MODULE 5

CST 304- COMPUTER GRAPHICS & IMAGE PROCESSING

SYLLABUS

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.

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.

- The principal **objective** of enhancement is to process an image so that the result is more suitable than the original image for a specific application.
- Image enhancement approaches fall into two broad categories: spatial domain methods and frequency domain methods.
- The term **spatial domain** refers to the image plane itself, and approaches in this category are based on direct manipulation of pixels in an image.
- **Frequency domain** processing techniques are based on modifying the Fourier transform of an image.

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

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

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

- The simplest form of T is when the neighborhood is of size 1*1 (that is, a single pixel).
- In this case, g depends only on the value of f at (x, y), and T becomes a gray-level (also called an intensity or mapping) transformation function of the form

$$s = T(r)$$

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

Some Basic Gray Level Transformations

Image enhancement

- Enhancing an image provides better contrast and a more detailed image as compare to non enhanced image.
- Image enhancement has very applications.
- It is used to enhance medical images, images captured in remote sensing, images from satellite e.t.c
- The transformation function has been given below

$$s = T(r)$$

- where r is the pixels of the input image and s is the pixels of the output image.
- T is a transformation function that maps each value of r to each value of s.
- Image enhancement can be done through gray level transformations.

There are three basic types of functions used frequently for image enhancement:

1. Linear (negative and identity transformations).
2. Logarithmic (log and inverse-log transformations).
3. Power-law (nth power and nth root transformations).

- The identity function is the trivial case in which output intensities are identical to input intensities.

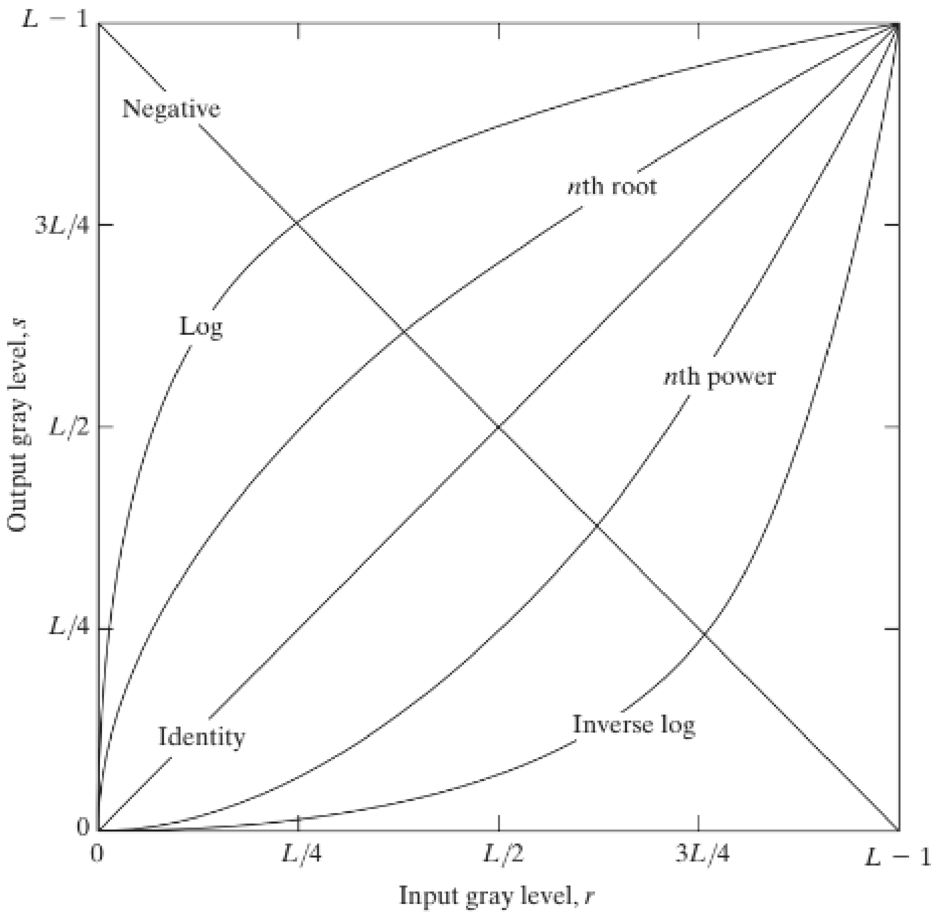
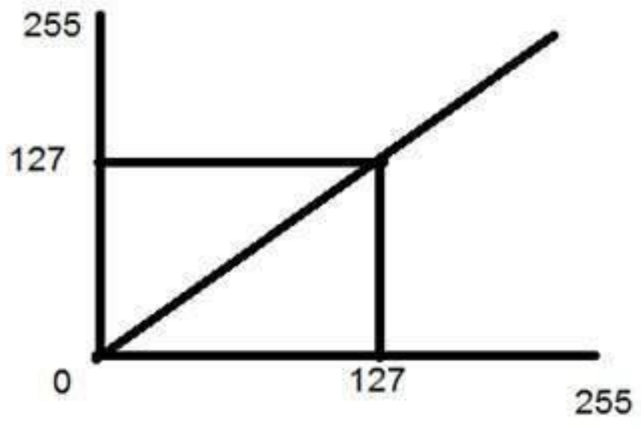


FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.

Linear transformation

- Linear transformation includes simple identity and negative transformation.
- Identity transformation is shown by a straight line.
- In this transition, each value of the input image is directly mapped to each other value of output image.
- That results in the same input image and output image.
- And hence is called identity transformation.

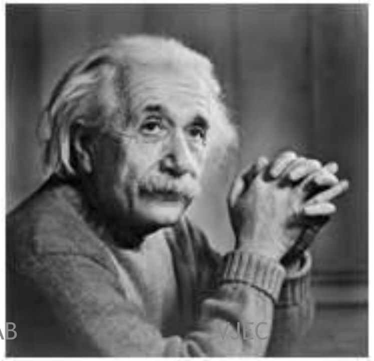
Identity transformation



Negative transformation

- The second linear transformation is negative transformation, which is invert of identity transformation.
- In negative transformation, each value of the input image is subtracted from the L-1 and mapped onto the output image.
- The result is somewhat like this.

INPUT IMAGE



OUTPUT IMAGE

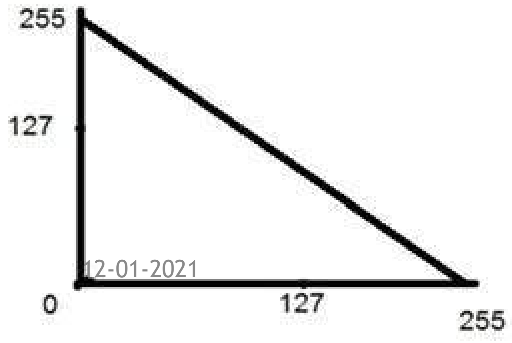
- In this case the following transition has been done.

s = (L – 1) – r

- Since the input image of Einstein is an 8 bpp image, so the number of levels in this image are 256.
- Putting 256 in the equation, we get this

s = 255 – r

- So each value is subtracted by 255 and the result image has been shown above.
- So what happens is that, the lighter pixels become dark and the darker picture becomes light.
- And it results in image negative.
- It has been shown in the graph below.



Logarithmic transformations

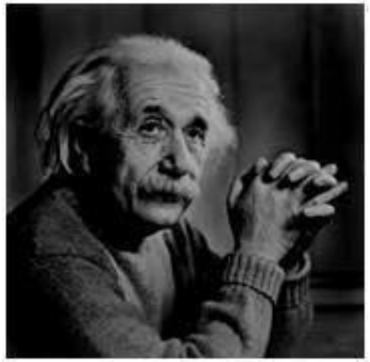
- Logarithmic transformation further contains two type of transformation.
- Log transformation and inverse log transformation.
- The log transformations can be defined by this formula

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

- Where s and r are the pixel values of the output and the input image and c is a constant it is assumed that $r >= 0$.
- The value 1 is added to each of the pixel value of the input image because if there is a pixel intensity of 0 in the image, then log (0) is equal to infinity.
- So 1 is added, to make the minimum value at least 1.

