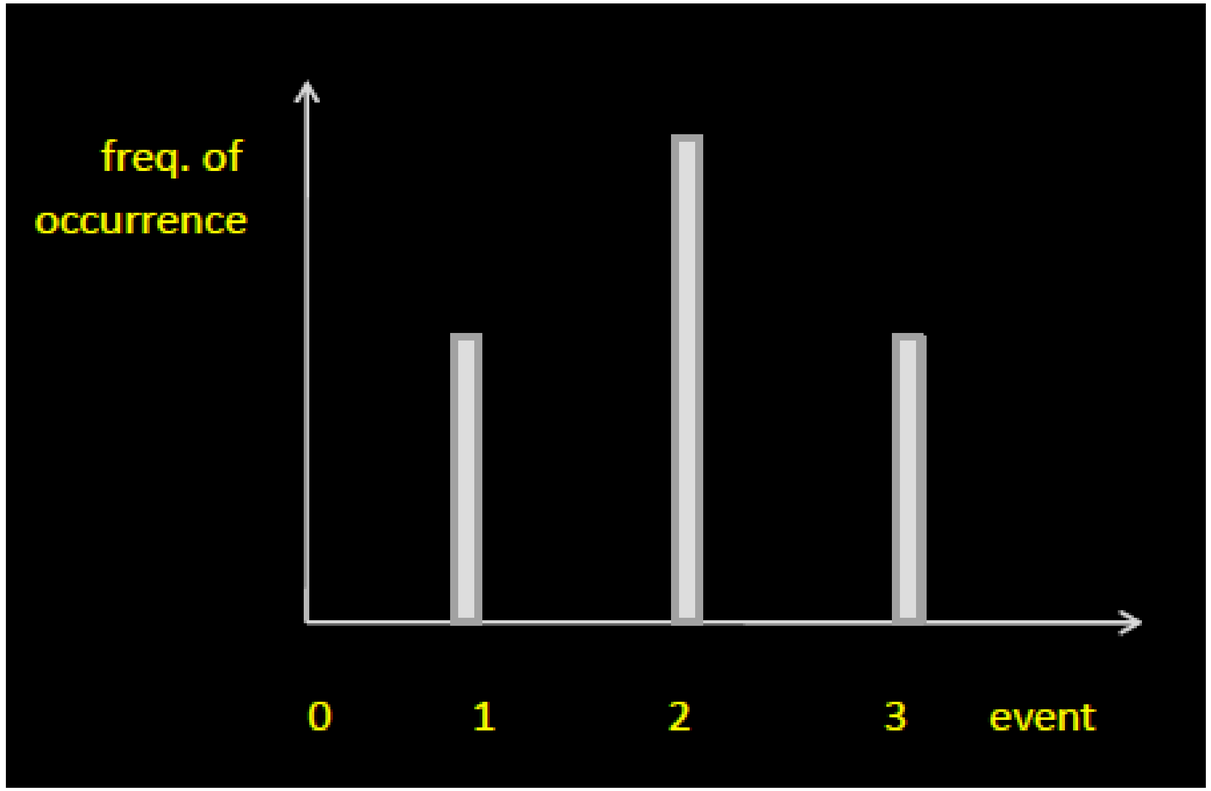
CST 304
Computer Graphics & Image Processing
Module 5

Module - 5 (Image Enhancement in Spatial Domain and Image Segmentation)

- Basic gray level transformation functions - Log transformations, Power-Law transformations
- Contrast stretching. Histogram equalization. Basics of spatial filtering - Smoothing spatial filter-Linear and nonlinear filters, and Sharpening spatial filters-Gradient and Laplacian.
- Fundamentals of Image Segmentation. Thresholding - Basics of Intensity thresholding and Global Thresholding. Region based Approach - Region Growing, Region Splitting and Merging.
- Edge Detection - Edge Operators- Sobel and Prewitt.

Gray level histogram

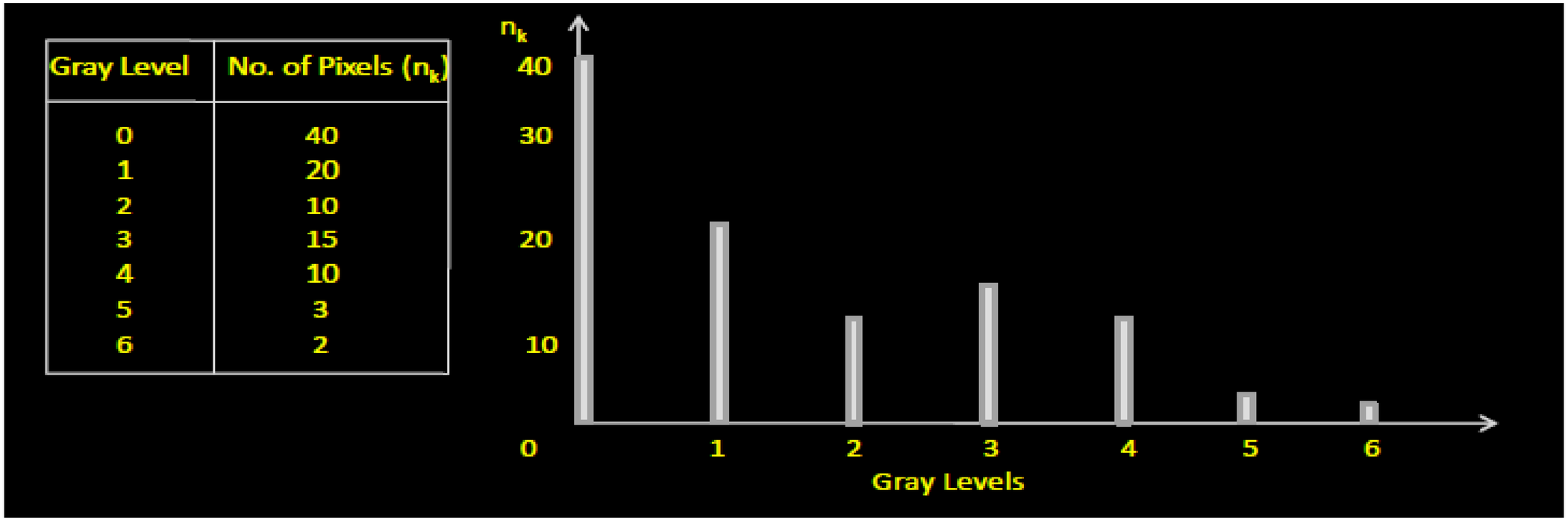
- In Statistics, Histogram is a graphical representation showing a visual impression of the distribution of data.
- An Image Histogram is a type of histogram that acts as a graphical representation of the lightness/color distribution in a digital image.
- It plots the number of pixels for each intensity value.
- Histogram of images provide a global description of their appearance.
- Histogram of an image represents relative frequency of occurrence of various gray levels.



Gray level histogram

- Histogram can be plotted in two ways:
- **First Method:**
  - X-axis has Gray levels & Y-axis has No. of pixels in each gray levels.
  - The histogram of a digital image with gray levels in the range [0, L-1] is a discrete function $\mathbf{h(r_k) = n_k}$, where r_k is the kth gray level and n_k is the number of pixels in the image having gray level r_k.

Gray Level	No. of Pixels (n_k)
0	40
1	20
2	10
3	15
4	10
5	3
6	2



Gray level histogram

- **Second Method:**
  - common practice to normalize a histogram
  - X-axis has gray levels & Y-axis has probability of occurrence of gray levels.

$$P(\mu_k) = n_k / n$$

μ_k – gray level,

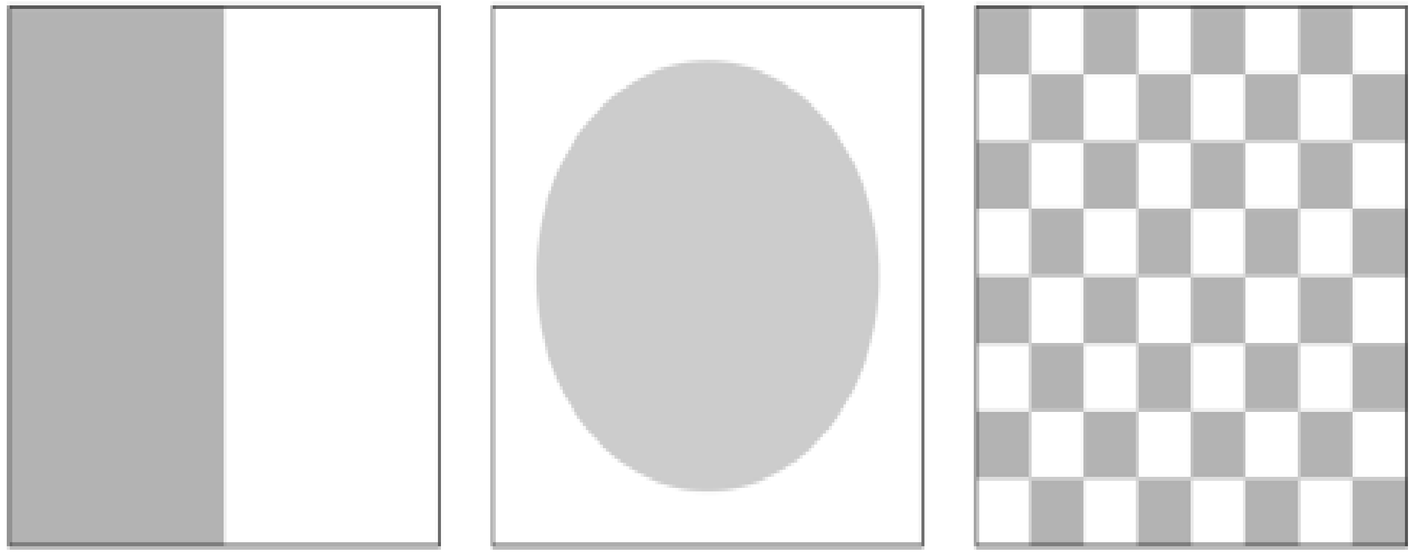
n_k–no. of pixels in kth gray level,

n –total number of pixels in an image

  - sum of all components of a normalized histogram is equal to 1.
  - Advantage of 2nd method: Maximum value plotted will always be 1.
  - White–1, Black–0.

Gray level histogram

- Different images can have same histogram
  - 3 images below have same histogram
  - Half of pixels are gray, half are white, so Same histogram = same statisics
  - Can we reconstruct image from histogram? No !



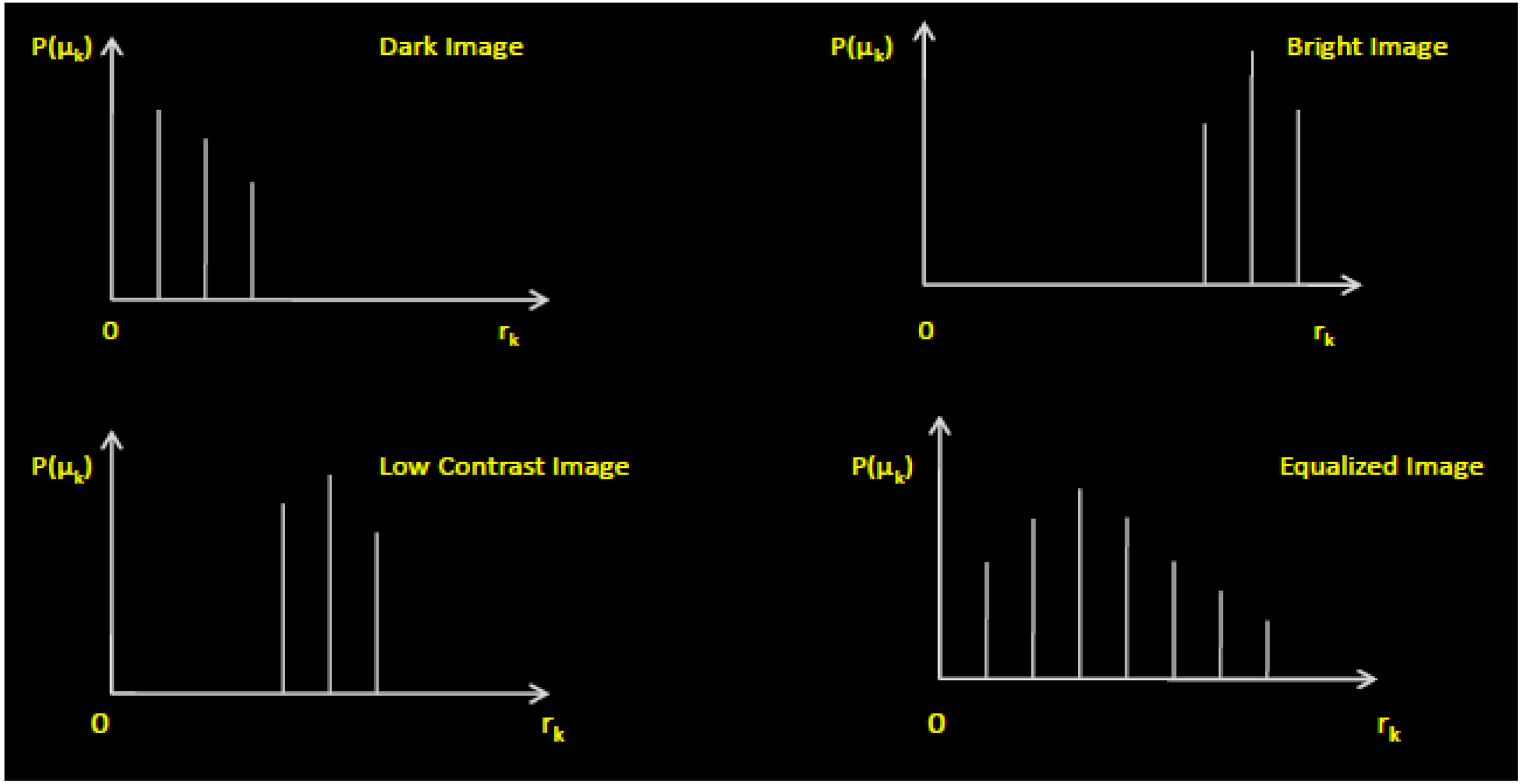
Gray level histogram

Why Histograms?

- Histograms are the basis for numerous spatial domain processing techniques
- Histogram manipulation can be used effectively for image enhancement
- Histograms can be used to provide useful image statistics
- Information derived from histograms are quite useful in other image processing applications, such as image compression and segmentation.

Gray level histogram

Types of Histograms



Gray level histogram

Types of Histograms

- In **dark Image** the components of the histogram are concentrated on the low (dark) side of gray scale
- The components of the histogram of the **bright image** are biased toward the high side of gray scale.
- A **low contrast image** histogram will be narrow and centered towards the middle of the gray scale.
- Components of the histogram in the **high-contrast image** cover a broad range of the gray scale and, further, that the distribution of pixels is not too far from uniform, with very few vertical lines being much higher than the others.
- It is reasonable to conclude that an image whose pixels tend to occupy the entire range of possible gray levels and, in addition, tend to be distributed uniformly, will have an appearance of high contrast and will exhibit a large variety of gray tones. The last graph is the best image; a high contrast image.
- Aim is to transform the first 3 histograms into the 4th type. That is try to increase the dynamic range of the image .This is called **Histogram Processing**

Gray level histogram

Histogram Processing

- There are two methods of enhancing contrast.
  - ✓ Histogram Stretching
  - ✓ Histogram Equalization

Histogram Processing - Histogram Stretching

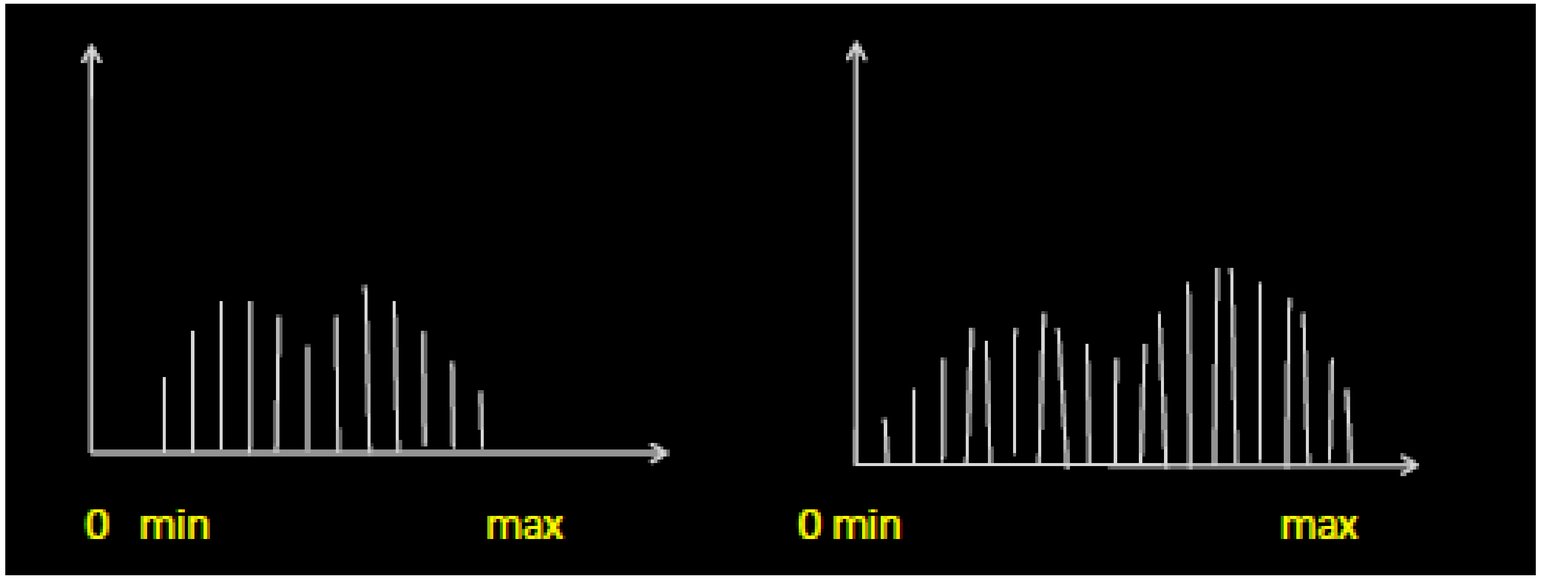
- Process of histogram stretching increases the dynamic range of the image and hence improves the contrast of the image.
- In this process, the basic shape of the histogram is not modified, but the entire range of image of histogram values is stretched.
- Before stretching process, it is necessary to specify the upper and lower limits for pixels of normalized range.
- Limit values depends on type of image. (eg: 8 bit image ,limit values ranges between 0 & 255)
- Simplest histogram function: $S = T(r) = ((S_{max}-S_{min}) / (r_{max}-r_{min})) (r - r_{min}) + S_{min}$

S_{max}– maximum limit value of image,

S_{min}– minimum limit value of image,

r_{max}– highest pixel value of image,

r_{min}– lowest pixel value of image



Histogram Processing - Histogram Stretching

Ex. 1) Perform Histogram Stretching so that the new image has a dynamic range of 0 to 7 [0, 7].

