Chapter 2

Image Enhancement and Filter in Spatial Domain

What Is Image Enhancement?

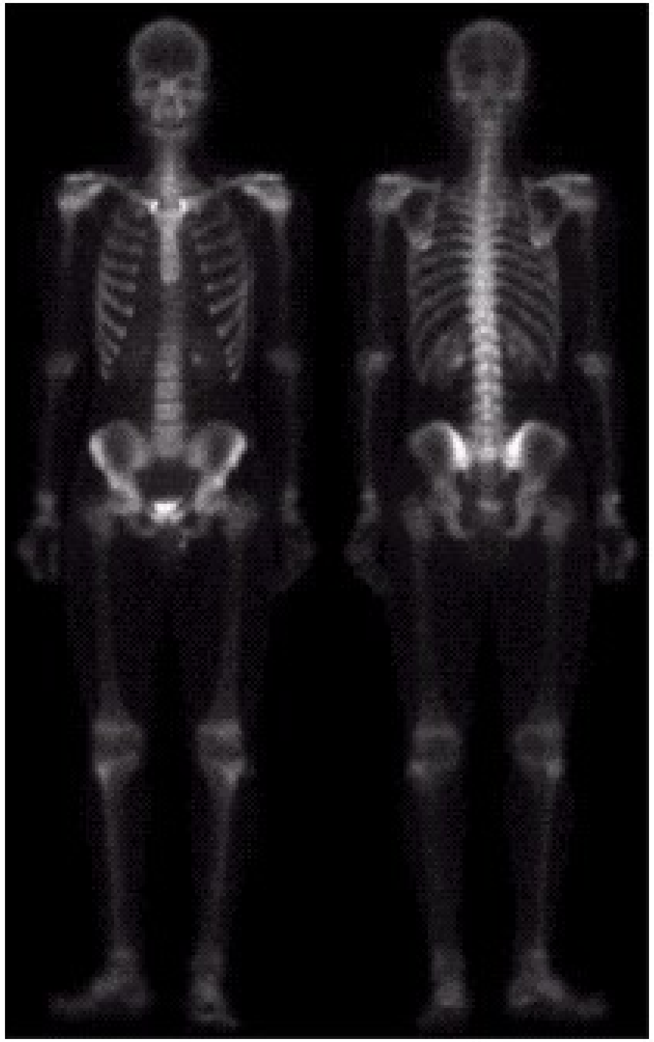
Enhancement is the process of manipulating an image so that the result is more suitable than the original for a specific application

The reasons for doing this include:

- Highlighting interesting detail in images
- Removing noise from images
- Making images more visually appealing

Image Enhancement Examples

Image Enhancement Examples (cont…)



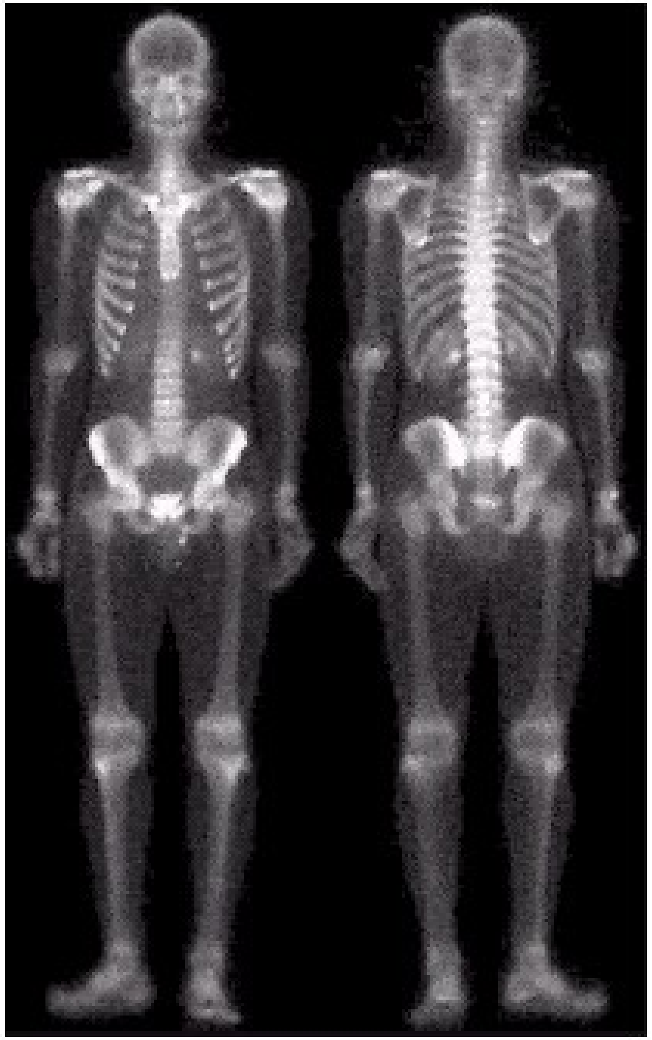
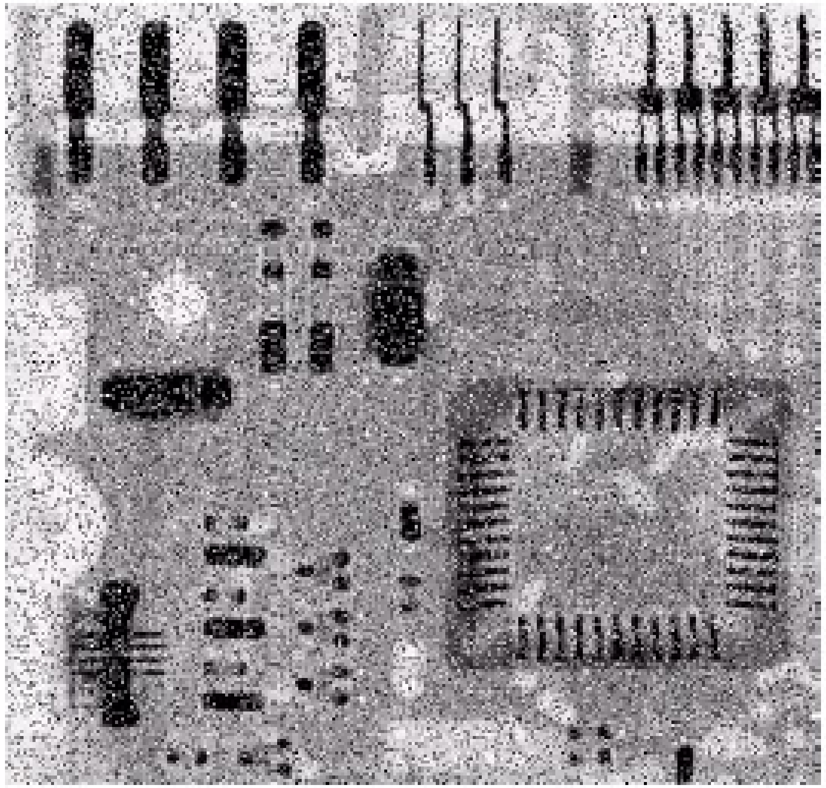


Image Enhancement Examples (cont…)



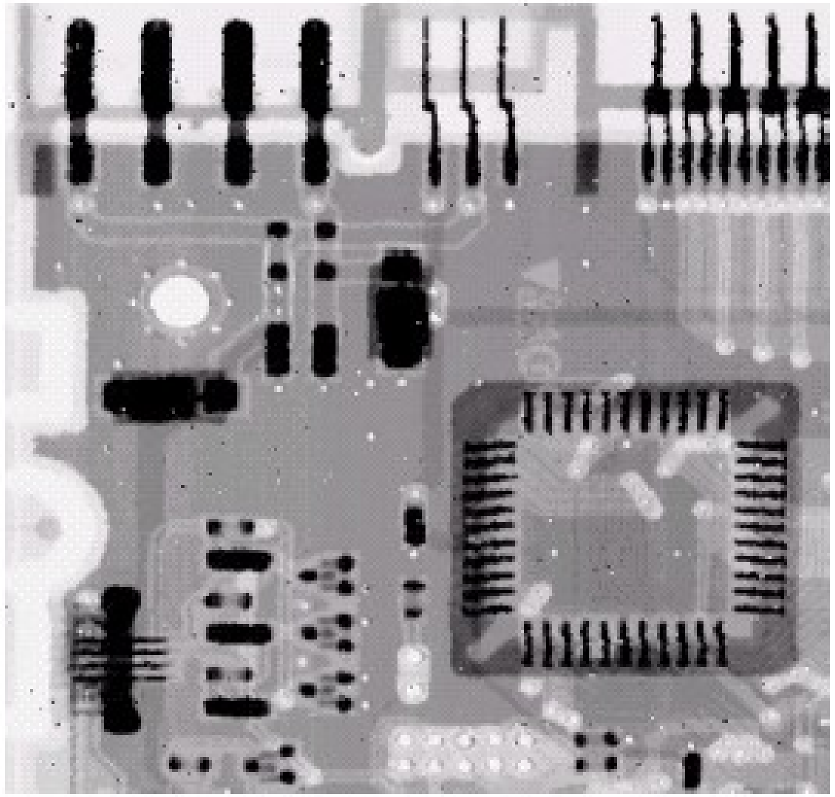


Image Enhancement Examples (cont…)

Spatial & Frequency Domains

There are two broad categories of image enhancement techniques

- **Spatial domain techniques**

  The term spatial domain refers to the image plane itself, and image processing methods in this category are based on direct manipulation of pixels in an image

- **Frequency domain techniques**

  It involves first transforming an image into the frequency domain, doing the processing there, and obtaining the inverse transform to bring the results back into the spatial domain

  For the moment we will concentrate on techniques that operate in the spatial domain

Basic Spatial Domain Image Enhancement

Most spatial domain enhancement operations can be reduced to the form

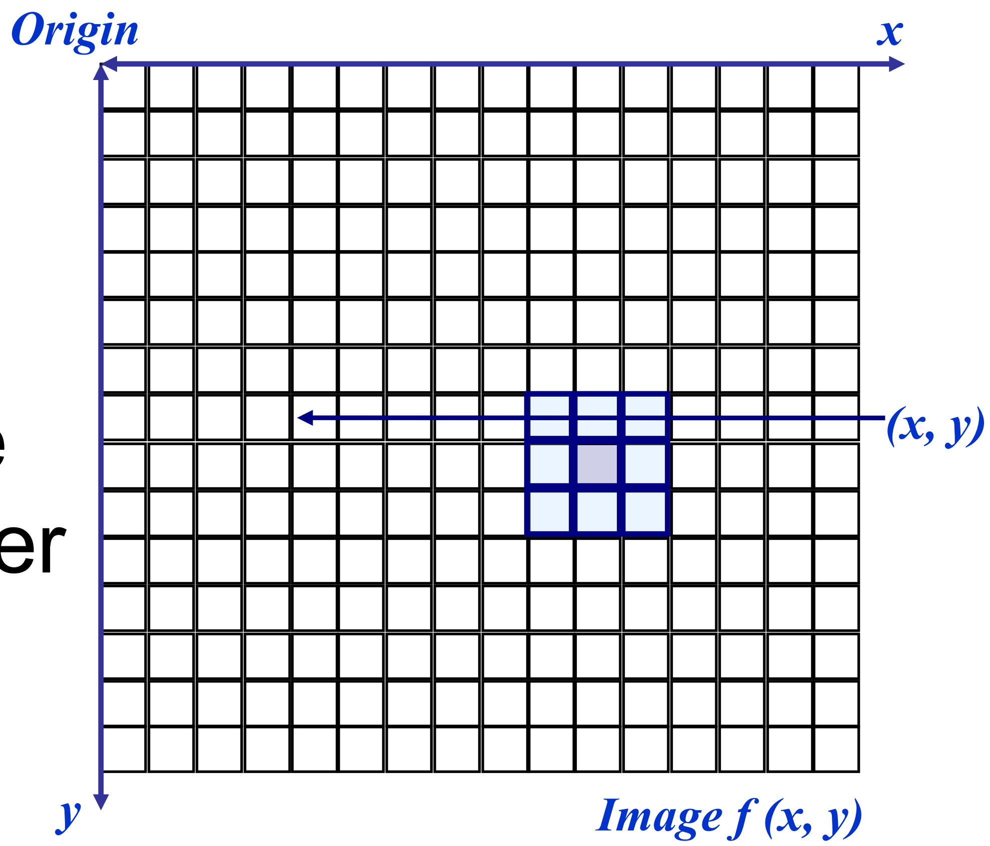
$$g(x, y) = T[f(x, y)]$$

Where,

$f(x, y)$ is the input image,

$g(x, y)$ is the processed image

T is some operator defined over some neighbourhood of (x, y)



Point Processing

- The simplest spatial domain operations occur when the neighbourhood is simply the pixel itself
- In this case T is referred to as a *grey level transformation function* or a *point processing operation*
- Point processing operations take the form

$$s = T(r)$$

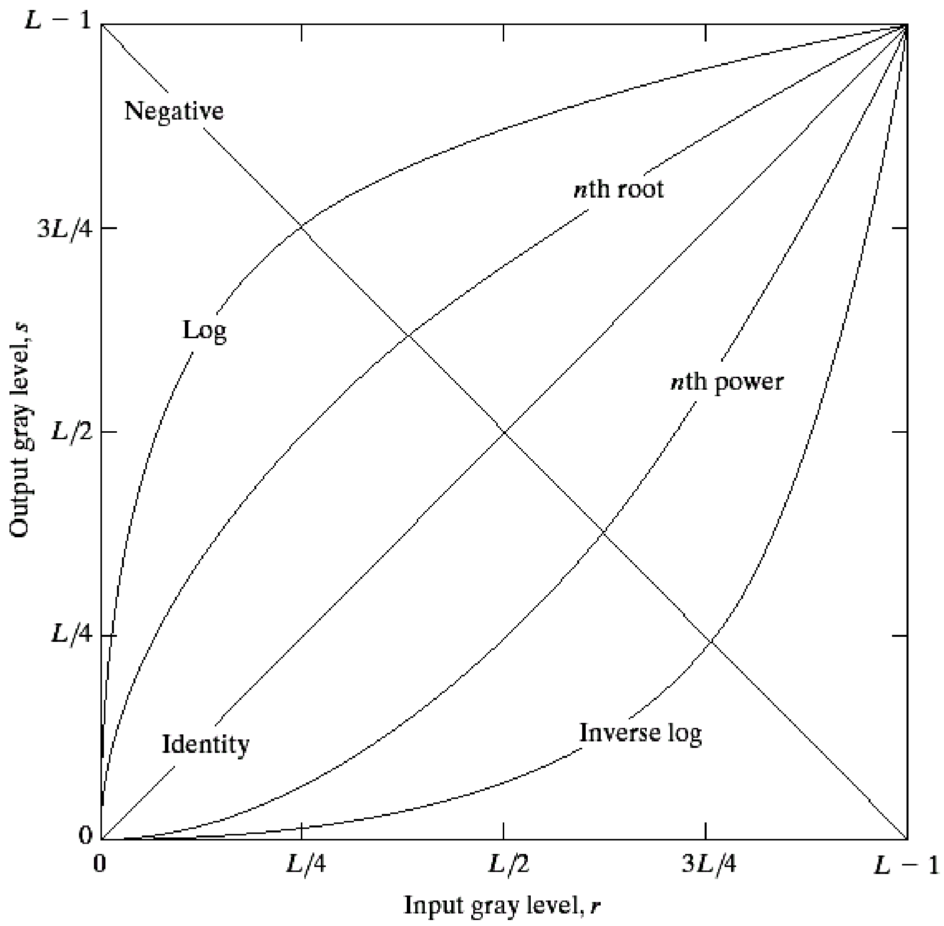
where s refers to the processed image pixel value and r refers to the original image pixel value

Basic Grey Level Transformations

There are many different kinds of grey level transformations

Three of the most common are shown here

- Linear
  - Negative/Identity
- Logarithmic
  - Log/Inverse log
- Power law
  - n^{th} power/n^{th} root



Point Processing Example: Digital Negative Images

Negative images are useful for enhancing white or grey detail embedded in dark regions of an image

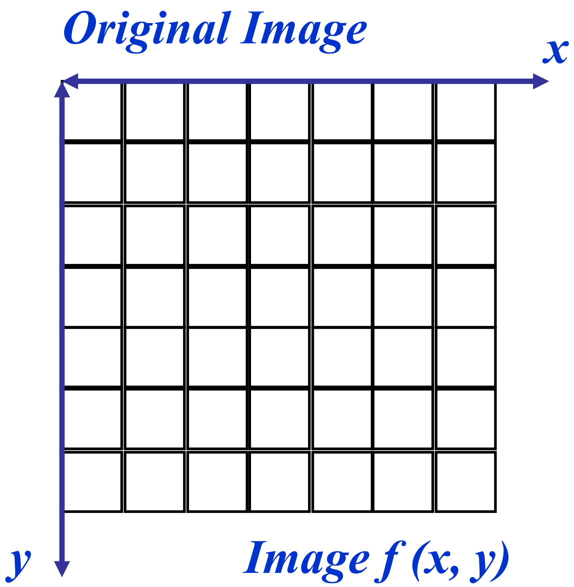
- Note how much clearer the tissue is in the negative image of the mammogram below

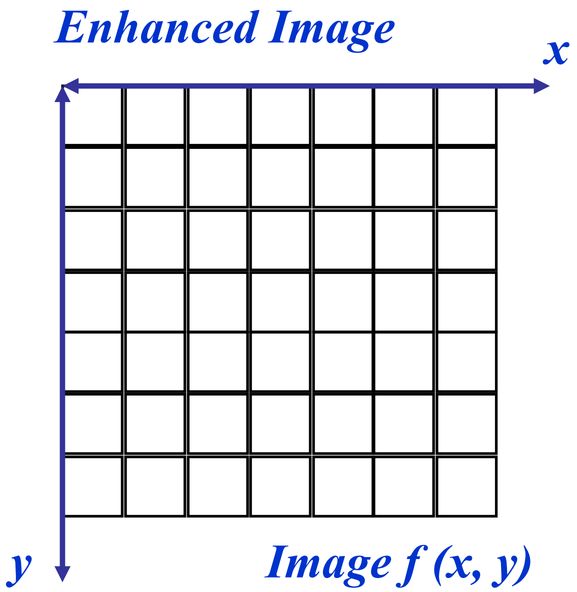
Original Image

$s = 1.0 - r$

Negative Image

Point Processing Example: Digital Negative





$$s = intensity_{max} - r$$

Point Processing Example: Digital Negative

Obtain the digital negative image of the 3 Bit image as shown below:

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

For obtaining the negative image :

f2(x,y) =L – 1 – f1(x,y)

Where ,

L : Total No. of Gray Levels

f1(x,y) : Original Image.

For a 3 - Bit image : $L= 2^3 = 8$

f2(x,y) = 8 – 1 – r

= 7 - r

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

Logarithmic Transformations

- The general form of the log transformation is

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

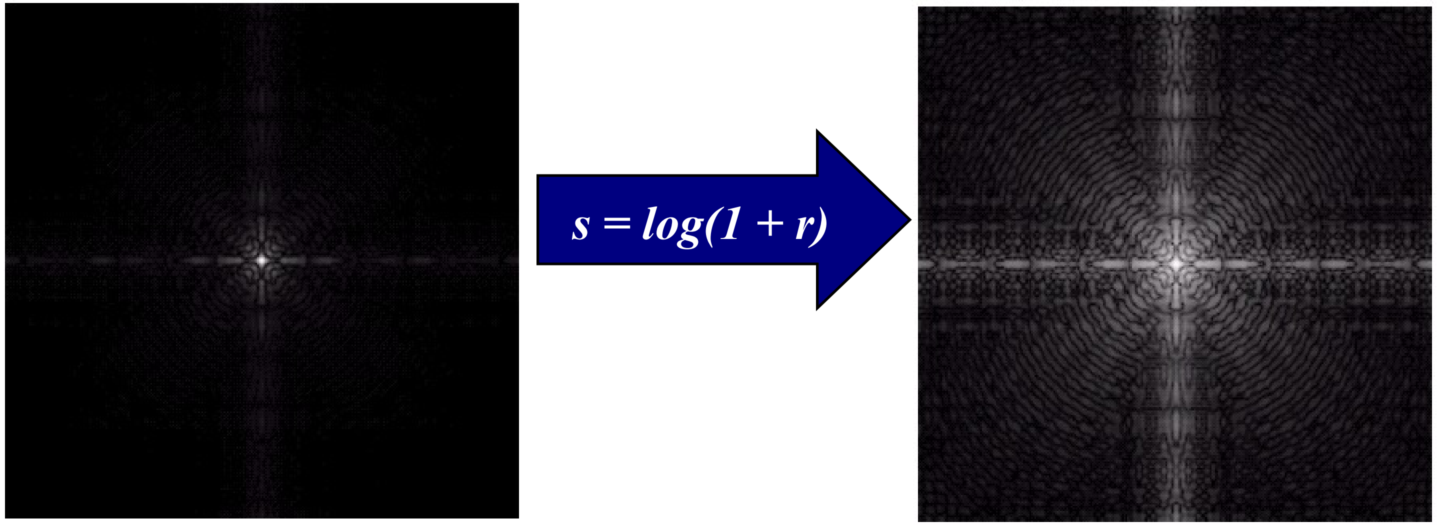
The log transformation maps a narrow range of low input grey level values into a wider range of output values

- The inverse log transformation performs the opposite transformation
- Compresses the dynamic range of images with large variations in pixel values

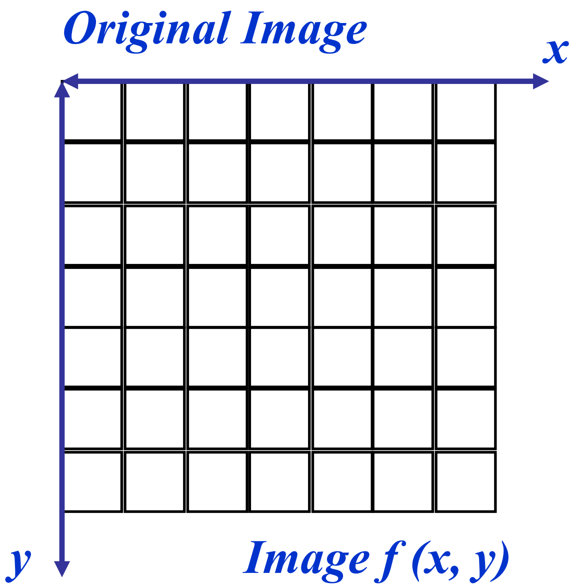
Logarithmic Transformations

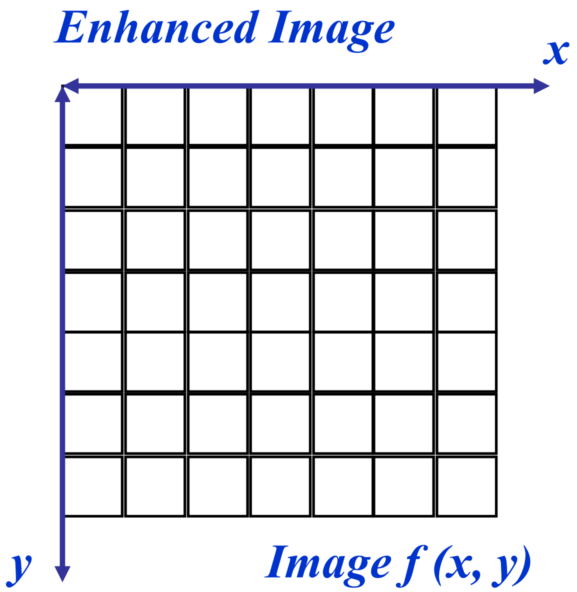
Log functions are particularly useful when the input grey level values may have an extremely large range of values

In the following example the Fourier transform of an image is put through a log transform to reveal more detail



Logarithmic Transformations





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

We usually set c to 1 Grey levels must be in the range [0.0, 1.0]