- During log transformation, the dark pixels in an image are expanded as compare to the higher pixel values.
- The higher pixel values are kind of compressed in log transformation.
- This result in following image enhancement.
- The value of c in the log transform adjust the kind of enhancement you are looking for.

INPUT IMAGE



OUTPUT IMAGE

The inverse log transform is opposite to log transform.

Problem

- Enhance the give image using logarithmic transformations for c= 1 and c= L/log10(1+L)

Input image

110	120	90
91	94	98
90	91	99

C=1

r	s
110	Log10(1+110)=2.04= 2
120	Log10(1+120)=2.08= 2
90	Log10(1+90)=1.95= 2
91	Log10(1+91)=1.96= 2
94	Log10(1+94)=1.97= 2
98	Log10(1+98)=1.99= 2
90	Log10(1+90)=2.04= 2
91	Log10(1+91)=1.96= 2
99	Log10(1+99)=2.04= 2

Output image

2	2	2
2	2	2
2	2	2

2^{nd} case

- Take the highest value from the input image 120
- We know that $L= 2^n = 128$

So n= 7

Calculate $c= 128/(\log_{10}(1+128)) = 60.66 = 61$

So **output image – perform round off**

2.04*61	2.08*61	1.95*61
1.96*61	1.97*61	1.99*61
1.96*61	2*61	2*61

Power – Law transformations

- There are further two transformation is power law transformations, that include nth power and nth root transformation.
- These transformations can be given by the expression:

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

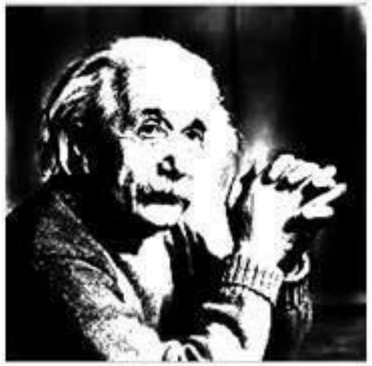
- This symbol γ is called gamma, due to which this transformation is also known as gamma transformation.
- Variation in the value of γ varies the enhancement of the images.

- Different display devices / monitors have their own gamma correction, that’s why they display their image at different intensity.
- This type of transformation is used for enhancing images for different type of display devices.
- The gamma of different display devices is different.
- For example Gamma of CRT lies in between of 1.8 to 2.5, that means the image displayed on CRT is dark.

- The same image but with different gamma values has been shown here.

GAMMA=10

GAMMA=8



GAMMA=6

Problem

- Enhance the give image using power law transformations for c= 1 and γ =0.2

Input image

110	120	90
91	94	98
90	91	99

We know that

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

r	s
110	$110^{0.2}$ =
120	$120^{0.2}$ =
90	$90^{0.2}$ =
91	$91^{0.2}$ =
94	$94^{0.2}$ =
98	$98^{0.2}$ =
90	$90^{0.2}$ =
91	$91^{0.2}$ =
99	$99^{0.2}$ =

Output image

3	3	2
2	2	3
2	2	3

Contrast Stretching

- Contrast stretching is an Image Enhancement method which attempts to improve an image by stretching the range of intensity values.
- Here, we stretch the minimum and maximum intensity values present to the possible minimum and maximum intensity values.
- **Example:** If the minimum intensity value(r min) present in the image is 100 then it is stretched to the possible minimum intensity value 0.

- Likewise, if the maximum intensity value(r max) is less than the possible maximum intensity value 255 then it is stretched out to 255.
- (0–255 is taken as standard minimum and maximum intensity values for 8-bit images)
- **Note:** Contrast stretching is only possible if minimum intensity value and maximum intensity value are not equal to the possible minimum and maximum intensity values.
- Otherwise, the image generated after contrast stretching will be the same as input image.

- **General Formula for Contrast Stretching:**

$$s = (r - r_{min}) \frac{(I_{max} - I_{min})}{(r_{max} - r_{min})} + I_{min}$$

For I min = 0 and I max = 255 (for standard 8-bit grayscale image)

$$s = 255 \times \frac{(r - r_{min})}{(r_{max} - r_{min})}$$

where,

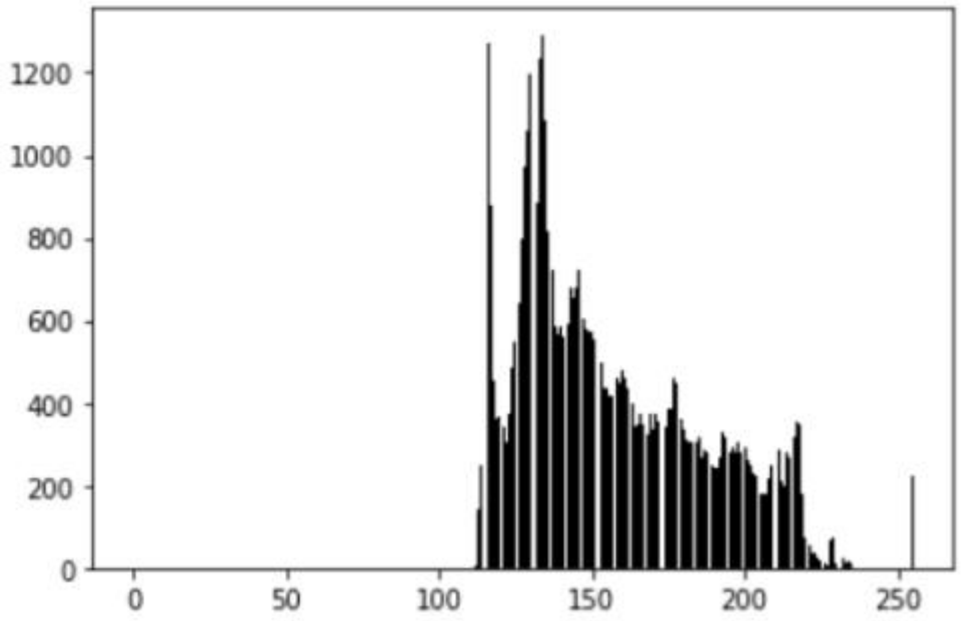
r = current pixel intensity value

r min = minimum intensity value present in the whole image

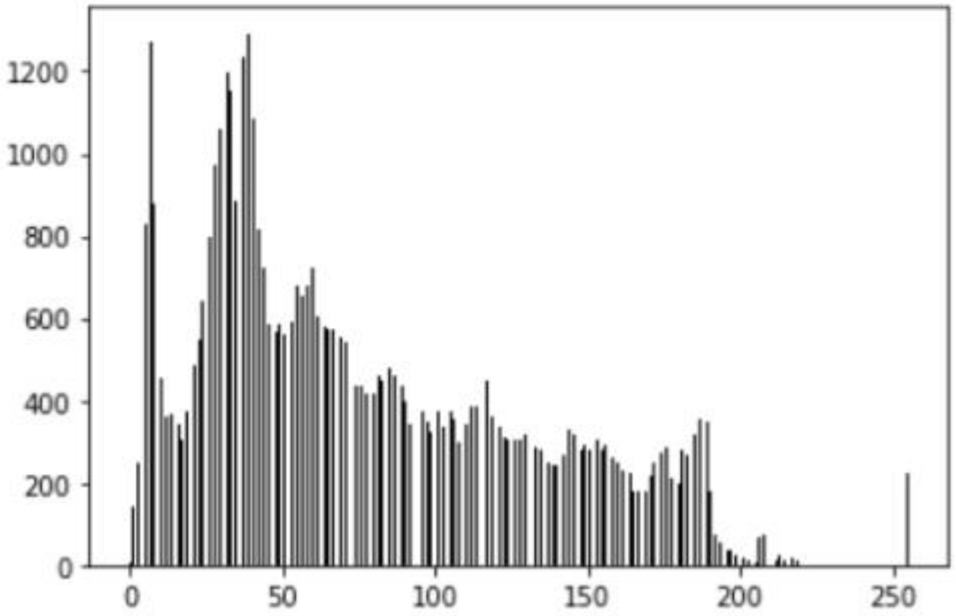
r max = maximum intensity value present in the whole image

Note: Output intensity value s should be rounded up to nearest integer value.