Gray Levels	0	1	2	3	4	5	6	7
No. of Pixels	0	0	50	60	50	20	10	0

Sol^n:- $r_{min} = 2$; $r_{max} = 6$; $s_{min} = 0$; $s_{max} = 7$;

$$slope = ((s_{max} - s_{min})/(r_{max} - r_{min})) = ((7 - 0)/(6 - 2)) = 7 / 4 = 1.75.$$

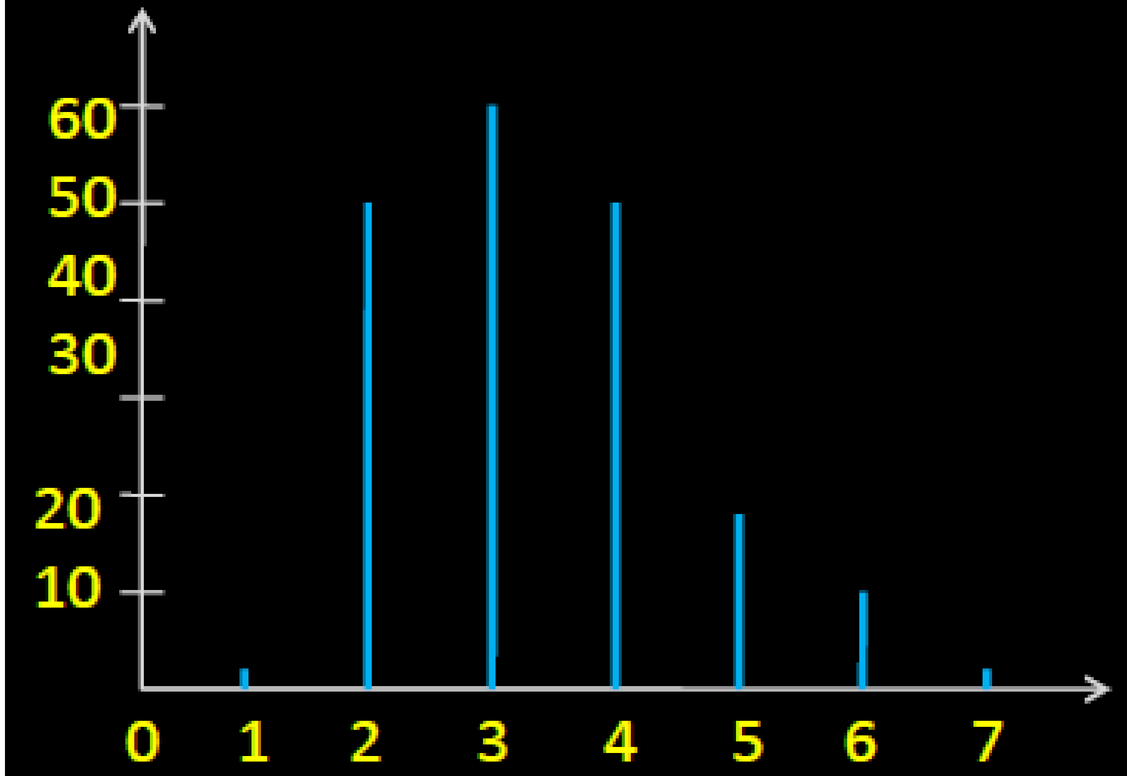
$S = (7 / 4)(r - 2) + 0;$

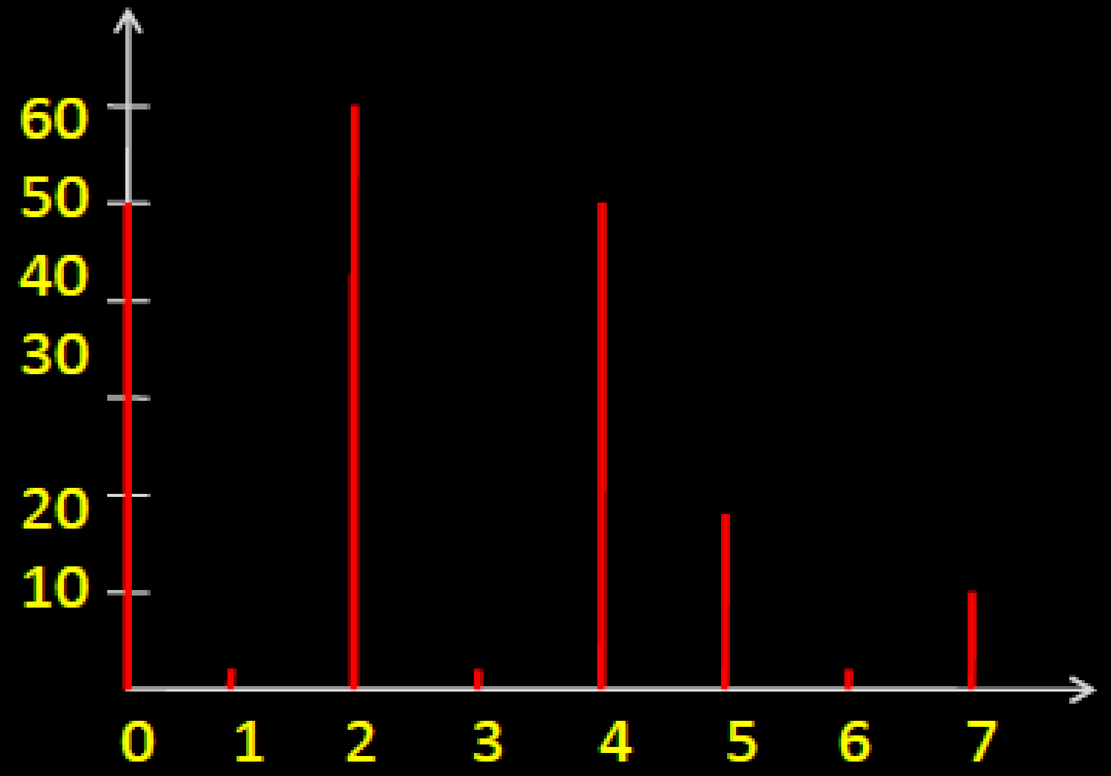
$S = (7 / 4)(r - 2)$

r	(7 / 4)(r - 2) = S
2	0 = 0
3	7/4 = 1.75 = 2
4	7/2 = 3.5 = 4
5	21/4 = 5.25 = 5
6	7 = 7

Histogram Processing - Histogram Stretching

Gray Levels	0	1	2	3	4	5	6	7
No. of Pixels	50	0	60	0	50	20	0	10





Histogram Processing - Histogram Equalization

- **Equalization** is a process that attempts to spread out the gray levels in an image so that they are evenly distributed across their range.
- It reassigns the brightness values of pixels based on image histogram.
- The histogram of resultant image is made as flat as possible.
- Provides more visually pleasing results across a wide range of images.
- Steps to perform Histogram Equalization:-
  - Find the running sum of histogram values.
  - Normalise the values from step1 by dividing by the total number of pixels
  - Multiply the values from step 2 by the maximum gray level value and round
  - Map the gray level values to the results from step 3 using a one to one correspondence.

Histogram Processing - Histogram Equalization

- Example:

  Q. Perform histogram equalization of the image.

  - Maximum value is found to be 5. We need a minimum of 3 bits to represent the number.
  - There are 8 possible gray levels from 0 to 7
  - Histogram of input image:-

$$\begin{bmatrix} 4 & 4 & 4 & 4 & 4 \\ 3 & 4 & 5 & 4 & 3 \\ 3 & 5 & 5 & 5 & 3 \\ 3 & 4 & 5 & 4 & 3 \\ 4 & 4 & 4 & 4 & 4 \end{bmatrix}$$

Gray level	0	1	2	3	4	5	6	7
No of pixels	0	0	0	6	14	5	0	0

Histogram Processing - Histogram Equalization

Step1: Compute the running sum of histogram values.

- Running sum is otherwise known as Cumulative Frequency Distribution(CFD).

Gray level	0	1	2	3	4	5	6	7
No of pixels	0	0	0	6	14	5	0	0
CFD	0	0	0	6	20	25	25	25

Gray level	0	1	2	3	4	5	6	7
No of pixels	0	0	0	6	14	5	0	0
CFD	0	0	0	6	20	25	25	25

Histogram Processing - Histogram Equalization

Step 2 : Divide the running sum obtained in step1 by the total number of pixels (which is 25 in this case).

Gray level	0	1	2	3	4	5	6	7
No of pixels	0	0	0	6	14	5	0	0
CFD	0	0	0	6	20	25	25	25
CFD/ total no of pixels	0/25	0/25	0/25	6/25	20/25	25/25	25/25	25/25

Histogram Processing - Histogram Equalization

Step3: Multiply the values from step 2 by the maximum gray level value (which is 7 in this case). The result is then rounded to closest integer.

Gray level	0	1	2	3	4	5	6	7
No of pixels	0	0	0	6	14	5	0	0
CFD	0	0	0	6	20	25	25	25
CFD/ total no of pixels	0/25	0/25	0/25	6/25	20/25	25/25	25/25	25/25
Multiply the values by max gray level	0/25*7	0/25*7	0/25*7	6/25*7	20/25*7	25/25*7	25/25*7	25/25*7
Rounded value	0	0	0	2	6	7	7	7

Histogram Processing - Histogram Equalization

Step 4: Map the gray level values to the results from step 3 using a one to one correspondence

Original image

$$\begin{bmatrix} 4 & 4 & 4 & 4 & 4 \\ 3 & 4 & 5 & 4 & 3 \\ 3 & 5 & 5 & 5 & 3 \\ 3 & 4 & 5 & 4 & 3 \\ 4 & 4 & 4 & 4 & 4 \end{bmatrix}$$

Histogram equalized image

$$\begin{bmatrix} 6 & 6 & 6 & 6 & 6 \\ 2 & 6 & 7 & 6 & 2 \\ 2 & 7 & 7 & 7 & 2 \\ 2 & 6 & 7 & 6 & 2 \\ 6 & 6 & 6 & 6 & 6 \end{bmatrix}$$

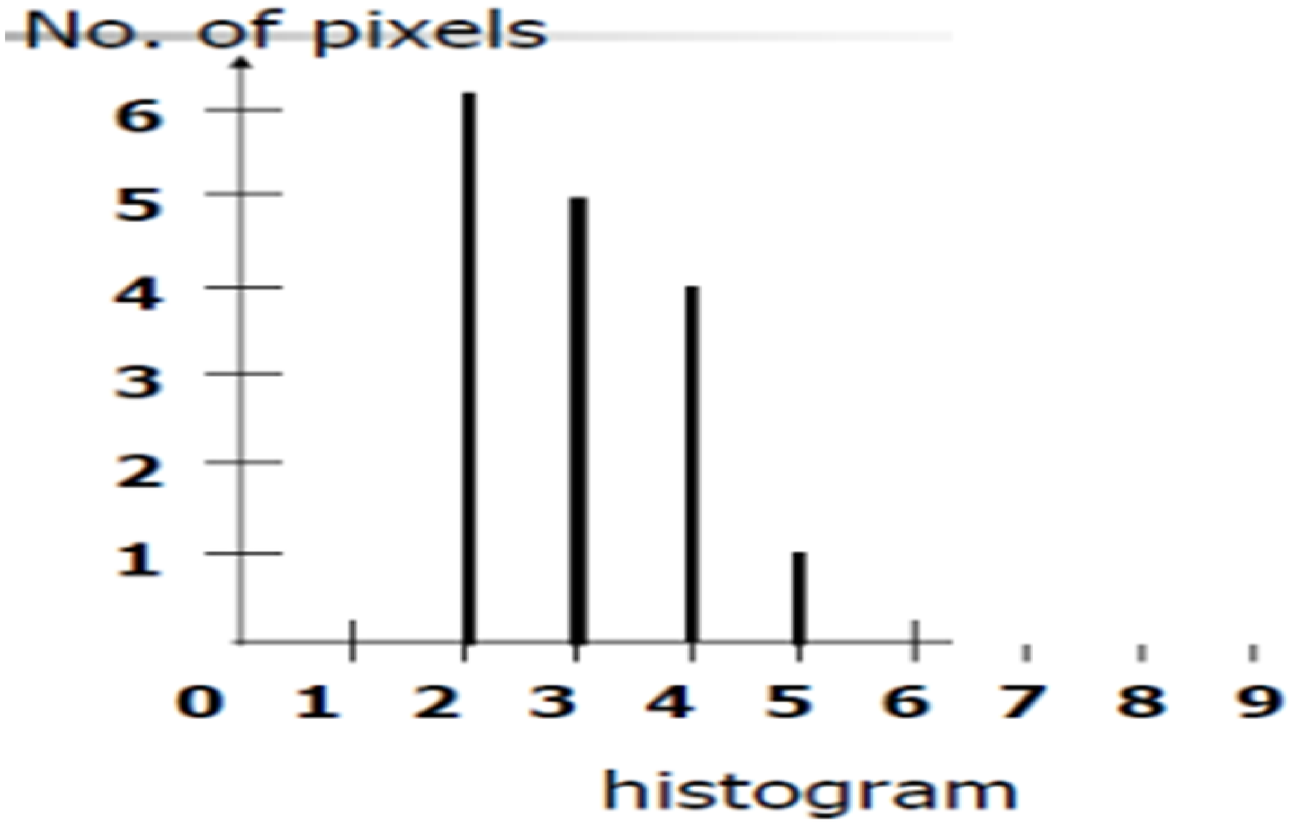
Original gray level	Histogram equalized level
0	0
1	0
2	0
3	2
4	6
5	7
6	7
7	7

Tutorial :- Histogram Equalization

2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4

4x4 image

Gray scale = [0,9]



Gray Level(j)	0	1	2	3	4	5	6	7	8	9
No. of pixels	0	0	6	5	4	1	0	0	0	0

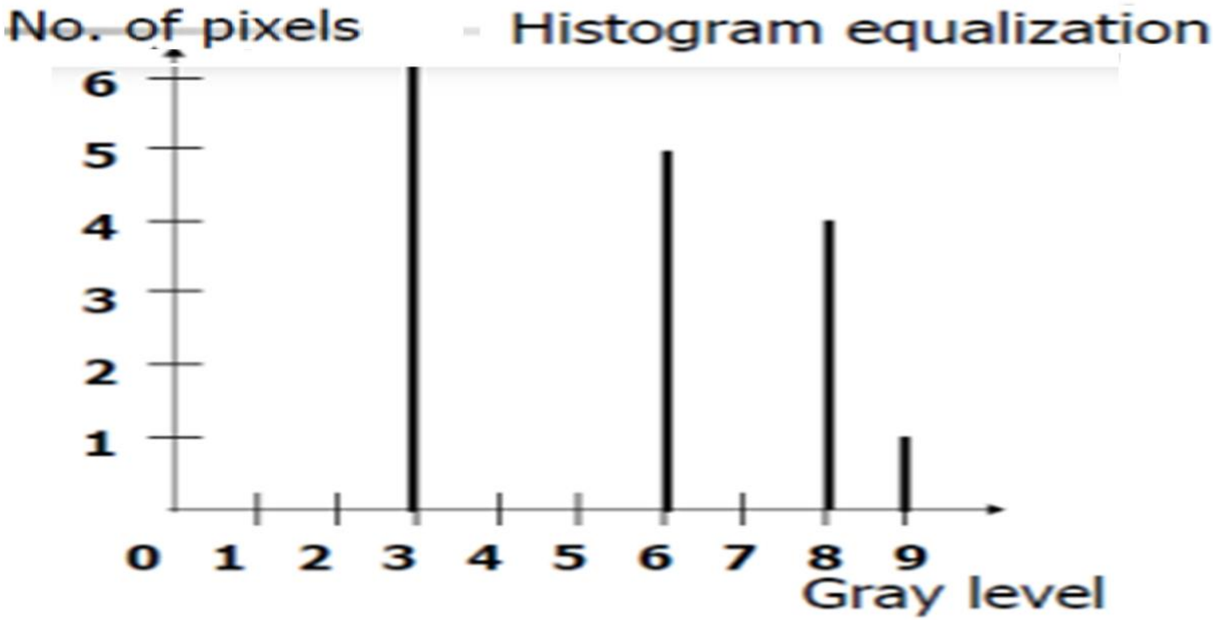
Tutorial :- Histogram Equalization

Gray Level(j)	0	1	2	3	4	5	6	7	8	9
No. of pixels	0	0	6	5	4	1	0	0	0	0
$\sum_{j=0}^{k} n_j$	0	0	6	11	15	16	16	16	16	16
$s = \sum_{j=0}^{k} \frac{n_j}{n}$	0	0	6 / 16	11 / 16	15 / 16	16 / 16	16 / 16	16 / 16	16 / 16	16 / 16
$s \times 9$	0	0	3.3 ≈3	6.1 ≈6	8.4 ≈8	9	9	9	9	9

3	6	6	3
8	3	8	6
6	3	6	9
3	8	3	8

Output image

Gray scale = [0,9]



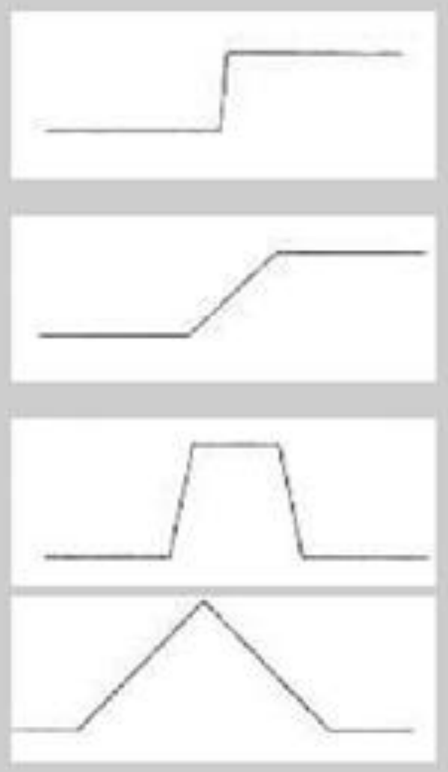
Edge Detection

Edge

- Sudden changes or discontinuities in an image are called as edges.
- Significant transitions in an image are called as edges
- Edges are significant local changes of intensity in an image.