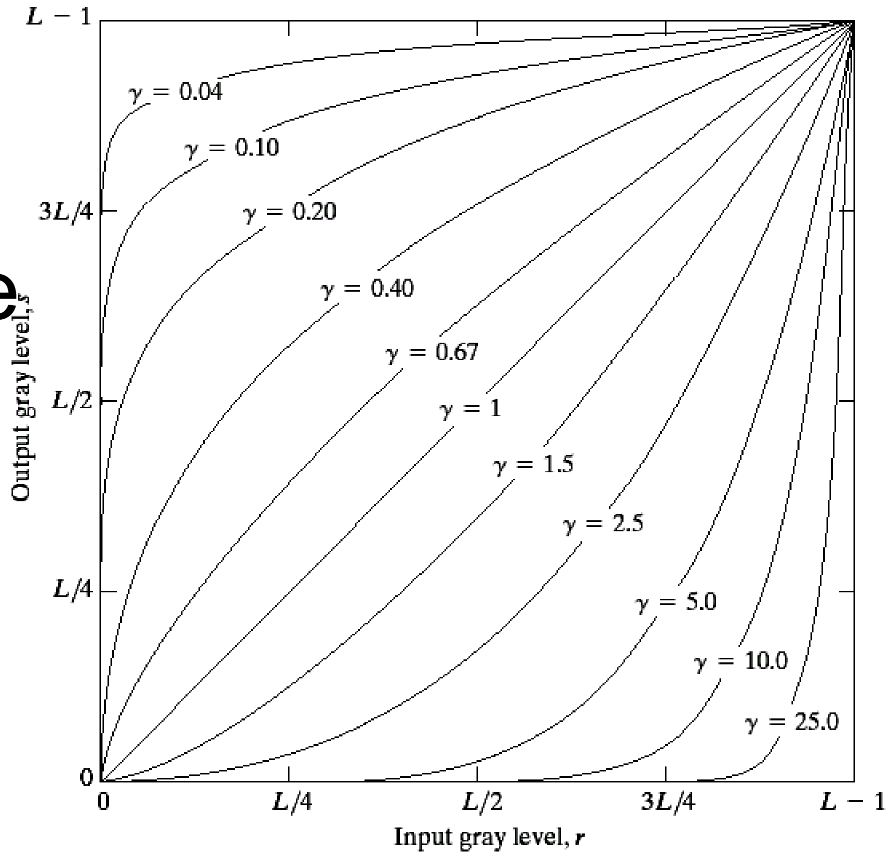
Power Law Transformations

Power law transformations have the following form

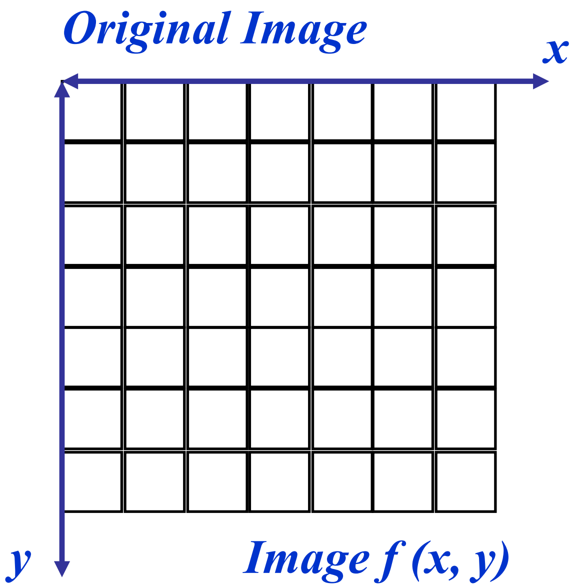
$$s = c * r^{\gamma}$$

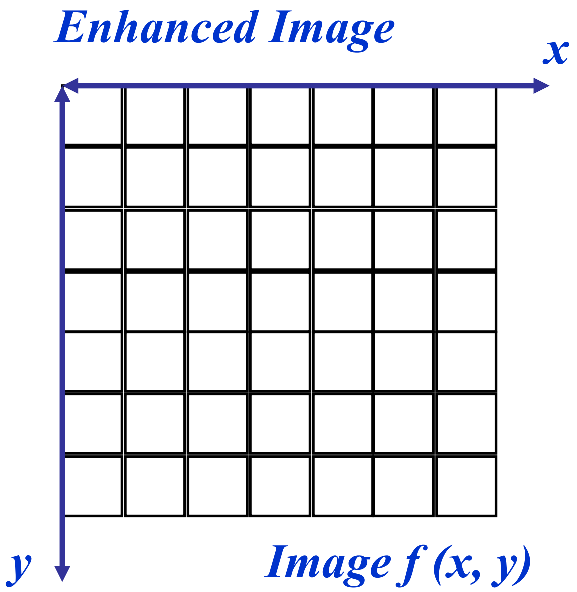
Map a narrow range of dark input values into a wider range of output values or vice versa

Varying γ gives a whole family of curves



Power Law Transformations



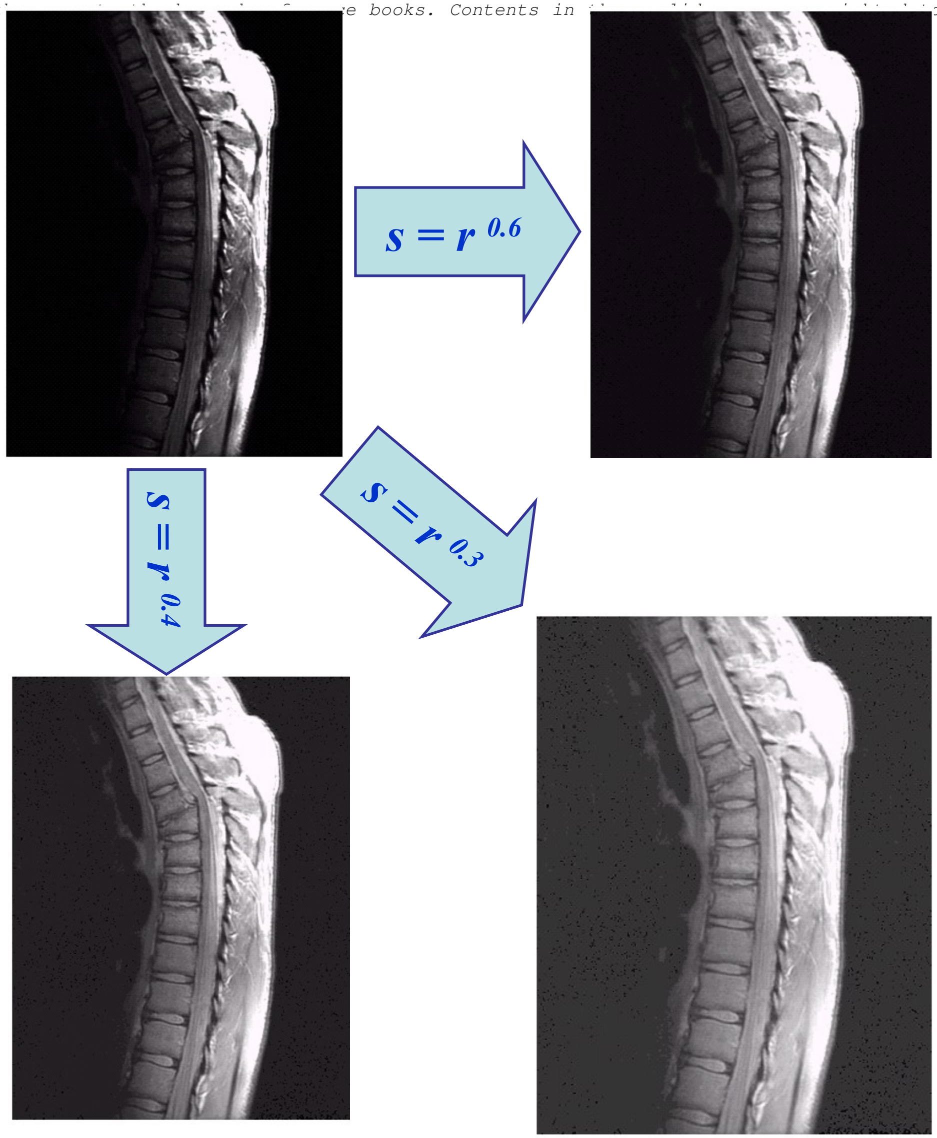


$$s = r^{\gamma}$$

We usually set c to 1 Grey levels must be in the range [0.0, 1.0]

Power Law Example (cont…)

The images to the right show a magnetic resonance (MR) image of a fractured human spine



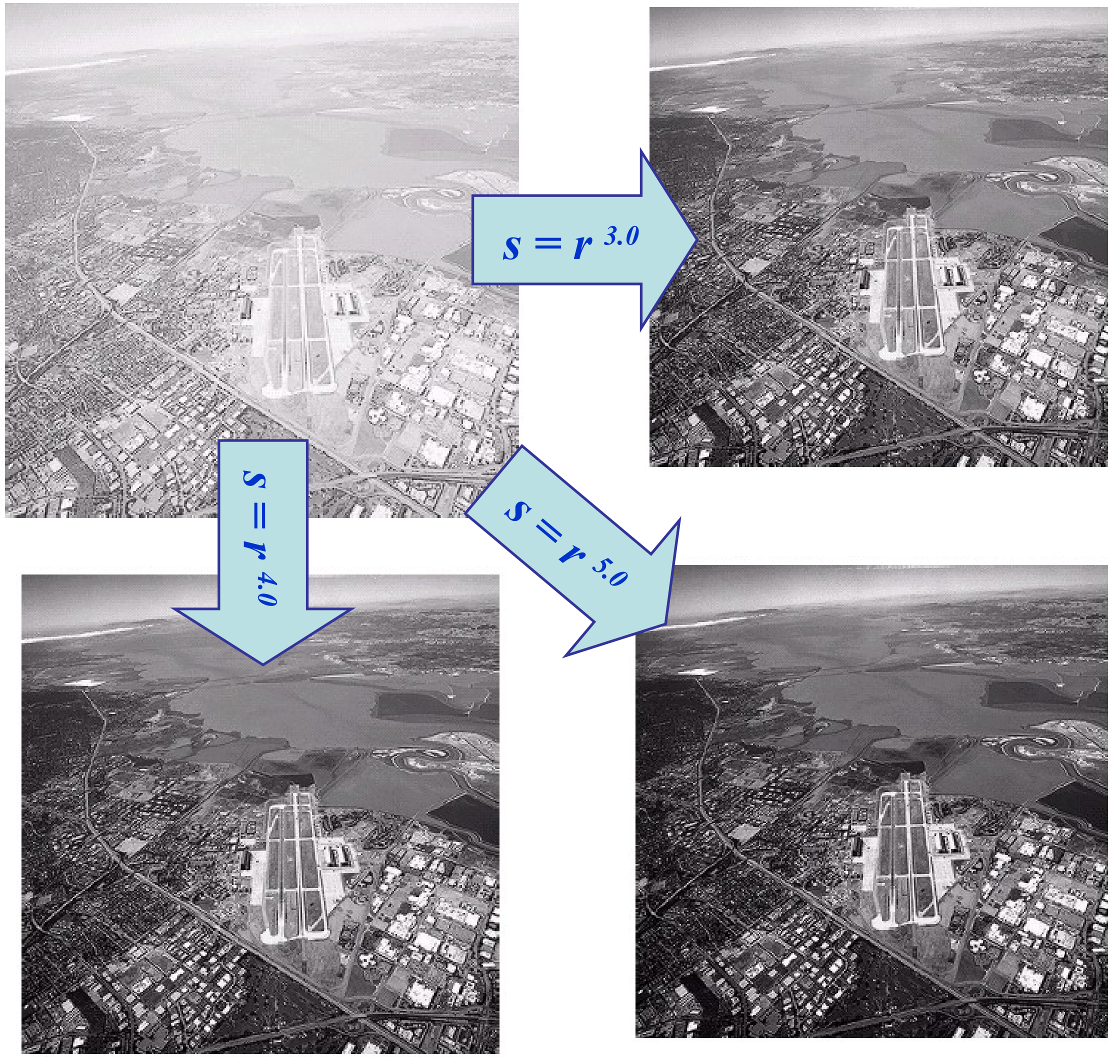
Different curves highlight different detail

Power Law Transformations (cont…)

An aerial photo of a runway is shown

This time power law transforms are used to darken the image

Different curves highlight different detail



Gamma Correction

- Many devices used for image capture, display and printing respond according to a power law
  - The exponent in the power-law equation is referred to as gamma
  - The process of correcting for the power-law response is referred to as gamma correction
  - Example: – CRT devices have an intensity-to-voltage response that is a power function (exponents typically range from 1.8-2.5)

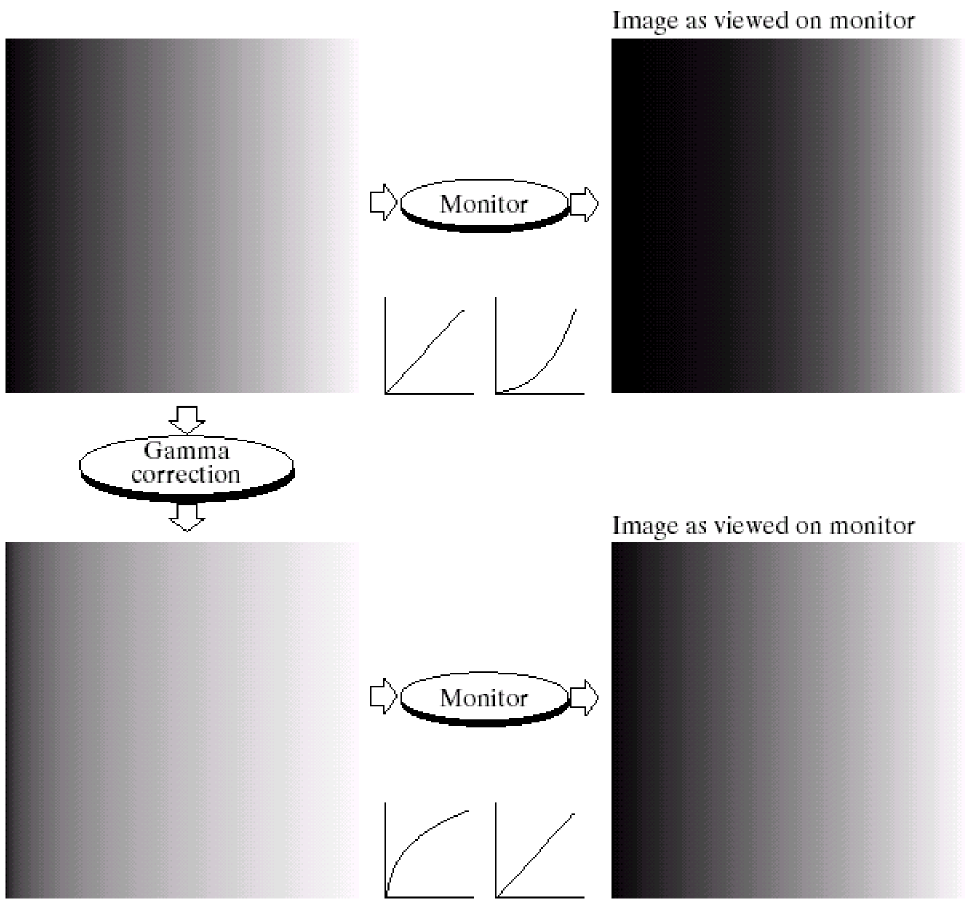
- Gamma correction in this case could be achieved by applying the transformation $s=r^{1/2.5}=r^{0.4}$

Gamma Correction

Many of us might be familiar with gamma correction of computer monitors

Problem is that display devices do not respond linearly to different intensities

Can be corrected using a log transform



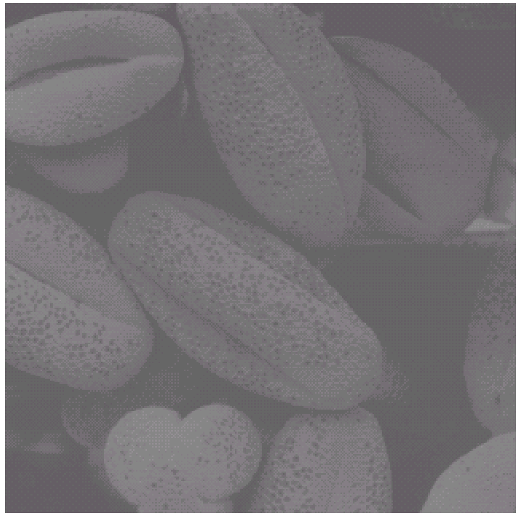
Effects of Reducing Gamma

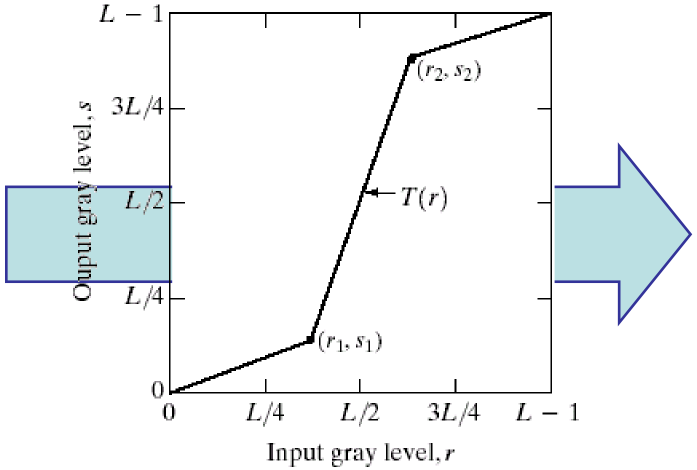
When the γ is reduced too much, the image begins to reduce contrast to the point where image started to have slight "Wash-out" look, especially in the background

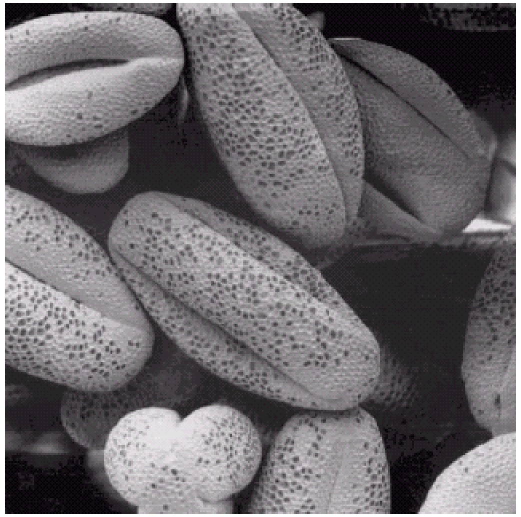
Piecewise Linear Transformation Functions

Rather than using a well defined mathematical function we can use arbitrary user-defined transforms

The images below show a contrast stretching linear transform to add contrast to a poor quality image







Piecewise Linear Transformation Functions

- Piece-wise Linear Transformation is type of gray level transformation that is used for image enhancement.
- It is a spatial domain method.
- It is used for manipulation of an image so that the result is more suitable than the original for a specific application.
- Types
  - Gray Level Slicing
  - Contrast Stretching
  - Bit plane Slicing

Gray Level Slicing

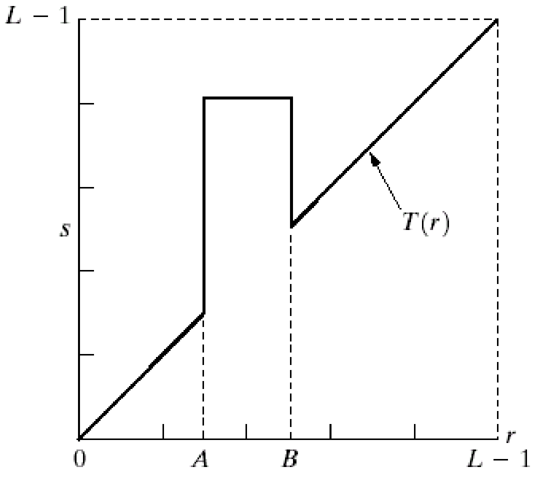
- Used to highlight a specific range of intensities in an image that might be of interest
- Two common approaches
  - Set all pixel values within a range of interest to one value (white) and all others to another value (black)

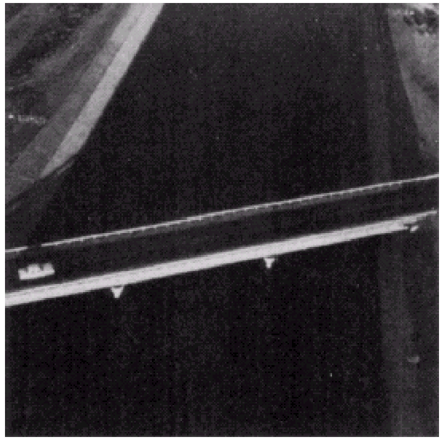
    (Produces a binary image)
  - Brighten (or darken) pixel values in a range of interest and leave all others unchanged

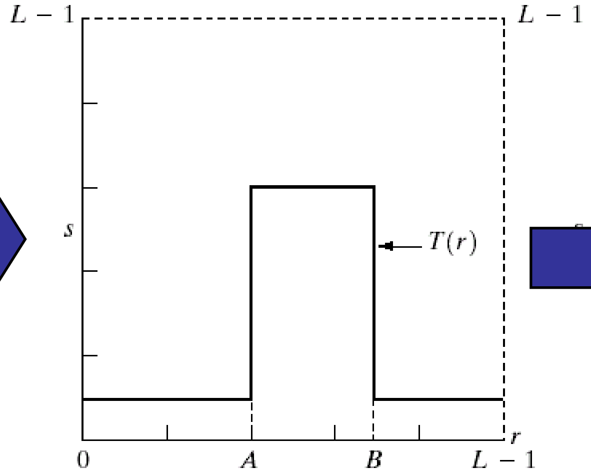
Gray Level Slicing

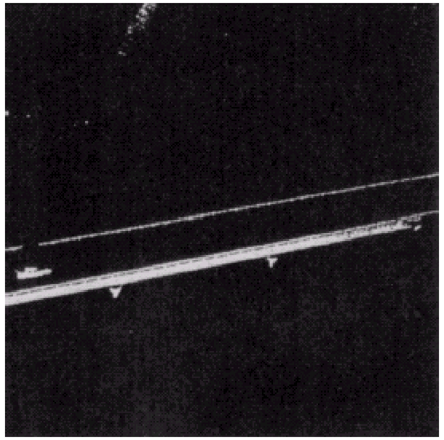
Highlights a specific range of grey levels

- Similar to thresholding
- Other levels can be suppressed or maintained
- Useful for highlighting features in an image



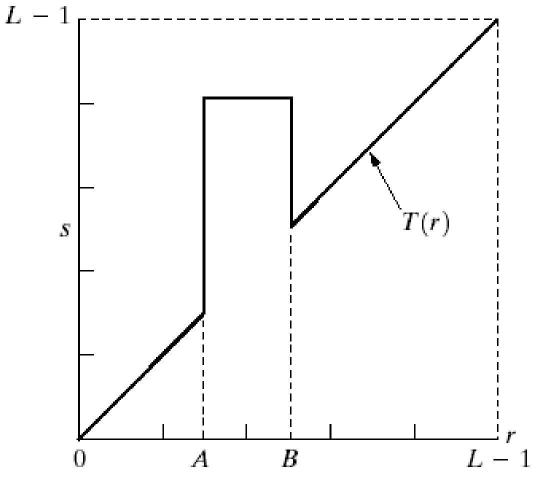


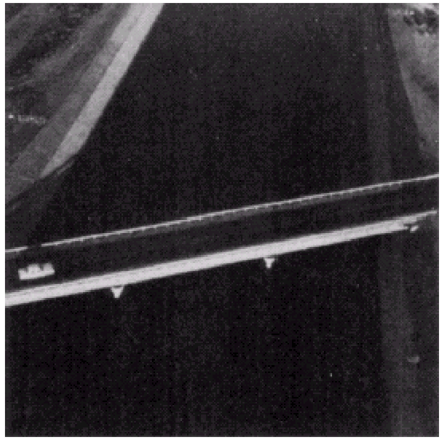


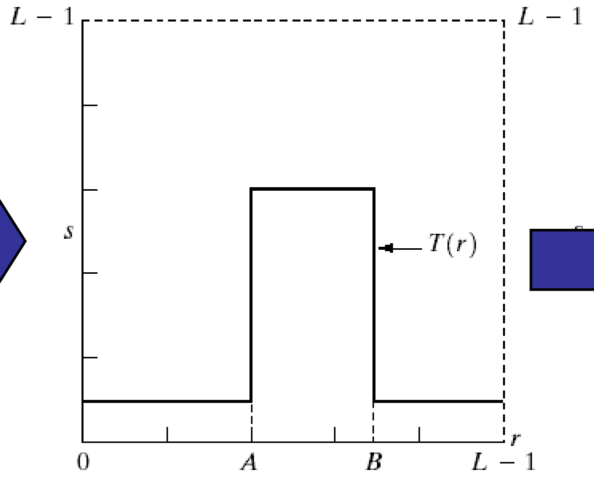


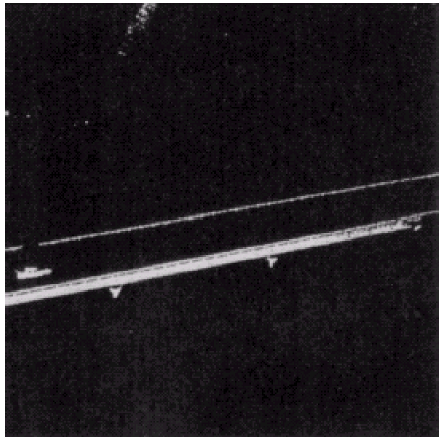
Gray Level Slicing

Highlights a specific range of grey levels
- Similar to thresholding
- Other levels can be suppressed or maintained
- Useful for highlighting features in an image