Input Image before Contrast Stretching with its histogram



Input Image after Contrast Stretching with its histogram



Histogram equalization

- **Histogram**
- Histogram is a graphical representation of the intensity distribution of an image.
- In simple terms, it represents the number of pixels for each intensity value considered.
- **Histogram Equalization**
- Histogram Equalization is a computer image processing technique used to improve contrast in images.
- It accomplishes this by effectively spreading out the most frequent intensity values, i.e. stretching out the intensity range of the image.

- This method usually increases the global contrast of images when its usable data is represented by close contrast values.
- This allows for areas of lower local contrast to gain a higher contrast.
- **Histogram** of an image can be drawn plotting pixel intensities Vs frequency of the pixel intensities or probability of the pixel intensity.

Use of Histogram Equalization

- In digital image processing, the contrast of an image is enhanced using this technique.

- It is used to increase the spread of the histogram.

- If the histogram represents the digital image, then by spreading the intensity values over a large dynamic range we can improve the contrast of the image.

Algorithm

1. Find the frequency of each value represented on the horizontal axis of the histogram i.e. intensity in the case of an image.
2. Calculate the probability density function for each intensity value.
3. After finding the PDF, calculate the cumulative density function for each intensity's frequency.
4. The CDF value is in the range 0-1, so we multiply all CDF values by the largest value of intensity.
5. Round off the final values to integer values.

Problem

- **Perform the histogram equalization for an 8*8 image shown below.**

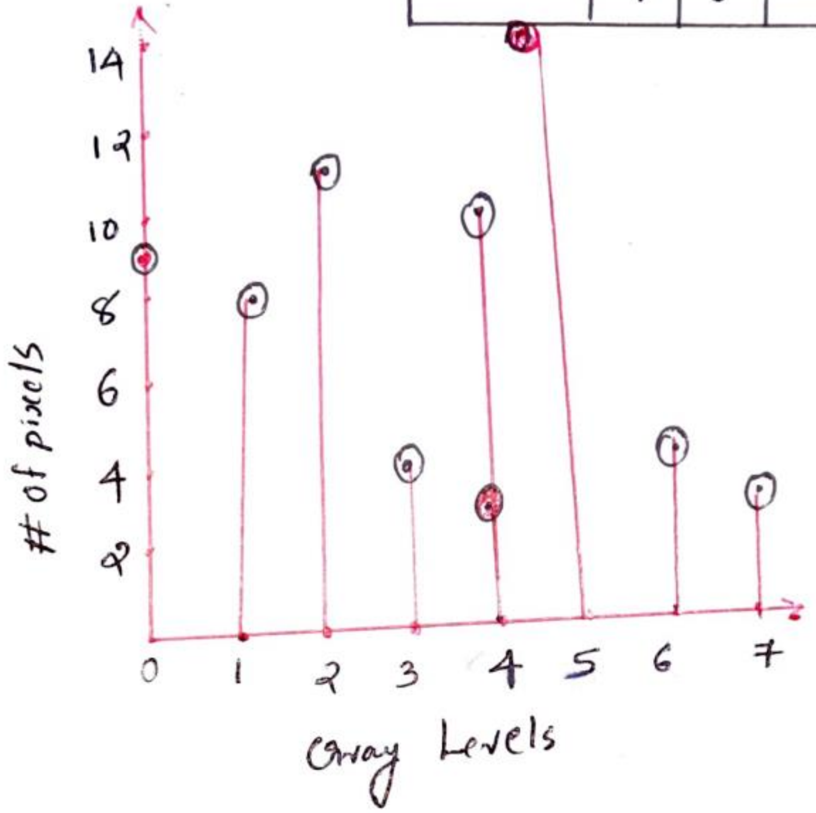
Gray levels	0	1	2	3	4	5	6	7
No. of pixels	9	8	11	4	10	15	4	3

Plot the histogram for the input image

- X axis : gray levels (0,1,2,3,4,5,6,7)
- Y axis : No. of pixels in each level

Gray levels	0	1	2	3	4	5	6	7
No. of pixels	9	8	11	4	10	15	4	3

Gray levels	0	1	2	3	4	5	6	7
# of pixels	9	8	11	4	10	15	4	3



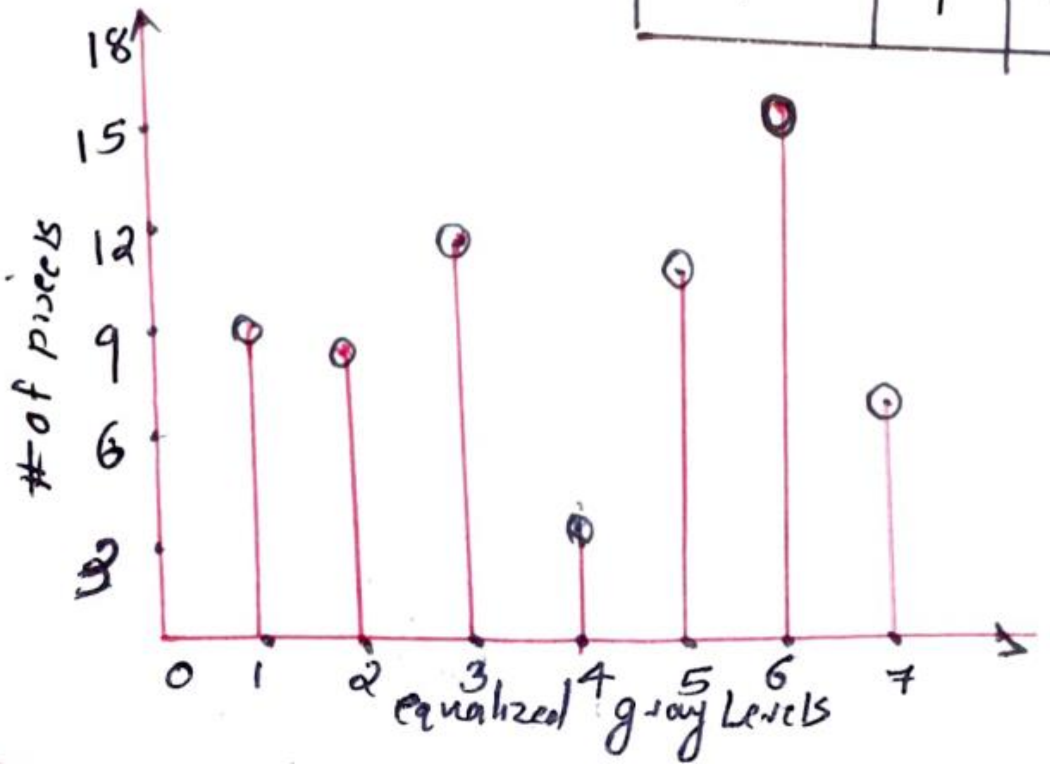
Gray levels(rk)	No. of pixels (nk)	Probability density function(PDF) nk/n	Cumulative density function(CDF) Sk	Multiply with largest intensity(7) Sk*7	Histogram equalized levels (round off)
0	9	9/64	9/64	0.9843	1
1	8	8/64	17/64	1.859	2
2	11	11/64	28/64	3.062	3
3	4	4/64	32/64	3.5	4
4	10	10/64	42/64	4.593	5
5	15	15/64	57/64	6.234	6
6	4	4/64	61/64	6.671	7
7	3	3/64	1	7	7

n=64

Plot histogram for the equalized image

Equalized gray levels	1	2	3	4	5	6	7
No.of pixels	9	8	11	4	10	15	7 (4+3)

Equalized gray Level	1	2	3	4	5	6	7
# of pixels	9	8	11	4	10	15	7



Problem 2

Perform histogram equalization for the following image

f(x,y)