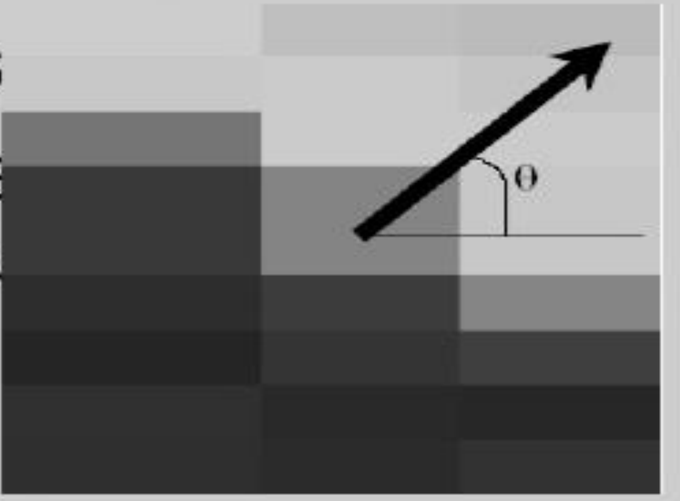
Edge Types

- Step Edge
- Ramp Edge
- Ridge
- Roof



Gradient based Edge Detection

- Best used for abrupt discontinuities
- Perform better in less noised image
- Magnitude of the gradient - strength of the edge .
- Direction - opposite of the edge direction.



$$|G| = \sqrt{Gx^2 + Gy^2} \approx |Gx| + |Gy| \qquad \theta = \tan^{-1}\left(\frac{Gy}{Gx}\right)$$

Edge Detection

- **Edge detection -** An image processing technique for finding the boundaries of objects within images
- It works by detecting discontinuities in brightness
- Used for image segmentation and data extraction in areas like image processing, computer vision, etc.
- Generally edges are of three types:
  - Horizontal Edges
  - Vertical Edges
  - Diagonal Edges

Edge Detection

Edge detection methods

- Prewitt Operator
- Sobel Operator
- Roberts Operator

- All these masks find edges. Some find horizontally and vertically, some find in one direction only and some find in all the directions.

Edge Detection

Gradient based Edge detection methods

- Prewitt Operator : Prewitt operator is used for detecting edges horizontally and vertically.
- Sobel Operator: The Sobel operator is very similar to Prewitt operator. It is also a derivate mask which calculates edges in both horizontal and vertical direction.
- Roberts Operator: This method is based on differences between adjacent pixels. Here +1 and -1 are explicitly used to find the edges. The difference is called as forward differences.

Edge Detection

Prewitt Operator

- It detects two types of edges: Horizontal edges & Vertical Edges
- Edges are calculated by using difference between corresponding pixel intensities of an image.
- Changes in a image can only be calculated using differentiation. So these operators are also called as derivative operators or derivative masks.
- All the derivative masks should have the following properties:
  - Opposite sign should be present in the mask.
  - Sum of mask should be equal to zero.
  - More weight means more edge detection.
- Prewitt operator provides us two masks one for detecting edges in horizontal direction and another for detecting edges in an vertical direction.

Edge Detection

Prewitt Operator

- Vertical direction

-1	0	1
-1	0	1
-1	0	1

- When you will convolve this mask on an image, it will give you the vertical edges in an image.
- When we apply this mask on the image it prominent vertical edges.
- It simply works like as first order derivate and calculates the difference of pixel intensities in a edge region.
- As the center column is of zero so it does not include the original values of an image but rather it calculates the difference of right and left pixel values around that edge.
- This increase the edge intensity and it become enhanced comparatively to the original image.
- Above mask will find the edges in vertical direction and it is because the zeros column in the vertical direction.

Edge Detection

Prewitt Operator

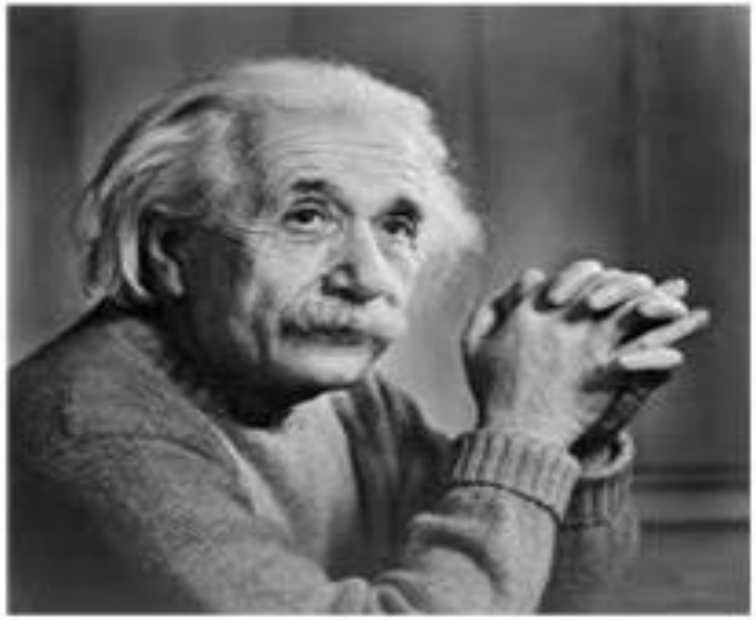
- Horizontal direction

-1	-1	-1
0	0	0
1	1	1

- Above mask will find edges in horizontal direction and it is because that zeros column is in horizontal direction.
- When you will convolve this mask onto an image it would prominent horizontal edges in the image.
- As the center row of mask is consist of zeros so it does not include the original values of edge in the image but rather it calculate the difference of above and below pixel intensities of the particular edge.
- Thus increasing the sudden change of intensities and making the edge more visible.
- In the picture on which we apply vertical mask, all the vertical edges are more visible than the original image.
- Similarly in the picture which we apply the horizontal mask ,all the horizontal edges are visible.
- So in this way we can detect both horizontal and vertical edges from an image.

Edge Detection

Prewitt Operator



- Applying Vertical Mask

Applying Horizontal Mask

Edge Detection

Sobel operator

- Very similar to Prewitt operator.
- also detects two types of edges: Horizontal edges & Vertical Edges
- Edges are calculated by using difference between corresponding pixel intensities of an image.
- Major difference is that in Sobel operator the coefficients of masks are not fixed and they can be adjusted according to our requirement unless they do not violate any property of derivative masks.
- provides us two masks one for detecting edges in horizontal direction and another for detecting edges in an vertical direction.

Edge Detection

Sobel operator

- Vertical direction

-1	0	1
-2	0	2
-1	0	1

- This mask works exactly same as the Prewitt operator vertical mask.
- There is only one difference that is it has “2” and “-2” values in center of first and third column.
- This give more weightage to the pixel values around the edge region. This increase the edge intensity and it become enhanced comparatively to the original image.
- When applied on an image this mask will highlight the vertical edges.
- It simply works like as first order derivate and calculates the difference of pixel intensities in a edge region.
- As the center column is of zero so it does not include the original values of an image but rather it calculates the difference of right and left pixel values around that edge.

Edge Detection

Sobel operator

- Horizontal direction

-1	-2	-1
0	0	0
1	2	1

- Above mask will find edges in horizontal direction and it is because that zeros column is in horizontal direction. On convolving this mask onto an image it would prominent horizontal edges in the image.
- The only difference between it is that it have 2 and -2 as a center element of first and third row.
- As the center row of mask is consist of zeros so it does not include the original values of edge in the image but rather it calculate the difference of above and below pixel intensities of the particular edge. Thus increasing the sudden change of intensities and making the edge more visible.
- Also if you compare the result of sobel operator with Prewitt operator, you will find that sobel operator finds more edges or make edges more visible as compared to Prewitt Operator as more weight has been allotted to the pixel intensities around the edges.
- apply more weight to mask, more edges will be visible.

Edge Detection

Sobel operator

- there are no fixed coefficients in sobel operator, so here is another weighted operator

-1	0	1
-5	0	5
-1	0	1

- This will give out more edges as compared to previous mask because we have allotted more weight in the mask.

Edge Detection

Roberts cross operator

- The Robert's cross operator performs a simple, quick to compute edges on an image.
- As a differential operator, the idea behind the Roberts cross operator is to approximate the gradient of an image through discrete differentiation which is achieved by computing the sum of the squares of the differences between diagonally adjacent pixels.
- According to Roberts, an edge detector should have the following properties:
  - the produced edges should be well-defined
  - the background should contribute as little noise as possible
  - the intensity of edges should correspond as close as possible to what a human would
- With these criteria in mind Roberts proposed the following equations:

$$y_{i,j} = \sqrt{x_{i,j}}$$

$$z_{i,j} = \sqrt{(y_{i,j} - y_{i+1,j+1})^2 + (y_{i+1,j} - y_{i,j+1})^2}$$

where x is the initial intensity value in the image, z is the computed derivative and i, j represent the location in the image

Edge Detection

Roberts cross operator

- The results of this operation will highlight changes in intensity in a diagonal direction.
- In order to perform edge detection with the Roberts operator we first convolve the original image, with the following 2x2 convolution masks M_x and M_y.

1	0
0	-1

M_x

0	1
-1	0

M_y

- One mask is simply the other rotated by 90 degrees. Thus the Robert's cross operator uses the diagonal directions to calculate the gradient vector.

Edge Detection

Roberts cross operator

- Let I(x,y) be a point in the original image and G_x(x,y) be a point in an image formed by convolving with the first kernel and G_y(x,y) be a point in an image formed by convolving with the second kernel.

- The gradient can then be defined as:

$$\nabla I(x,y) = G(x,y) = \sqrt{G_x^2 + G_y^2}.$$

- Direction of gradient can then be defined as:

$$\Theta(x,y) = \arctan\left(\frac{G_y(x,y)}{G_x(x,y)}\right) - \frac{3\pi}{4}.$$

Smoothing Linear Filters

- Smoothing
  - elimination of high frequencies
  - usually performed using averaging (either weighted or non-weighted)
- Example: Smoothing using average 3x3 filters

$$H=\frac{1}{9}\begin{pmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{pmatrix}$$

non-weighted

$$H=\frac{1}{10}\begin{pmatrix}1 & 1 & 1\\ 1 & 2 & 1\\ 1 & 1 & 1\end{pmatrix}$$

weighted

$$H=\frac{1}{16}\begin{pmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{pmatrix}$$

weighted

Neighborhood processing in spatial domain: Here, to modify one pixel, values of the immediate neighboring pixels also considered. For this purpose, 3X3, 5X5, or 7X7 neighborhood mask can be considered.

f(x-1, y-1)	f(x-1, y)	f(x-1, y+1)
f(x, y-1)	f(x, y)	f(x, y+1)
f(x+1, y-1)	f(x+1, y)	f(x+1, y+1)

Low Pass filtering: It is also known as the smoothing filter. It removes the high-frequency content from the image. It is also used to blur an image.

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

High Pass Filtering: It eliminates low-frequency regions while retaining or enhancing the high-frequency components

A high pass filtered image may be computed as the difference between the original image and a low pass filtered version of that image as follows

High pass = Original – Low pass

Multiplying the original by an amplification factor yields a high boost or high frequency-emphasis filter.

$$\begin{aligned} Highboost &= A(Original) - Lowpass \\ &= (A-1)(Original) + Original - Lowpass \\ &= (A-1)(Original) + Highpass \end{aligned}$$

A high pass filtering mask is as shown.

-1/9	-1/9	-1/9
-1/9	8/9	-1/9
-1/9	-1/9	-1/9

Median Filtering: It is also known as nonlinear filtering. It is used to eliminate salt and pepper noise. Here the pixel value is replaced by the median value of the neighboring pixel.

SPATIAL CONVOLUTION AND CORRELATION

Spatial filtering

- Filtering refers to accepting(passing) or rejecting certain frequency components. This effectively smoothens or sharpens the image.
  - E.g. Low pass filter, high pass filter, etc.
- If the operation performed on the image pixels is linear, then the filter is called a linear spatial filter, otherwise nonlinear.

Spatial Filters

Spatial filters can be classified by effect into:

Smoothing Spatial Filters: also called lowpass filters. They include:

Averaging linear filters

Order-statistics nonlinear filters.

Sharpening Spatial Filters: also called highpass filters.

eg: Laplacian linear filter.

Spatial Filters

Smoothing Spatial Filters

Used for blurring and for noise reduction.

Blurring is used in preprocessing steps to:

- remove small details from an image prior to (large) object extraction
- bridge small gaps in lines or curves.

Noise reduction can be accomplished by blurring with a linear filter and also by nonlinear filtering.

Averaging linear filters

The response of averaging filter is simply the average of the pixels contained in the neighborhood of the filter mask.

The output of averaging filters is a smoothed image with reduced "sharp" transitions in gray levels.

Noise and edges consist of sharp transitions in gray levels. Thus smoothing filters are used for noise reduction; however, they have the undesirable side effect that they blur edges.

Linear Spatial Filtering (Convolution)

The process consists of moving the filter mask from pixel to pixel in an image. At each pixel (x,y), the response is given by a sum of products of the filter coefficients and the corresponding image pixels in the area spanned by the filter mask.

For the 3×3 mask shown in the previous figure, the result (or response), R, of linear filtering is:

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

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

In general, linear filtering of an image *f* of size *M*× *N* with a filter mask of size *m*× *n* is given by the expression:

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

where a = (*m* - 1)/2 and b = (*n* - l)/2. To generate a complete filtered image this equation must be applied for x = 0,1, 2,..., *M*-1 and y = 0,1, 2,..., *N*-1.

Nonlinear Spatial Filtering

- This operation also consists of moving the filter mask from pixel to pixel in an image.
- The filtering operation is based conditionally on the values of the pixels in the neighborhood, and they do not explicitly use coefficients in the sum-of-products manner.
- For example, noise reduction can be achieved effectively with a nonlinear filter whose basic function is to compute the median gray-level value in the neighborhood in which the filter is located.
- Computation of the median is a nonlinear operation.

Linear Spatial Filtering

Use the following 3×3 mask to perform the convolution process on the shaded pixels in the 5×5 image below. Write the filtered image.

0	1/6	0
1/6	1/3	1/6
0	1/6	0

3×3 mask

30	40	50	70	90
40	50	80	60	100
35	255	70	0	120
30	45	80	100	130
40	50	90	125	140

5×5 image

Solution:

$$0 \times 30 + \frac{1}{6} \times 40 + 0 \times 50 + \frac{1}{6} \times 40 + \frac{1}{3} \times 50 + \frac{1}{6} \times 80 + 0 \times 35 + \frac{1}{6} \times 255 + 0 \times 70 = 85$$

$$0 \times 40 + \frac{1}{6} \times 50 + 0 \times 70 + \frac{1}{6} \times 50 + \frac{1}{3} \times 80 + \frac{1}{6} \times 60 + 0 \times 255 + \frac{1}{6} \times 70 + 0 \times 0 = 65$$

$$0 \times 50 + \frac{1}{6} \times 70 + 0 \times 90 + \frac{1}{6} \times 80 + \frac{1}{3} \times 60 + \frac{1}{6} \times 100 + 0 \times 70 + \frac{1}{6} \times 0 + 0 \times 120 =$$

$$0 \times 40 + \frac{1}{6} \times 50 + 0 \times 80 + \frac{1}{6} \times 35 + \frac{1}{3} \times 255 + \frac{1}{6} \times 70 + 0 \times 30 + \frac{1}{6} \times 45 + 0 \times 80 = 118$$

and so on ...

Filtered image =

30	40	50	70	90
40	**85**	**65**	**61**	100
35	**118**	**92**	**58**	120
30	**84**	**77**	**89**	130
40	50	90	125	140

Weighted average filter has different coefficients to give more importance (weight) to some pixels at the expense of others. The idea behind that is to reduce blurring in the smoothing process.

Averaging linear filtering of an image f of size $M \times N$ with a filter mask of size $m \times n$ is given by the expression given here.

To generate a complete filtered image this equation must be applied for

x = 0,1, 2,..., M-1 and y = 0,1, 2,..., N-1.

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

Effects of Averaging linear filter

Figure 6.2 Effect of averaging filter. (a) Original image. (b)-(f) Results of smoothing with square averaging filter masks of sizes n = 3,5,9,15, and 35, respectively.

Averaging linear filter

Effects of averaging linear filter are:

- Blurring which is increased whenever the mask size increases.
- Blending (removing) small objects with the background. The size of the mask establishes the relative size of the blended objects.
- Black border because of padding the borders of the original image.
- Reduced image quality.

The effects of median filter are:

1. Noise reduction
2. Less blurring than averaging linear filter

❑ Order-statistics filters

Nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in the neighborhood, and then replacing the value of the center pixel with the value determined by the ranking result.

Examples include Max, Min, and Median filters.

- Median filter: Computes the median gray-level value of the neighbourhood, (noise reduction).
- Max filter: Used to find the brightest points in an image $R=\max\{ z|k=1,2,...,9\}$
- Min filter: Used to find the dimmest points in an image $R=\min\{ z|k=1,2,...,9\}$

Median filter

It replaces the value at the center by the median pixel value in the neighborhood, (i.e. the middle element after they are sorted). Median filters are particularly useful in removing impulse noise (also known as salt-and-pepper noise).

Salt = 255, pepper = 0 gray levels.

In a 3×3 neighborhood the median is the 5th largest value, in a 5×5 neighborhood the 13th largest value, and so on.

- Suppose that a 3×3 neighborhood has gray levels (10, 20, 0, 20, 255, 20, 20, 25, 15).

These values are sorted as (0,10,15,20,20,20,20,25,255), which results in a median of 20 that replaces the original pixel value 255 (salt noise).