Gray Level Slicing

Perform gray level slicing with Background and Without Background on the 3 bit image shown below. (Assume r1 =2 and r2= 4)

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

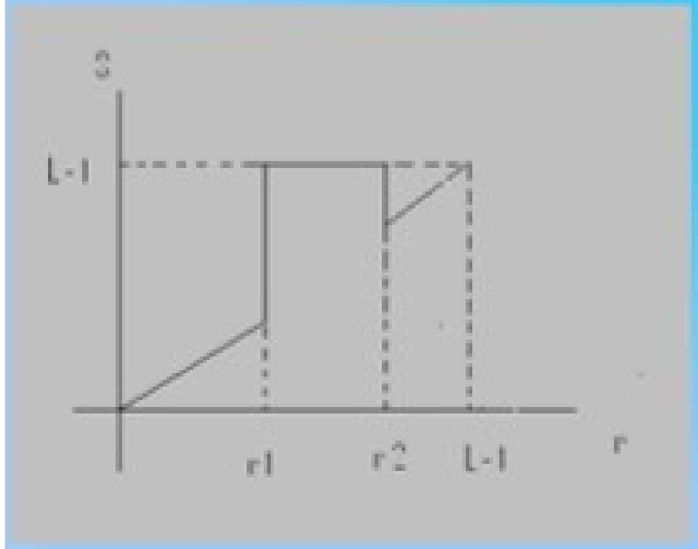
With Background

S=7 if $r_1 <= r <= r_2$

S = S (otherwise) if $0 < r < r_1$ and if $r > r_2$

$r_1 = 2$ and $r_2 = 4$

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



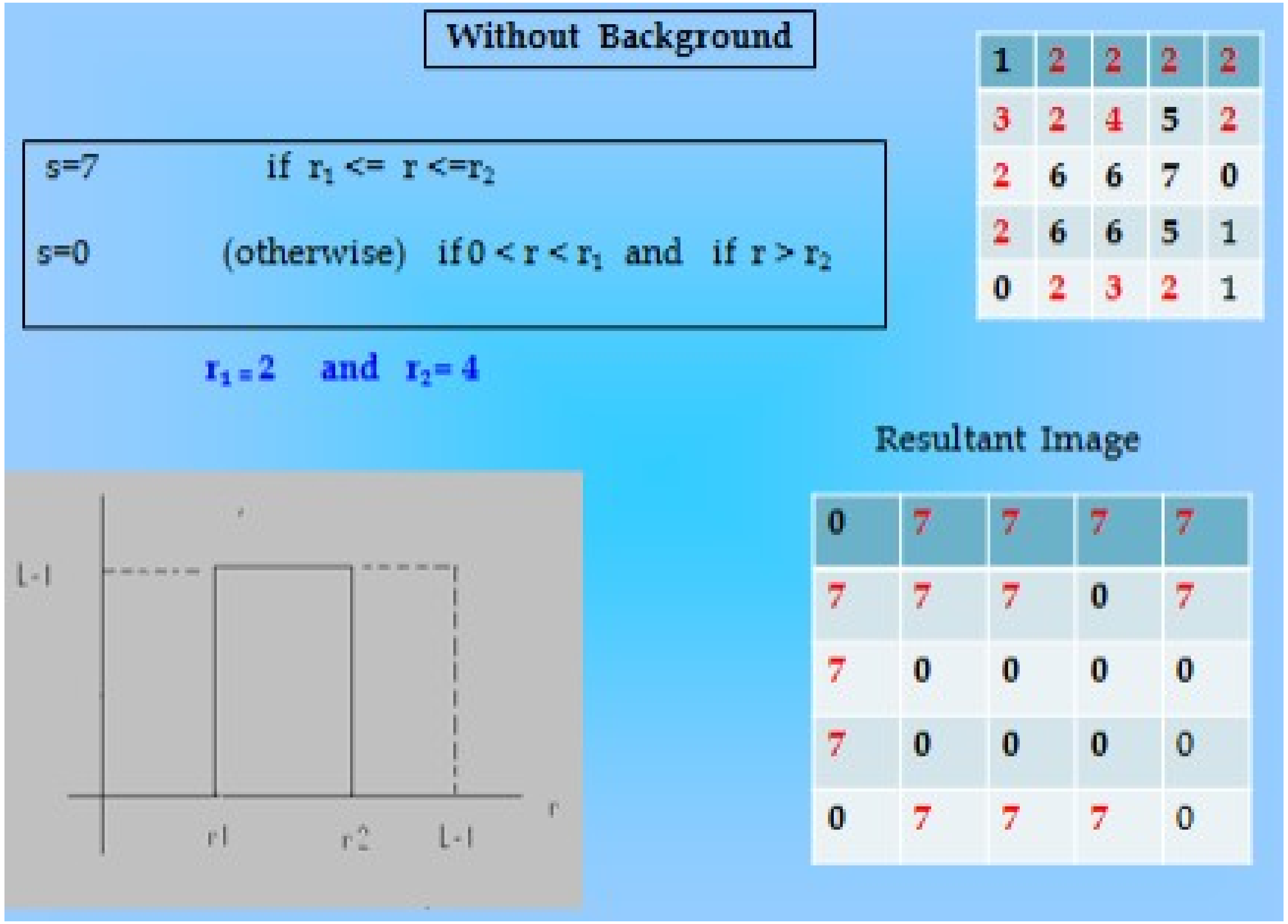
Resultant Image

1	7	7	7	7
7	7	7	5	7
7	6	6	7	0
7	6	6	5	1
0	7	7	7	1

Gray Level Slicing

Perform gray level slicing with Background and Without Background on the 3 bit image shown below. (Assume r1 =2 and r2= 4)

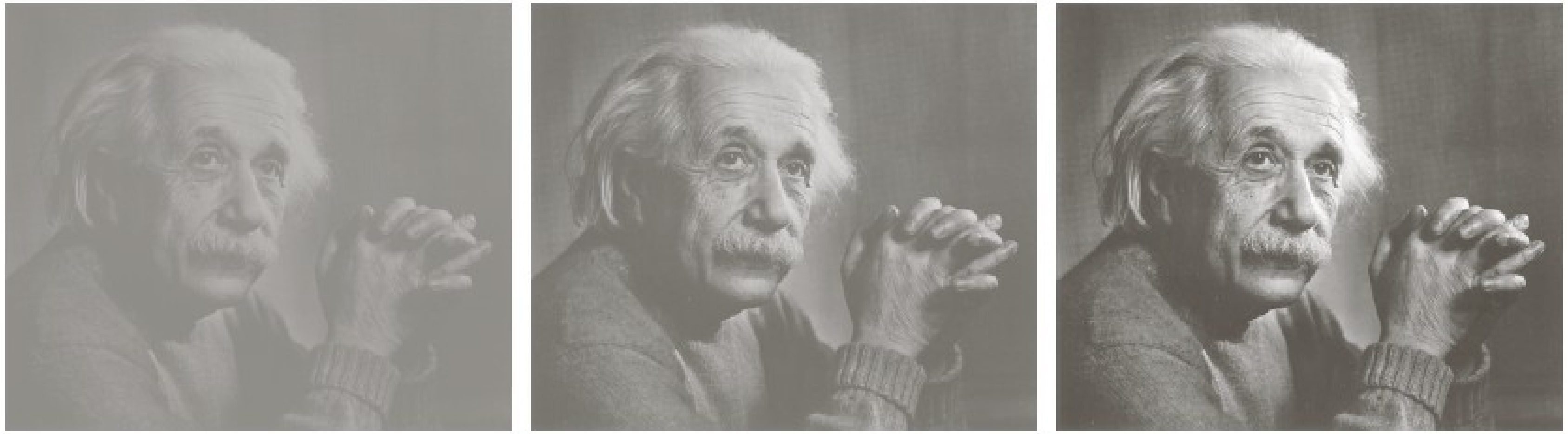
1	2	2	2	2
3	2	4	5	2
2	6	6	7	0
2	6	6	5	1
0	2	3	2	1



Contrast

Contrast is the difference between maximum and minimum pixel intensity

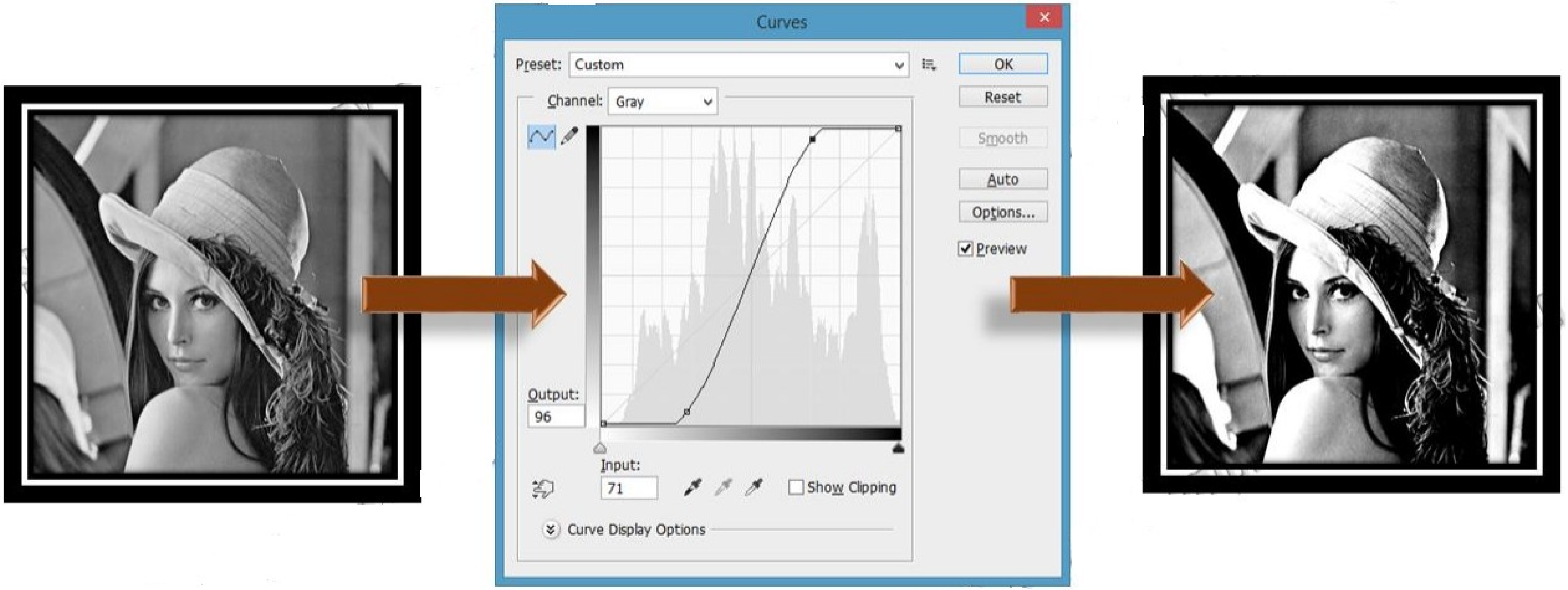
Contrast is the difference in luminance or color that makes an object distinguishable from other objects within the same field of view.



Contrast Stretching

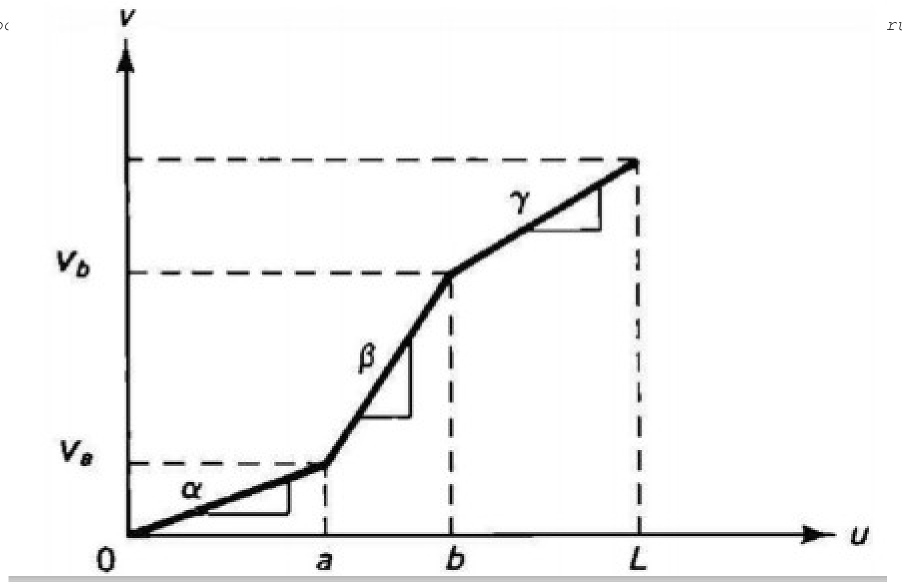
Contrast stretching is an image enhancement technique that tries to improve the contrast by stretching the intensity values of an image to fill the entire dynamic range.

Contrast Stretching

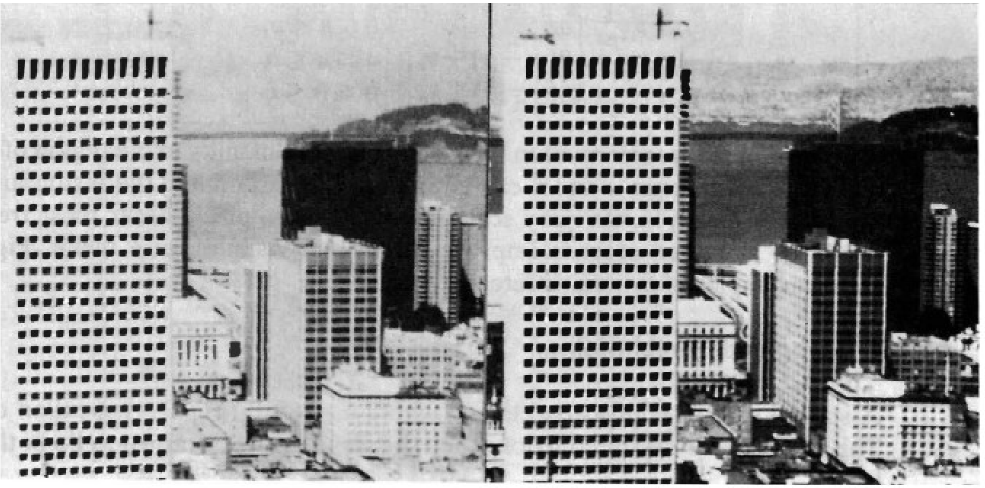


Contrast Stretching

The slopes *alpha, beta, gamma* determine the relative contrast stretch



$$f(u) = \begin{cases} \alpha u, & 0 \le u < a \\ \beta(u-a) + v_a, & a \le u < b \\ \gamma(u-b) + v_b, & b \le u < L \end{cases}$$



Clipping

A special case of contrast stretching where a = γ = 0 is called clipping

$$f(u) = \begin{cases} \alpha u, & 0 \le u < a \\ \beta(u-a) + v_a, & a \le u < b \\ \gamma(u-b) + v_b, & b \le u < L \end{cases}$$

$$\Rightarrow$$

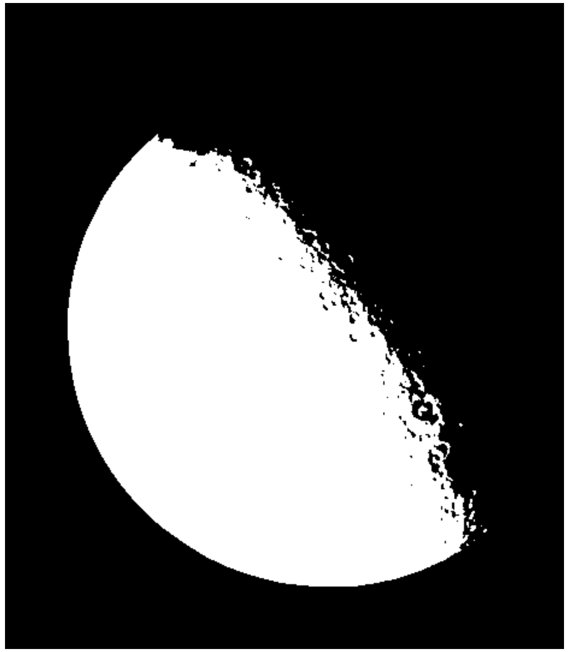
$$f(u) = \begin{cases} 0, & 0 \le u < a \\ \alpha u, & a \le u \le b \\ L, & u \ge b \end{cases}$$

For a = b = t, this is called THRESHOLDING (The output becomes binary)

Point Processing Example: Significance of Thresholding

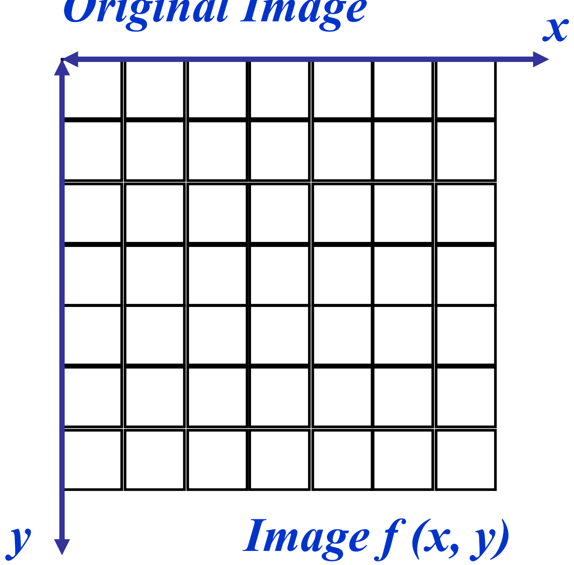
Thresholding transformations are particularly useful for segmentation in which we want to isolate an object of interest from a background

$$s = \begin{cases} 1.0 & r > \text{threshold} \\ 0.0 & r <= \text{threshold} \end{cases}$$

