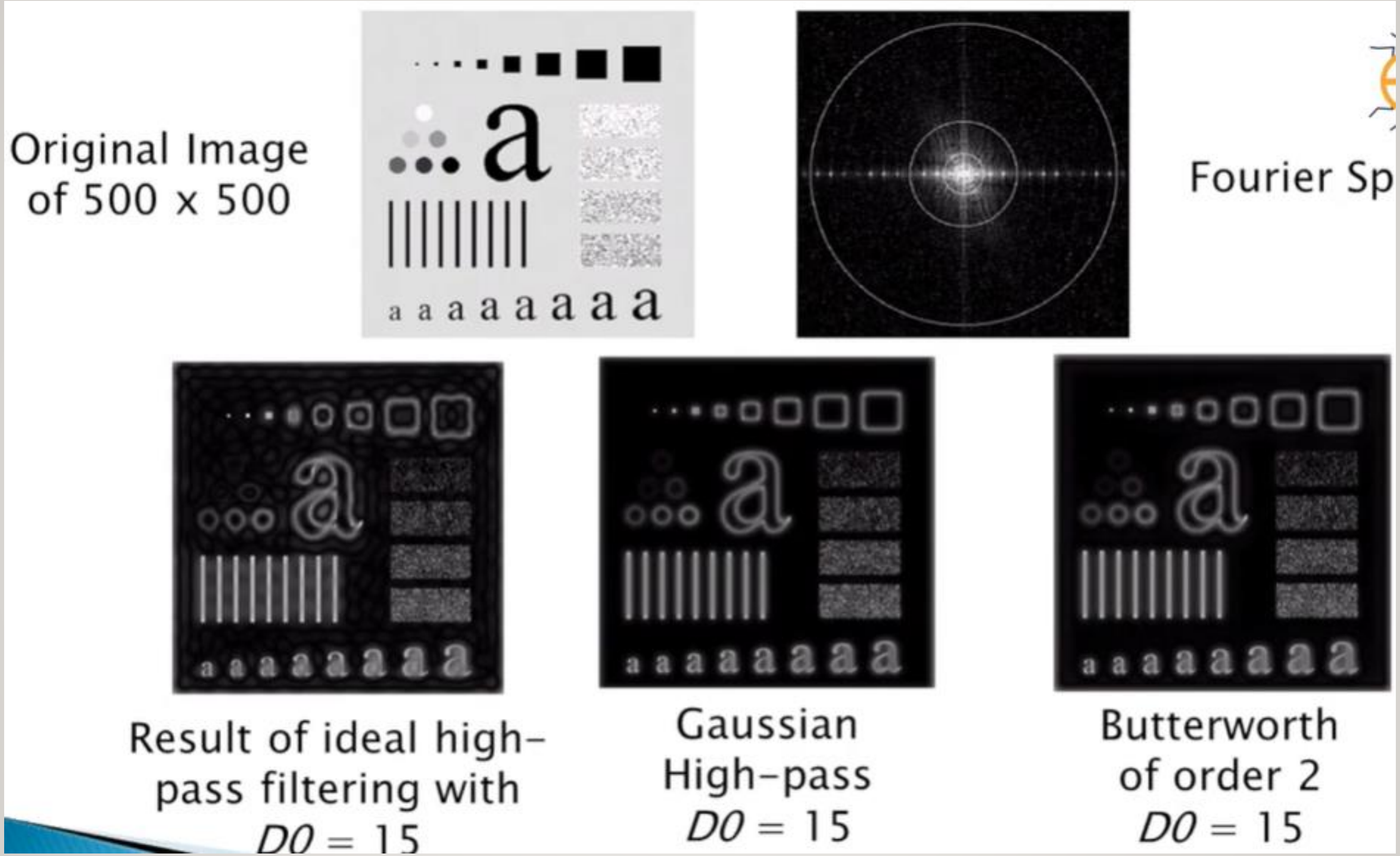
Spatial Representation of typical Ideal frequency domain high-pass filter with gray-level profile.



Spatial Representation of typical Butterworth frequency domain high-pass filter with gray-level profile.



Spatial Representation of typical Gaussian frequency domain high-pass filter with gray-level profile.


Original Image
of 500 x 500
Fourier Sp
Result of ideal high-
pass filtering with
D0 = 15
Gaussian
High-pass
D0 = 15
Butterworth
of order 2
D0 = 15