Nonlinear Median Filter

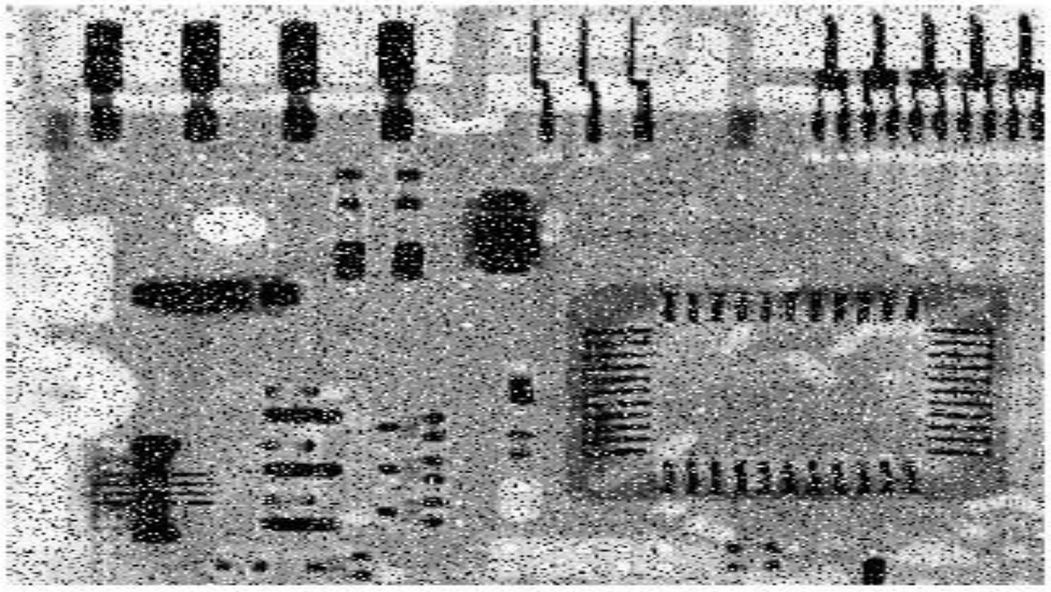
101	86	99
100	106	103
91	102	109

f (x,y)

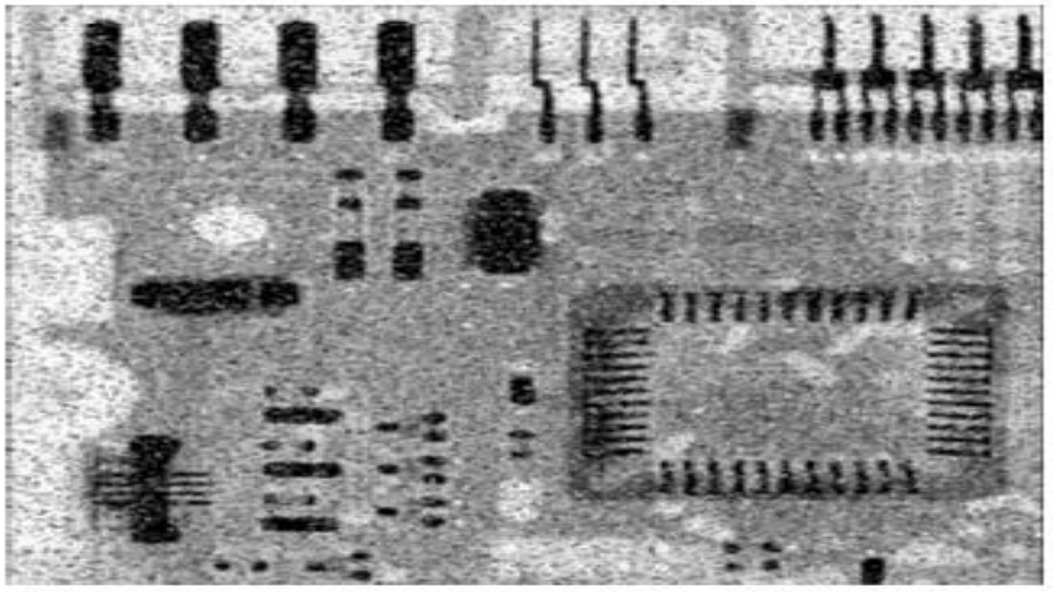
86
91
99
100
101
102
103
106
109

	101	

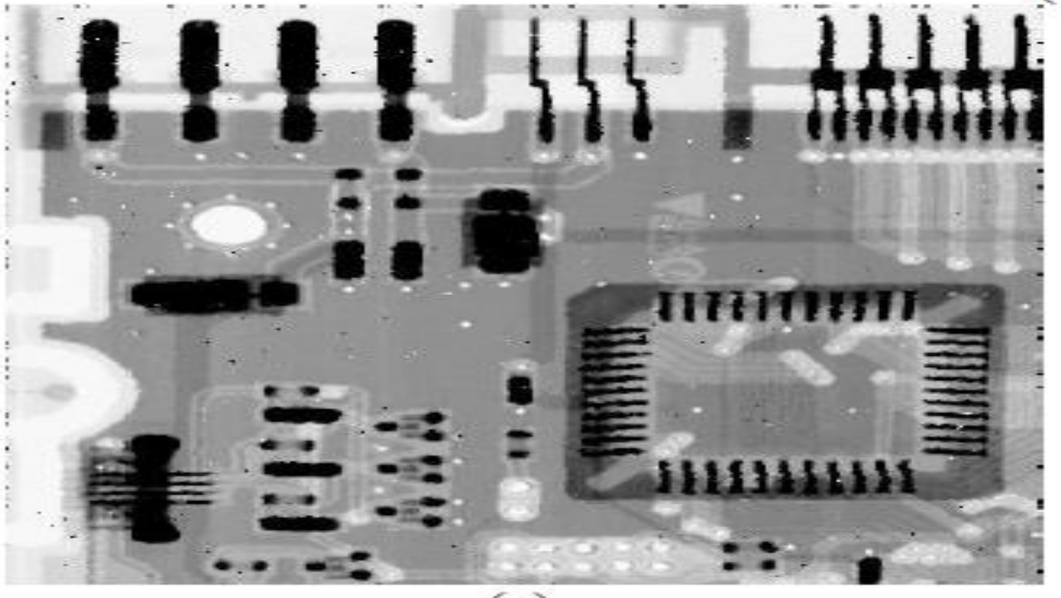
g (x,y)



(a)



(b)



(c)

Figure 6.3 Effect of median filter. (a) Image corrupted by salt & pepper noise. (b) Result of applying 3×3 standard averaging filter on (a). (c) Result of applying 3×3 median filter on (a).

Example:

Consider the following 5×5 image:

20	30	50	80	100
30	20	80	100	110
25	255	70	0	120
30	30	80	100	130
40	50	90	125	140

Apply a 3×3 median filter on the shaded pixels, and write the filtered image.

Solution

20	30	50	80	100
30	20	80	100	110
25	255	70	0	120
30	30	80	100	130
40	50	90	125	140

Sort:
20, 25, 30, 30, **30**, 70, 80, 80, 255

20	30	50	80	100
30	20	80	100	110
25	255	70	0	120
30	30	80	100	130
40	50	90	125	140

Sort
0, 20, 30, 70, **80**, 80, 100, 100, 255

20	30	50	80	100
30	20	80	100	110
25	255	70	0	120
30	30	80	100	130
40	50	90	125	140

Sort
0, 70, 80, 80, **100**, 100, 110, 120, 130

Filtered Image =

20	30	50	80	100
30	20	80	100	110
25	**30**	**80**	**100**	120
30	30	80	100	130
40	50	90	125	140

Sharpening Spatial Filters

Sharpening aims to highlight fine details (e.g. edges) in an image, or enhance detail that has been blurred through errors or imperfect capturing devices.

Image blurring can be achieved using averaging filters, and hence sharpening can be achieved by operators that invert averaging operators. In mathematics, averaging is equivalent to the concept of integration, and differentiation inverts integration. Thus, sharpening spatial filters can be represented by partial derivatives.

Partial derivatives of digital functions

The first order partial derivatives of the digital image $f(x,y)$ are:

$$\frac{\partial f}{\partial x} = f(x+1,y) - f(x,y) \quad \text{and} \quad \frac{\partial f}{\partial y} = f(x,y+1) - f(x,y)$$

❖ The first derivative must be:

- zero along flat segments (i.e. constant gray values).
- non-zero at the outset of gray level step or ramp (edges or noise)
- non-zero along segments of continuing changes (i.e. ramps).

Partial Derivatives

The second order partial derivatives of the digital image $f(x,y)$ are:

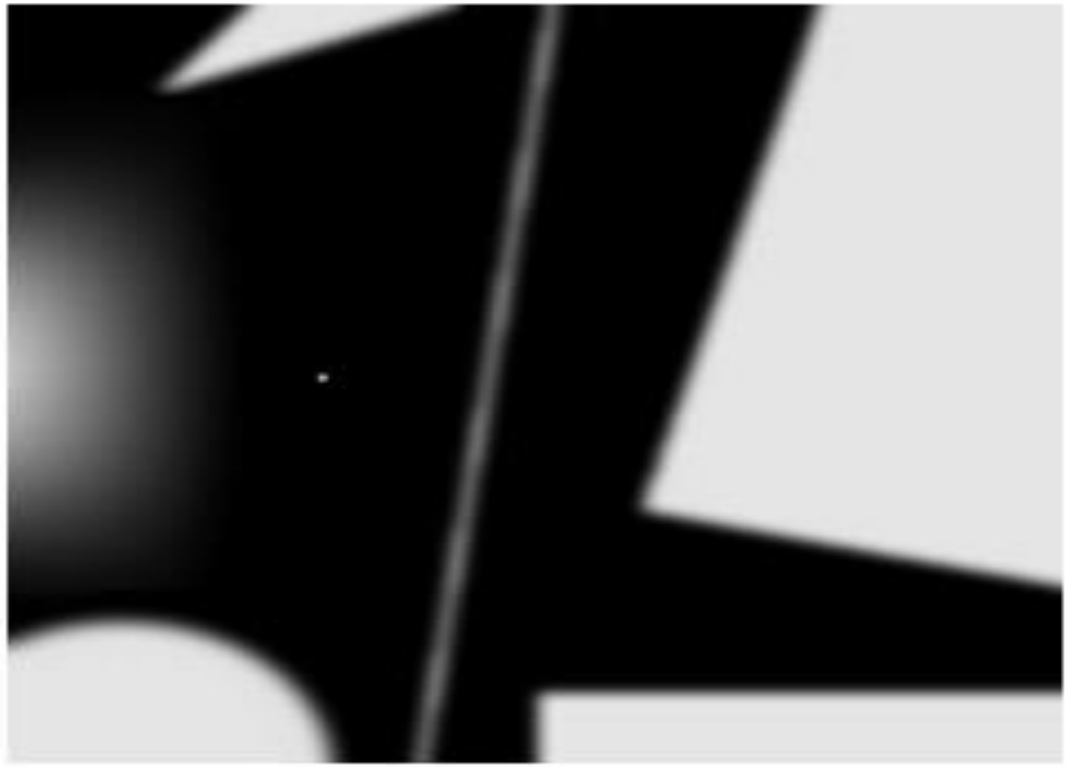
$$\frac{\partial^2 f}{\partial x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y)$$

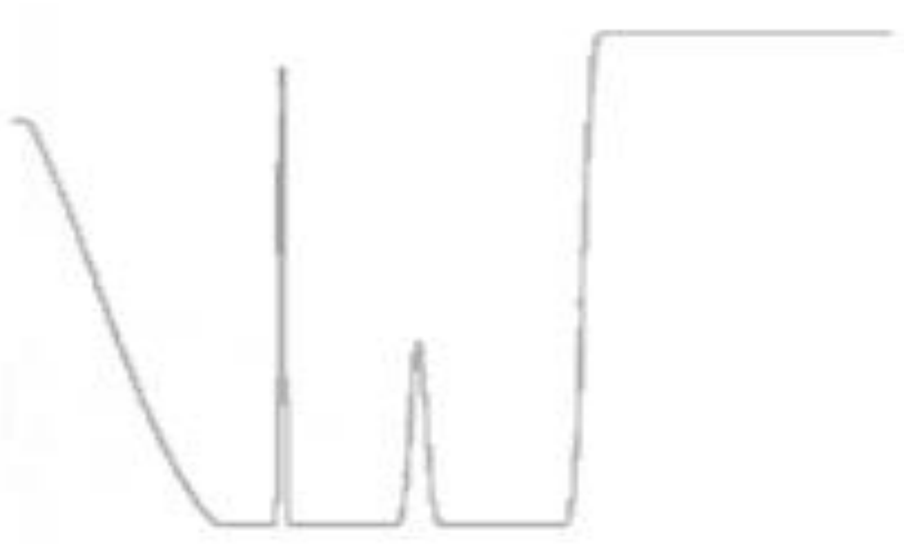
$$\frac{\partial^2 f}{\partial y^2} = f(x,y+1) + f(x,y-1) - 2f(x,y)$$

❖ The second derivative must be:

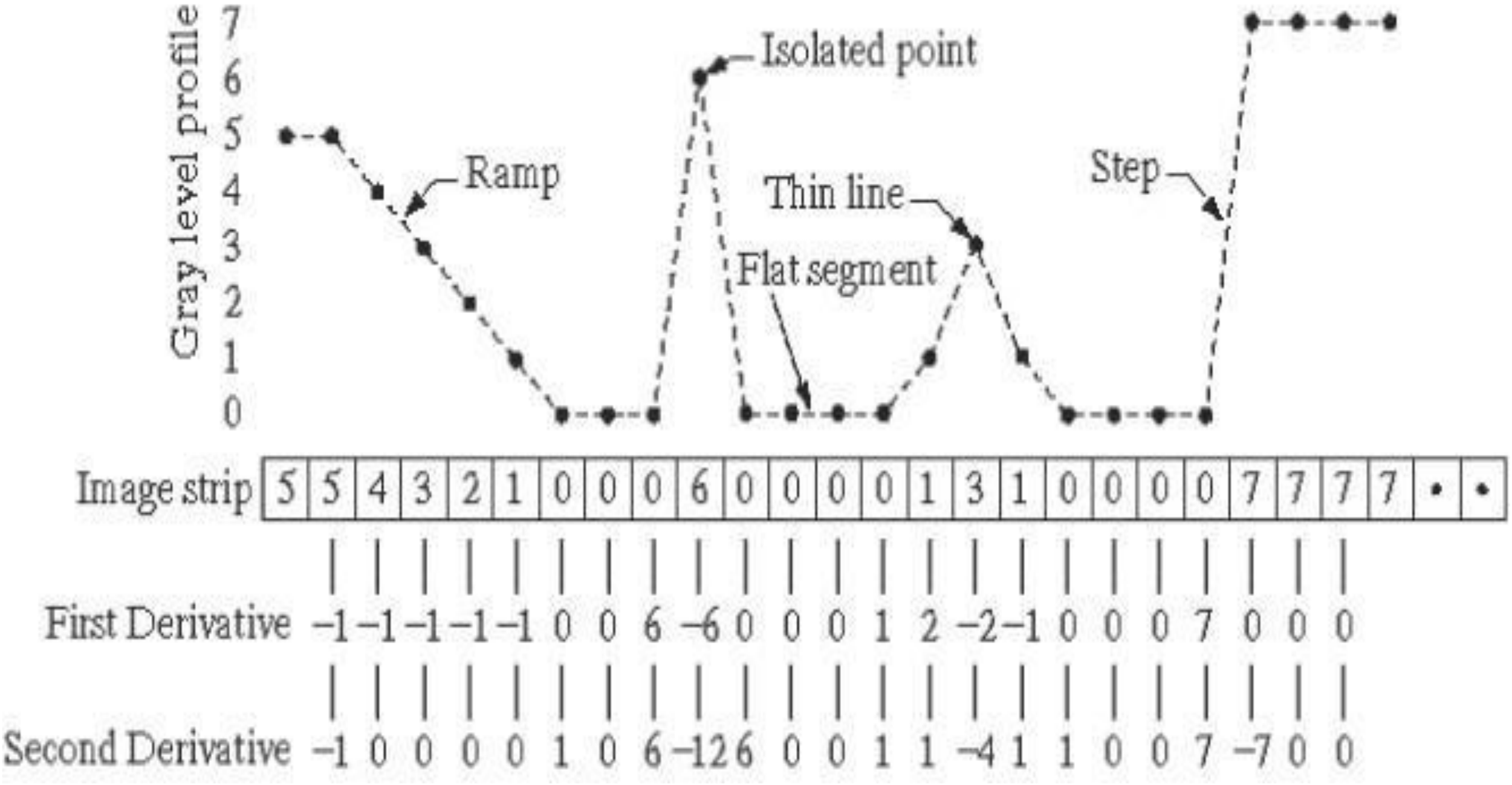
- zero along flat segments.
- nonzero at the outset and end of a gray-level step or ramp;
- zero along ramps of constant slope

Partial Derivatives





Partial Derivatives



Partial Derivatives

❑ Conclusion:

- 1st derivative detects thick edges while 2nd derivative detects thin edges.
- 2nd derivative has much stronger response at gray-level step than 1st derivative.

Thus, we can expect a second-order derivative to enhance fine details (thin lines, edges, including noise) much more than a first-order derivative.

The Laplacian Filter

The Laplacian operator of an image $f(x,y)$ is:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

This equation can be implemented using the 3×3 mask:

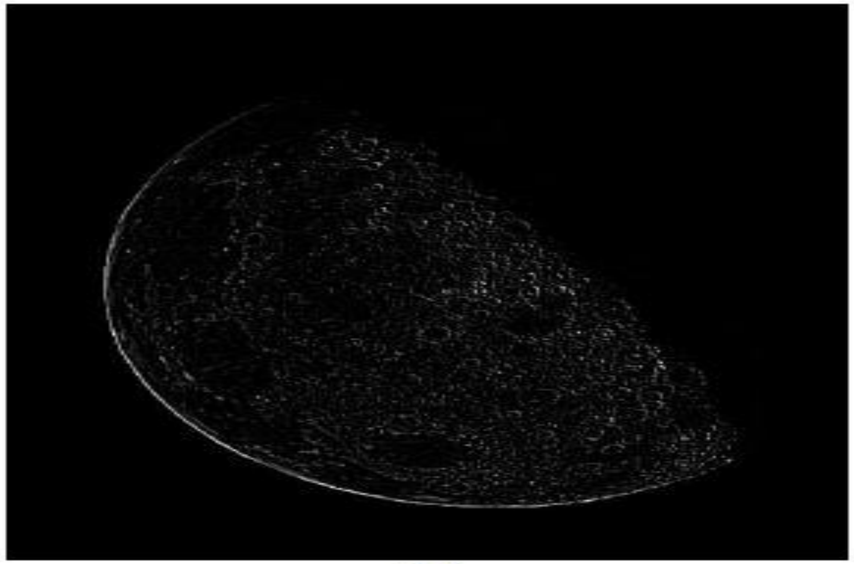
$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

- Since the Laplacian filter is a linear spatial filter, it can be applied using the same mechanism of the convolution process.
- This will produce a laplacian image that has grayish edge lines and other discontinuities, all superimposed on a dark, featureless background.
- Background features can be "recovered" while still preserving the sharpening effect of the Laplacian operation simply by adding the original and Laplacian images.
- The main disadvantage of the Laplacian operator – it produces double edges.