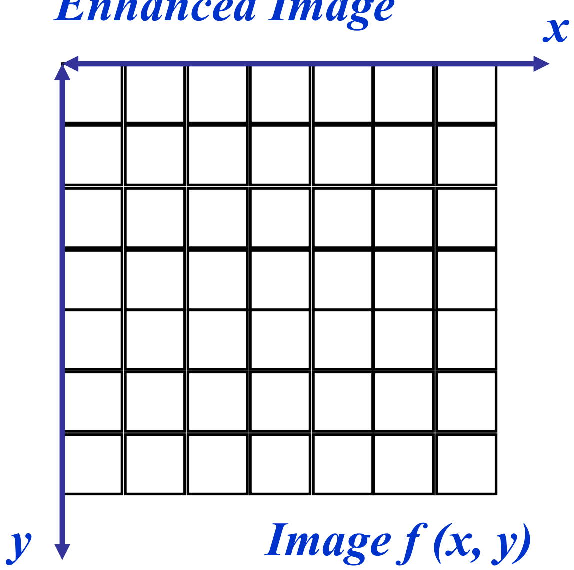


Point Processing Example: Thresholding

Original Image

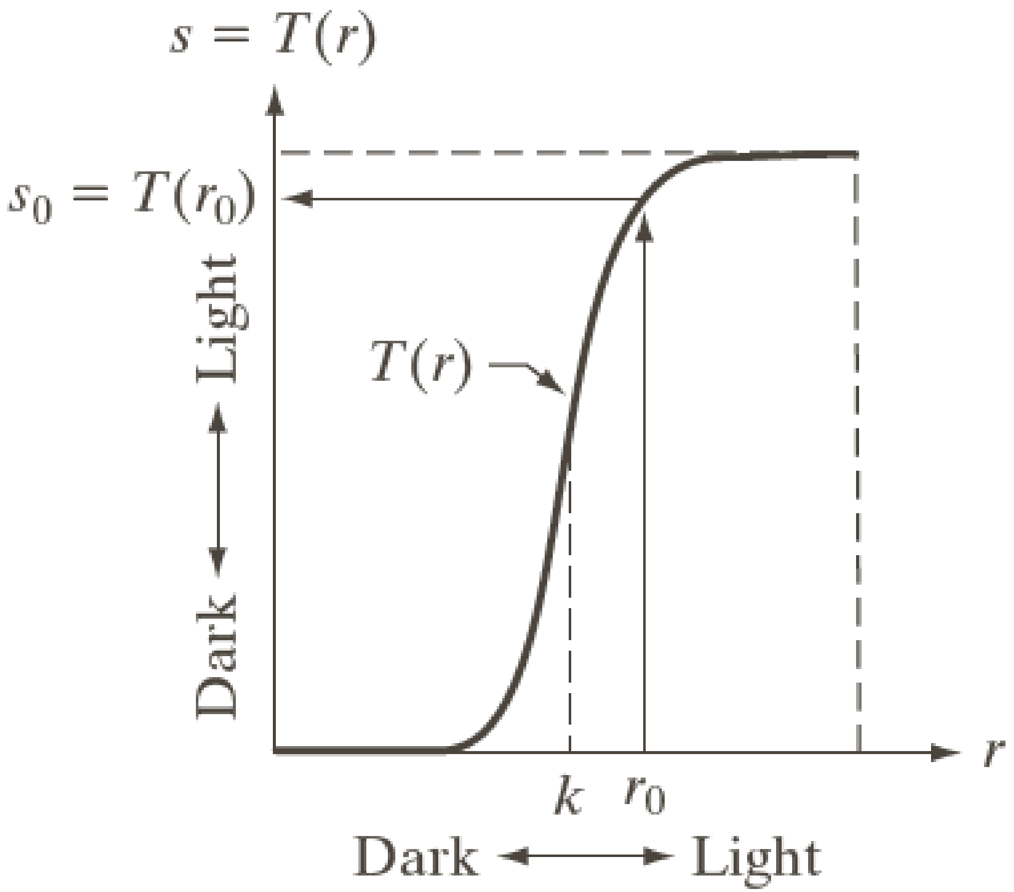


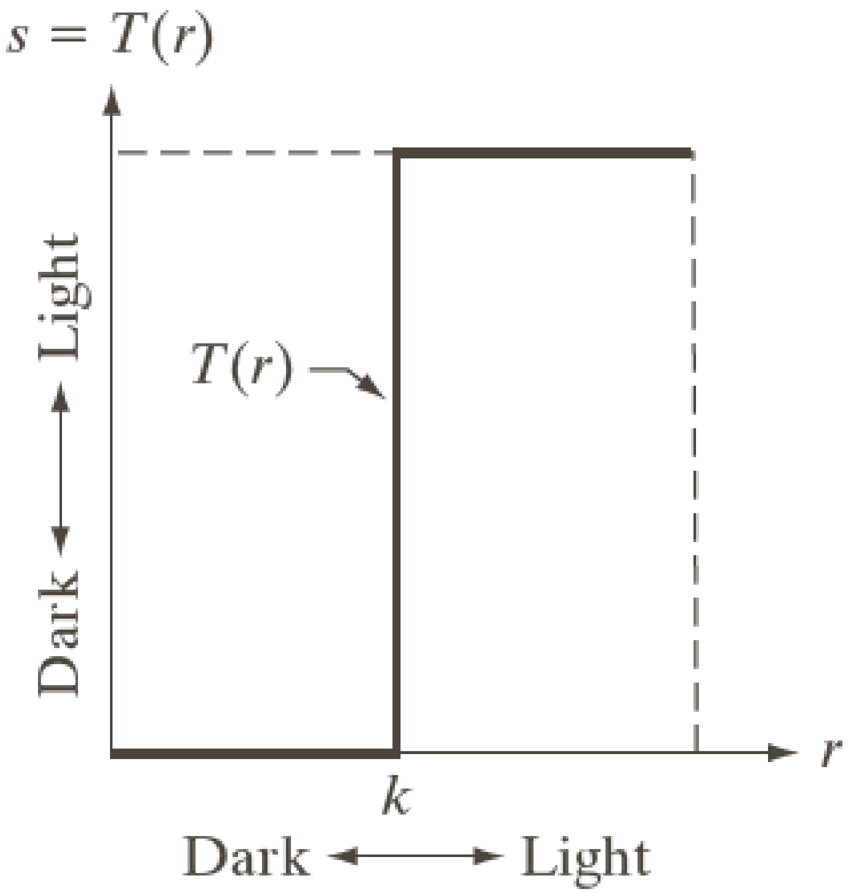
Enhanced Image



$$s = \begin{cases} 1.0 & r > threshold \\ 0.0 & r <= threshold \end{cases}$$

Intensity Transformations

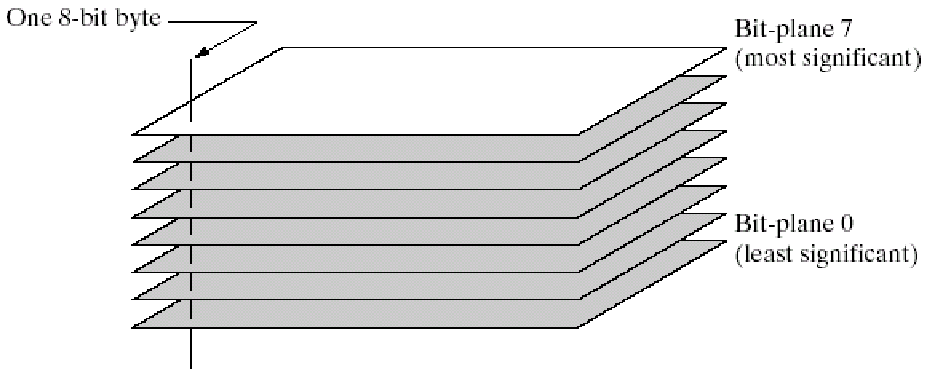


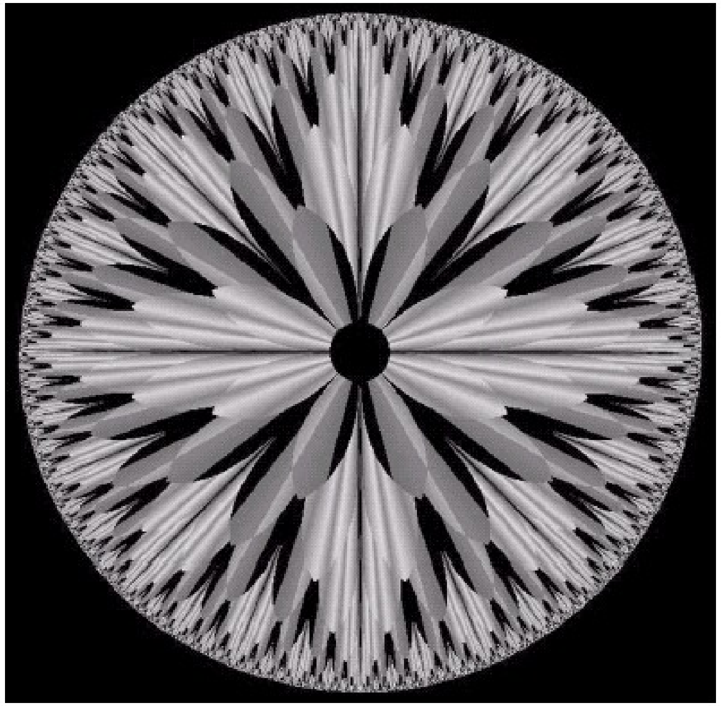


Bit Plane Slicing

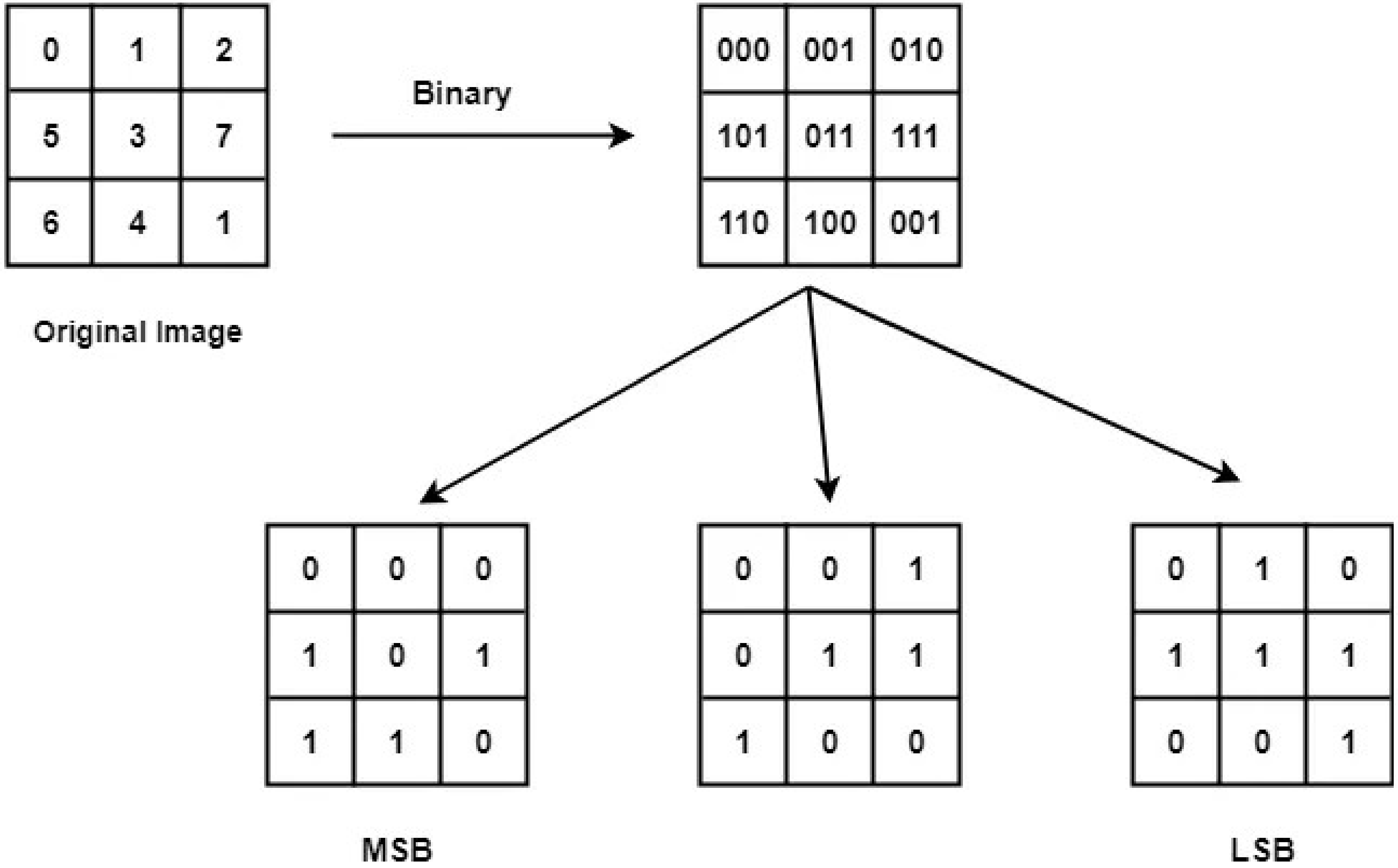
Often by isolating particular bits of the pixel values in an image we can highlight interesting aspects of that image

- Higher-order bits usually contain most of the significant visual information
- Lower-order bits contain subtle details





Bit Plane Slicing



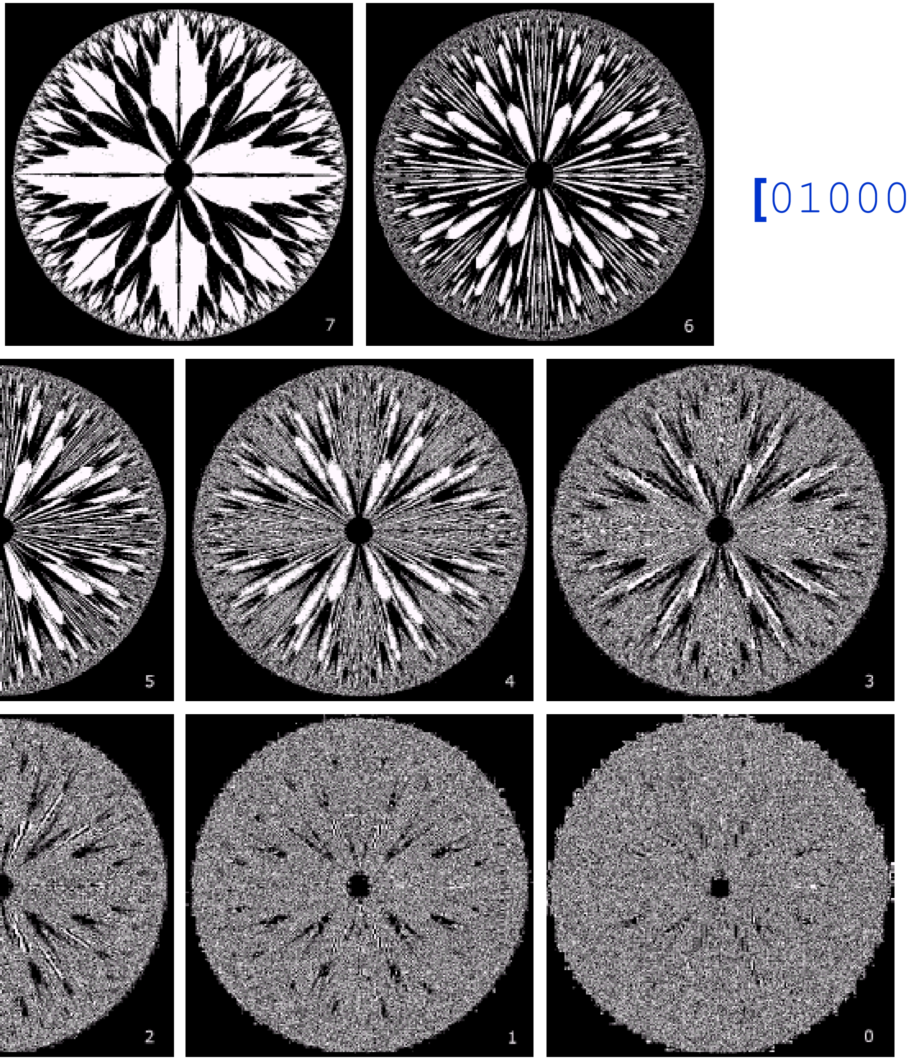
Bit Plane Slicing

[10000000]

[01000000]

[00100000]

[00001000]



[00000100]

[00000001]

Bit Plane Slicing

Bit Plane Slicing

Reconstructed image using only bit planes 8 and 7

Reconstructed image using only bit planes 8, 7 and 6

Reconstructed image using only bit planes 7, 6 and 5

Histogram of Image

The histogram of a digital image with gray levels in the range [0, L-1] is a discrete function.

$$H(r_k) = n_k$$

Where r_k is the kth intensity value and nk is the number of pixels in f with intensity r_k

Histogram of an image provides a global description of the appearance of an image.

Histogram of an image represents the relative frequency of occurrence of various gray levels in an image.

Use of Histogram

- Image enhancements
- Image statistics
- Image compression
- Image segmentation
- Popular tool in real-time image processing
- A plot of this function for all values of k provides a global description of the appearance of the image (gives useful information for contrast enhancement)

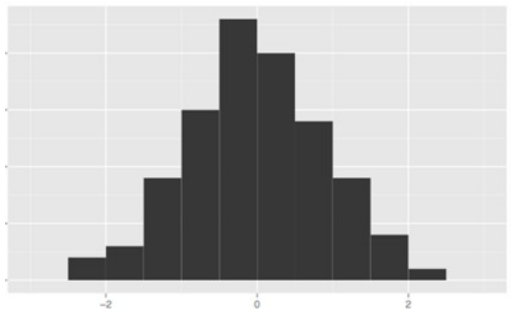
Histogram of Image

Histograms commonly viewed in plots as

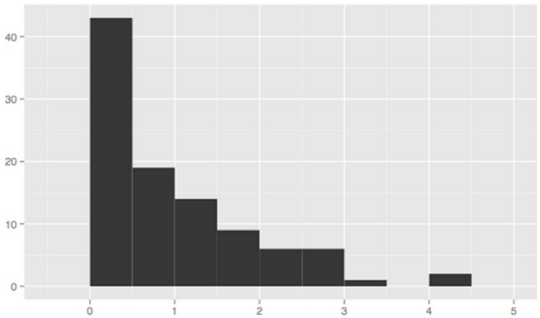
- $h(r_k) = n_k$ versus r_k
- $p(r_k) = n_k / MN$ versus r_k

where MN is the total number of pixels

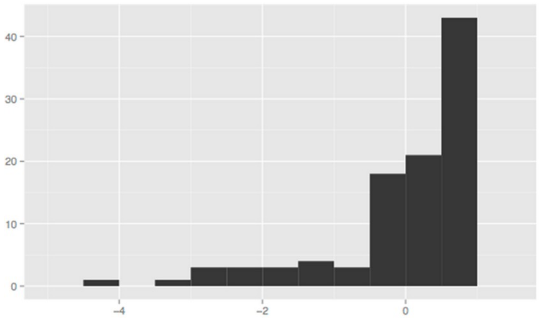
Histogram of Image



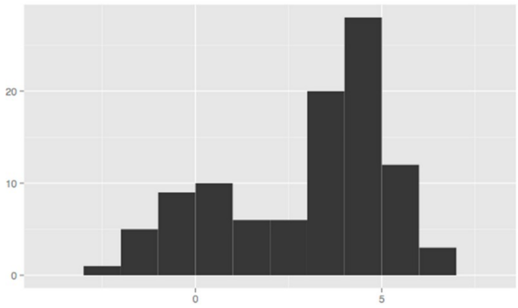
Symmetric, unimodal



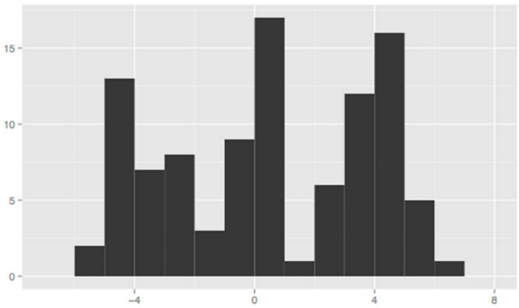
Skewed, right



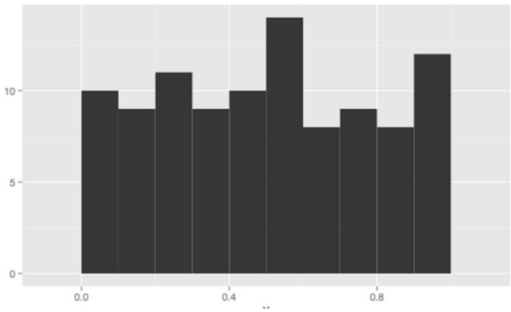
Skewed, left



Bimodal



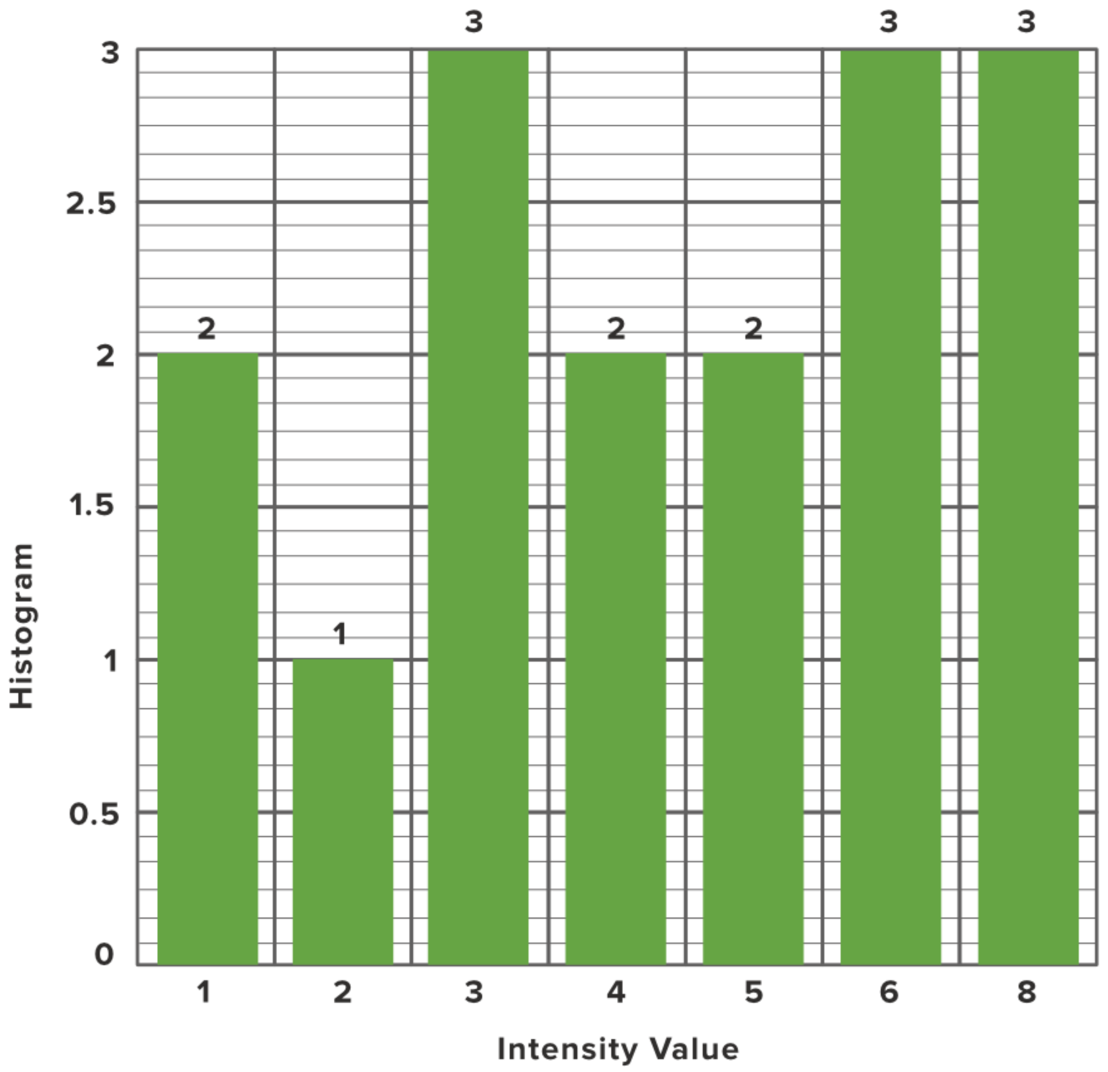
Multimodal



Symmetric

Histogram of Image

$$\begin{bmatrix} 3 & 6 & 6 & 8 \\ 5 & 3 & 1 & 4 \\ 8 & 6 & 5 & 1 \\ 4 & 8 & 2 & 3 \end{bmatrix}$$

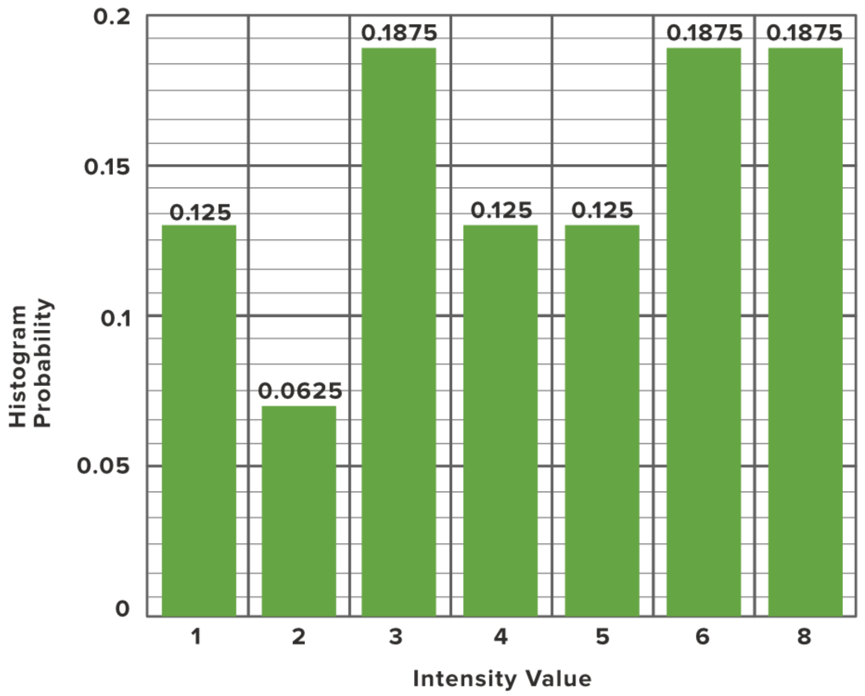


Histogram of Image

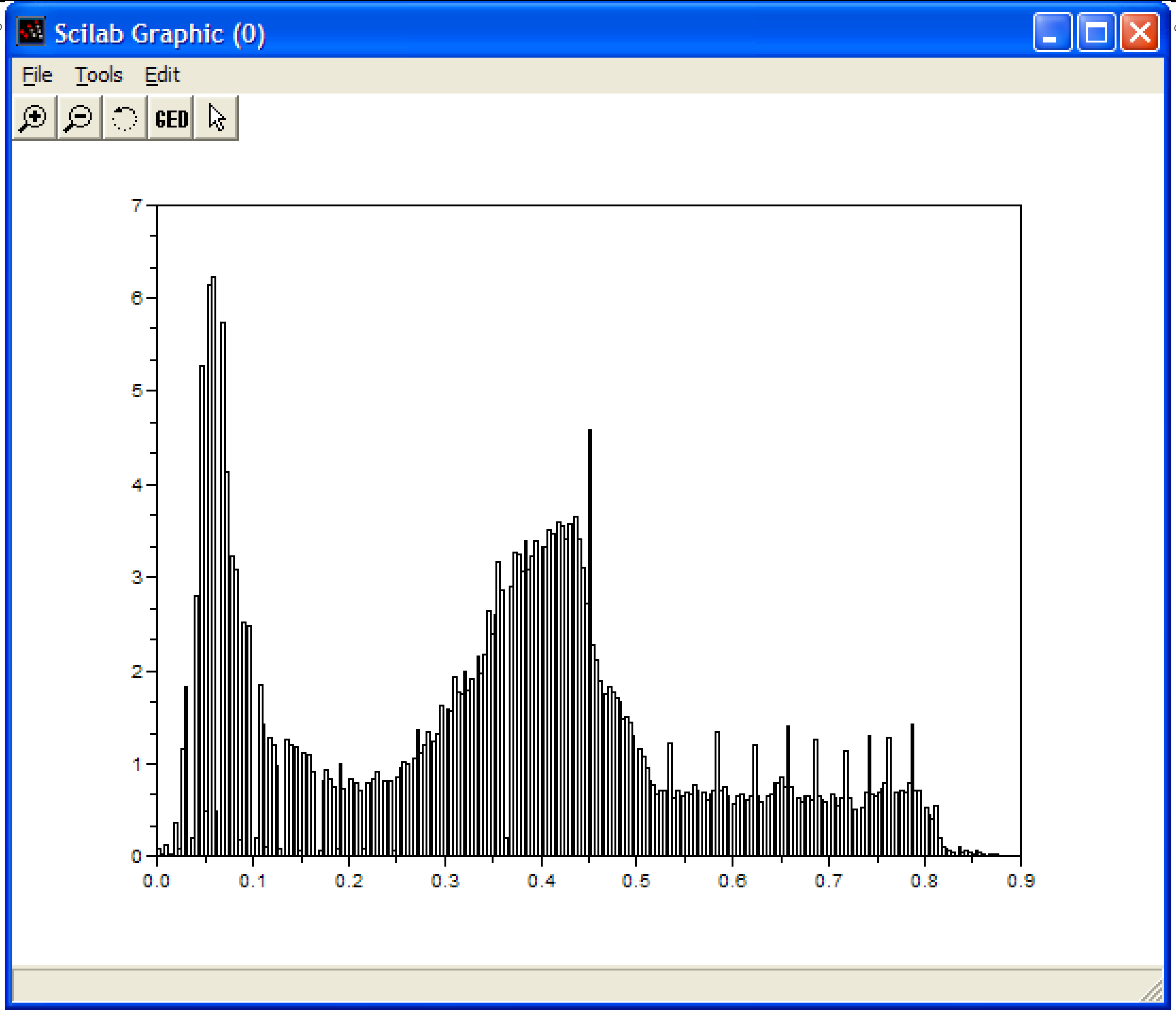
$$\begin{bmatrix} 3 & 6 & 6 & 8 \\ 5 & 3 & 1 & 4 \\ 8 & 6 & 5 & 1 \\ 4 & 8 & 2 & 3 \end{bmatrix}$$