1	2	1	1	1
2	5	3	5	2
2	5	5	5	2
2	5	3	5	2
1	1	1	2	1

Here maximum intensity is 5

We know that $L = 2^n$

$2^0 = 1$

$2^1 = 2$

$2^2 = 4$

$2^3 = 8$

So L= 8

(0 to 7)

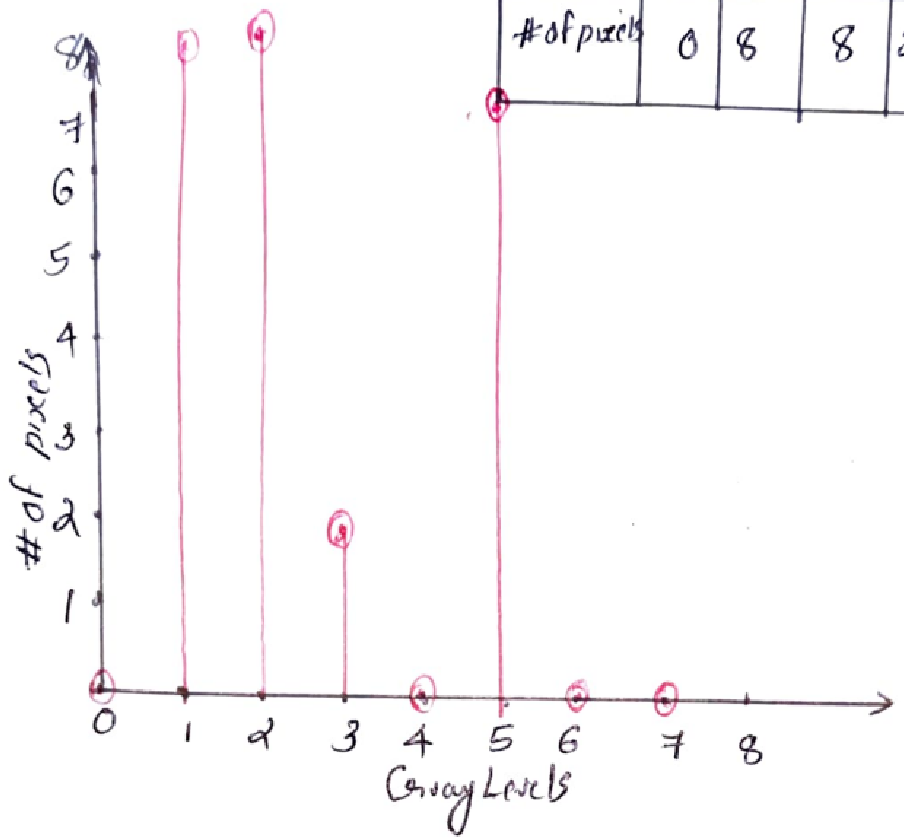
Solution

Gray levels (rk)	0	1	2	3	4	5	6	7
Number of pixels(nk)	0	8	8	2	0	7	0	0

Plot the histogram for the input image

- X axis : gray levels (0,1,2,3,4,5,6,7)
- Y axis : No. of pixels in each level

Gray Levels	0	1	2	3	4	5	6	7
# of pixels	0	8	8	2	0	7	0	0



Gray levels(rk)	No. of pixels (nk)	Probability density function(PDF) nk/n	Cumulative density function(CDF) Sk	Multiply with largest intensity(7) Sk*7	Histogram equalized levels (round off)
0	0	0	0	0	0
1	8	8/25	8/25	2.24	2
2	8	8/25	16/25	4.48	4
3	2	2/25	18/25	5.04	5
4	0	0	18/25	5.04	5
5	7	7/25	25/25= 1	7	7
6	0	0	1	7	7
7	0	0	1	7	7

n=25

1	2	1	1	1
2	5	3	5	2
2	5	5	5	2
2	5	3	5	2
1	1	1	2	1

Input image

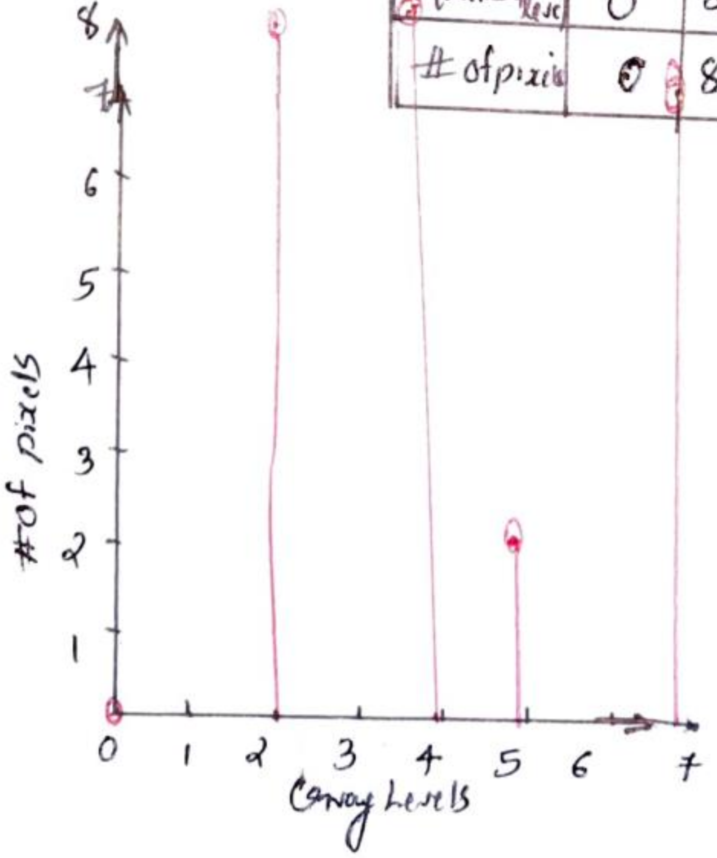
2	4	2	2	2
4	7	5	7	4
4	7	7	7	4
4	7	5	7	4
2	2	2	4	2

Output image

Gray levels (rk)	0	2	4	5	7
Number of pixels(nk)	0	8	8	2	7

Plot the histogram for the equalized image

equalized level	0	2	4	5	7
# of pixels	0	8	8	2	7



Homework

- Perform histogram equalization for the given image and also draw the histogram for the input and equalized image

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

Fundamental of spatial filtering

- The name filter is borrowed from frequency domain processing where filtering refers to accepting (passing) or rejecting certain frequency components.
- Filtering used to modify or enhancing an image.

Filters in frequency domain:

- Low pass filter: passes low frequencies: used for smoothing (blurring) on the image.
- High pass filter: passes high frequencies: used for sharpening the image.
- Band pass: passes frequencies within a band.
- Band reject: reject frequencies within a band.

- Many image enhancement techniques are based on spatial operations which are performed on **local neighbourhood of an input pixels.**

Filters in spatial domain:

- Spatial filters used different masks which are also known as kernels, templates or windows.

- There is a one-to-one correspondence between linear spatial filters and filters in frequency domains.

- **Spatial filters** can be classified into two types based on the basis nature of responses.

- Linear and nonlinear filtering. (Frequency domain filters just for linear filtering).

- Filtering creates a new pixel with coordinates equal to the coordinates of the center of the neighborhood, and whose value is the result of the filtering operations.