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

The Laplacian Filter

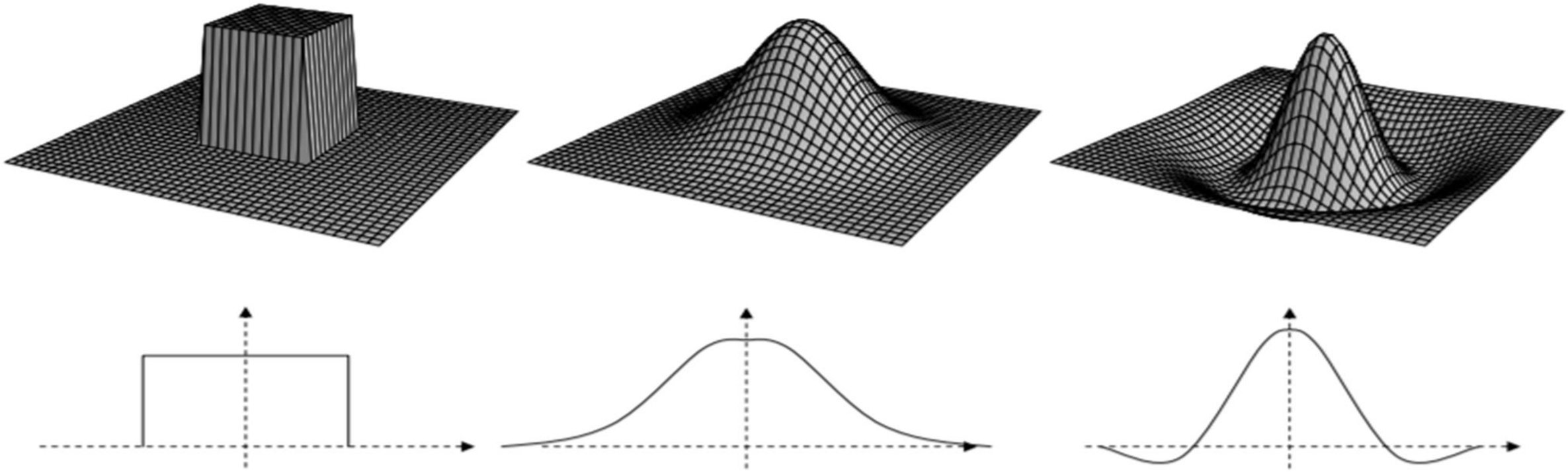
(a)



(b)

(c)

Figure 6.5 Example of applying Laplacian filter. (a) Original image. (b) Laplacian image. (c) Sharpened image.



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

(a)

0	1	2	1	0
1	3	5	3	1
2	5	9	5	2
1	3	5	3	1
0	1	2	1	0

(b)

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

(c)

SPATIAL CONVOLUTION AND CORRELATION

Correlation & Convolution

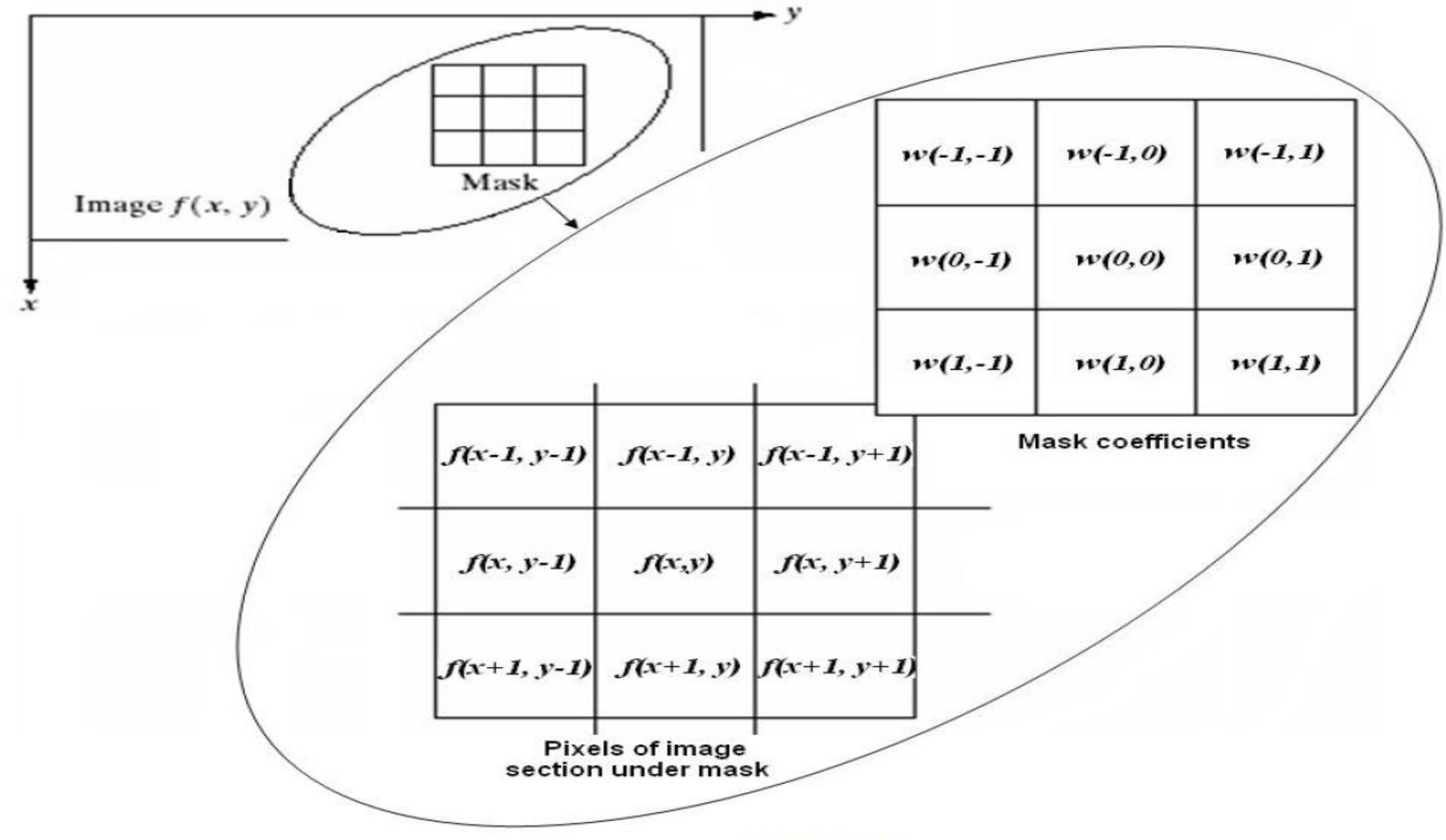
- Correlation & Convolution are two closely related concepts used in linear spatial filtering.
- Both are used to extract information from images.
- Both are basically linear and shift invariant operations.
- Term linear indicates that a pixel is replaced by the linear combination of its neighbours.
- Term shift variant means that same operations are performed at every point in the image.
- Correlation: It is a process of moving a filter mask over an image & computing the sum of products at each location.
- Convolution: mechanics are same, except that the filter is first rotated by 180˚.

SPATIAL CONVOLUTION AND CORRELATION

Spatial convolution

- Method which replaces a pixel with the weighted average of the pixel and its neighbours of the convolution mask.
- Convolution is basically mathematical operation where each value in the output is expressed as the sum of values in the input multiplied by a set of weighting coefficients.
- An image can be either smoothened or sharpened by convolving the image with respect to low pass and high pass filter respectively.
- Application areas - image filtering, image segmentation, feature extraction etc.

The mechanism of linear spatial filtering



f(x-1, y-1)	f(x-1, y)	f(x-1, y+1)
f(x, y-1)	f(x, y)	f(x, y+1)
f(x+1, y-1)	f(x+1, y)	f(x+1, y+1)

W(-1,-1)	W(-1,0)	W(-1,1)
W(0,-1)	W(0,0)	W(0,1)
W(1,-1)	W(1,0)	W(1,1)

Linear Spatial Filtering

- For linear spatial filtering, the response is given by a sum of products of the filter coefficients and the corresponding image pixels in the area spanned by the filter mask. For the 3 X 3 mask shown in the previous figure, the result (response), of linear filtering with the filter mask at a point (x,y) in the image is:

g(x,y) = w (-1,-1) f(x-1, y-1) + w(-1,0) f (x-1, y) + … + w(0,0) f(x,y) + … + w(1,0) f(x+1, y) + w(1,1) f(x+1, y+1)

which we see is the sum of products of the mask coefficients with the corresponding pixels directly under the mask.

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

Original image

		a	b	c	
		d	x	f	
		g	h	i	

⊗

Filter mask

1	3	1
2	4	-2
-3	1	4

1a	3b	1c
2d	4x	-2f
-3g	1h	4i

a	b	c
d	y	f
g	h	i

Output image

y = 1a+3b+1c+2d+4x-2f-3g+1h+4i

Linear Spatial Filtering

- There are two closely related concepts that must be understood clearly when performing linear spatial filtering.
- One is correlation; the other is convolution.
- Correlation is the process of passing the filter mask w by the image and computing the sum of products at each location
- Mechanically, convolution is the same process, except that w is rotated by 180 degrees prior to passing it by f.

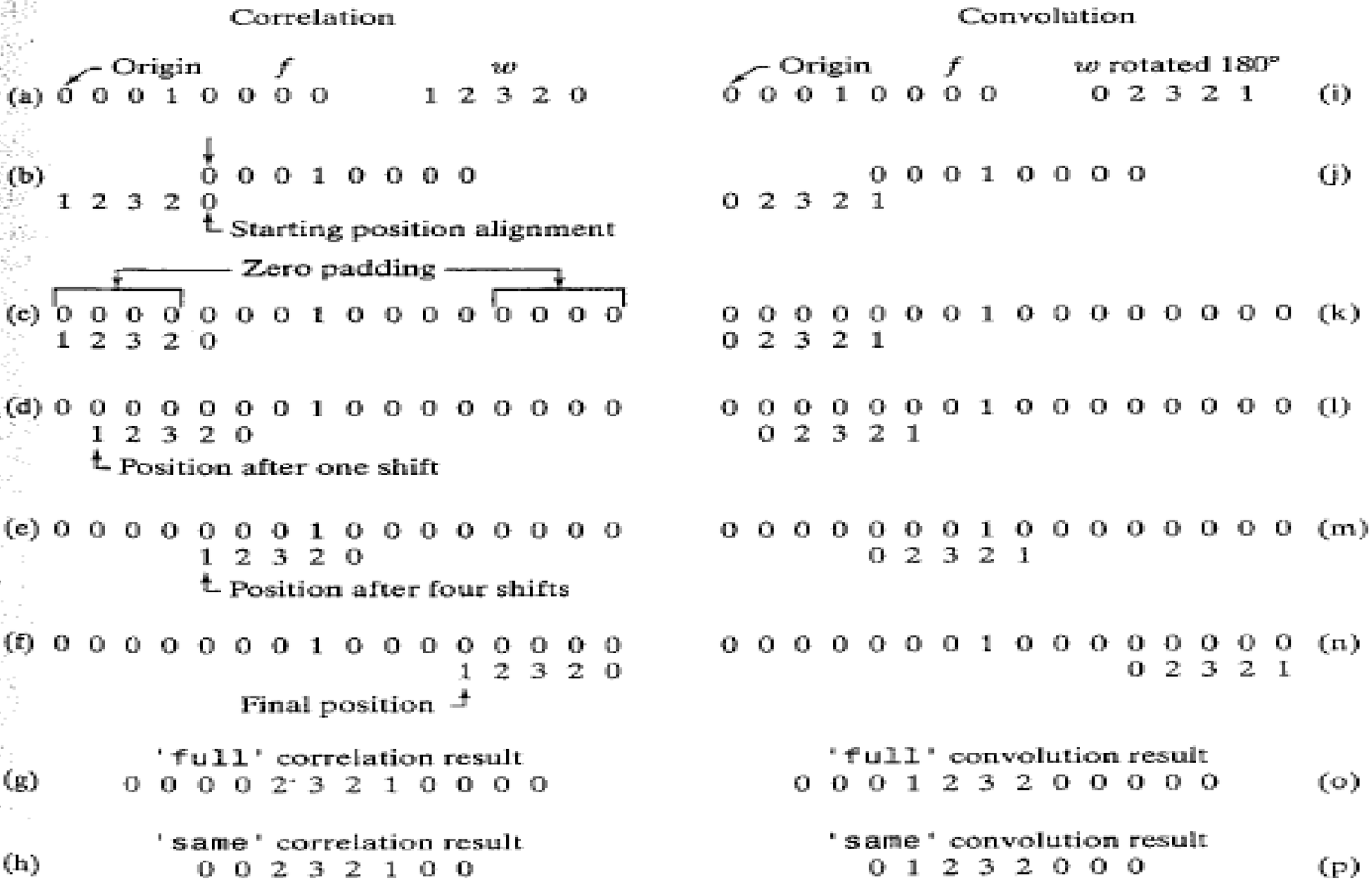


FIGURE 3.13 Illustration of one-dimensional correlation and convolution.

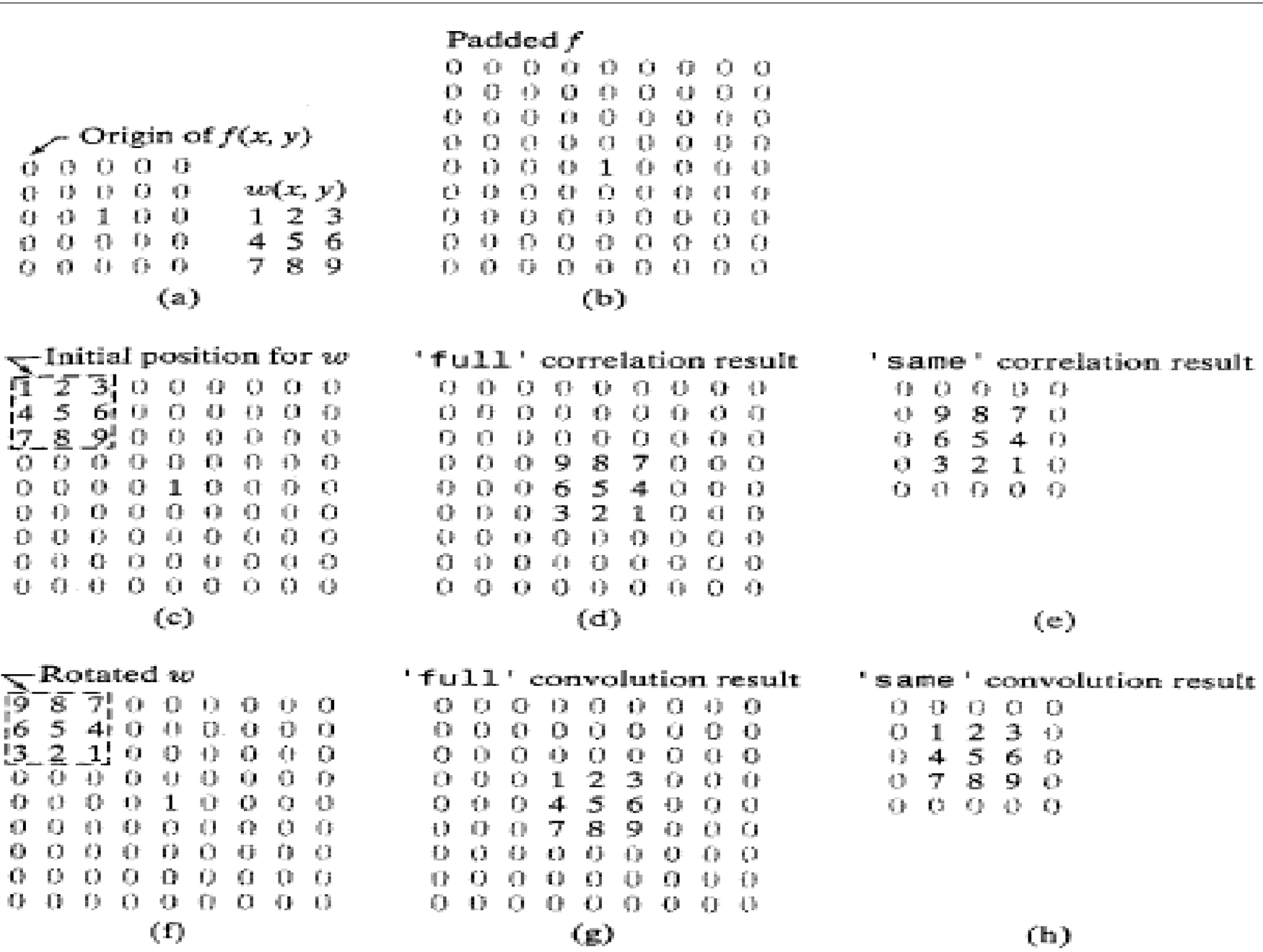


FIGURE 3.14 Illustration of two-dimensional correlation and convolution. The 0s are shown in gray to simplify viewing.

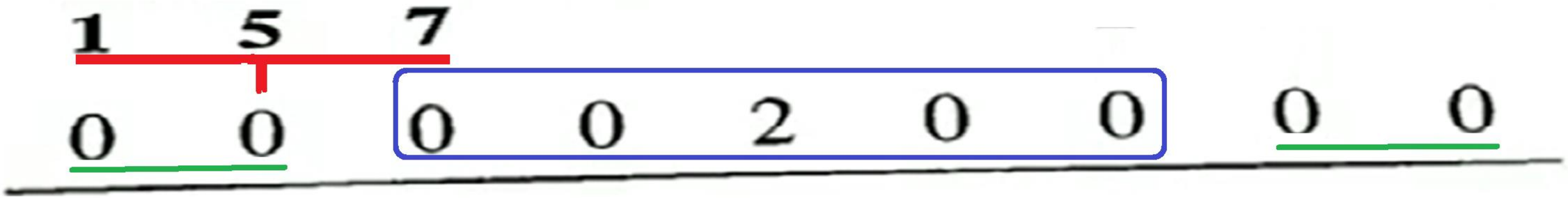
Exercise - Spatial convolution

F = { 0,0,2,0,0} and the mask or Kernel is { 7 5 1 }

Perform the Convolution Process ?

In Convolution Process we have to rotate the Kernel by 180°

Zero padding process for convolution



(1*0)+(5*0)+(7*0) = 0

Convolution process

(a) Initial position

Template

1 5 7

0 0 | 0 0 2 0 0 | 0 0

0

Output is produced in the centre pixel.

(b) Position after one shift

Template is shifted by one bit.

1 5 7

0 0 | 0 0 2 0 0 | 0 0

0 0

Output produced is zero.

(c) Position after two shifts

Template is shifted again.

1 5 7

0 0 | 0 0 2 0 0 | 0 0

0 0 14

Output produced is 14.

(d) Position after three shifts

Template is shifted again.

1 5 7

0 0 | 0 0 2 0 0 | 0 0

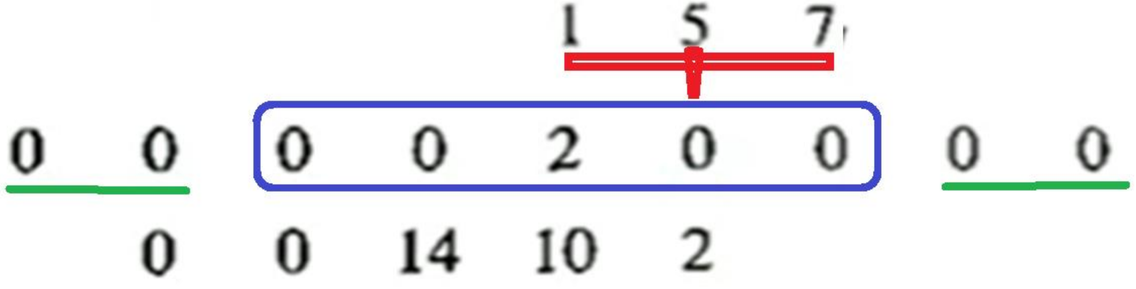
0 0 14 10

Output produced is 10.