Intensity Value/ Gray Level	#/Occurrence of intensity value/ number of pixels	Probability
1	2	0.125
2	1	0.0625
3	3	0.1875
4	2	0.125
5	2	0.125
6	3	0.1875
8	3	0.1875

$$P(r_k) = \frac{n_k}{n}$$

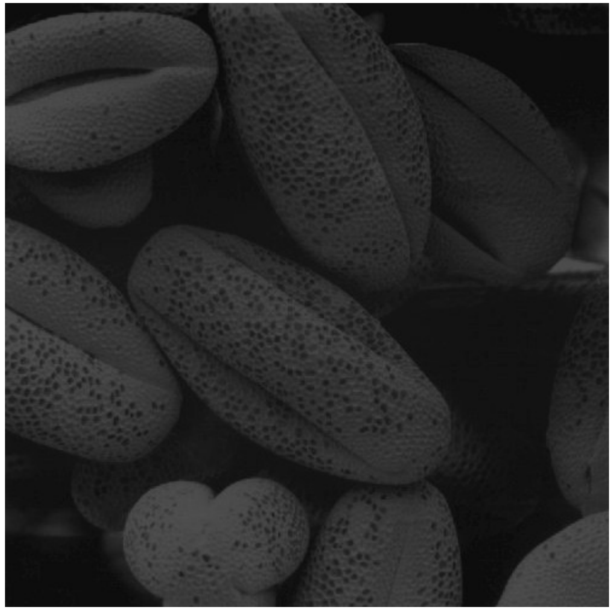


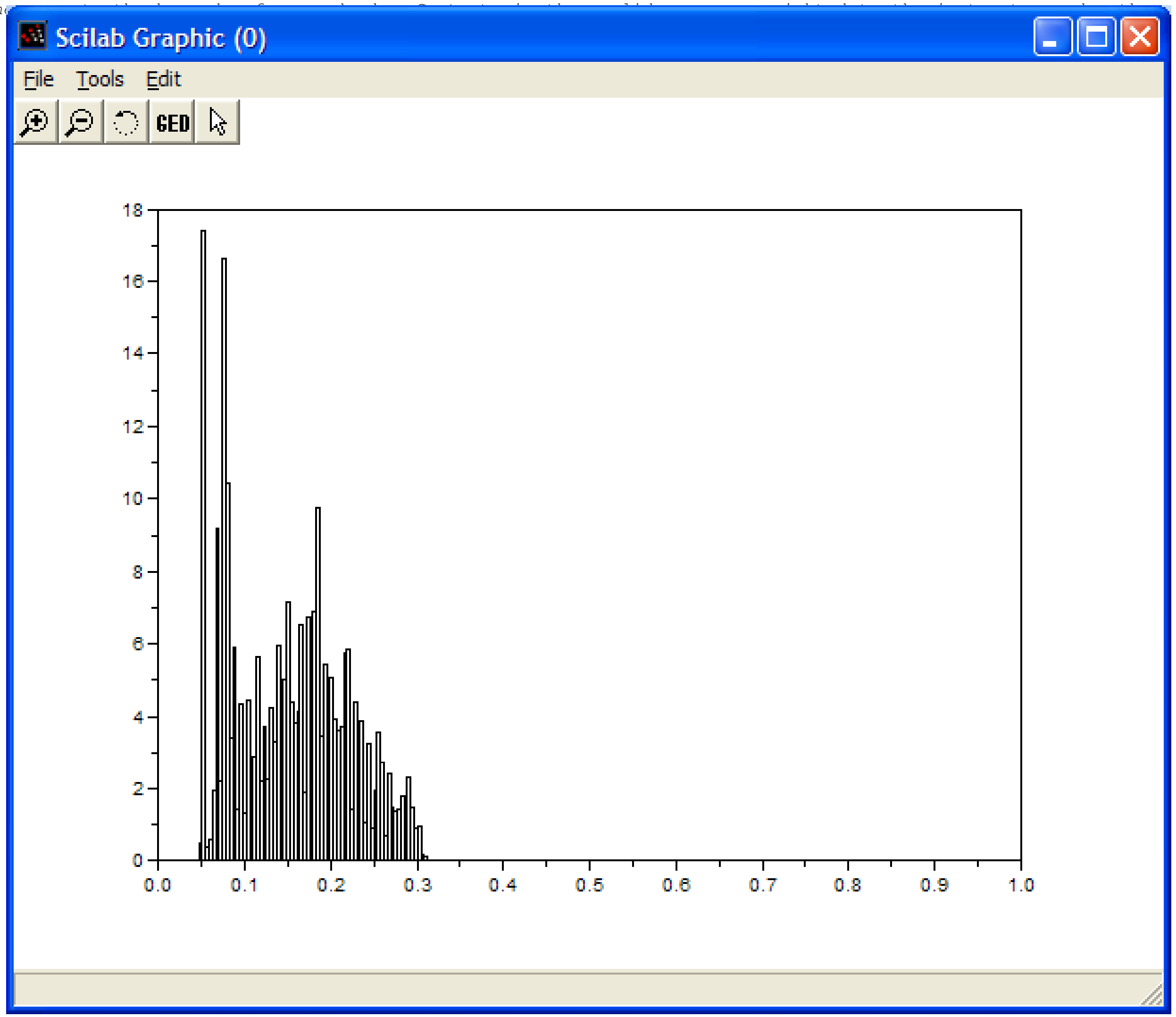
Histogram of Image



Histogram of Image

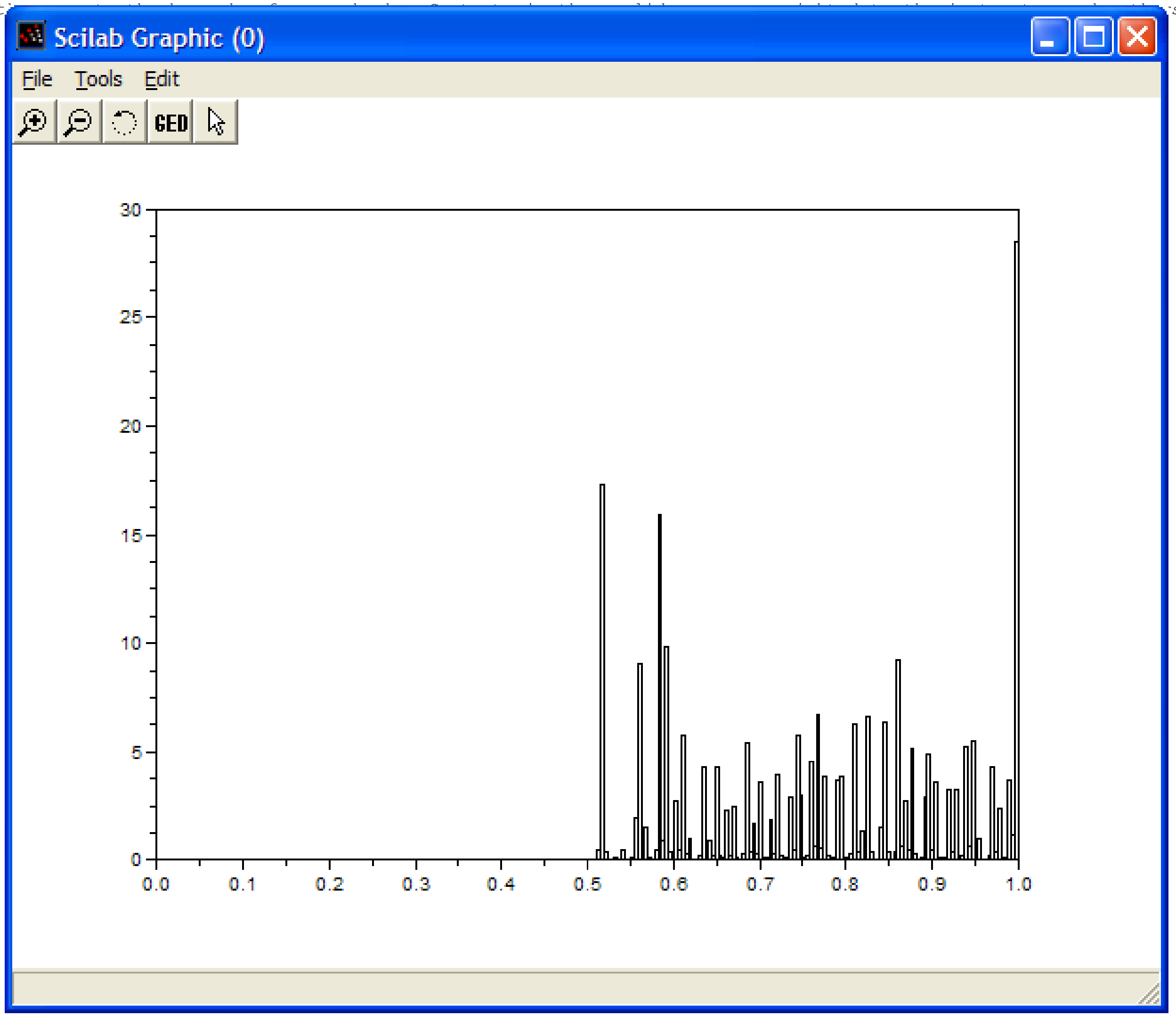
In dark Image, the components of the histogram are concentrated on the dark (low) side of the gray scale





Histogram of Image

In Bright Image, the components of the histogram are concentrated on the bright (high) side of the gray scale



Histogram Equalization

- Histogram equalization is the process of increasing the contrast in an image by spreading the histogram out to be approximately uniformly distributed.
- The gray level of an image that has been subjected to histogram equalization are spread out. The increase of dynamic range produces an increase in contrast.

For Continuous Image:

$$S = T(r) = (L-1)\int_0^r P_r(w)\,dw$$

For Discrete Image

$$S_k = T(r_k) = (L-1)\sum_{j=0}^{k} P_r(r_j)$$

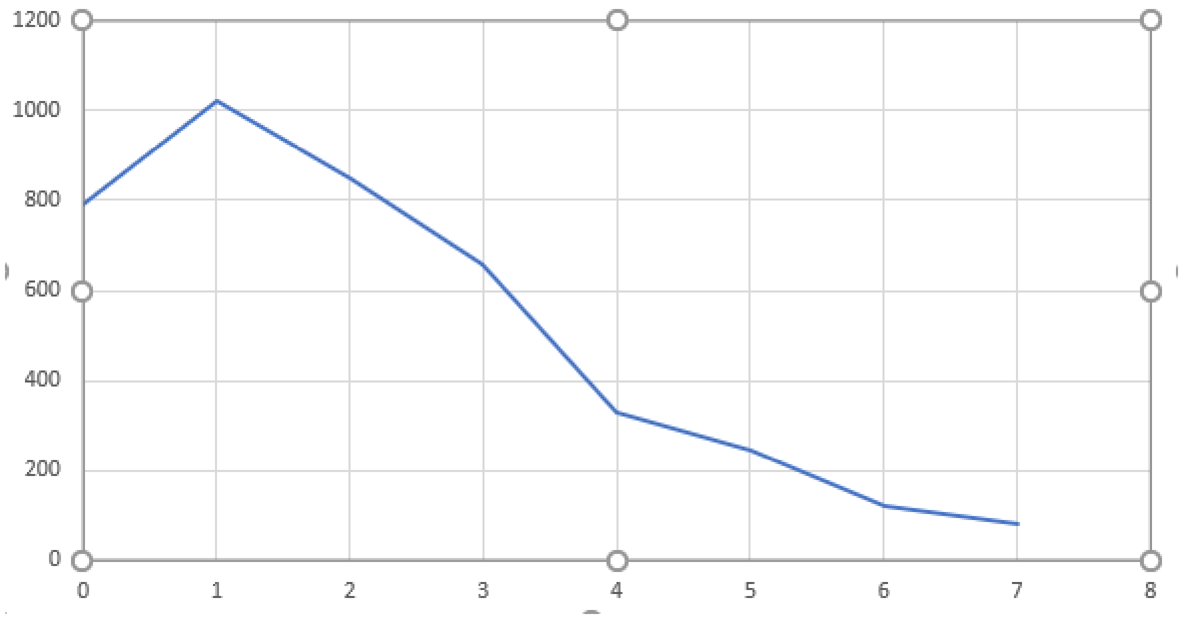
Histogram Equalization

Suppose a 3bit image [L=8) of size 64 *64 has the intensity distribution. Shown as following table. Get the histogram equation transformation

Gray Level	0	1	2	3	4	5	6	7
No. Of Pixel	790	1023	850	656	329	245	122	81

Histogram Equalization

Gray Level	0	1	2	3	4	5	6	7
No. Of Pixel	790	1023	850	656	329	245	122	81

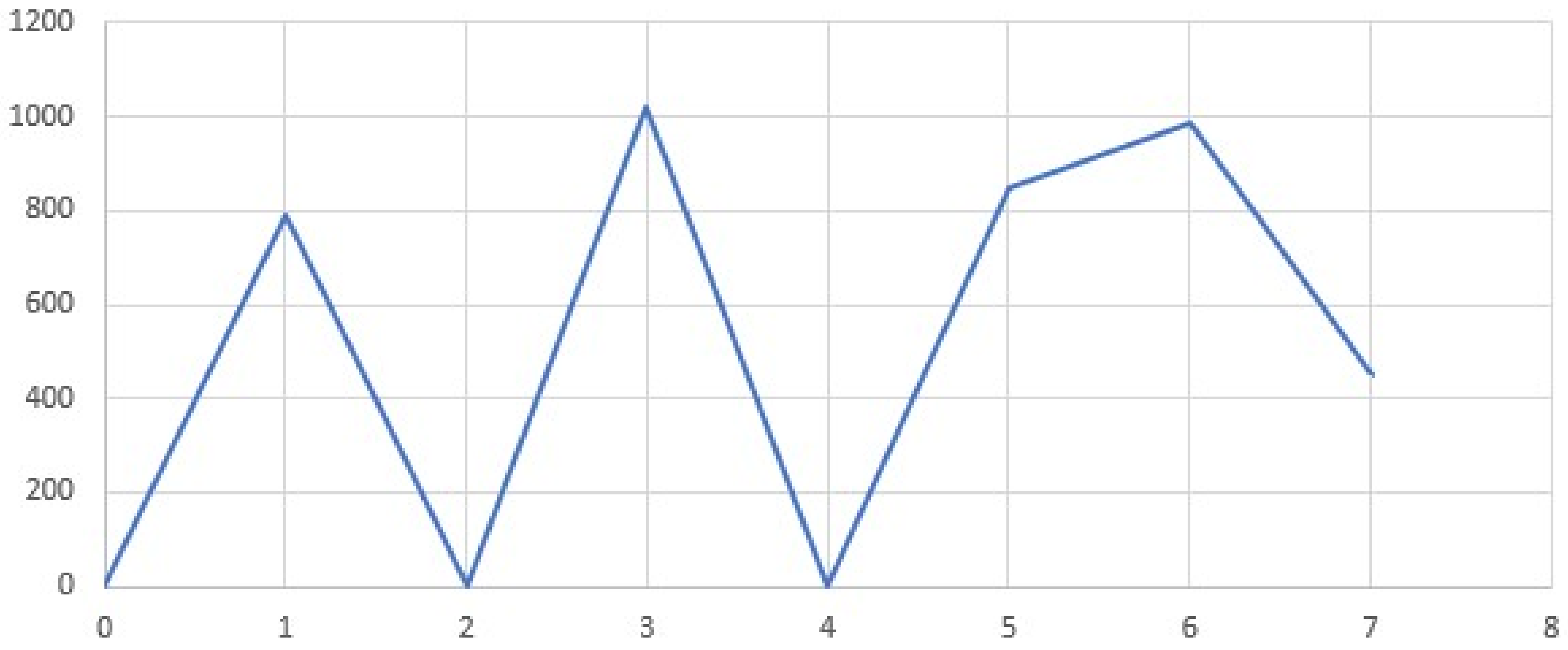


Histogram Equalization

r_k	No. of Pixel	PDF ($P(r_k)$) (n_k/N)	CDF	(L-1)*CDF	S_k
0	790	0.19	0.19	1.33	1
1	1023	0.25	0.44	3.08	3
2	850	0.21	0.65	4.55	5
3	656	0.16	0.81	5.67	6
4	329	0.08	0.89	6.23	6
5	245	0.06	0.95	6.65	7
6	122	0.03	0.98	6.86	7
7	81	0.02	1	7	7

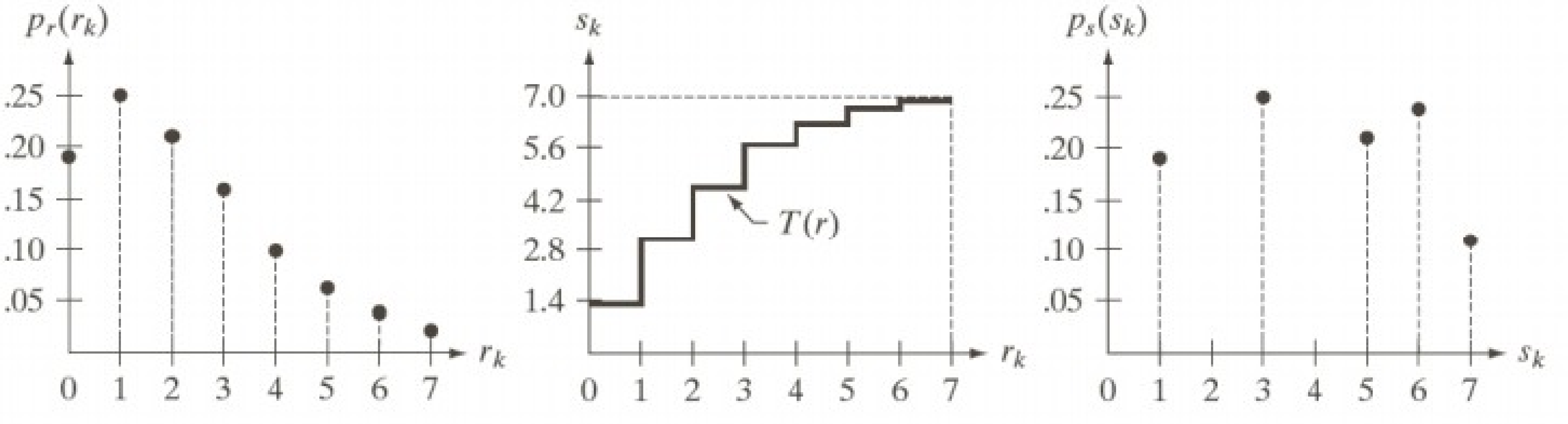
Histogram Equalization

Gray Level	0	1	2	3	4	5	6	7
No. Of Pixel	0	790	0	1023	0	850	656+329	245+122+81

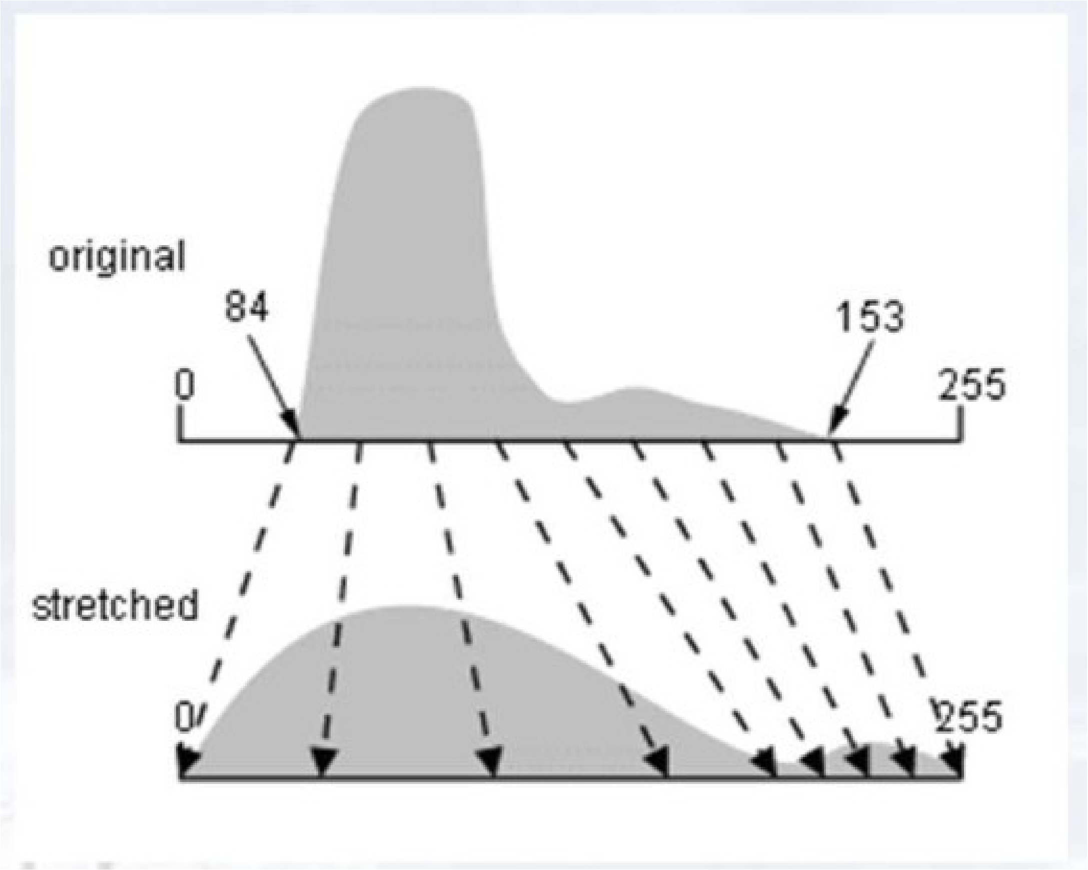


Histogram Equalization

Gray Level	0	1	2	3	4	5	6	7
No. Of Pixel	0	790	0	1023	0	850	656+329	245+122+81



Histogram Stretching



Histogram Stretching

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

S_{min} = Minimum Gray Level of o/p Image

S_{max} = Maximum Gray Level of o/p Image

r_{max} = Maximum Gray Level of i/p Image

r_{min} = Minimum Gray Level of i/p Image

Histogram Stretching

Perform histogram stretching. Show that the new image has dynamic range of entire range 0-7

Gray Level	0	1	2	3	4	5	6	7
No. Of Pixel	0	0	50	60	50	20	10	0

Here,

S_{min} =0

S_{max} =7

r_{min} =2

r_{max} =6

$$When\ r = 2 \qquad S = \frac{7-0}{6-2}(2-2)+0=0$$

$$When\ r = 3 \qquad S = \frac{7-0}{6-2}(3-2)+0=2$$

$$When\ r = 4 \qquad S = \frac{7-0}{6-2}(4-2)+0=4$$

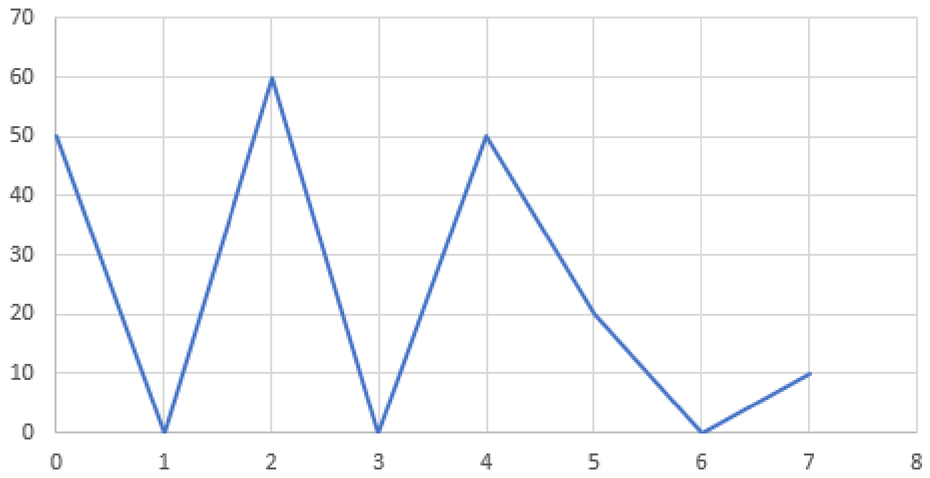
$$When\ r = 5 \qquad S = \frac{7-0}{6-2}(5-2)+0=5$$

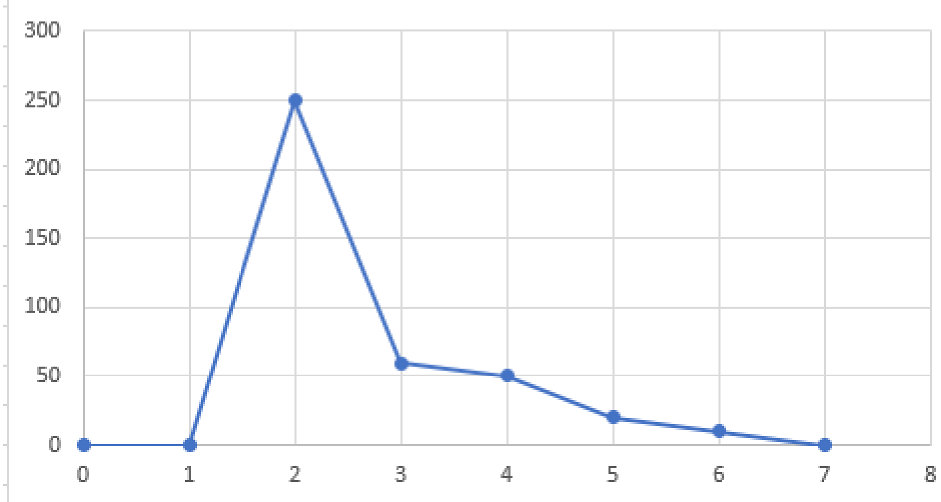
$$When\ r = 6 \qquad S = \frac{7-0}{6-2}(6-2)+0=7$$

Histogram Stretching

Gray Level	0	1	2	3	4	5	6	7
No. Of Pixel	50	0	60	0	50	20	0	10

Gray Level	0	1	2	3	4	5	6	7
No. Of Pixel	0	0	50	60	50	20	10	0





Histogram Statistics for Image Enhancement

Let r denote a discrete random variable representing intensity values in [0, L-1] and let $p(r_i)$ denote the normalized histogram component corresponding to value r

.