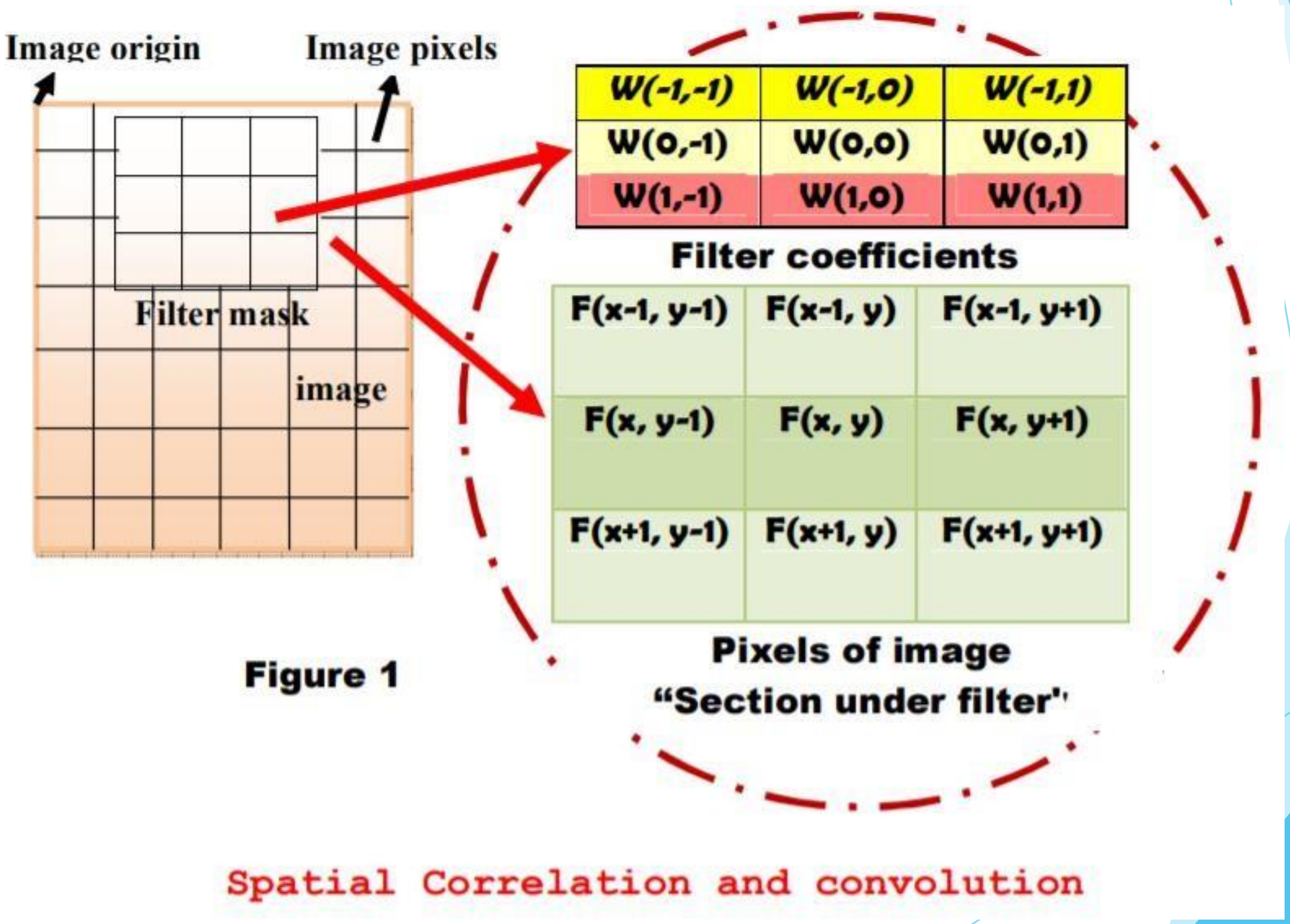
- If the operation performed on the image pixels is linear then the filter is called a linear spatial filter other wise the filter is non linear.

Spatial filtering

- The subimage is called a **filter, mask, kernel, template, or window,** with the first three terms being the most prevalent terminology.
- The values in a filter subimage are referred to as **coefficients,** rather than pixels.
- The concept of filtering has its roots in the use of the Fourier transform for signal processing in the so-called frequency domain.

- Here ,we are interested in filtering operations that are performed directly on the pixels of an image.
- We use the term spatial filtering to differentiate this type of process from the more traditional frequency domain filtering.

- Consider linear spatial filtering using a 3*3 neighborhood.
- At any point (x, y) in the image, the response **g(x, y)** of the filter is the sum of products of the filter coefficients and the image pixels values.



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

Observe that the center coefficient of the filter, $w(0,0)$ aligns with the pixel at location(x,y).

General mask of size $m * n$:

Assume that

$$m = 2a + 1$$

$$\textit{and}$$

$$n = 2b + 1$$

(where a, b are **positive integers**).

(Odd filters)

In general, linear spatial filtering of an image of size $M * N$ with a filter 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 x and y are varied so that each pixel in w visits every pixel in f.

Spatial Filtering techniques

- Spatial Filtering technique is used directly on pixels of an image.
- Mask is usually considered to be added in size so that it has specific center pixel.
- This mask is moved on the image such that the center of the mask traverses all image pixels.

Classification on the basis of linearity:

There are two types:

1. Linear Spatial Filter
2. Non-linear Spatial Filter

General Classification:

- **Smoothing Spatial Filter:** Smoothing filter is used for blurring and noise reduction in the image.
- Blurring is pre-processing steps for removal of small details and Noise Reduction is accomplished by blurring.
- **Types of Smoothing Spatial Filter:**

1. Linear Filter (Mean Filter)
2. Order Statistics (Non-linear) filter

Mean Filter

- Linear spatial filter is simply the **average of the pixels contained in the neighborhood of the filter mask.**
- The idea is replacing the value of every pixel in an image by the average of the grey levels in the neighborhood define by the filter mask.

Types of Mean filter:

i) Averaging filter(mean filter /box filter /average filter):

- It is used in reduction of the detail in image. All coefficients are equal.

(ii) Weighted averaging filter:

- In this, pixels are multiplied by different coefficients.
- Center pixel is multiplied by a higher value than average filter.

Order Statistics Filter:

- It is based on the ordering the pixels contained in the image area encompassed by the filter.
- It replaces the value of the center pixel with the value determined by the ranking result.
- Edges are better preserved in this filtering.
- **Types of Order statistics filter:**

  i) Minimum filter:

  - 0th percentile filter is the minimum filter.
  - The value of the center is replaced by the smallest value in the window.

(ii) Maximum filter:

- 100th percentile filter is the maximum filter.
- The value of the center is replaced by the largest value in the window.

(iii) Median filter:

- Each pixel in the image is considered.
- First neighboring pixels are sorted and original values of the pixel is replaced by the median of the list.

Box filter / Mean / average filter

- It is used for blurring and noise reduction.
- Blurring – removal of small details from an image , prior to object extraction(Preprocessing)
- Noise reduction – can be implemented by blurring with linear/non linear filter.

Mask = 1/9 *

1	1	1
1	1	1
1	1	1

- It is an example for 3*3 filter .
- Replacing each pixel in an image by the average gray level value of the filter mask.
- Generally m*n mask is normalized to 1/m*n

Application

- Noise reduction

Side effect

- It produce blurred edges .(edges are characterized by the sharp transition in gray levels)

Problem 1

- A 5x5 image patch is shown below. Compute the value of the pixel at the position (3x3) if it is smoothened by a 3x3 average filter.
- Pixel value at the position (3x3) = 5

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 2 \\ 5 & 6 & 7 & 8 & 4 \\ 4 & 3 & 2 & 1 & 2 \\ 8 & 7 & 6 & 5 & 3 \\ 1 & 5 & 3 & 7 & 6 \end{pmatrix}$$

1/9 *

1	1	1
1	1	1
1	1	1

2	1	2
6	5	3
3	7	6

$$1/9*[1*2+1*1+1*2+1*6+1*5+1*3+1*3+1*7+1*6] = 35/9=3.88= 4$$

So the pixel value 5 is changed to 4

Problem 2

- A 4x4 image patch is shown below. Compute the value of the pixel at the position (2x2) if it is smoothened by a 2x2 average filter.

$$\begin{pmatrix} 0 & 1 & 2 & 3 \\ 5 & 6 & 7 & 8 \\ 4 & 3 & 2 & 1 \\ 8 & 7 & 6 & 5 \end{pmatrix}$$

Pixel value at the position (2*2) = 2

1/4	1	1		2	1
	1	1		6	5

1/4[1*2+1*1+1*6+1*5]= ¼[14] = 7

So the pixel value 2 is changed to 7

Weighted average filter

- In this, pixels are multiplied by different coefficients.
- Center pixel is multiplied by a higher value than average filter.