(1*0)+(5*0)+(7*2) = 14

(e) Position after four shifts

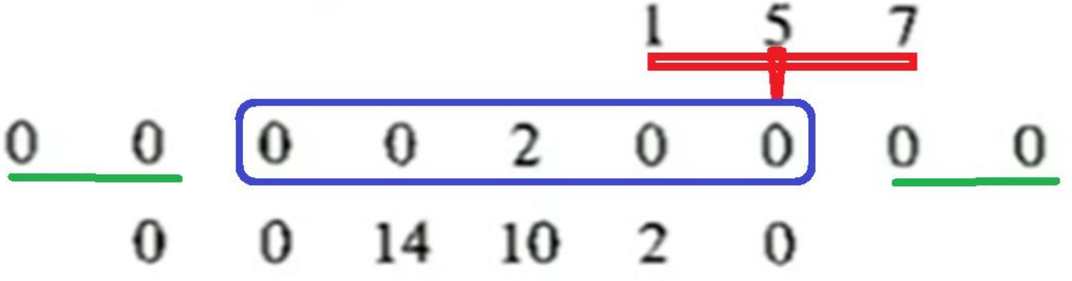
Template is shifted again.



Output produced is 2.

(f) Position after five shifts

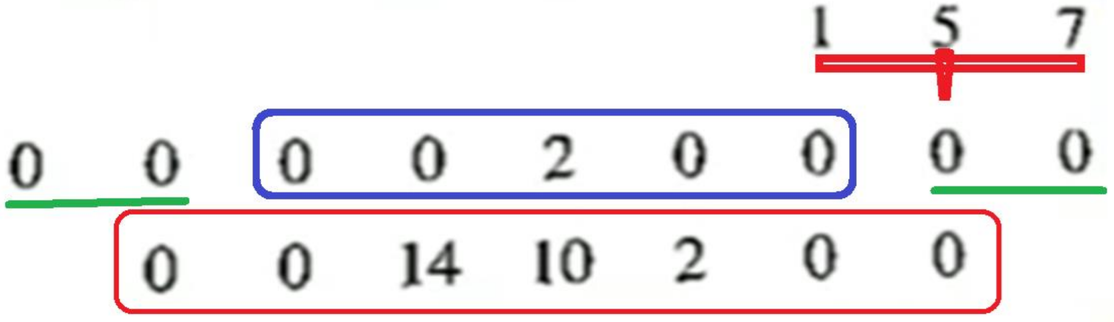
Template is shifted again.



Output produced is 0.

(g) Final position

Template is shifted again.



Output produced is 0. Further shift crosses the range. Hence the process is stopped.

Correlation process

(a) Initial position

Template

7 5 1

0 0 0 0 2 0 0 0 0

0

Output produced is 0.

(b) Position after one shift

Template is shifted by one bit.

7 5 1

0 0 0 0 2 0 0 0 0

0 0

Output produced is 0.

(c) Position after two shifts

Template is shifted again.

7 5 1

0 0 0 0 2 0 0 0 0

0 0 2

Output produced is 2.

(d) Position after three shifts

Template is shifted again.

7 5 1

0 0 0 0 2 0 0 0 0

0 0 2 10

Output produced is 10.

(e) Position after four shifts

Template is shifted again.

				7	5	1		
0	0	0	0	2	0	0	0	0
	0	0	2	10	14			

Output produced is 14.

(f) Position after five shifts

Template is shifted again.

					7	5	1	
0	0	0	0	2	0	0	0	0
	0	0	2	10	14	0		

Output produced is 0.

(g) Final position

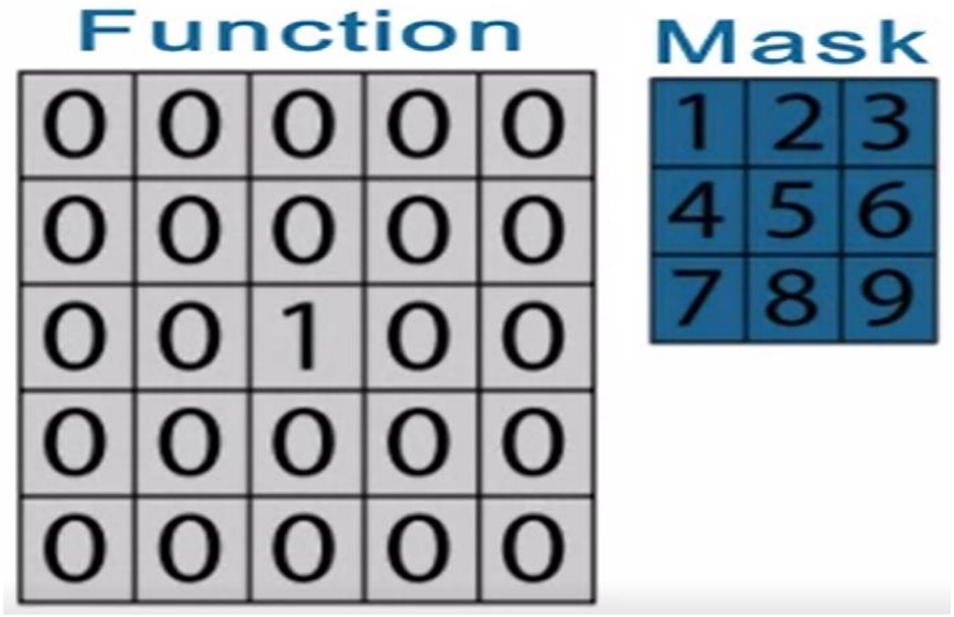
Template is shifted again.

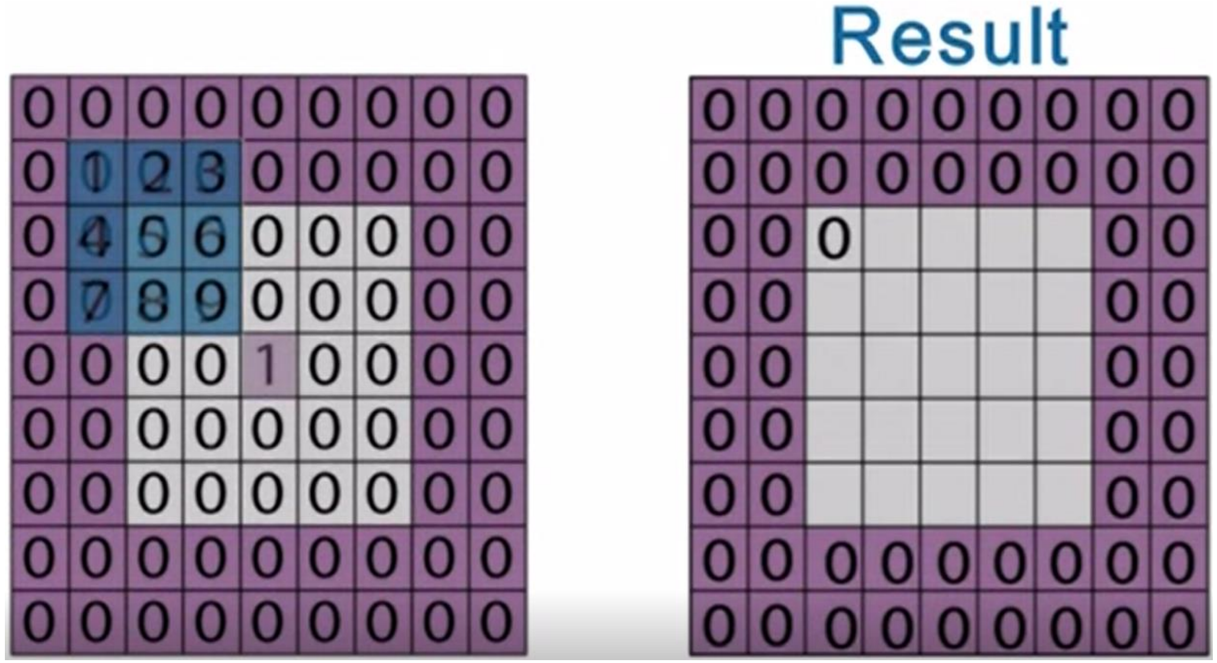
						7	5	1
0	0	0	0	2	0	0	0	0
	0	0	2	10	14	0	0	

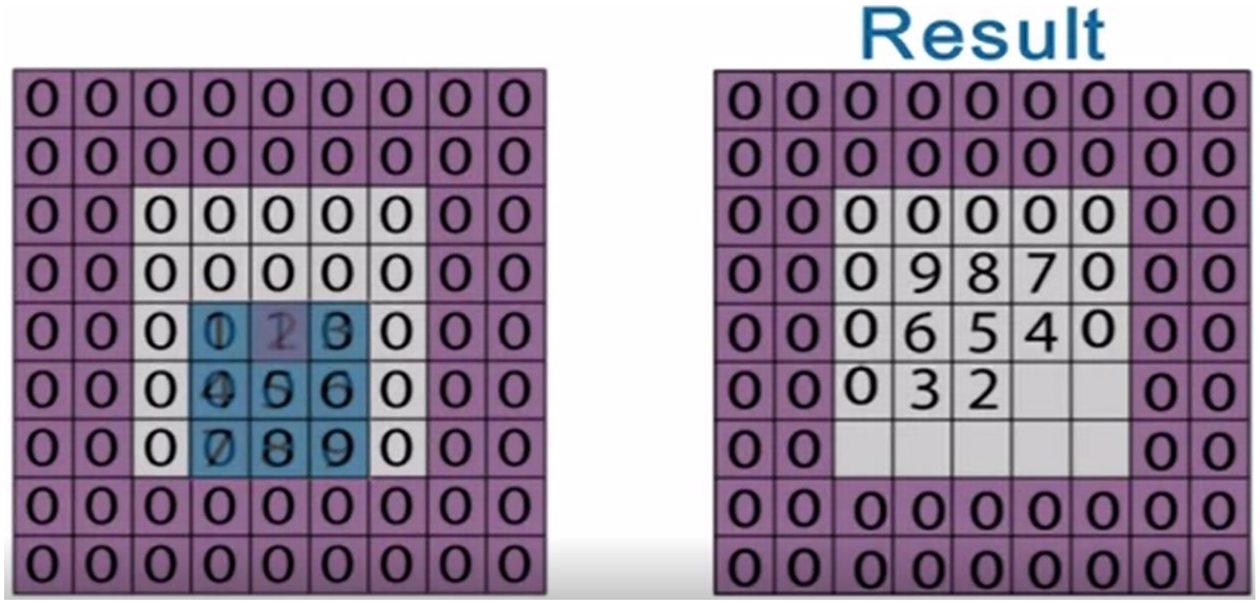
Output produced is 0.

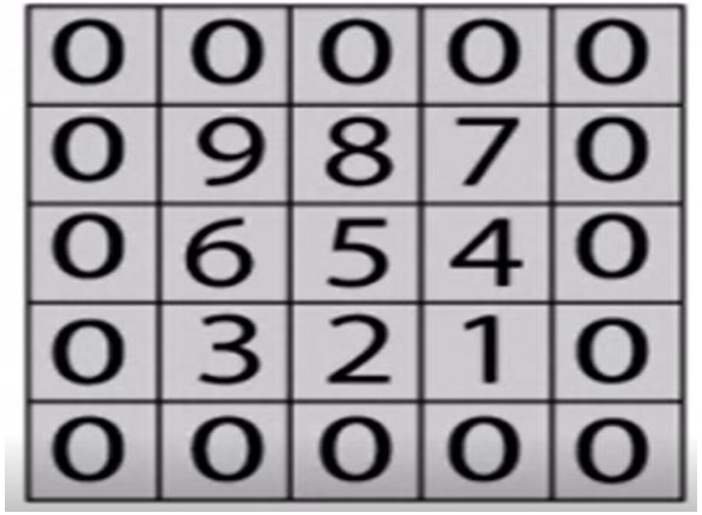
So in the final position, the output produced is [0 0 2 10 14 0 0].

Spatial Correlation









Spatial Convolution

Function

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

Mask

9	8	7
6	5	4
3	2	1

Cropped Convolution Result

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

M x N pixels and G = number of gray levels (G = 2^k)

It is assumed that discrete levels are equally spaced between 0 to G-1 in the gray scale.

Therefore the number of bits required to store a digitized image of size **M x N** is

b = M x N x k .

Q. Find the number of bits required to store a 128 x 128 image with 256 gray levels.

128 x 128 image with 256 gray levels (ie 8 bits/pixel)

128 x 128 x 8 = 131072 bits

= 131072 /8 = 16384 bytes = 17000 bytes

Image resolution

A resolution can be defined as the total number of pixels in an image. Clarity of an image does not depends on number of pixels, but on the spatial resolution of the image.

Gray level resolution

Gray level resolution refers to the predictable or deterministic change in the shades or levels of gray in an image. It is equal to the number of bits per pixel.

Sampling is done on x axis and quantization is done in Y axis.
So that means digitizing the gray level resolution of an image is done in quantization.

Image Segmentation

Aim: To partition an image into a collection of set of pixels

– Meaningful regions (coherent objects)

– Linear structures (line, curve, …)

– Shapes (circles, ellipses,…)

- Content based image retrieval
- Machine Vision
- Medical Imaging applications (tumor delineation,..)
- Object detection (face detection,...)
- 3D Reconstruction
- Object/Motion Tracking
- Object-based measurements such as size and shape
- Object recognition (face recognition,...)
- Fingerprint recognition,
- Video surveillance

Segmentation is an approach for Feature Extraction in an image

Features of Image: Points, lines, edges, corner points, regions, etc

Geometrical attributes: orientation, length, curvature, area, diameter, perimeter, etc

Topological attributes: overlap, adjacency, common end point, parallel, vertical, etc

Image Segmentation

Image Segmentation divides an image into regions that are connected and have some similarity within the region and some difference between adjacent regions.

- The goal is usually to find individual objects in an image.
- For the most part there are fundamentally two kinds of approaches to segmentation: discontinuity and similarity.

– Similarity may be due to pixel intensity, color, texture, histogram, features etc.

– Differences are sudden changes (discontinuities) in any of these, but especially sudden changes in intensity along a boundary line, which is called an edge.

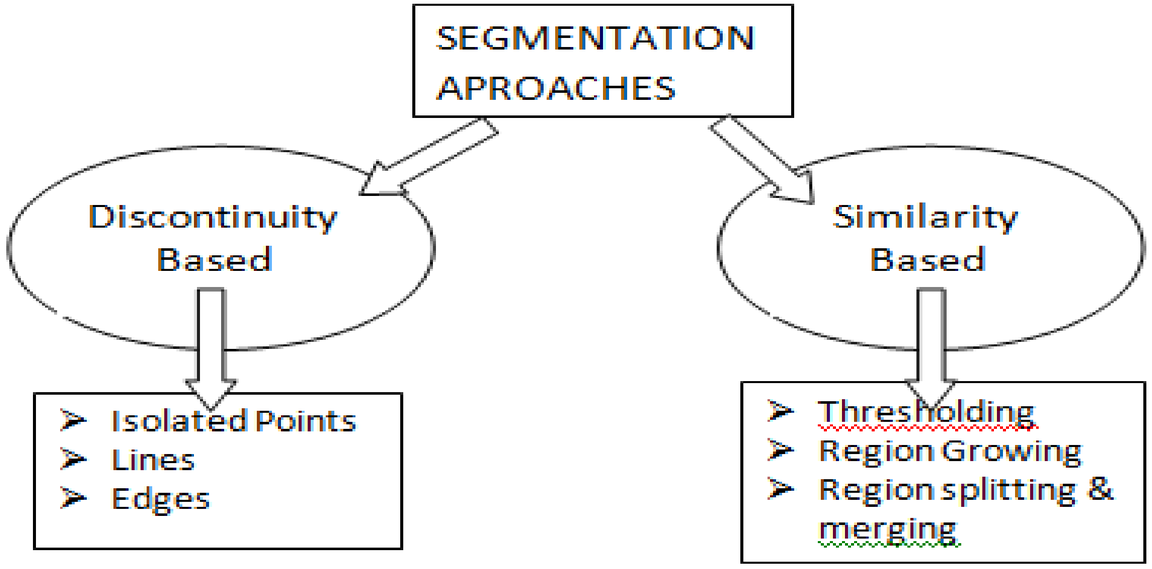


Image Segmentation

Approaches in Image Segmentation

- Similarity approach: This approach is based on detecting similarity between image pixels to form a segment, based on a threshold.
  ML algorithms like clustering are based on this type of approach to segment an image.
- Discontinuity approach: This approach relies on the discontinuity of pixel intensity values of the image.
  Line, Point, and Edge Detection techniques use this type of approach for obtaining intermediate segmentation results which can be later processed to obtain the final segmented image.

Detection of Discontinuities

- There are three kinds of discontinuities of intensity: points, lines and edges.
- The most common way to look for discontinuities is to scan a small mask over the image.
- The mask determines which kind of discontinuity to look for.

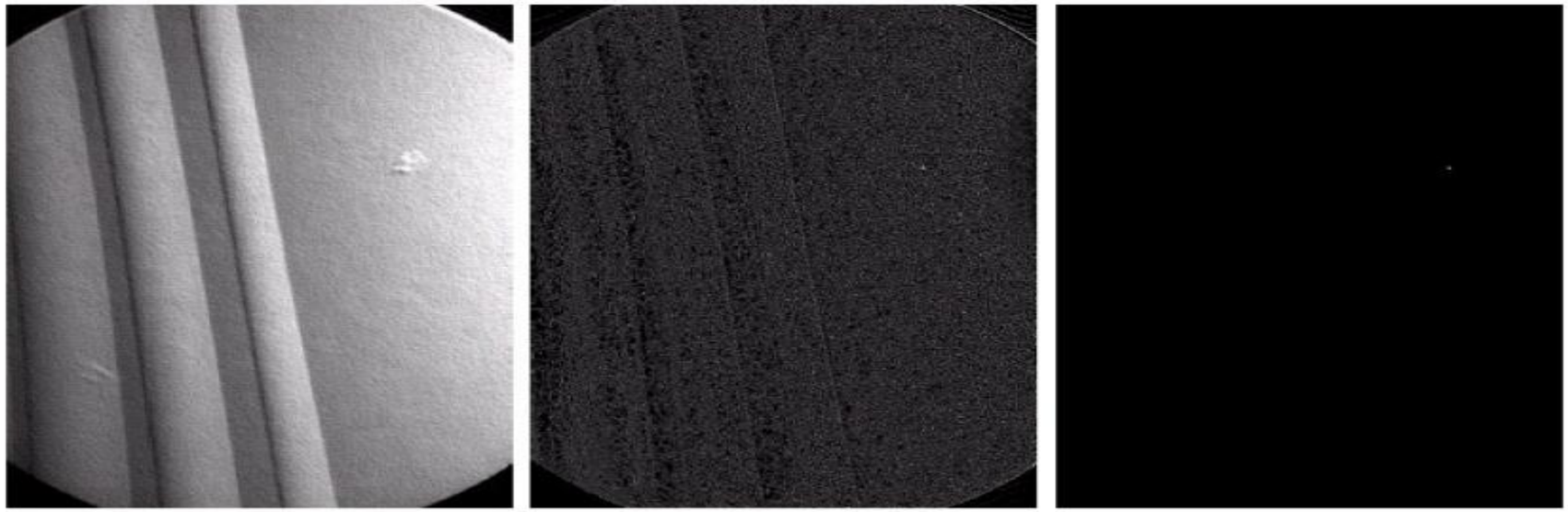
Detection of Discontinuities

Point Detection

$|R| \geq T$ where T : a nonnegative threshold

−1	−1	−1
−1	8	−1
−1	−1	−1

Point detection mask



(b) X-ray image of a turbine blade with a porosity. (c) Result of point detection. (d) Result of using Eq. (10.1-2). (Original image courtesy of X-TEK Systems Ltd.)