We may view $p(r_i)$ as an estimate of the probability that r_i occurs in the image

$$\mu_n = \sum_{i=1}^{l-1}(r_i - m)^n \, p(r_i)$$

n^{th} moment about its mean

m is the mean(average intensity)

$$m = \sum_{i=1}^{l-1} r_i \, p(r_i)$$

$$\mu_l = \sum_{i=1}^{l-1}(r_i - m)^2 \, p(r_i) = \sigma^2$$

Global mean and variance is computed on an entire image and useful for gross adjustment in overall intensity and contrast. A more powerful use of these parameters is in local enhancement, where local mean and variance are used as the basis for making changes.

Local Histogram Processing

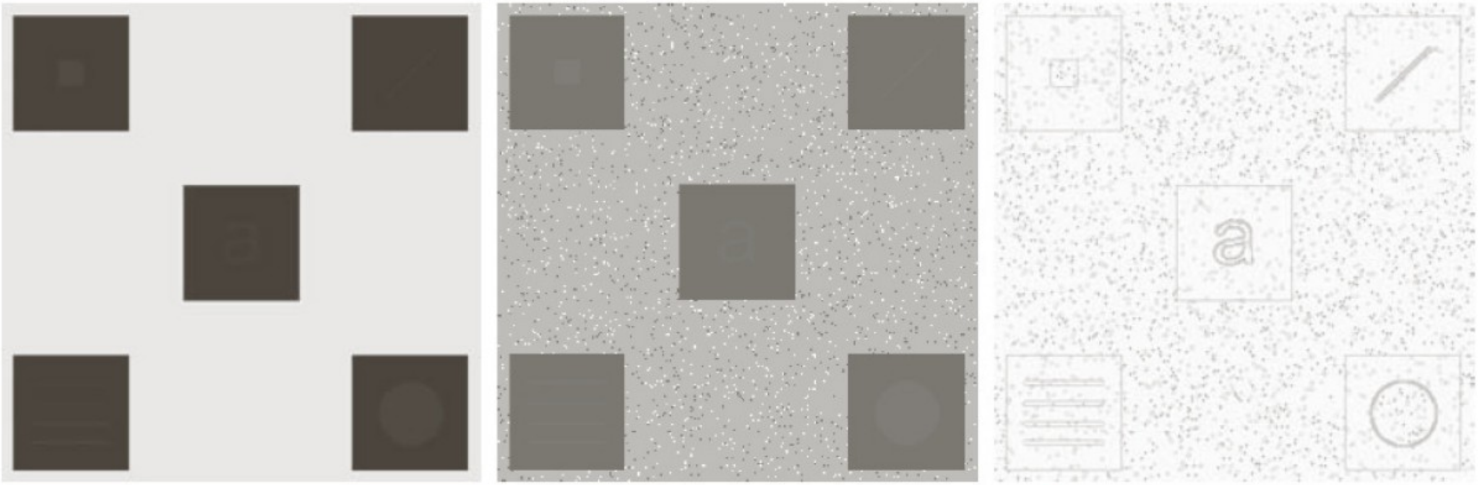
Problem:

- global spatial processing not always desirable

Solution:

-apply point-operations to a pixel neighborhood with a sliding window

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9



a b c

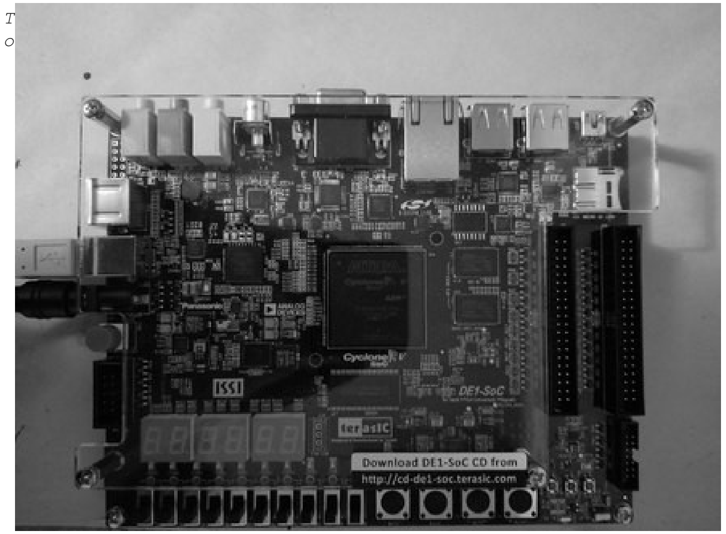
FIGURE 3.26 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization applied to (a), using a neighborhood of size 3×3.

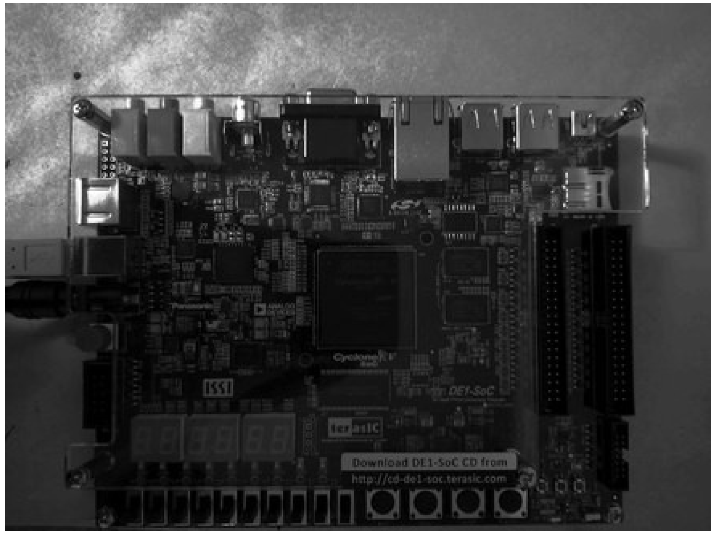
Histogram Matching

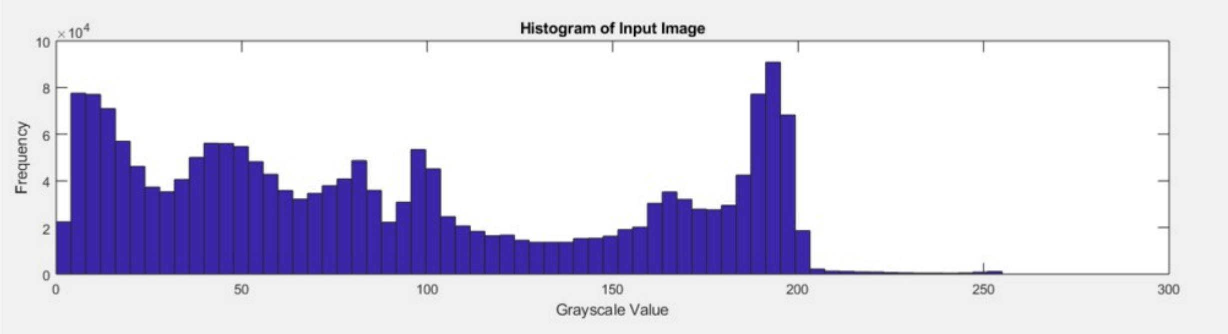
Histogram matching or histogram specification is the transformation of an image so that its histogram matches a specified histogram. The well-known histogram equalization method is a special case in which the specified histogram is uniformly distributed.

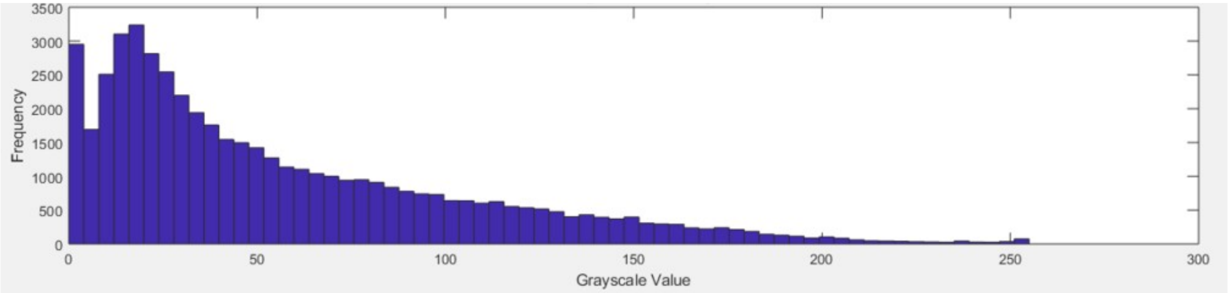
- It is possible to use histogram matching to balance detector responses as a relative detector calibration technique.
- It can be used to normalize two images, when the images were acquired at the same illumination, but by different sensors, atmospheric conditions or global illumination.

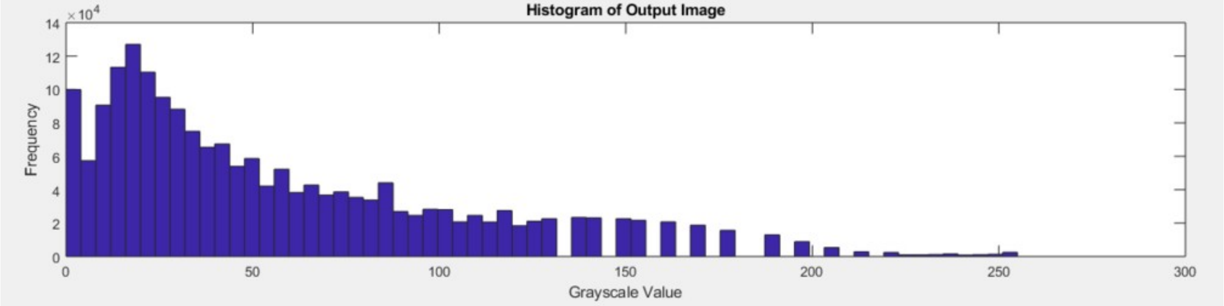
Histogram Matching











Histogram Matching

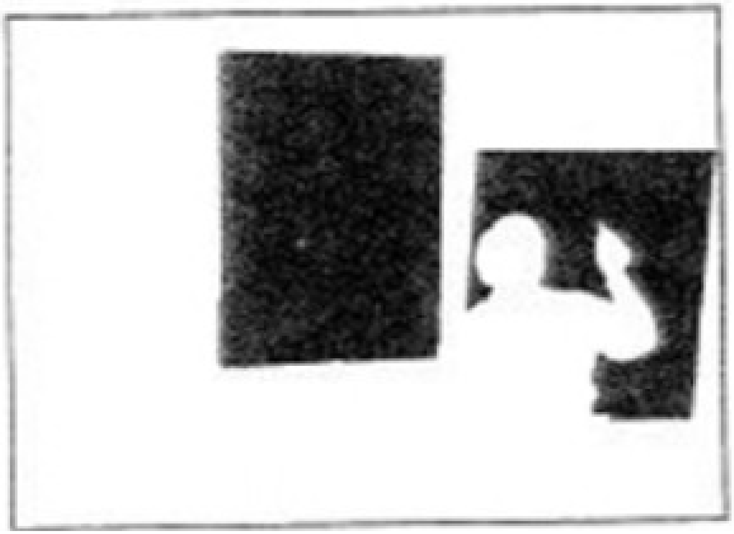
Image Enhancement using Arithmetic Operations

Arithmetic/logic operations involving images are performed on a pixel-by-pixel basis between two or more images (this excludes the logic operation NOT, which is performed on a single image).

As an example, subtraction of two images resulted in a new image whose pixel at co-ordinates (x , y) is the difference between the pixels in that same location in the two images being subtracted.

Image Enhancement using Arithmetic Operations

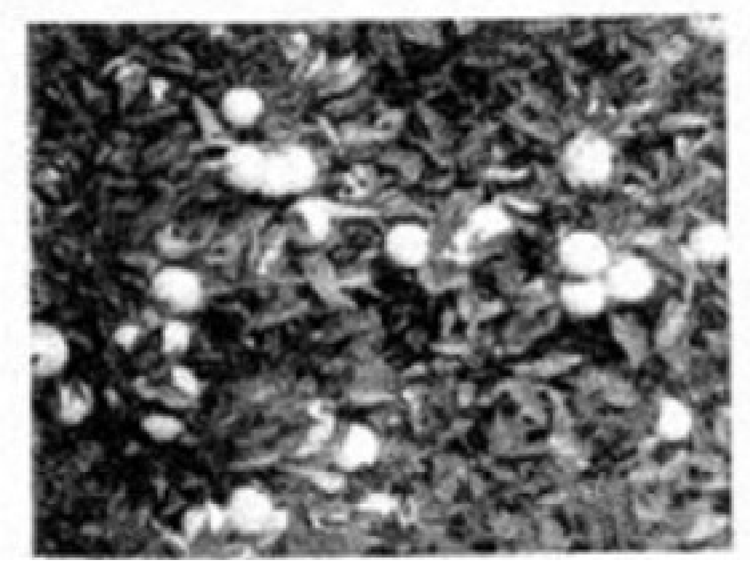
Original Image



AND image Mask

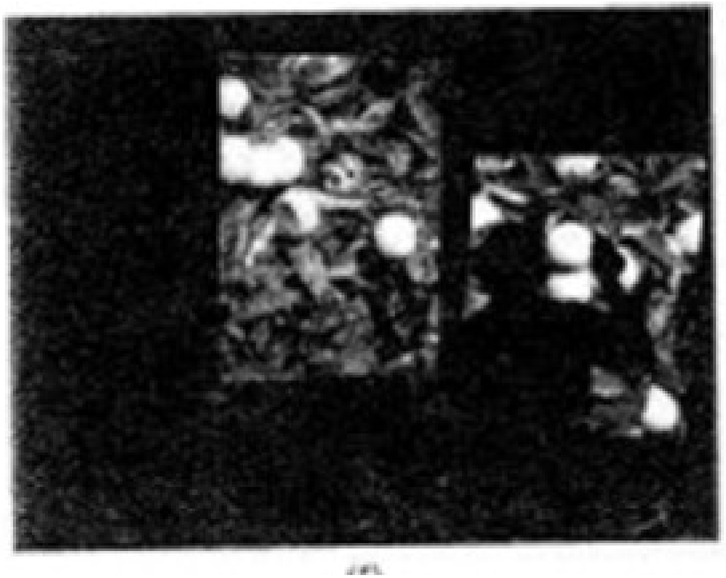
Result of AND

Image Enhancement using Arithmetic Operations



Original Image

OR image Mask



Result of OR

Image Enhancement using Arithmetic Operations
Image Subtraction

- Image subtraction or pixel subtraction is a process whereby the digital numeric value of one pixel or whole image is subtracted from another image.

- The difference between two images f(x,y) and h(x,y) expressed as

$$g(x,y) = f(x,y) - h(x,y)$$

- The key usefulness of subtraction is the enhancement of differences between images

Image Enhancement using Arithmetic Operations
Image Subtraction

Image Enhancement using Arithmetic Operations
Image Averaging

- Consider a noisy image g(x,y) formed by the addition of noise h(x,y) to an original image f(x,y) expressed as

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

The assumption is that at every pair of coordinates (x,y) the noise is uncorrelated has zero average value.

- The objective of the procedure is to reduce the noise content by adding a set of noisy images.

Image Enhancement using Arithmetic Operations
Image Averaging

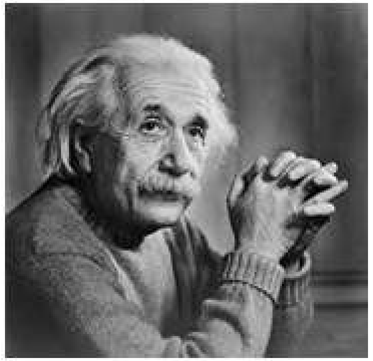
Magnification

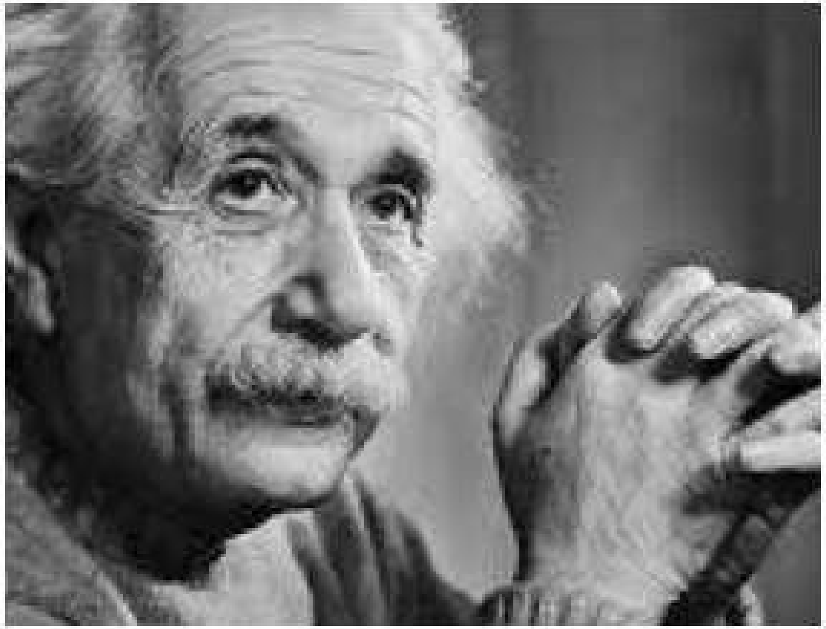
- Image Magnification refers to the proportional increase in the dimensions of a image relative to the actual dimensions.

- Types
  - Replication
  - Interpolation

Magnification

- Enlarging a picture in a sense that the details in the image became more visible and clear.
- It involves two steps:
- \- Creating new pixel location
- \- Assigning Gray Level to those new locations





Magnification

Pixel Replication(Zero Order Hold)

- Replication is a zero order hold where each pixel along a scan line is repeated and then each scan line is repeated.

- This is equivalent to taking M*N image and interlace it by rows and columns of zeros to obtain a 2M * 2N matrix.

Magnification

Pixel Replication(Zero Order Hold)

$$\begin{bmatrix} 69 & 50 & 80 \\ 45 & 60 & 66 \\ 30 & 55 & 80 \end{bmatrix} = \begin{bmatrix} 69 & 69 & 50 & 50 & 80 & 80 \\ 45 & 45 & 60 & 60 & 66 & 66 \\ 30 & 30 & 55 & 55 & 80 & 80 \end{bmatrix} = \begin{bmatrix} 69 & 69 & 50 & 50 & 80 & 80 \\ 69 & 69 & 50 & 50 & 80 & 80 \\ 45 & 45 & 60 & 60 & 66 & 66 \\ 45 & 45 & 60 & 60 & 66 & 66 \\ 30 & 30 & 55 & 55 & 80 & 80 \\ 30 & 30 & 55 & 55 & 80 & 80 \end{bmatrix}$$

Original image | image with rows expanded | image with rows and columns expanded

Magnification
Zero Interlace

Suppose a 2*2 Image :

Step 1: Adding Zero after every pixel in every scan line

1	0	3	0
4	0	2	0

Step 2: Adding the scan line with complete zero after the every scan line. (Inserting the row of Zero is called Zero Interlace)

1	0	3	0
0	0	0	0
4	0	2	0
0	0	0	0

Linear Interpolation

Linear Interpolation is derived from the Zero-Interlace technique

In this method, instead of replicating each pixel, average of the two adjacent pixel along the row is taken and placed between the two pixels

Accordingly, the same operation is then performed along the column

Magnification
Linear Interpolation

Step 1: Zero Interlace

1	0	3	0
0	0	0	0
4	0	2	0
0	0	0	0

Step 2: Interpolate Row

1	2	3	1.5
0	0	0	0
4	3	2	1
0	0	0	0

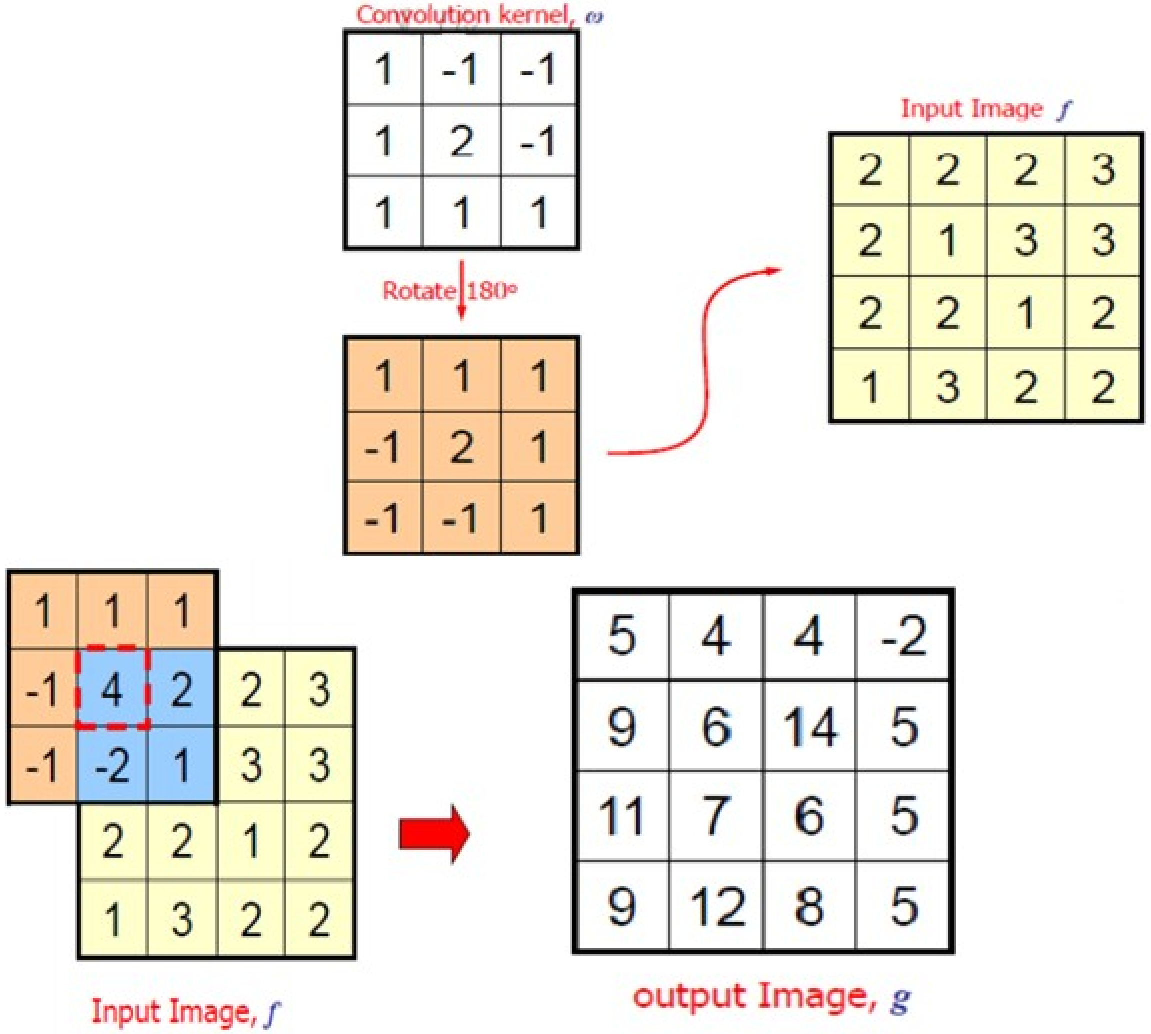
Step 3 : Interpolate Columns

1	2	3	1.5
2.5	2.5	2.5	1.25
4	3	2	1
2	1.5	1	0.5

Convolution

- Convolution involves calculating the weighted sum of a neighborhood of pixels.
- The weights are taken from a convolution kernel. Each value from the neighborhood of pixels is multiplied with its opposite on the matrix.
- For example, the top-left of the neighbor is multiplied by the bottom-right of the kernel.