Mask = 1/16 *

1	2	1
2	4	2
1	2	1

- The general implementation for filtering an M*N image with a weighted averaging filter of size m*n (m and n odd) is given by the expression

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

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

Application

- This is used to reduce blurring during smoothing process.

Problem 1

- A 5x5 image patch is shown below. Compute the value of the pixel at the position (3x3) if it is smoothened by a 3x3 weighted average filter.
- Pixel value at the position (3x3) = 5

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 2 \\ 5 & 6 & 7 & 8 & 4 \\ 4 & 3 & 2 & 1 & 2 \\ 8 & 7 & 6 & 5 & 3 \\ 1 & 5 & 3 & 7 & 6 \end{pmatrix}$$

1/16 *	1	2	1		2	1	2
	2	4	2		6	5	3
	1	2	1		3	7	6

1/16*[1*2+2*1+1*2+2*6+4*5+2*3+1*3+2*7+1*6] = 67/16=4.18= 4

So the pixel value 5 is changed to 4

Problem 2

- A 4x4 image patch is shown below. Compute the value of the pixel at the position (2x2) if it is smoothened by a 2x2 weighted average filter

$$\begin{pmatrix} 0 & 1 & 2 & 3 \\ 5 & 6 & 7 & 8 \\ 4 & 3 & 2 & 1 \\ 8 & 7 & 6 & 5 \end{pmatrix}$$

Order statistics filter / Non linear filters

- Order static filters are the non linear filter whose response is based on the ordering or ranking the pixels in the image encompassed by the filter.
- These filters replaces the value of the center pixel value determined by ranking result.

Median filter

- Replace the value of a pixel by the median of the gray level in the neighborhood of that pixel.

Application

- Most popular filter.
- Excellent noise reduction capability.
- Less blurring compared to linear spatial filter.
- Most effective for impulse noise.(meaning salt and pepper noise)
- Salt and pepper noise means , white and black noise on image.

Problem

- Apply median filters for the given image

10	20	20
20	15	20
20	25	100

Rank pixels in the increasing value of intensity

10,15,20,20,**20**,20,20,25,100

Find the median = 20

That means the pixel 15 is replaced by 20

Max filters

- It is used for finding brightest point in image.

R= MAX $\{Z_K | K=1,2,3,4\ldots\ldots\ldots..9\}$ if we consider 3*3 pixels

- Apply max filters for the given image

10	20	20
20	15	20
20	25	100

Brightest point = 100

Min filters

- It is used for finding brightest point in image.

R= MIN $\{Z_K | K=1,2,3,4..........9\}$ if we consider 3*3 pixels

- Apply max filters for the given image

10	20	20
20	15	20
20	25	100

Darkest point = 10

Sharpening Spatial Filters

- The principal objective of sharpening is to highlight fine detail in an image or to enhance detail that has been blurred, either in error or as a natural effect of a particular method of image acquisition.

Applications

- Electronic printing
- Medical imaging.
- Industrial inspection.
- Autonomous guidance in military systems.

- In the case of smoothening filters , image blurring could be accomplished in the spatial domain by pixel averaging in a neighborhood.

- Since averaging is analogous to **integration.**

- But sharpening could be accomplished by **spatial differentiation.**

- Fundamentally, the strength of the response of a derivative operator is proportional to the degree of discontinuity of the image at the point at which the operator is applied.

- Thus, image differentiation enhances edges and other discontinuities (such as noise) and deemphasizes areas with slowly varying gray-level values.
- we consider in some detail sharpening filters that are based on first- and second-order derivatives, respectively.
- In particular, we are interested in the behavior of these derivatives in areas of constant gray level (flat segments), at the onset and end of discontinuities (step and ramp discontinuities), and along gray-level ramps.
- These types of discontinuities can be used to model noise points, lines, and edges in an image.

- The derivatives of a digital function are defined in terms of differences.
- There are various ways to define these differences.

- However, we require that any definition we use for a **first derivative**

1. must be zero in flat segments (areas of constant gray-level values)
2. must be nonzero at the onset of a gray-level step or ramp
3. must be nonzero along ramps.

- Similarly, any definition of a **second derivative**

1. must be zero in flat areas;
2. must be nonzero at the onset and end of a gray-level step or ramp
3. must be zero along ramps of constant slope.

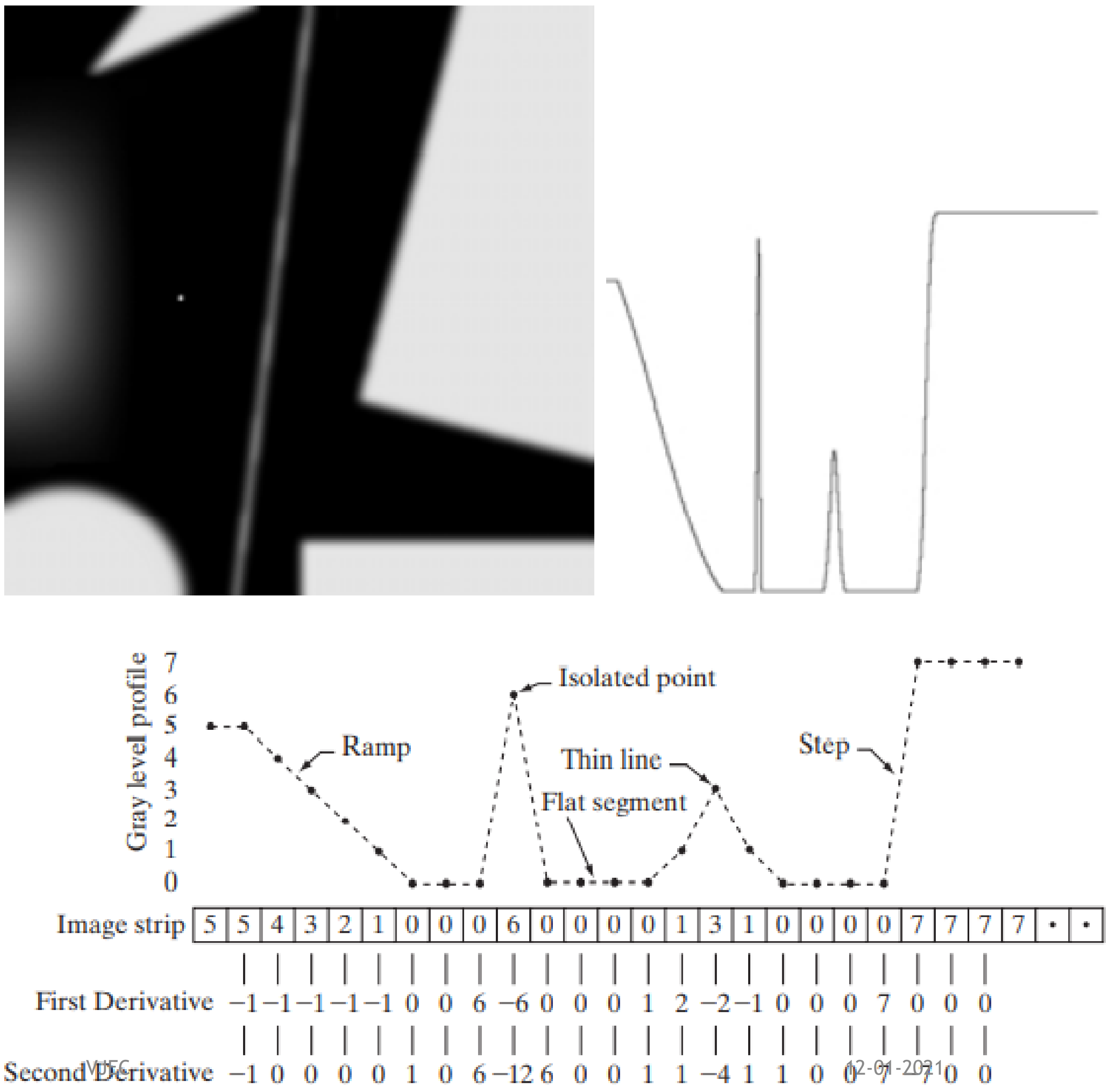
- Since we are dealing with digital quantities whose values are finite, the maximum possible gray-level change also is finite, and the shortest distance over which that change can occur is between adjacent pixels.