Detection of Discontinuities

Line Detection

Only slightly more common than point detection is to find a one pixel wide line in an image.

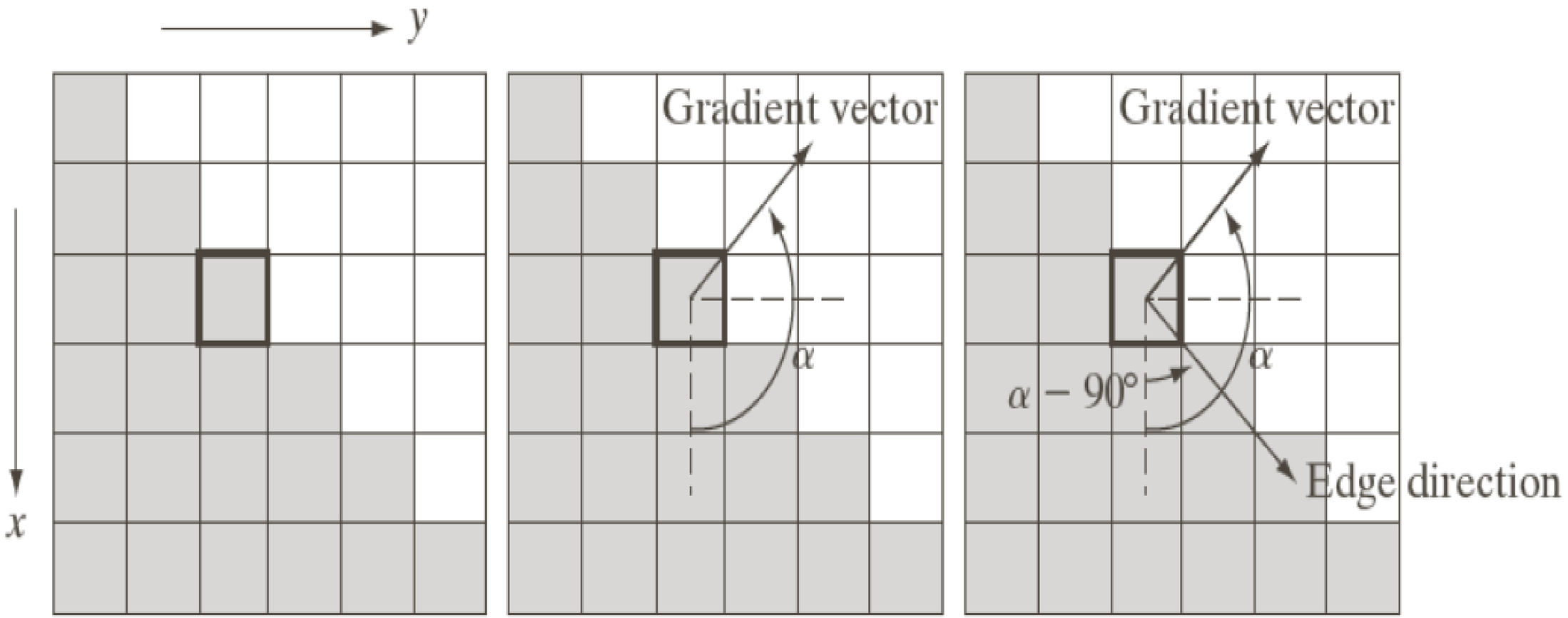
- For digital images the only three point straight lines are only horizontal, vertical, or diagonal (+ or –45°).

Line masks

−1	−1	−1	−1	−1	2	−1	2	−1	2	−1	−1
2	2	2	−1	2	−1	−1	2	−1	−1	2	−1
−1	−1	−1	2	−1	−1	−1	2	−1	−1	−1	2
Horizontal			+45°			Vertical			−45°		

Detection of Discontinuities

Edge Detection - Gradient Operators



Using the gradient to determine edge strength and direction at a point. Note that the edge is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square in the figure represents one pixel.

Detection of Discontinuities

Gradient Operators

−1	0
0	1

0	−1
1	0

Roberts

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Prewitt

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel

Detection of Discontinuities
Gradient Operators

0	1	1	−1	−1	0
−1	0	1	−1	0	1
−1	−1	0	0	1	1

Prewitt

0	1	2	−2	−1	0
−1	0	1	−1	0	1
−2	−1	0	0	1	2

Sobel

Prewitt and Sobel masks for detecting diagonal edges.

Detection of Discontinuities
Gradient Operators

(a) Original image.

(b) $|G_x|$, component of the gradient in x-direction.

(c) $|G_y|$, component in y-direction.

(d) Gradient image, $|G_x| + |G_y|$.

$$\nabla f \approx |G_x| + |G_y|$$

Image Segmentation Techniques

- Threshold Based Segmentation
- Edge Based Segmentation
- Region-Based Segmentation
- Clustering Based Segmentation
- Artificial Neural Network Based Segmentation

Threshold Based Segmentation

❑ Image thresholding segmentation is a simple form of image segmentation.

- It is a way to create a binary or multi-color image, based on setting a threshold value on the pixel intensity of the original image.

In this thresholding process:

- First consider the intensity histogram of all the pixels in the image.
- Then set a threshold to divide the image into sections.

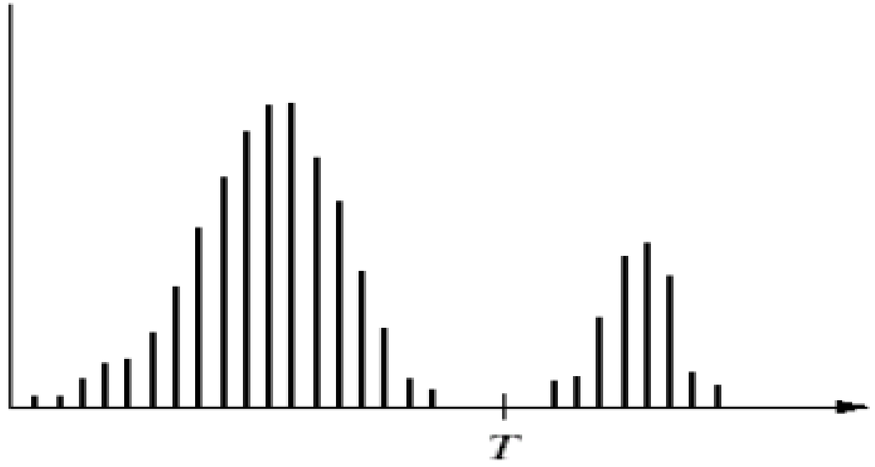
Eg:- Considering image pixels ranging from 0 to 255, we set a threshold of 60. So all the pixels with values less than or equal to 60 will be provided with a value of 0(black) and all the pixels with a value greater than 60 will be provided with a value of 255(white).

Thresholding

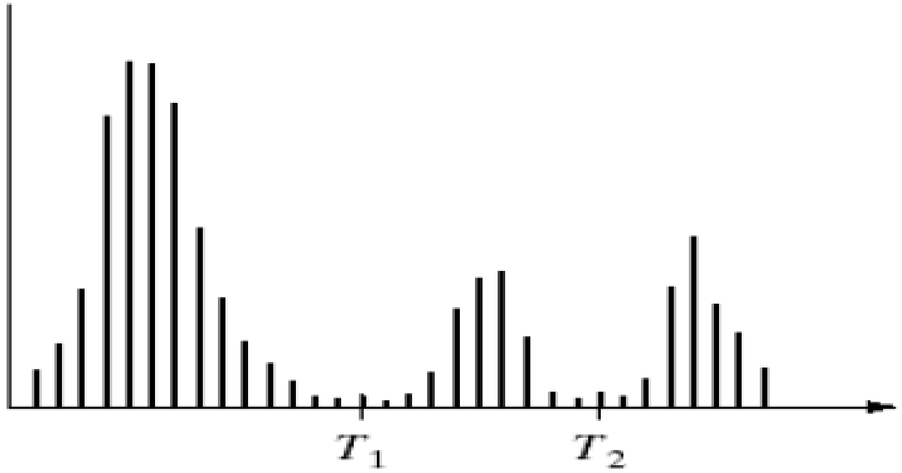
Assumptions:

1. the intensity values are different in different regions
2. within each region, which represents the corresponding object in a scene, the intensity values are similar
3. the range of intensity levels covered by objects of interest is different from the background

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{if } f(x,y) \le T \end{cases}$$



Single threshold

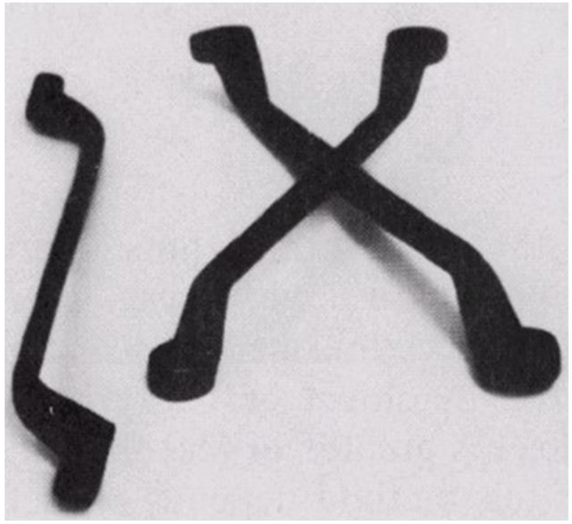


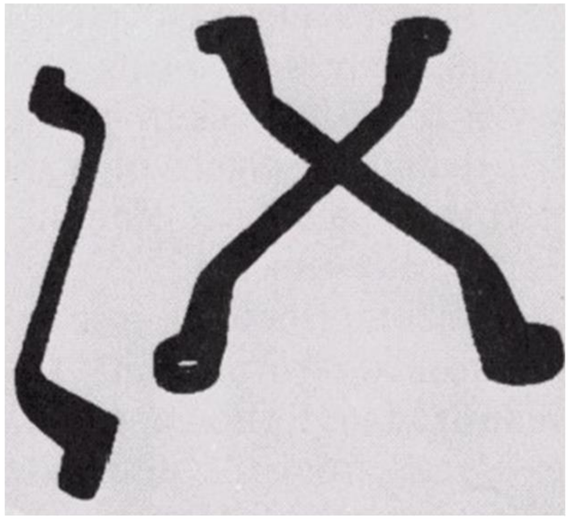
Multiple threshold

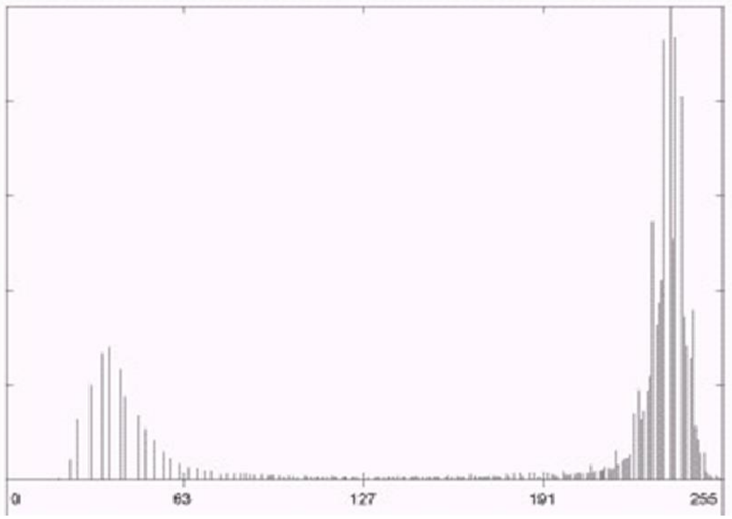
Types of Thresholding

- Global or Fixed Thresholding:
  A threshold value is selected where the threshold value depends only on the pixel intensities in the image. In fixed (or global) thresholding, the threshold value is held constant throughout the image.
- Dynamic or Adaptive or Local thresholding:
  Threshold depends on pixel value and pixel position (both f(x,y) and p(x,y).
  So, the threshold for different pixels in the image will be different. The image is divided into overlapping sections which are thresholded one by one.
- Optimal thresholding: estimate that how much is the error incorporated if we choose a particular threshold. Then, you choose that value of the threshold whereby which your average error will be minimized.

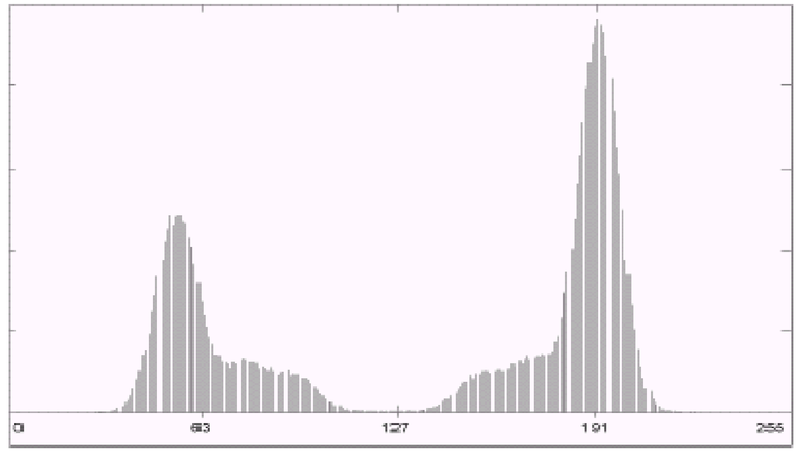
Basic Global Thresholding







Global Thresholding with T midway between the maximum and minimum gray levels

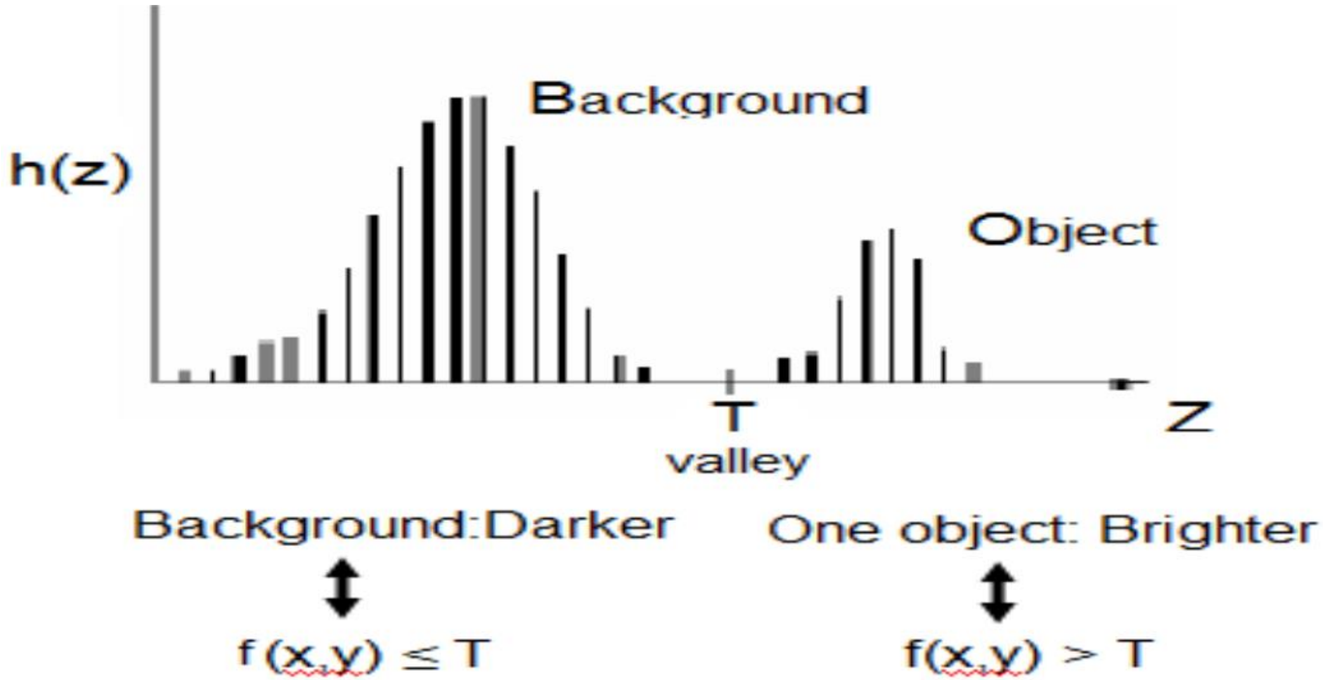


Global Thresholding with T estimated by iteration

IntensityThresholding

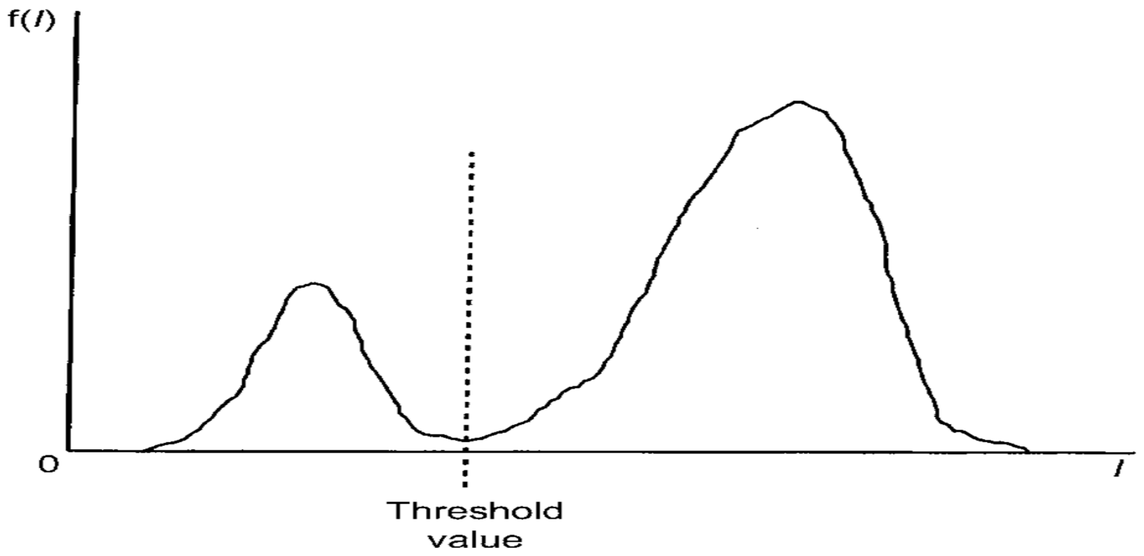
- Image Binarization applies often just one global threshold T for mapping a Scalar image I into a Binary image.

$$J(x,y) = \begin{cases} 0 & \text{if } I(x,y) < T \\ 1 & \text{otherwise.} \end{cases}$$



- The global threshold can be identified by an optimization strategy aiming at creating “large” connected regions and at reducing the number of small-sized regions (*artifacts*).
- Most frequently employed method for determining threshold is based on histogram analysis of intensity level s.