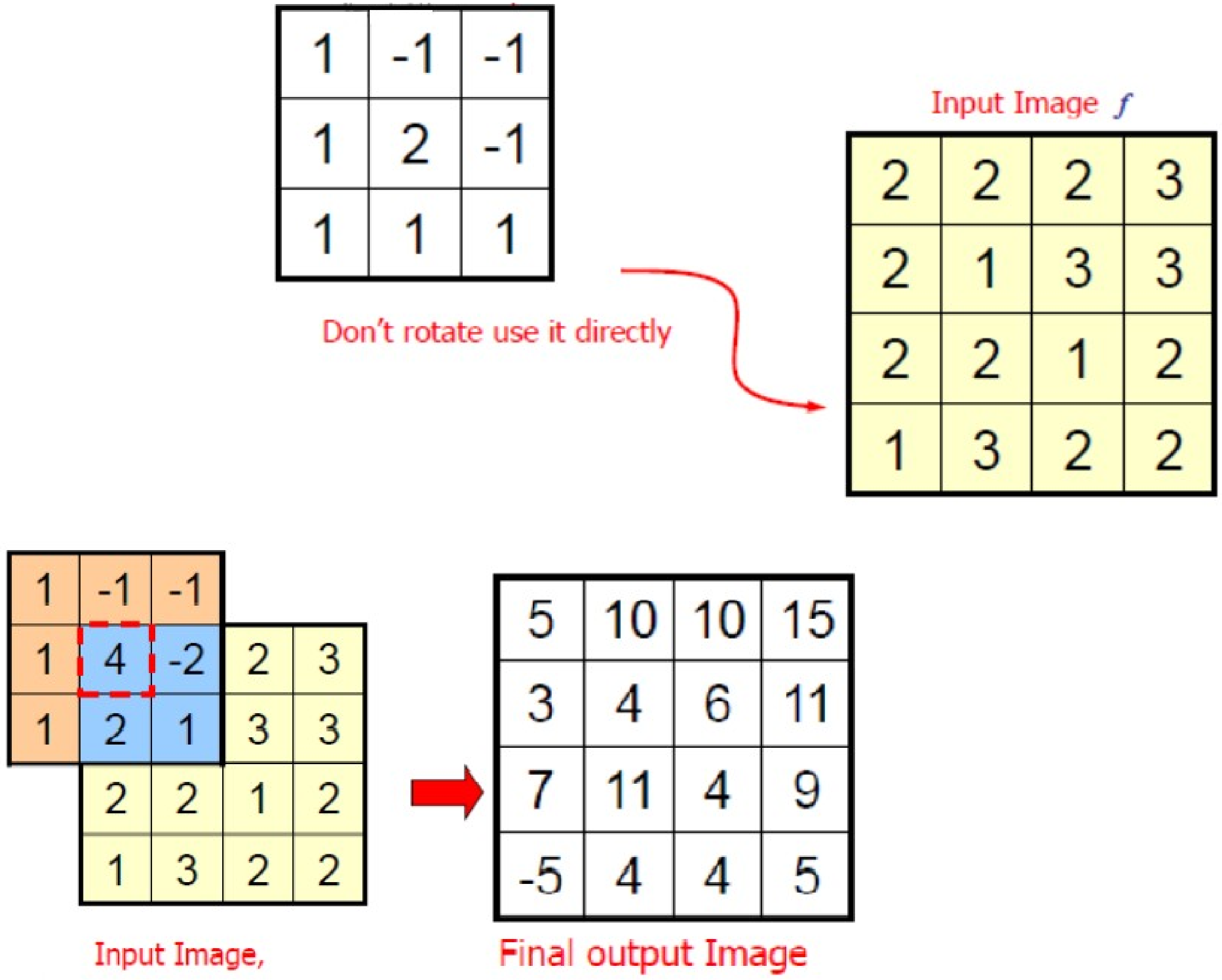
Convolution Example



Correlation

Correlation is nearly identical to convolution with only a minor difference, where instead of multiplying the pixel by the opposite in the kernel, you multiply it by the equivalent (i.e. top-left multiplied by top-left)

Correlation Example



Spatial Filtering

- Filtering is a technique for modifying or enhancing an image.
  For example, we can filter an image to emphasize certain features or remove other features.

- Image processing operations implemented with filtering include smoothing, sharpening, and edge enhancement.

- Spatial filtering refers to some neighborhood operations working with the values of the image pixels in the neighborhood and the corresponding values of a sub image that has the same dimensions as the neighborhood

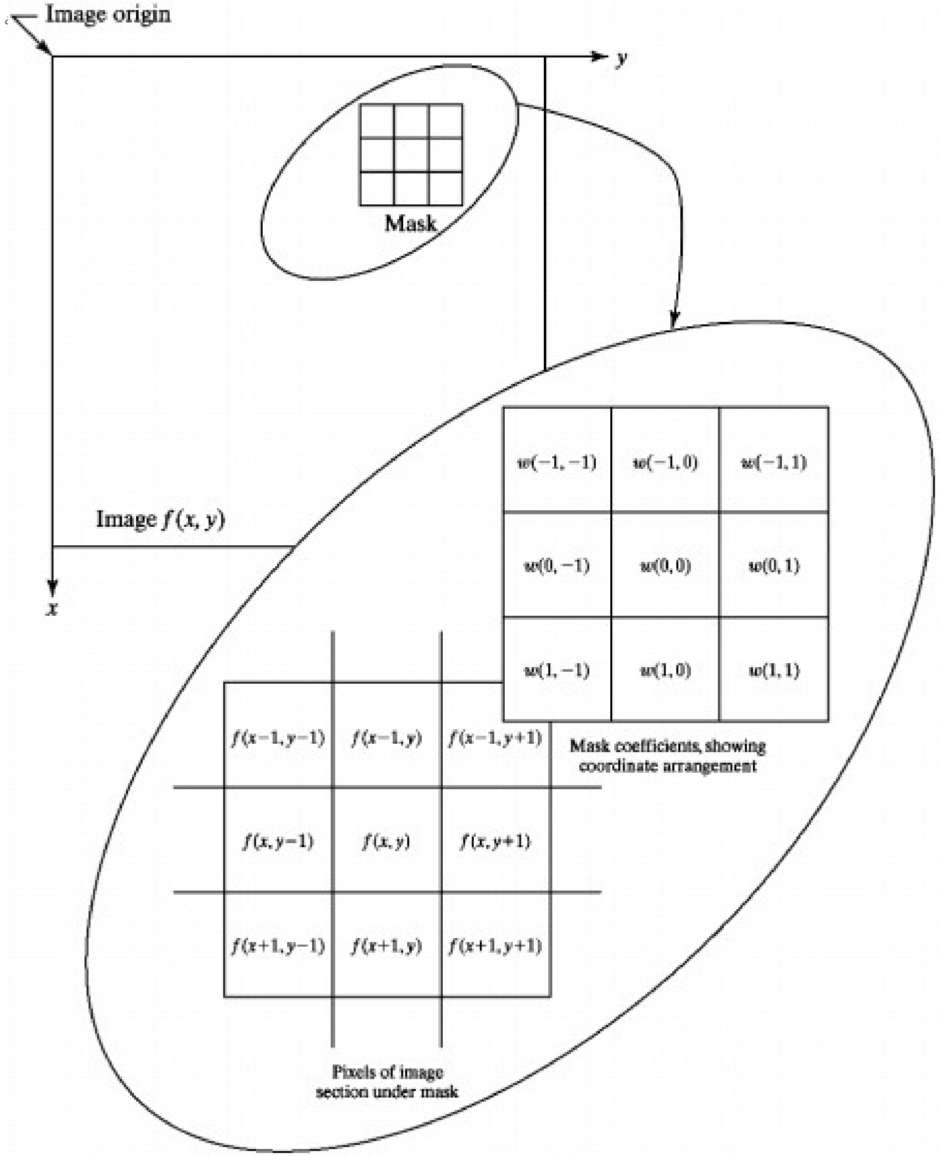
Neighbourhood Processing
Spatial masks

The sub image used in spatial filtering is called a filter, mask, kernel, template or a window

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

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

where a =(m - 1)/2, b =(n - 1)/2



Spatial Averaging

- In spatial averaging, each pixel is replaced by the weighted average of its neighborhood pixel

$$v(m,n) = \sum_{K,L\,\in w} a(k,l)\, y(m-K, n-L)$$

$y(m-K, n-L)$ = input image
$v(m,n)$ = output image
w = choosen window
$a(k,l)$ = filter Weight

$$\begin{array}{cc} & L \\ K & \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} \end{array}$$

- A common class of spatial average filter has all equal weight

$$v(m,n) = \frac{1}{N_w} \sum_{K,L\,\in w} y(m-K, n-L)$$

$\frac{1}{N_w}$ is the number of pixel in window.

- Spatial average is used for noise smoothing , low pass filtering and sub sampling of the image

Spatial Filter Types

- Linear Filter

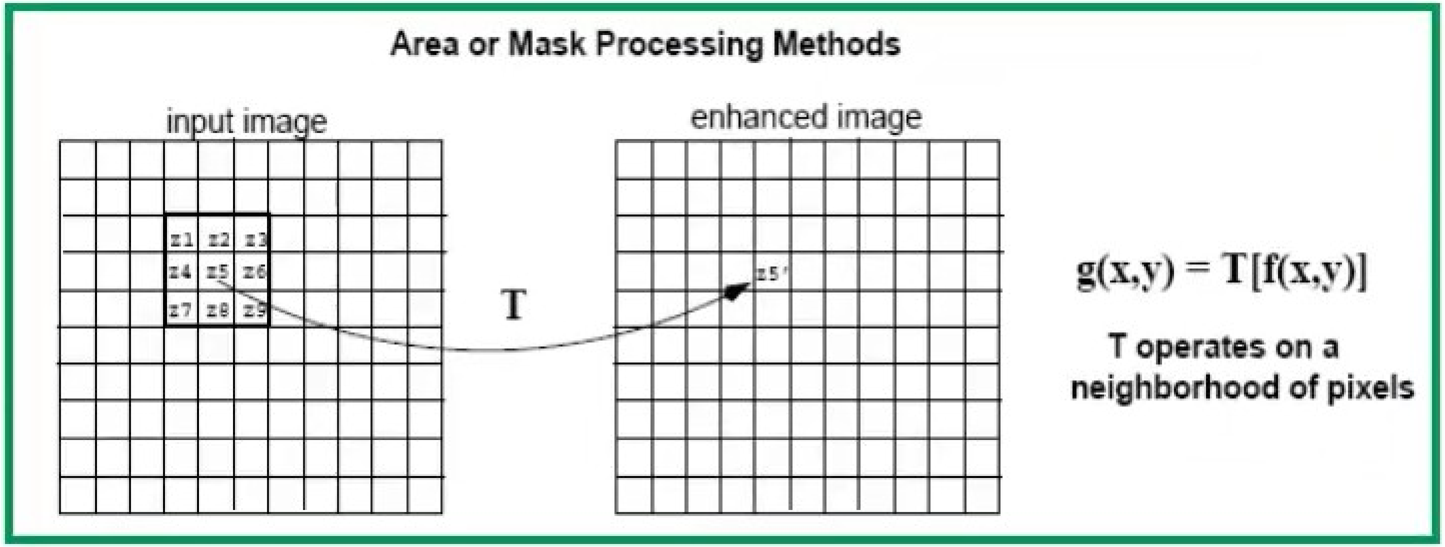
  Linear filtering is the filtering method in which the value of output pixel is linear combinations of the neighboring input pixels.

- Non Linear Filter

  A non-linear filtering is one where output is not the linear combination of the inputs

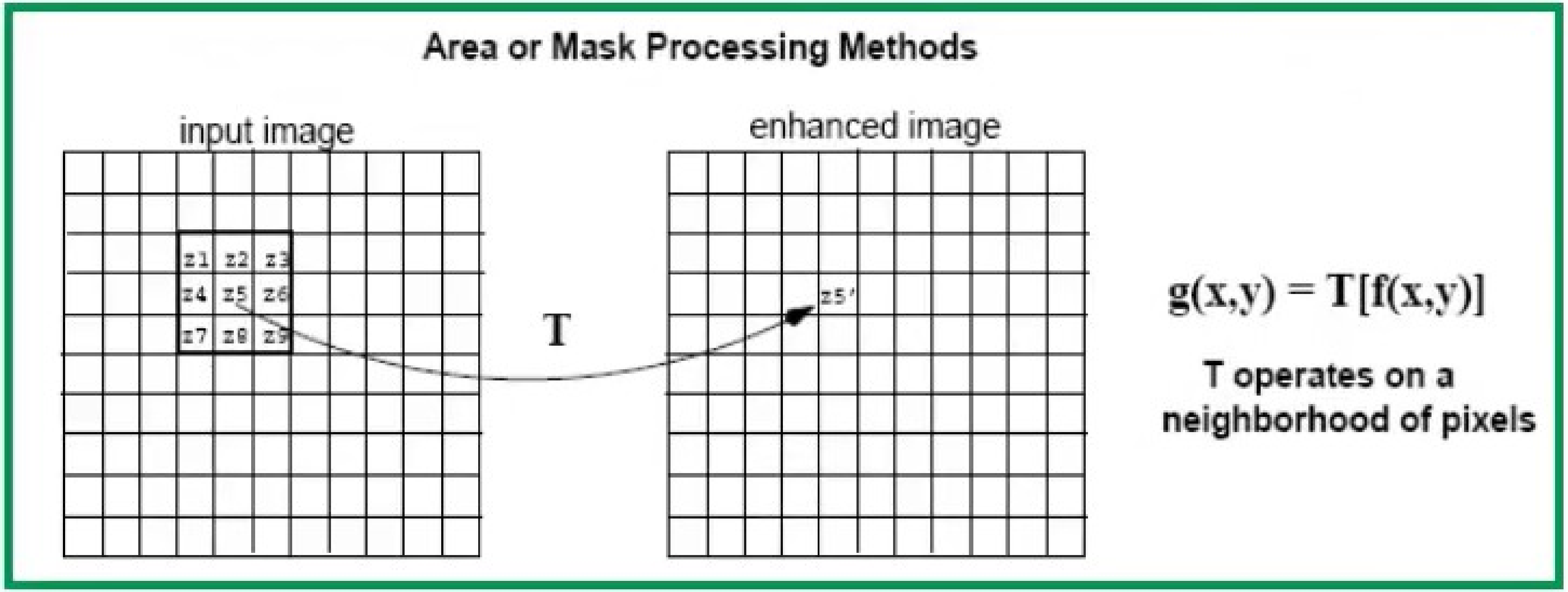
Spatial Filter Linear

$$z'_5 = 5z_1 - 3z_2 + z_3 - z_4 - 2z_5 - 3z_6 + z_8 - z_9 - 9z_7$$



Spatial Filter
Non Linear

$$z'_5 = \max(z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8, z_9)$$



Types of Linear Filter

- Low Pass Filter / Smoothing Filter
  - Attenuate (or eliminate) high frequency component such as characterized by edges and sharp details in an image.
  - Net effect is image blurring.
  - The simplest low-pass filter just calculates the average of a pixel and all of its eight immediate neighbors. The result replaces the original value of the pixel. The process is repeated for every pixel in the image.

Types of Linear Filter

- Low Pass Filter

Before and After Low-Pass Filter

Types of Linear Filter

❑Low Pass Filter

To perform a low-pass filter by simply averaging adjacent pixels, the following kernel is used:

+1/9	+1/9	+1/9
+1/9	+1/9	+1/9
+1/9	+1/9	+1/9

0	+1/8	0
+1/8	+1/2	+1/8
0	+1/8	0

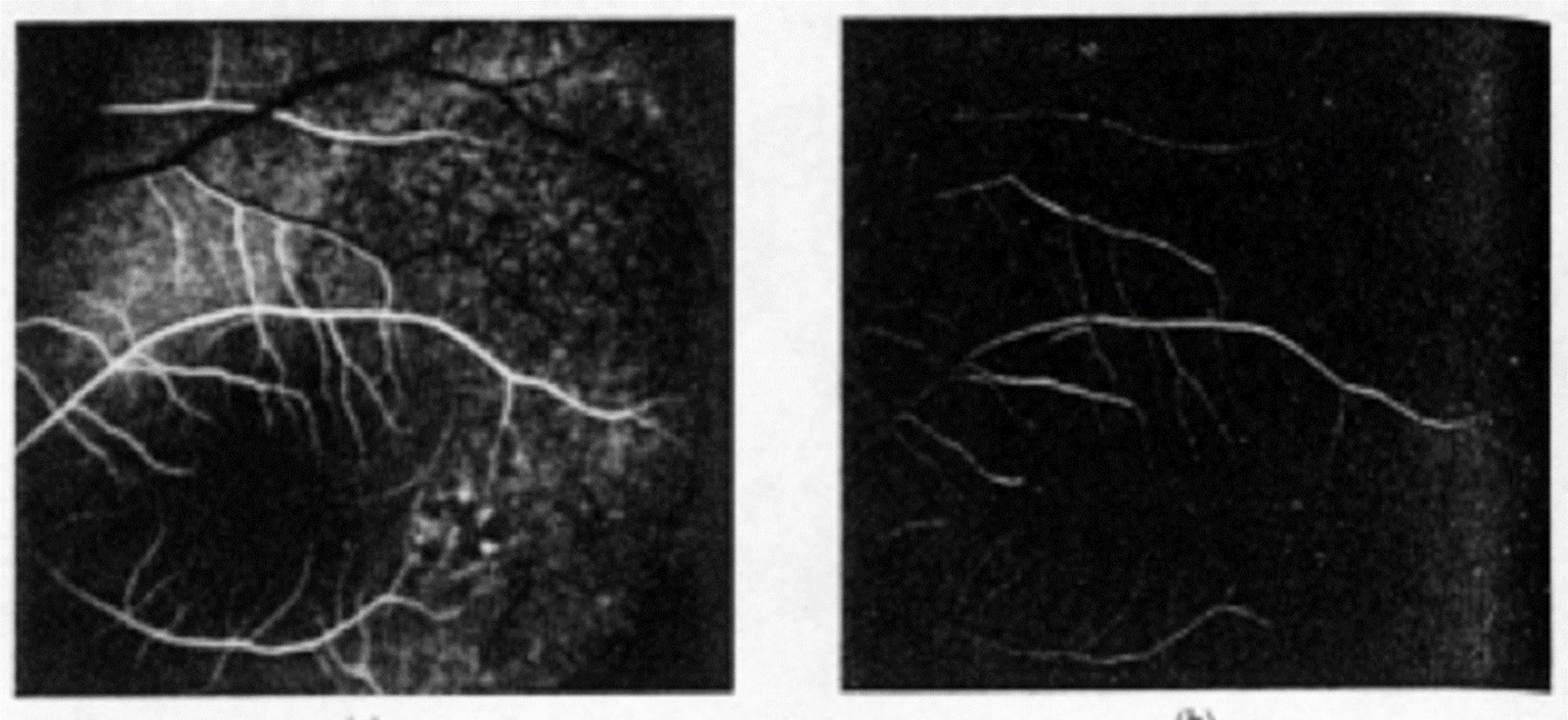
Types of Linear Filter

- High Pass Filter / Sharpening Filter
  - Attenuate (or eliminate) low frequency components such as slowly varying characteristics.
  - Net effect is a sharpening of edges and other details
  - These filters emphasize fine details in the image – exactly the opposite of the low-pass filter.

0	-1/4	0
-1/4	+2	-1/4
0	-1/4	0

Types of Linear Filter

- High Pass Filter

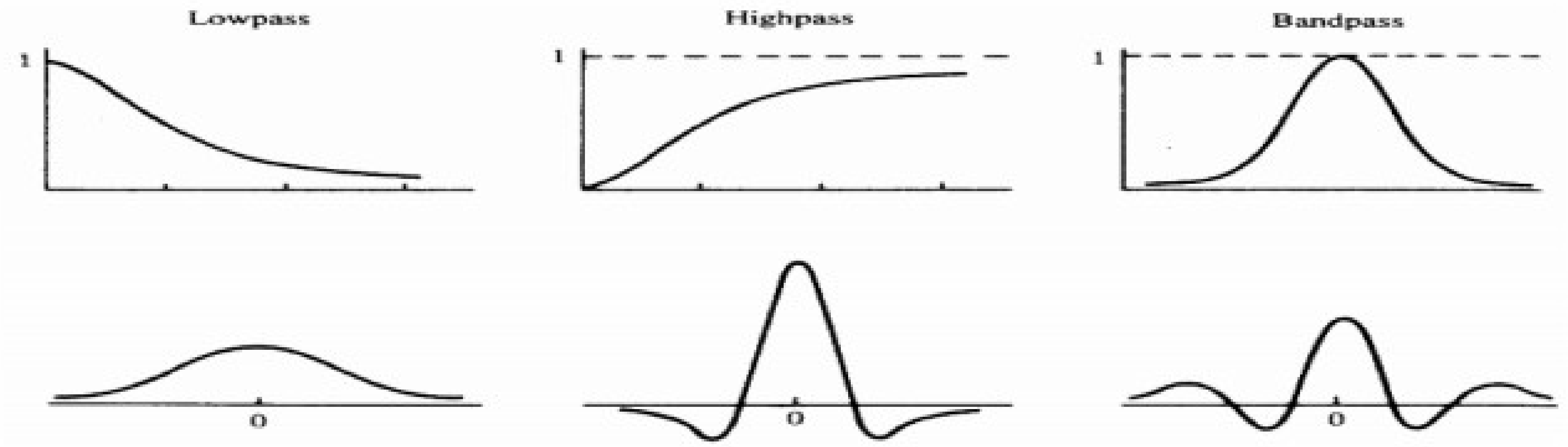


(a) (b)

Types of Linear Filter

Band Pass Filter

- Attenuate (or eliminate) a given frequency range.
- It is used primarily for image restoration



Sharpening Filters (High Boost)

It is often desirable to emphasize high frequency components representing the image details (by means such as sharpening) without eliminating low frequency components representing the basic form of the signal.