- A basic definition of the first-order derivative of a one dimensional function f(x) is the difference.

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

- Similarly, we define a second-order derivative as the difference

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



a b
c

FIGURE 3.38
(a) A simple image. (b) 1-D horizontal gray-level profile along the center of the image and including the isolated noise point.
(c) Simplified profile (the points are joined by dashed lines to simplify interpretation).

Conclusion

- The first-order derivatives produce “thick” edges and second order derivatives, much finer ones.
- Next we encounter the isolated noise point.
- Here, the response at and around the point is much stronger for the second- than for the first-order derivative.
- A second-order derivative is much more aggressive than a first-order derivative in enhancing sharp changes.

- Thu, we can expect a second-order derivative to enhance fine detail (including noise) much more than a first-order derivative.

- In most applications, the second derivative is better suited than the first derivative for image enhancement because of the ability of the former to enhance fine detail.

Use of Second Derivatives for Enhancement-The Laplacian

- Here we consider in some detail the use of two-dimensional, second order derivatives for image enhancement.

Development of the method

- The laplacian operator of an image f(x, y) is defined as

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

2-D Laplacian

- Taking into account that we now have two variables, we use the following notation for the partial second-order derivative in the x-direction:

$$\frac{\partial^2 f}{\partial^2 x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y) \qquad (3.7\text{-}2)$$

and, similarly in the y-direction, as

$$\frac{\partial^2 f}{\partial^2 y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y) \qquad (3.7\text{-}3)$$

The digital implementation of the two-dimensional Laplacian in Eq. (3.7-1) is obtained by summing these two components:

$$\nabla^2 f = \left[f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)\right] - 4f(x, y). \qquad (3.7\text{-}4)$$

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

a b
c d

FIGURE 3.39 (a) Filter mask used to implement the digital Laplacian, as defined in Eq. (3.7-4). (b) Mask used to implement an extension of this equation that includes the diagonal neighbors. (c) and (d) Two other implementations of the Laplacian.

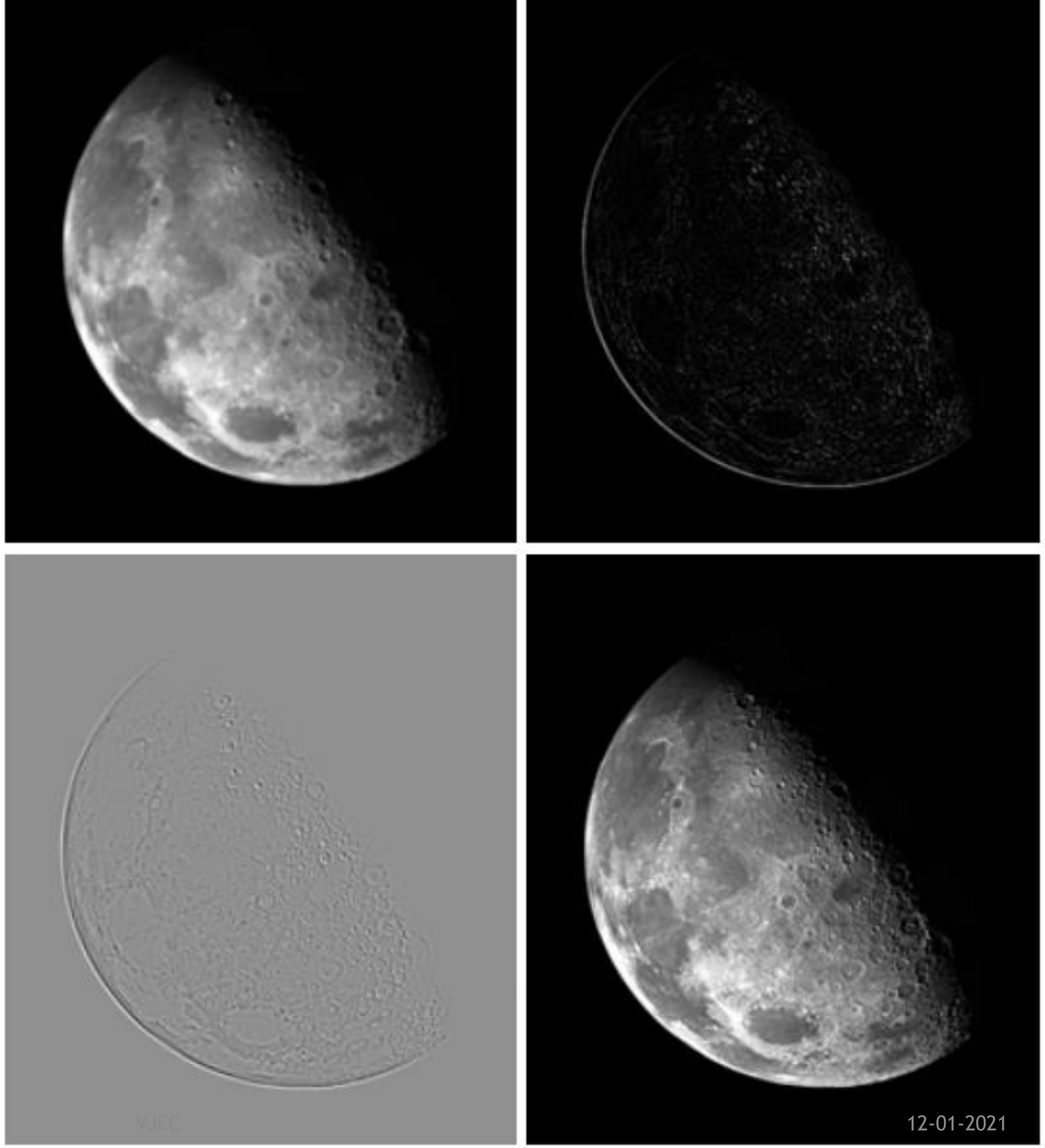
- Background features can be "recovered" while still preserving the sharpening effect of the Laplacian operation simply by adding the original and Laplacian images.
- If the definition used has a negative center coefficient, then we subtract, rather than add, the Laplacian image to obtain a sharpened result.
- Thus, the basic way in which we use the Laplacian for image enhancement is as follows:

$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y) & \text{if the center coefficient of the Laplacian mask is negative} \\ f(x, y) + \nabla^2 f(x, y) & \text{if the center coefficient of the Laplacian mask is positive.} \end{cases} \quad (3.7\text{-}5)$$

- Since the laplacian filter is a linear filter, we can apply it using the same mechanism of the convolution operation.

- This will produce a laplacian image that has grayish edge lines and other discontinuities, all superimposed on a dark , featureless background.

FIGURE 3.40
(a) Image of the North Pole of the moon.
(b) Laplacian-filtered image.
(c) Laplacian image scaled for display purposes.
(d) Image enhanced by using Eq. (3.7-5).
(Original image courtesy of NASA.)



Unsharp masking

- A process used for many years in the publishing industry to sharpen images consists of subtracting a blurred version of an image from the image itself.
- This process, called unsharp masking, is expressed as

$$f_s(x, y) = f(x, y) - \bar{f}(x, y)$$

- The origin of unsharp masking is in dark room photography, where it consists of clamping together a blurred negative to a corresponding positive film and then developing this combination to produce a sharper image.

Use of First Derivatives for Enhancement—The Gradient

- First derivatives in image processing are implemented using the magnitude of the gradient.
- For a function f(x, y), the gradient of f at coordinates (x, y) is defined as the two-dimensional column vector

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}.$$

The magnitude of this vector is given by

$$\begin{aligned} \nabla f &= \text{mag}(\nabla \mathbf{f}) \\ &= \left[G_x^2 + G_y^2\right]^{1/2} \\ &= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}. \end{aligned}$$

- Approximate the magnitude of the gradient by using absolute values instead of squares and square roots:

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

- First order derivatives mainly used for edge detection.