Thresholding



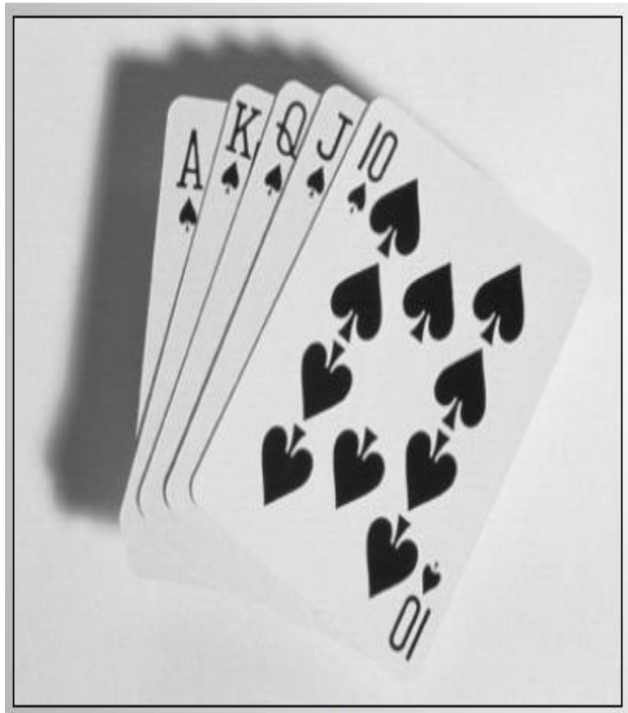
Peak on the left of the histogram corresponds to dark objects

Peak on the right of the histogram corresponds to brighter objects

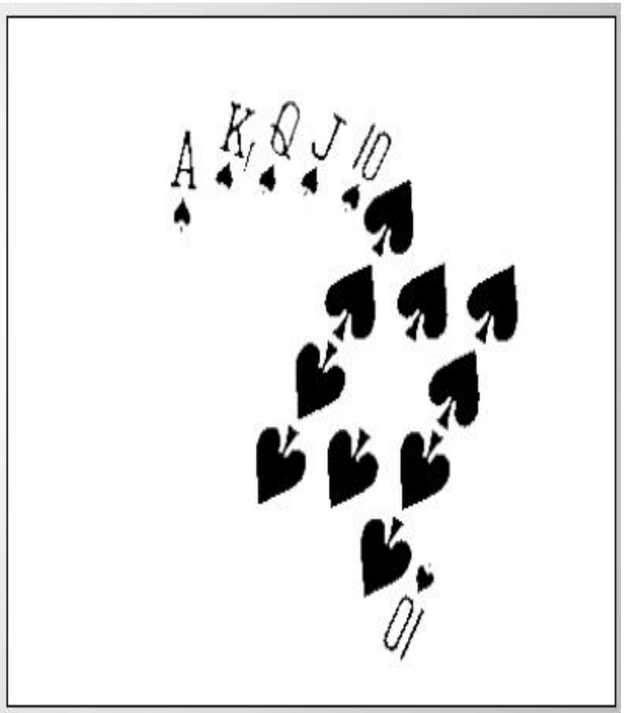
DIFFICULTIES

1. The valley may be so broad that it is difficult to locate a significant minimum
2. Number of minima due to type of details in the image
3. Noise
4. No visible valley
5. Histogram may be multi-modal

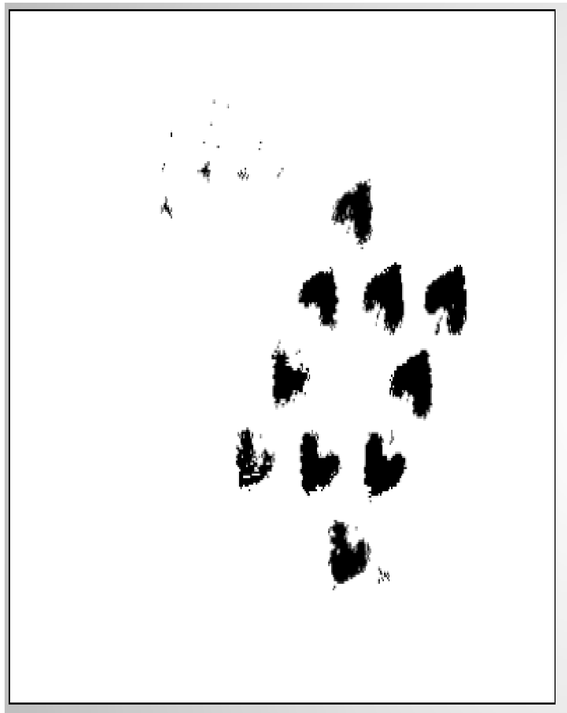
Thresholding



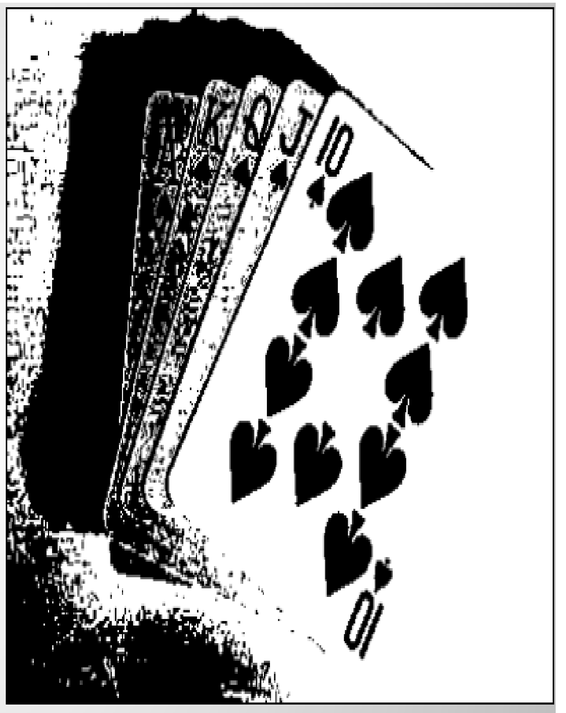
Original Image



Thresholded Image



Threshold Too Low



Threshold Too High

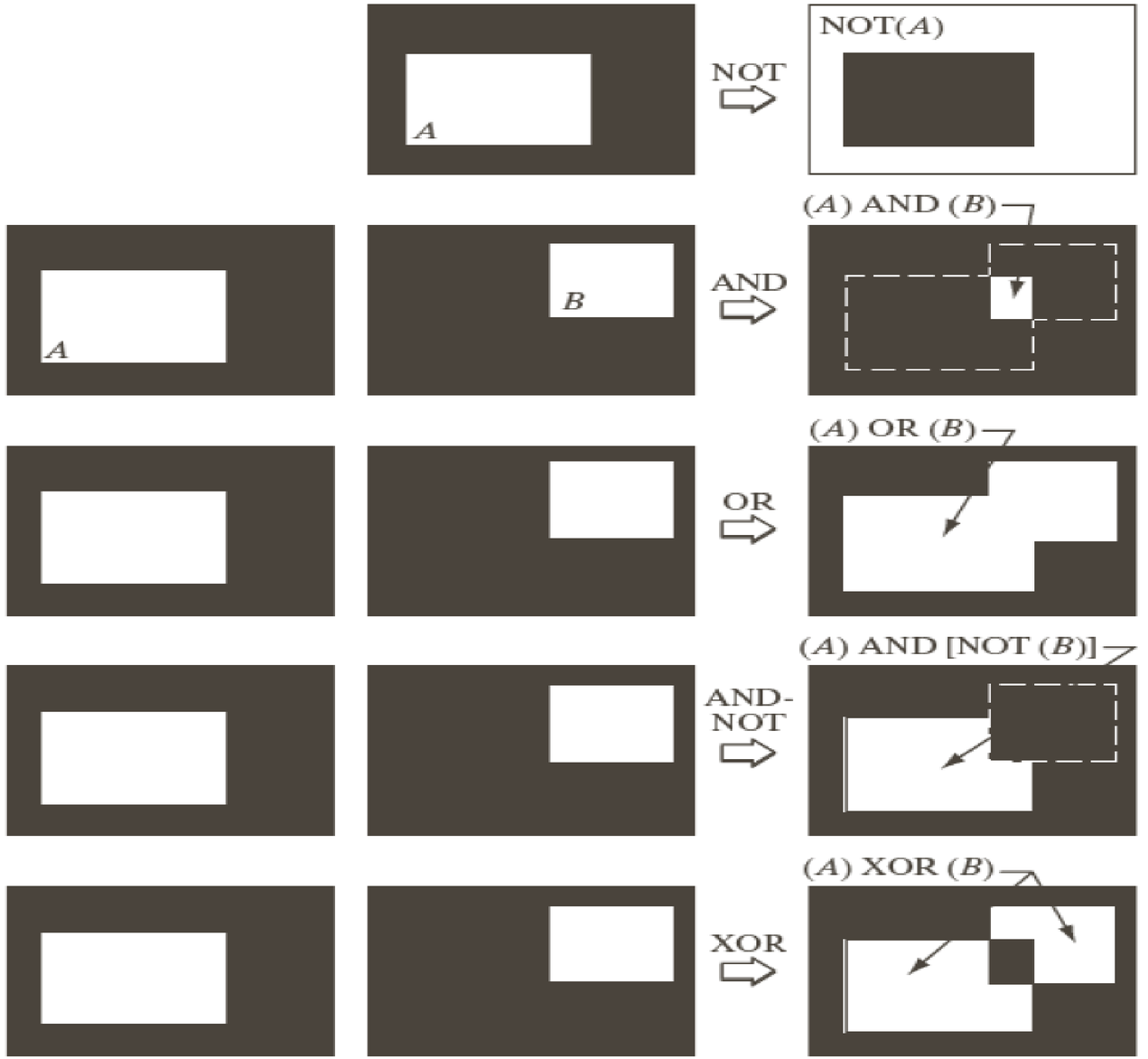
Thresholding

Advantages:

- Simple to implement
- Fast (especially if repeating on similar images)
- Good for some kinds of images (e.g., documents, controlled lighting)

Disadvantages:

- No guarantees of object coherency— may have holes, extraneous pixels, etc.
- (incomplete) solution: post-processing with morphological operators



The output pixels are set of elements not in A.All elements in A become zero and the others to 1

AND operation is the set of coordinates common to A and B

OR *The output pixels belong to either A or B or Both*

A *but not* in B

Exclusive OR *The output pixels belong to either A or B but not to Both*

Region-Based Segmentation - Basic Formulation

- Let R represent the entire image region.
- Segmentation is a process that partitions R into subregions, $R_1, R_2, \ldots, R_n$, such that

$$\text{(a)} \bigcup_{i=1}^{n} R_i = R$$

$$\text{(b)}\ R_i \text{ is a connected region}, i = 1,2,\ldots,n$$

$$\text{(c)}\ R_i \cap R_j = \phi \ \text{for all}\ i \text{ and } j, i \neq j$$

$$\text{(d)}\ P(R_i) = \text{TRUE for } i = 1,2,\ldots,n$$

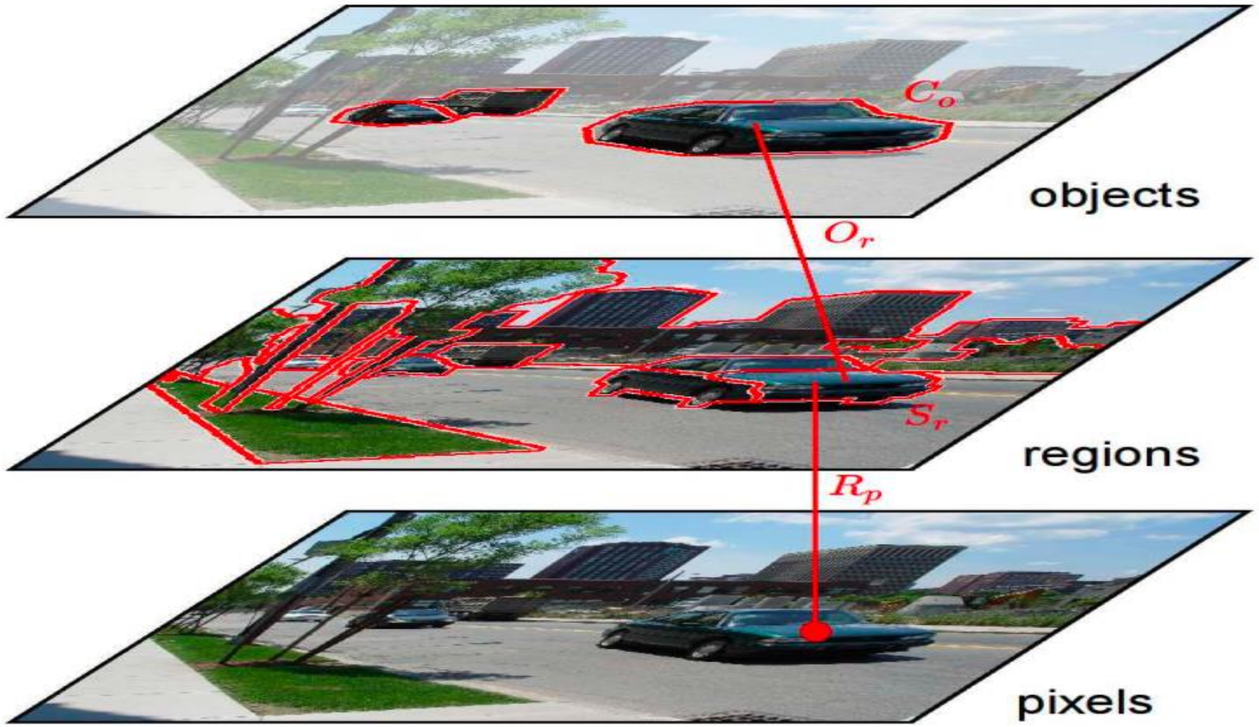
$$\text{(e)}\ P(R_i \cup R_j) = \text{FALSE for any adjacent regions } R_i \text{ and } R_j$$

where $P(R_k)$: a logical predicate defined over the points in set R_k

$P(R_k)$=TRUE if all pixels in R_k have the same gray level.

Region:

- A group of connected pixels with similar properties
- Closed boundaries
- Computation of regions is based on similarity
- Regions may correspond to objects in a scene or parts of objects
- Spatial proximity + Similarity



Region Growing (Seeded Segmentation)

1. Choose the seed pixel
2. Check the neighboring pixels and add them to the region if they are similar to the seed
3. Repeat step 2 for each of the newly added pixels; stop if no more pixels can be added

0	0	5	6	7
1	1	5	8	7
0	1	6	7	7
2	0	7	6	6
0	1	5	6	5

(a)

a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b

(b)

$$|\text{neighboring pixels} - \text{seed}| < Threshold$$

0	0	5	6	7
1	1	5	8	7
0	***1***	6	***7***	7
2	0	7	6	6
0	1	5	6	5

image, 2 seeds

a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b

result for $T = 4$

a	a	a	a	a
a	a	a	a	a
a	a	a	a	a
a	a	a	a	a
a	a	a	a	a

result for $T = 8$

Homogeneity criterion: maximum allowed absolute difference T within region

Region homogeneity

Examples of homogeneity criteria for region R:-

- difference between max and min grey-values in R is small
- difference between any pixel and mean grey-value in R is small
- variance of grey-values in R is small

Segmentation depends on:-

- properties used
- measure of similarity between properties
- similarity variation tolerance (threshold)

Region-Based Segmentation

❖ Goal: find coherent (homogeneous) regions in the image

- Coherent regions contain pixels which share some similar property

❖ Advantages:

- Region-based techniques are generally better in noisy images (where borders are difficult to detect)

❖ Drawbacks:

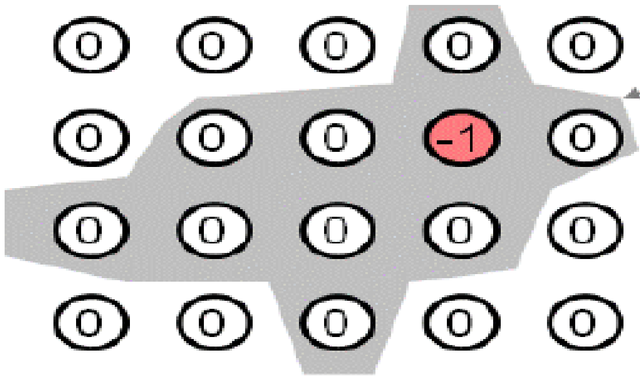
- The output of region-growing techniques is either over segmented (too many regions) or under segmented (too few regions)
- Can’t find objects that span multiple disconnected regions

Note that a complete segmenta:on of an image must sa:sfy a number of criteria:
1) All pixels must be assigned to regions
2)

Region Growing Implementation

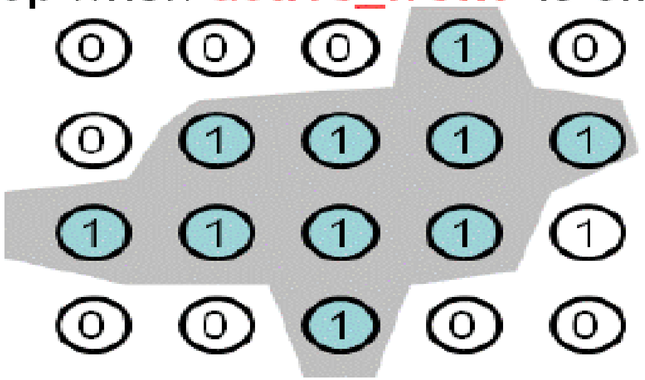
growRegion: red nodes are the "active_front" (queue or stack)

add seed into active_front



object with small *intensity* variation within the image

stop when active_front is empty



ALGORITHM:

Remove pixel *p* from active_front
and mark it as ***region[p]=1*** .
Add all neighbors *q* such that:

$$region[q] == 0,\ |I_p - I_q| < T$$

and set ***region[q]*** $= -1$.