In this case, the high-boost filter can be used to enhance high frequency component while still keeping the low frequency components

Sharpening Filters (High Boost)

- A high-pass filter may be computed as:

  High-pass = Original - Lowpass

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

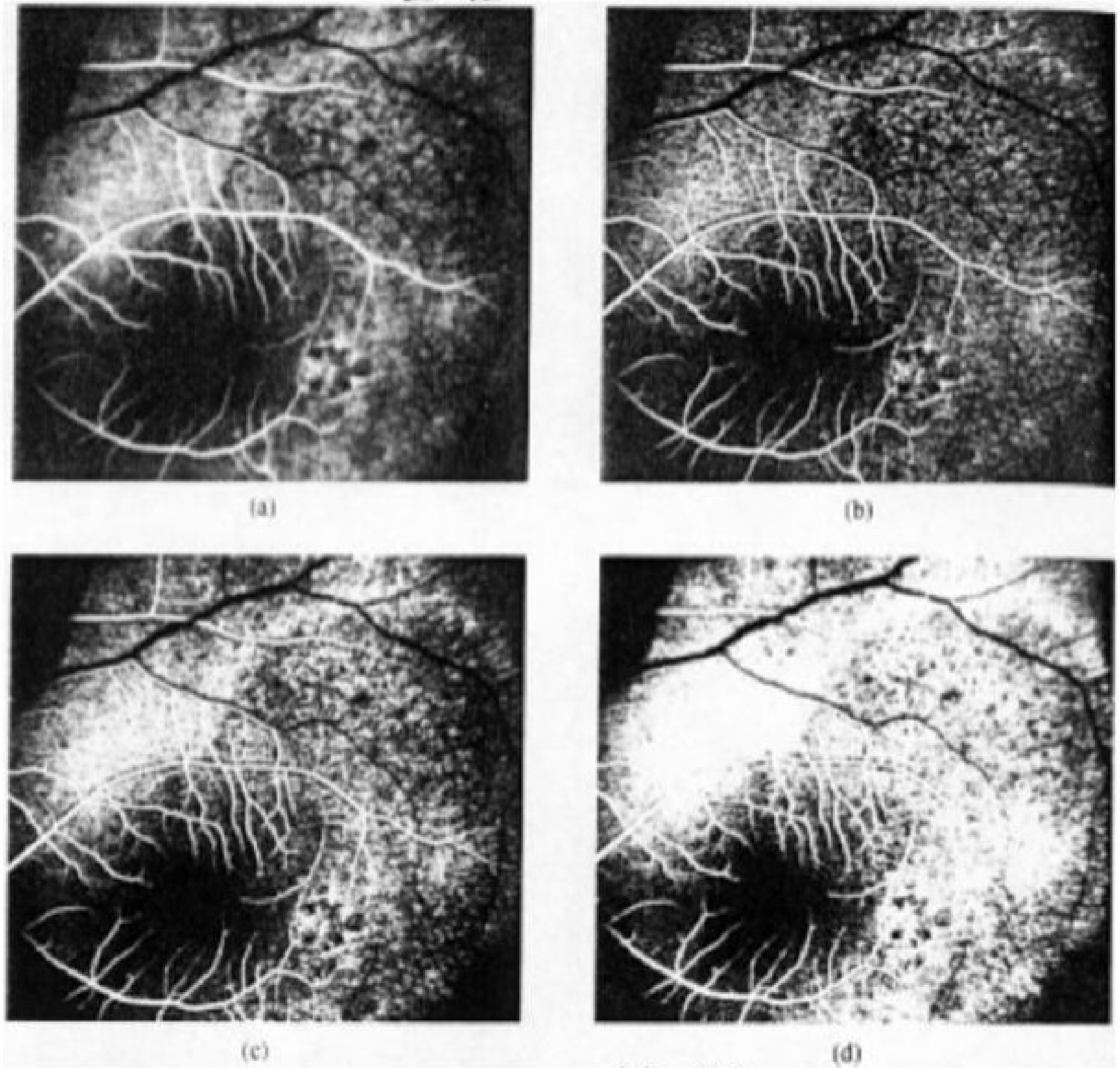
$$\begin{aligned} \text{High-boost} &= A(\text{Original}) - \text{Lowpass} \\ &= (A-1)(\text{Original}) + \text{Original} - \text{Lowpass} \\ &= (A-1)(\text{Original}) + \text{High-pass} \end{aligned}$$

  – If $A>1$, part of the original image is added to the high-pass result (partially restoring low frequency components)

  – Result looks more like the original image with a relative degree of edge enhancement that depends on the value of A

  – May be implemented with the center coefficient value $w=9A-1 (A \geq 1)$

Sharpening Filters (High Boost)



A=1.1

A=1.15

A=1.2

❑ Mean Filter

- Applies a arithmetic mean filter to an image.
- An arithmetic mean filter operation on an image removes short tailed noise such as uniform type noise from the image at the cost of blurring the image. The arithmetic mean filter is defined as the average of all pixels within a local region of an image.
- The arithmetic mean is defined as: $\bar{x} = \frac{1}{n}(x_1 + \cdots + x_n)$

5	3	6
2	1	9
8	4	7

5	3	6
2	5	9
8	4	7

Types of Non Linear Filter

Types of Non Linear Filter

Median Filter

- Applies a median filter to an image. The median filter is defined as the median of all pixels within a local region of an image.

- The median filter is normally used to reduce salt and pepper noise in an image, somewhat like the mean filter. However, it often does a better job than the mean filter of preserving useful detail in the image.

Types of Non Linear Filter

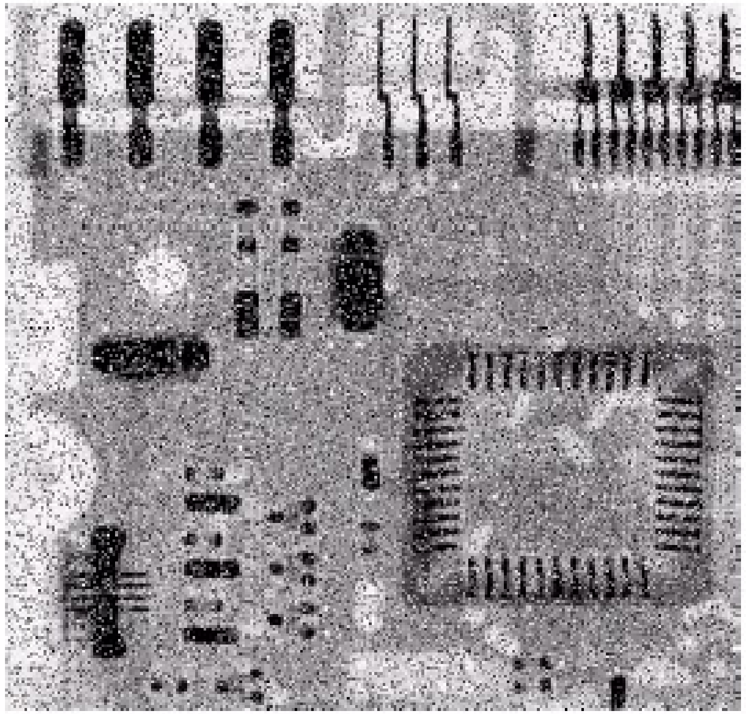
Median Filter

- Unfiltered Values in Order Like
  0,1,2,3,4,6,10,15,19,97
- Median = 6

6	2	0
3	97	4
19	15	10

6	2	0
3	6	4
19	15	10

Types of Non Linear Filter



Original Image With Noise

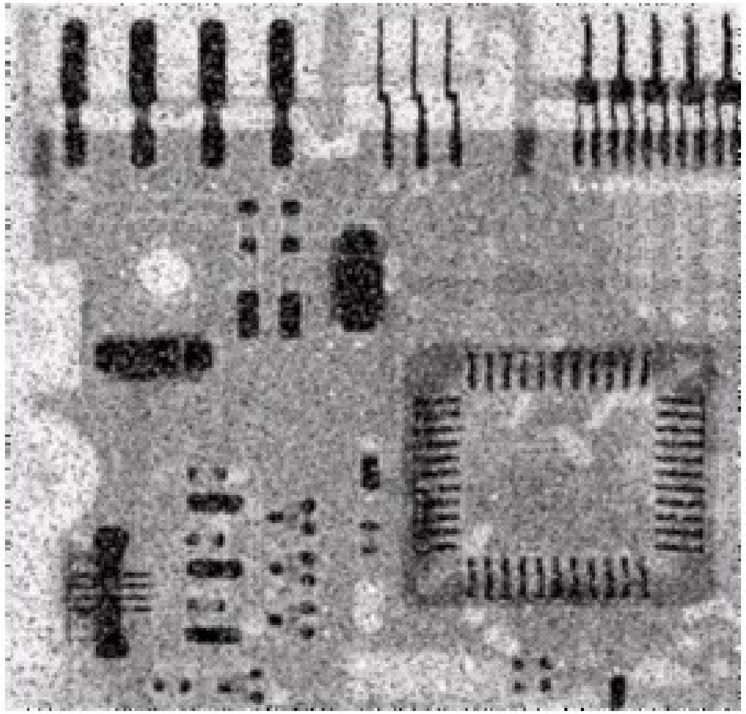


Image After Averaging Filter

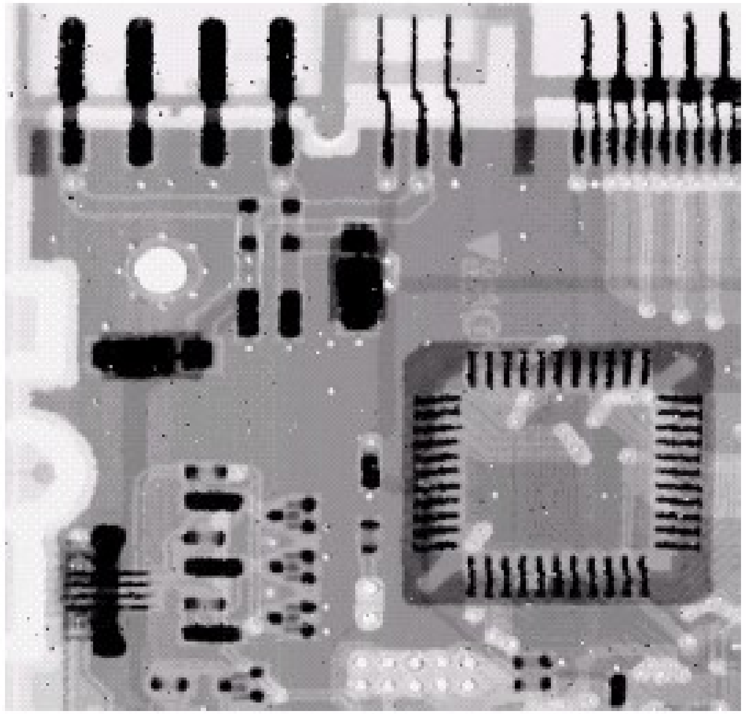


Image After Median Filter

Types of Non Linear Filter

❑ Max Filter

- Used to find the brightest points in an image

  $$R=\max\{z_k|k=1,2,\ldots,9\}$$

- Applies a maximum filter to an image. The maximum filter is defined as the maximum of all pixels within a local region of an image.
- The maximum filter is typically applied to an image to remove negative outlier noise.

Types of Non Linear Filter

- Max Filter

Types of Non Linear Filter

Min Filter

- Used to find the dimmest points in an image

  $R = \min\{z_k \mid k = 1, 2, \ldots, 9\}$

- Applies a minimum filter to an image. The minimum filter is defined as the minimum of all pixels within a local region of an image.

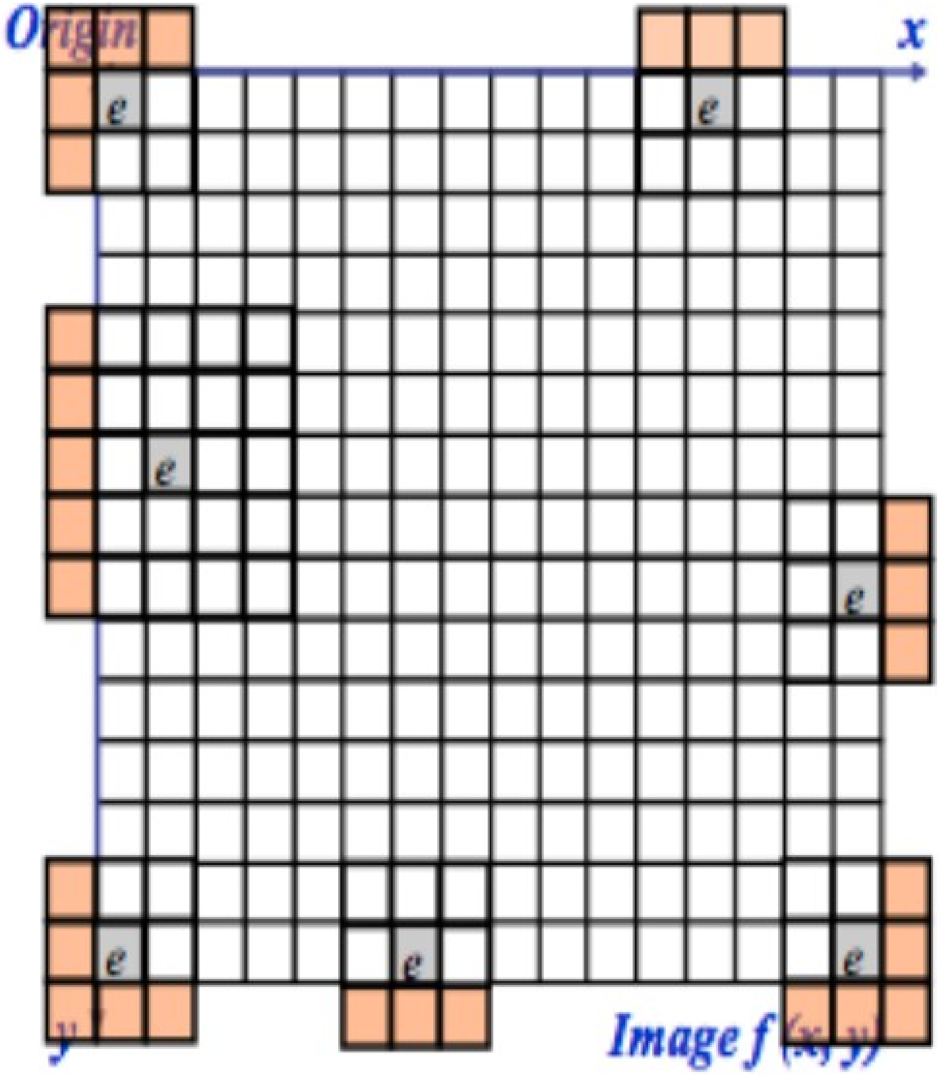
- The minimum filter is typically applied to an image to remove positive outlier noise.

Types of Non Linear Filter

- Min Filter

Spatial Filtering in the Edge

At the end of an image, we are missing pixels to form a neighborhood



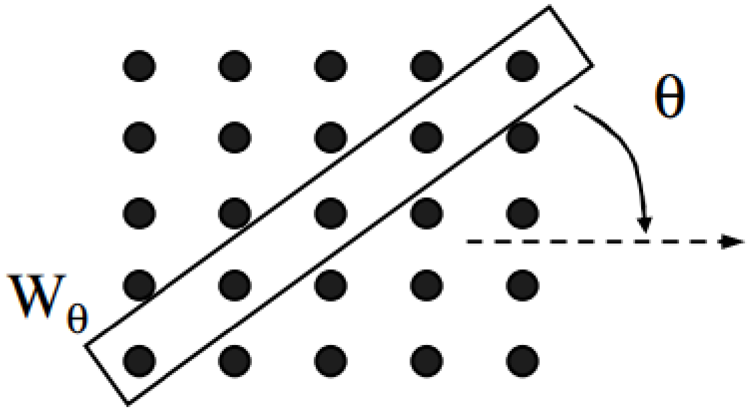
Spatial Filtering in the Edge

There are few approaches to dealing with the missing edge pixels

- Omit missing pixels
  - only works with some filter
  - can add extra code and slow down the processing
- Pad the image
  - typically with either all white or black pixels
- Replicate the border pixels
- Allow pixels wrap around the image

Directional Smoothing

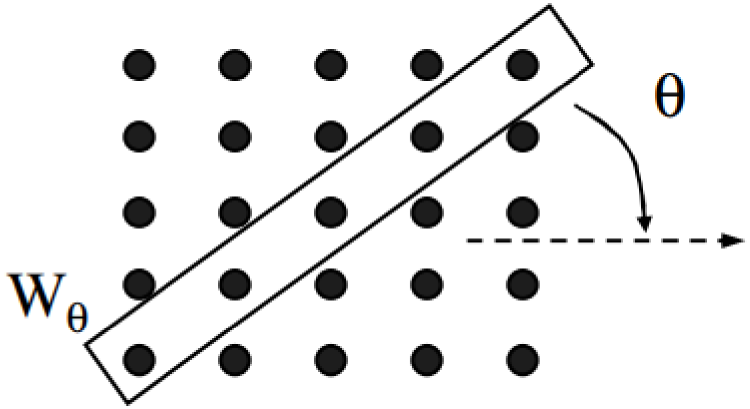
- Problems with simple spatial averaging mask
  - Edges get blurred
- Improvement
  - Restrict Smoothing to along edge direction
  - Avoid Filtering across edges
- Directional Smoothing
  - Computes spatial average along several directions
  - Take the result from the direction giving the smallest changes before and after filtering
- Other Solutions
  - Use more explicit edge detection and adapt filtering accordingly



Directional Smoothing

Spatial average, are calculated in several directions as

$$= \frac{1}{N_\theta} \sum_{(k,l) \in w_\theta} \sum y(m-k, n-l)$$



And a direction θ^* is found such that

$$\left| y(m,n) - v(m,n;\theta^k) \right|$$

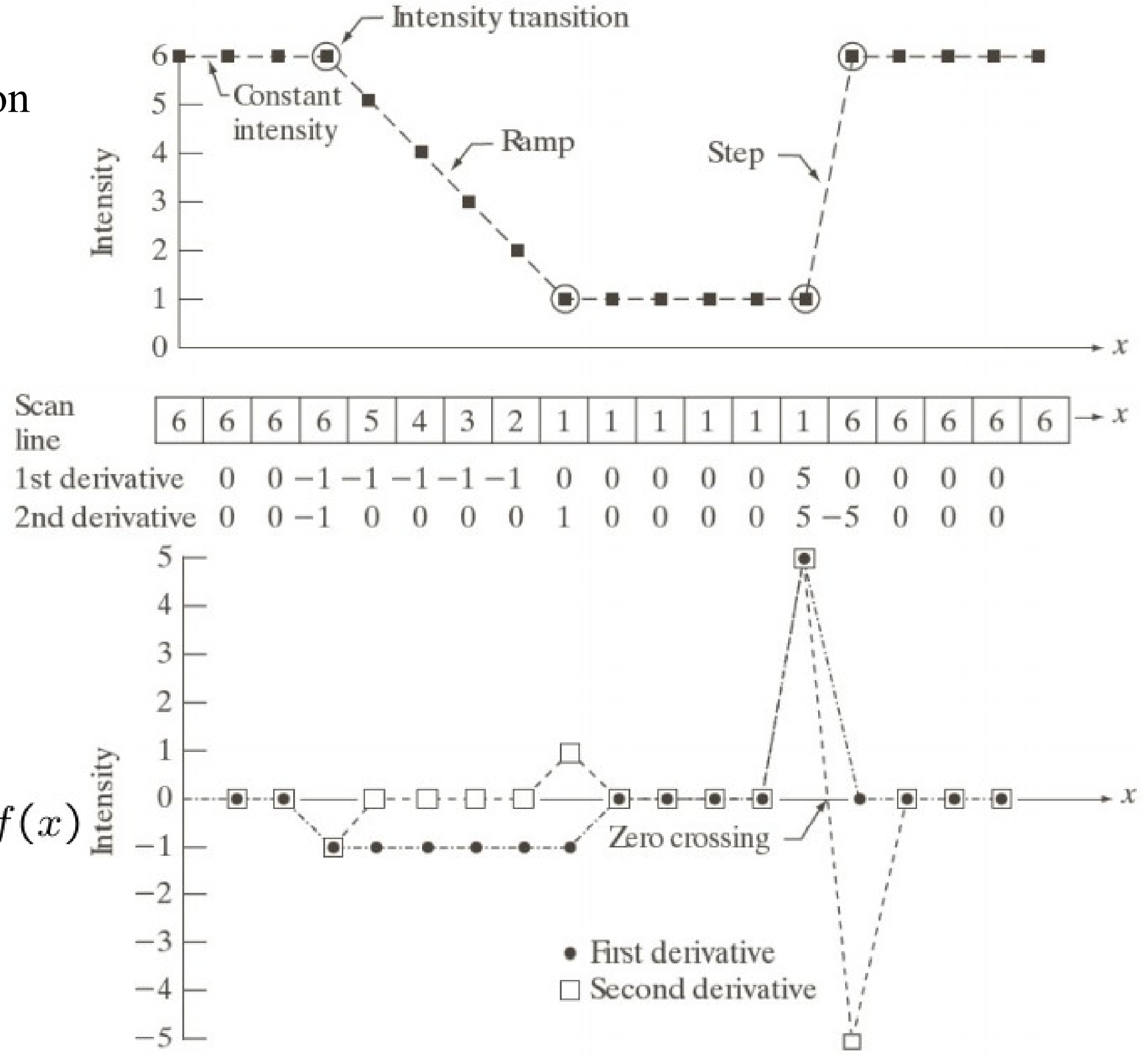
Then $v(m,n) = v(m,n,\theta^k)$ gives the desired result

Spatial Differentiation

Differentiation measures the rate of change of function

$$
\begin{aligned}
f'(x) &= \frac{\partial f}{\partial x} \\
&= f(x+1) - f(x)
\end{aligned}
$$

$$
\begin{aligned}
f''(x) &= \frac{\partial^2 f}{\partial x^2} \\
&= \frac{\partial f}{\partial x}(f'(x) - f'(x-1)) \\
&= f(x+1) + f(x-1) - 2f(x)
\end{aligned}
$$



Spatial Differentiation

Second derivative is more useful for image enhancement than First Derivative

- Stronger response to fine details
- Simpler Implementation

Spatial Differentiation
First Derivative

For a function f(x, y) the gradient of f at coordinates (x, y) is given as the column vector:

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

The magnitude of this vector is given by:

$$\nabla f = mag(\nabla f)$$

$$= \sqrt{G_x^2 + G_y{}^2}$$

$$= \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

For practical reasons this can be simplified as:

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

Spatial Differentiation
First Derivative

The magnitude of the gradient at z_5 can be approximated in number of ways. The simplest is to use the difference $(z_5 - z_8)$ in the x-direction and $(z_5 - z_6)$ in the y-direction

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$$\nabla f = \sqrt{(z_5 - z_8)^2 + (z_5 - z_6)^2}$$

or approximated as

$$\nabla f \approx |z_5 - z_8| + |z_5 - z_6|$$

A cross difference may also be used to approximate the magnitude of the gradient.

$$\nabla f = \sqrt{(z_5 - z_9)^2 + (z_6 - z_8)^2}$$

$$\nabla f \approx |z_5 - z_9| + |z_6 - z_8|$$

0	1
-1	0

1	0
0	-1

This can be implemented by taking the absolute value of the response of the following two masks (the Roberts cross-gradient operators) and summing the results

Spatial Differentiation
First Derivative

Extension to a 3*3 mask yields the following

$$\nabla f \approx |(z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)| + |(z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)|$$

The difference between the first and third rows & first and third columns approximates the derivative in the x-direction and y-direction respectively. The Prewitt operator masks may be used to implement the above approximation.

-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

Robert cross operator

- The Robert cross operator performs a simple, quick to compute, 2-D spatial gradient measurement on image. The output pixels values at each point represent the absolute magnitude of the spatial gradient of the input at that point.

- The operator consists of a pair of 2×2 convolution masks kernels as shown. One kernel is simply rotated by the other by 90 degree.

- It is very similar to the Sobel operator

Gx

1	0
0	-1

Gy

0	1
-1	0

- These masks are designed to respond maximally to edges running at 45° of the pixel grid, Then these are applied to the input image to produce separate measurements of gradient component in each orientation(Gx and Gy)
- The gradient magnitude is given by

$$|G| = (G_x^2 + G_y^2)^{1/2}$$

- The angle is given by

$$\theta = \arctan(G_y / G_x) - 3\pi/4$$

Spatial Differentiation
Second Derivative

- The first sharpening filter, we will look at is Laplacian.
- It is isotropic(property which has the same value when measured in different directions) and one of the simplest sharpening filter. A Laplacian filter is one of edge detectors used to compute the second spatial derivatives of an image. It measures the rate at which the first derivatives changes. In other words, Laplacian filter highlights the regions where the pixel intensities dramatically change
- The Laplacian is defined as follow

0	1	0
1	-4	1
0	1	0

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

where
the partial 2nd order derivative in the x direction is defined as: $\frac{\delta^2 f}{\delta x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y)$

and in the y direction as : $\frac{\delta^2 f}{\delta y^2} = f(x,y+1) + f(x,y-1) - 2f(x,y)$

So, the Laplacian can be given as follows: $\nabla^2 f = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$

Spatial Differentiation Second Derivative

- Applying the Laplacian to an image, we get a new image that highlights edges and other discontinuities .
- The result of an image is not an enhanced image. For this, subtract the Laplacian result from the original image to generate our final sharpened enhanced image

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

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

$$g(x,y) = 5f(x,y) - f(x+1,y) - f(x-1,y) - f(x,y+1) - f(x,y-1)$$

Spatial Differentiation

First Derivative	Second Derivative
Produce thicker edges	Produces fine details
Stronger response to gray level step	Produces Double response at step changes in gray level