Three operators used for edge detection

- Robert operator
- Sobel operator
- Prewitt operator

Robert Operator:

- One of the first edge detectors, also know as cross gradient.
- This gradient-based operator computes the sum of squares of the differences between diagonally adjacent pixels in an image through discrete differentiation.
- Then the gradient approximation is made.
- It uses the following 2 x 2 kernels or masks –

$$M_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad M_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Advantages:

- Detection of edges and orientation are very easy
- Diagonal direction points are preserved

Limitations:

- Very sensitive to noise
- 2*2 masks are not easy to implement.
- No. of calculations are more.
- Not very accurate in edge detection

Sobel Operator:

- It is a discrete differentiation operator.
- It computes the gradient approximation of image intensity function for image edge detection.
- At the pixels of an image, the Sobel operator produces either the normal to a vector or the corresponding gradient vector.
- It uses two 3 x 3 kernels or masks which are convolved with the input image to calculate the vertical and horizontal derivative approximations respectively

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad M_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Advantages:

- Simple and time efficient computation
- Very easy at searching for smooth edges

Limitations:

- Diagonal direction points are not preserved always
- Highly sensitive to noise
- Not very accurate in edge detection
- Detect with thick and rough edges does not give appropriate results

Prewitt Operator:

- This operator is almost similar to the sobel operator.
- It also detects vertical and horizontal edges of an image.
- It is one of the best ways to detect the orientation and magnitude of an image.
- It uses the kernels or masks

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad M_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

- ***Advantages:***

- Good performance on detecting vertical and horizontal edges
- Best operator to detect the orientation of an image

- ***Limitations:***
- The magnitude of coefficient is fixed and cannot be changed
- Diagonal direction points are not preserved always

Image segmentation

- **Image segmentation** is the division of an image into regions or categories, which correspond to different objects or parts of objects.
- Every pixel in an image is allocated to one of a number of these categories.

A good segmentation is typically one in which:

- Pixels in the same category have similar grey scale of multivariate values and form a connected region.
- Neighboring pixels which are in different categories have dissimilar values.

Segmentation can be done based on discontinuity and similarity.

1. Partition an image based on abrupt changes in intensity, such as edges example **: edge detection**

2. Partition an image into region that are similar according to a set of predefined criteria.

examples: **Thresholding, Region growing , region splitting and merging.**

Characteristics of Segmentation Process

Let **R** represent the entire image region **and Segmentation is partitioning R** into n subgroups R_i

- $\bigcup_{i=1}^{n} Ri = R$ $\quad$ $i = 1, 2, --n$ $\quad$ R
- Ri should be connected region :i=1,2,3,……..,n
- $Ri \cap R_1 = \emptyset\ (for\ all\ i\ and\ j): i \neq j$
- $P(R_i) = TRUE\ for\ i = 1,2,3, \ldots \ldots n$
- $P(R_i \cup R_j) = FALSE\ for\ i \neq j$

P(Ri) is a predicate that indicates some properties over an image

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

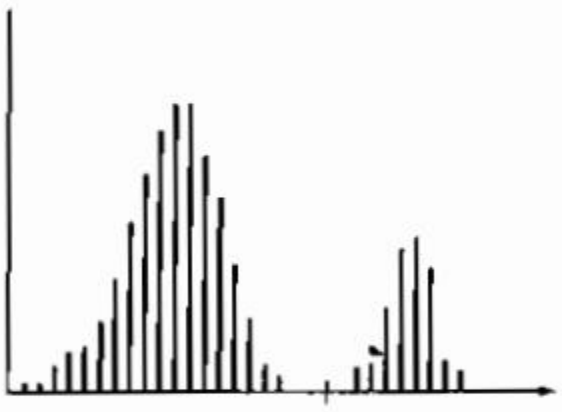
- For example, considering image pixels ranging from 0 to 255, we set a threshold of 60.
- So all the pixels with values less than or equal to 60 will be provided with a value of 0(black) and all the pixels with a value greater than 60 will be provided with a value of 255(white).
- Considering an image with a background and an object, we can divide an image into regions based on the intensity of the object and the background.
- But this threshold has to be perfectly set to segment an image into an object and a background.

The basics of intensity thresholding

- Consider an image f(x,y) composed of light objects on a dark background , in such a way that object and background pixels have intensity values grouped into two dominant modes.
- One way to extract the objects from background is to select a threshold T, that separates these modes.
- Any point (x,y) in the image at which f(x,y)> T is called an object point .
- Otherwise the point is called a background point.

- Segmented image g(x,y) is given by

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



Intensity histogram that can be portioned by a single threshold

Various thresholding techniques are :

Global thresholding:

- In this method, we use a bimodal image.
- A bimodal image is an image with 2 peaks of intensities in the intensity distribution plot.
- One for the object and one for the background.
- Then we deduce the threshold value for the entire image and use that global threshold for the whole image.
- A **disadvantage** of this type of threshold is that it performs really poorly during poor illumination in the image.

- When T is constant applicable over an entire image the process given in this equation is referred to as global thresholding.

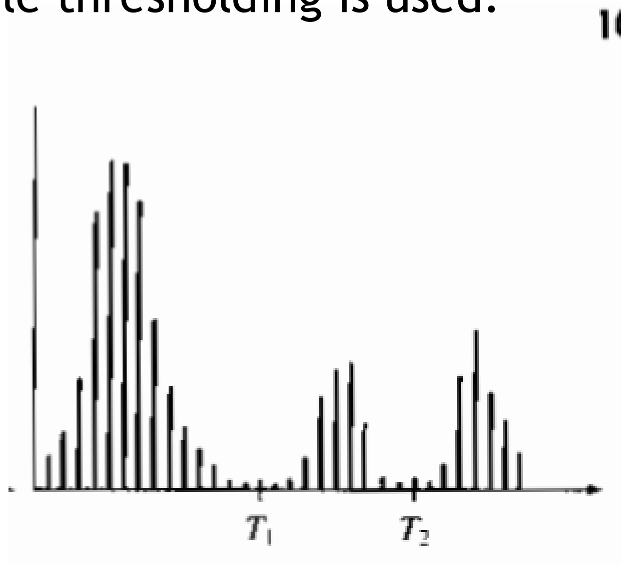
Local / regional thresholding

- It is used to denote variable thresholding in which the value of T at any point (x,y) in an image depends on properties of a neighborhood of (x,y).

Adaptive Thresholding:

- To overcome the effect of illumination, the image is divided into various subregions, and all these regions are segmented using the threshold value calculated for all these regions.
- Then these subregions are combined to image the complete segmented image.
- This helps in reducing the effect of illumination to a certain extent.

Figure shows a histogram with 3 dominant modes .
Example two types of light objects on a dark background
Here multiple thresholding is used.



Intensity histogram that can be portioned by a dual threshold

C – background a,b – two objects

That is, the segmented image is given by

$$g(x, y) = \begin{cases} a & \text{if } f(x, y) > T_2 \\ b & \text{if } T_1 < f(x, y) \leq T_2 \\ c & \text{if } f(x, y) \leq T_1 \end{cases}$$

Basic global thresholding algorithm

- Select initial threshold estimate T
- Segment the image using T

region G1 = (values >T) and region G2(values<=T)

- Compute the average intensities m1 and m2 of region G1 and G2 respectively.
- Set T= (m1+m2)/2
- Repeat step 2 through 4 until the difference in T in successive iteration is smaller than a predefined parameter T.

Problem

- Consider an image

5	3	9
2	1	7
8	4	2

T= 5+3+9+2+1+7+8+4+2/9
= 4.55 = 5

Segmenting the image
G1= 9,7,8 G2= 5,3,2,1,4,2
m1= 9+7+8/3 = 8 m2= 5+3+2+1+4+2/6 = 2.83= 3
Calculate New T = (8+3)/2 = 5.5 = 6
Difference is 1 (6-5=1)
Which is less than the initial value T. So algorithm stopped.